

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ / COMPUTER MODELING AND DESIGN AUTOMATION

DOI: <https://doi.org/10.23670/IRJ.2023.138.202>

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ РЕКОНСТРУКЦИИ ПОВЕРХНОСТИ ИЗ ОБЛАКА ТОЧЕК

Научная статья

Сырых А.С.^{1,*}, Медведев С.Н.²¹ORCID : 0000-0001-8326-3743;²ORCID : 0000-0002-6971-5069;^{1,2} Воронежский государственный университет, Воронеж, Российская Федерация

* Корреспондирующий автор (alexandr.syryh[at]gmail.com)

Аннотация

В статье представлен сравнительный анализ алгоритмов реконструкции поверхности из облака точек. Особое внимание уделено реконструкции поверхности в задаче фотограмметрии – технологии получения облака точек из набора изображений.

Основной целью исследования является оценка эффективности алгоритмов реконструкции поверхности с помощью вычислительного эксперимента. В статье приводится краткое описание рассматриваемых алгоритмов, а также описываются необходимые библиотеки, используемые при реализации программного обеспечения для проведения вычислительного эксперимента.

Представлена эталонная модель и производные от неё модели с разной плотностью облака точек (40%, 60% и 80%). Эксперимент проведен в несколько этапов, приведены выводы и результаты по каждому из них. Показано, что результаты эксперимента соответствуют теоретическим фактам, также отраженным в статье. В завершении дается ответ на вопрос о применимости алгоритмов в задаче фотограмметрии.

Ключевые слова: реконструкция поверхности, алгоритм Poisson Surface Reconstruction, алгоритм Ball-Pivoting, алгоритм Power Crust, качество модели.

A COMPARATIVE ANALYSIS OF ALGORITHMS FOR SURFACE RECONSTRUCTION FROM A POINT CLOUD

Research article

Sirikh A.S.^{1,*}, Medvedev S.N.²¹ORCID : 0000-0001-8326-3743;²ORCID : 0000-0002-6971-5069;^{1,2} Voronezh State University, Voronezh, Russian Federation

* Corresponding author (alexandr.syryh[at]gmail.com)

Abstract

The article presents a comparative analysis of algorithms for surface reconstruction from a point cloud. Particular attention is paid to surface reconstruction in the photogrammetry problem – the technology of obtaining a point cloud from a set of images.

The main objective of the study is to evaluate the efficiency of surface reconstruction algorithms using computational experiment. The work provides a brief description of the analysed algorithms and describes the necessary libraries used in the implementation of the software for the computational experiment.

The reference model and derived models with different point cloud densities (40%, 60% and 80%) are presented. The experiment was carried out in several stages, conclusions and results for each of them are presented. It is shown that the experimental results are consistent with the theoretical facts, also reported in the article. Finally, the question of applicability of the algorithms in the photogrammetry problem is answered.

Keywords: surface reconstruction, Poisson Surface Reconstruction algorithm, Ball-Pivoting algorithm, Power Crust algorithm, model quality.

Введение

Реконструкция поверхности – это процесс создания трехмерного представления поверхности из набора дискретных точек, часто полученных с помощью технологий трехмерного сканирования или визуализации. Это фундаментальная задача появляется в различных областях. Так, например, в медицинской визуализации [1] реконструкция поверхности может использоваться для создания 3D-моделей анатомических структур на основе медицинских изображений, таких как магнитно-резонансная томография и компьютерная томография. Это позволяет врачам визуализировать и изучать структуры в трехмерном пространстве, что может помочь в диагностике, планировании лечения и хирургическом моделировании. В робототехнике [2] реконструкция поверхности может использоваться для создания трехмерных моделей объектов, с которыми робот должен взаимодействовать, что позволяет ему планировать свои движения и взаимодействия. В архитектуре и строительстве [3] реконструкция поверхности может быть использована для создания 3D-моделей зданий или строительных площадок с помощью собранной информации с беспилотных летательных аппаратов или лазерных сканеров. Такая технология позволяет архитекторам и строителям визуализировать и планировать процесс строительства, дизайн ландшафта, что может помочь выявить потенциальные проблемы и оптимизировать процесс строительства. В виртуальной реальности и

видеоиграх [4] реконструкция поверхности может использоваться для создания трехмерных моделей игровых сред или сред виртуальной реальности на основе реальных объектов или концепт-арта. Это позволяет игрокам взаимодействовать с окружением в трехмерном пространстве и может улучшить впечатления от игры или виртуальной реальности.

Таким образом, реконструкция поверхности позволяет создавать детальные 3D-модели, которые можно использовать для визуализации, анализа, моделирования и производства.

Методы реконструкции поверхности и настройка или выбор алгоритмов под требуемые критерии является актуальной задачей. Методы реконструкции можно разделить на две категории: неявные и явные методы. Неявные методы представляют поверхность в виде непрерывной функции или уравнения, в то время как явные методы определяют поверхность как набор вершин, ребер и граней [5]. Неявные методы часто используются для гладких и непрерывных поверхностей, в то время как явные методы подходят для поверхностей с разрывами или резкими особенностями.

Существует несколько методов реконструкции поверхности, начиная от простых методов интерполяции и заканчивая продвинутой алгоритмами, которые могут обрабатывать большие и зашумленные наборы данных. В статье будут рассмотрены некоторые методы и алгоритмы, проанализированы их преимущества и недостатки, а также даны рекомендации по их использованию и применению.

Постановка задачи

Целью данной статьи является сравнение явных алгоритмов реконструкции поверхности: Ball-Pivoting (поворот шара) [6], Poisson Surface Reconstruction (реконструкция поверхности Пуассона) [7], Power Crust (силовая корка) [8], а также поиск наиболее подходящего алгоритма для задачи фотограмметрии [9]. Сравнение будет проведено на основе вычислительного эксперимента.

Под подходящим алгоритмом будем подразумевать алгоритм, который удовлетворяет следующим критериям:

1. Критерий топологии. Все исходные точки, которые подавались на вход, после реконструкции должны оставаться на тех же местах;

2. Критерий овражности. Модель не должна иметь овражностей в своей геометрии, если так не было задумано;

3. Критерий непрерывности. Модель не должна иметь разрывов в своей геометрии, если так не было задумано;

4. Критерий точности. Модель должна соответствовать реальному объекту.

Работу фотограмметрии [9] можно разделить на три этапа.

I. Получение снимков объекта;

II. Получение 3D модели из серии фотографий;

III. Пост-обработка получившейся 3D модели.

На втором этапе выделяют 4 шага:

1. Сопоставление изображений;

2. Калибровка камер и получение разреженного облака точек;

3. Получение плотного облака точек;

4. Реконструкция поверхности из плотного облака точек.

Как видно, реконструкция поверхности происходит на четвертом шаге второго этапа процесса фотограмметрии.

Алгоритм реконструкции поверхности для задачи фотограмметрии должен удовлетворять следующим требованиям:

– способность обрабатывать зашумленное облако точек;

– работа с плотностью точек 60% [10] (экстремальное значение – 40% [11]).

Описание алгоритмов реконструкции поверхности

Далее приведем краткое описание рассматриваемых алгоритмов реконструкции для лучшего восприятия их работы.

Алгоритм Ball-Pivoting (поворот шара) [6] – это метод реконструкции поверхности, который предполагает размещение набора шаров в облаке точек и последующее соединение их в сеть треугольников.

Алгоритм можно описать следующим образом:

1. Инициализация. Сначала случайным образом выбирается начальная точка из входного облака точек;

2. Выбор шара. Шар с фиксированным радиусом помещается в начальную точку, и алгоритм пытается определить соседние точки в пределах шара, которые могут образовать треугольник с начальной точкой. Шар постепенно поворачивается для поиска потенциальных точек-кандидатов;

3. Формирование треугольника. Если в пределах шара найдена подходящая точка-кандидат, то формируется треугольник, соединяющий начальную точку, найденную точку-кандидат и произвольную точку из облака точек, находящуюся в шаре. Этот образованный треугольник добавляется к существующей сетке;

4. Выбор поворотной точки. После формирования треугольника алгоритм определяет точку поворота по краям треугольника. Точка поворота должна соответствовать минимальному углу, для обеспечения качества результирующей поверхности;

5. Создание новых треугольников. Шаги 2-4 повторяются итеративно, при этом поворотная точка становится новой начальной точкой.

Алгоритм Poisson Surface Reconstruction (реконструкции поверхности Пуассона) [7] – это техника, используемая для создания гладких и замкнутых сеток из облака точек.

Алгоритм можно описать следующим образом:

1. Предварительная обработка. Расчет нормалей облака точек при помощи алгоритма Point Cloud Normal Estimation (оценка нормалей облака точек) [12];

2. Реконструкция Пуассона. Реконструкция Пуассона формулирует задачу реконструкции как задачу решения уравнения Лапласа с граничными условиями Дирихле, используя объемное представление, известное как знаковое поле расстояний (SDF) [14];

3. Вычисление знакового поля расстояний SDF. Сначала облако точек описывается минимальным объемным примитивом (кубом). После этого куб равномерно разбивается на воксели. Далее для каждого вокселя или ячейки в объемном представлении алгоритм вычисляет знаковое расстояние от этого вокселя до ближайшей точки входного облака точек. Знак указывает на то, находится ли этот воксель внутри или вне реконструируемой поверхности;

4. Неявное выделение поверхности. Поле SDF используется для выделения неявной поверхности, представляющей реконструированную поверхность, как изоповерхность SDF. После этого при помощи алгоритма Marching Cubes [15] происходит создание триангулированной сетки.

Алгоритм Power Crust (силовой корки) [8] – это алгоритм, который строит поверхность на основе диаграммы Вороного [16].

Алгоритм можно описать следующим образом:

1. Построение диаграммы Вороного. Алгоритм начинает работу с построения диаграммы Вороного для входных точек. Диаграмма Вороного делит пространство на регионы на основе близости к каждой точке, создавая сеть ячеек вокруг каждой точки;

2. Создание двудольного графа. В двудольном графе вершины представляют собой ячейки диаграммы Вороного, а ребра – грани диаграммы Вороного;

3. Вычисление медиальной оси двудольного графа. Медиальная ось представляет собой центральное ядро или скелет облака точек, которые равноудалены от двух или более точек входа;

4. Построение начальной поверхности. Алгоритм покрывает все входное облако точек путем соединения точек медиальной оси. Получается начальная поверхность, образованная сетью тетраэдров;

5. Уточнение поверхности. Алгоритм производит уточнение поверхности с помощью следующих шагов:

5.1. Подразделение. Поверхность подразделяется на более мелкие тетраэдры;

5.2. Классификация. Каждый тетраэдр классифицируется, как находящийся внутри, снаружи или на поверхности облака точек;

5.3. Реконструкция поверхности. Алгоритм строит новую поверхность, которая аппроксимирует облако точек, соединяя точки медиальной оси тетраэдров, классифицированных как «находящиеся на поверхности»;

5.4. Завершение. Процесс уточнения завершается при достижении желаемого уровня точности;

6. Получение выходной поверхности.

В заключение следует отметить, что алгоритмы Ball-Pivoting, Poisson Surface Reconstruction и Power Crust представляют собой значительные достижения в области реконструкции поверхности, предлагая уникальные подходы к восстановлению моделей по облаку точек. Эти алгоритмы играют важную роль в различных областях, таких как компьютерная графика, компьютерное зрение и робототехника, позволяя создавать точные и детальные представления поверхностей, которые вносят вклад в широкий спектр приложений и исследований.

Анализ алгоритмов реконструкции поверхности из облака точек на основе теории

Проанализируем рассмотренные выше алгоритмы на основе некоторых известных сведений, теории и предположений о них (табл. 1) [6], [7], [8].

Таблица 1 - Основные характеристики алгоритмов

DOI: <https://doi.org/10.23670/IRJ.2023.138.202.1>

	Плотность облака точек	Качество сетки	Чувствительность к параметрам	Детерминированный выход	Способность обрабатывать зашумленное облако точек	Сложность алгоритма
Ball-Pivoting	> 40 %	При качественном входе получается качественная выходная сетка	+	+	+	$O(n \log n)$
Poisson Surface Reconstruction	> 50 %	При качественном входе получается качественная	-	-	+	$O(n \log n)$

		выходная сетка				
Power Crust	> 60 %	При качественном входе получается качественная выходная сетка	+ (ограниченный контроль)	+	+	$O(n \log n)$

Можно сделать следующие выводы:

1. Все алгоритмы имеют одинаковую вычислительную сложность $O(n \log n)$;
 2. Алгоритм Poisson Surface Reconstruction не чувствителен к параметрам, однако имеет основной недостаток – недетерминированный выход;
 3. У всех алгоритмов получается качественная сетка при качественном входе. Однако только алгоритм Ball-Pivoting способен обрабатывать облако точек, начиная с 40% плотности;
 4. И, наконец, все алгоритмы способны обрабатывать зашумленное облако точек.
- Предложенные выводы сделаны на основе анализа имеющихся в открытом доступе источников с некоторой долей предположения, так как не все данные есть в открытом доступе и не всегда их можно обобщить. Целью дальнейшего вычислительного эксперимента будет подтверждение или опровержение данных выводов.

Вычислительный эксперимент

Для сравнения результатов работы алгоритма был проведен вычислительный эксперимент. Его суть заключалась в восстановлении поверхности модели из облака точек уже существующей модели и анализа получившегося результата. Входными данными вычислительного эксперимента были облака точек (вершины) разной плотности (40%, 60% и 80%) исходной модели низкополигонального стэнфордского кролика (Stanford Bunny) [17], изображенной на рис. 1.

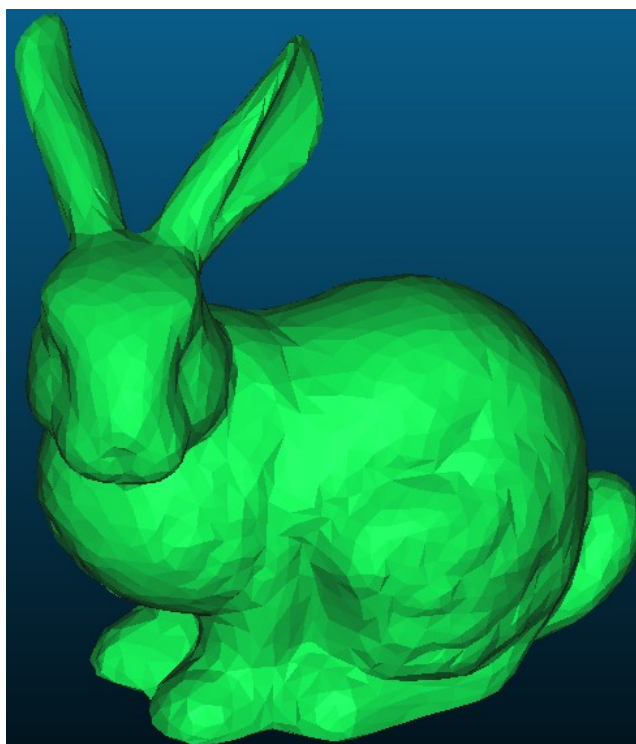


Рисунок 1 - Входная модель
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.2>

Выбор этой модели обоснован несколькими важными ее особенностями: у нее имеются области со сложной геометрией (уши), небольшие существенные области (лапы и хвост) и крупные мало детализированные области (туловище).

Общее количество точек в плотном облаке точек модели – 2503.

Сначала эксперимент был проведен на незашумленных данных, а после – на зашумленных.

На рис. 2 (а-в) показано незашумленное облако точек с разными плотностями.

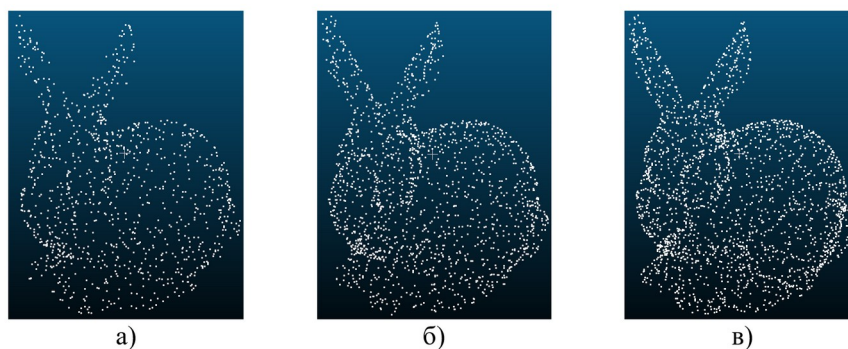


Рисунок 2 - Облако точек с плотностью 40% (а), 60% (б), 80% (в)
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.3>

А на рис. 3 (а-в) показано зашумленное облако точек с разными плотностями. Всего было добавлено 20% зашумленных точек от количества точек в текущем облаке. Шум был сгенерирован по нормальному закону распределения $\mu = 0, \sigma^2 = 0,05$.

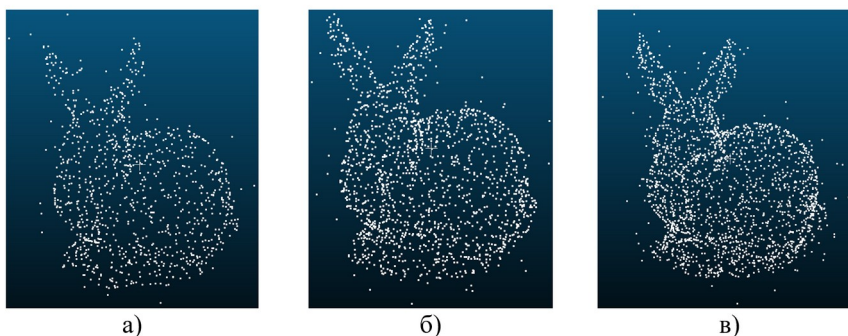


Рисунок 3 - Облако точек с плотностью 40% (а), 60% (б), 80% (в)
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.4>

Сравнение результатов работы алгоритмов производилось путем наложения полученной модели на исходную и нахождения расстояния между ними с помощью алгоритма Cloud to Mesh Signed Distances [18].

Эксперимент состоял из трех этапов:

1. Оценка работы каждого алгоритма на трех облаках точек разных плотностей на незашумленных данных;
2. Сравнение работы всех алгоритмов на одинаковых плотностях и выбор алгоритма, который лучше всего подходит под критерии, описанные в постановке задачи;
3. Проведение этапов 1 и 2 для зашумленных данных.

Для проведения вычислительного эксперимента была написана программа на языке Python. В функционал программы входит управление запуском алгоритмов, поиск расстояний между моделями, добавление шума к облаку точек и создание тепловой карты наложения моделей друг на друга. Также в функционал программы входит проверка на пустые полигоны модели с помощью алгоритма «Ray Casting» [19].

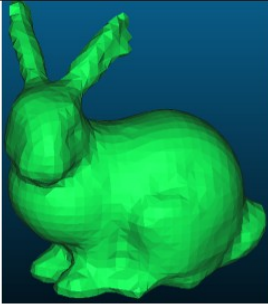
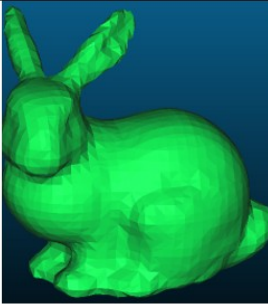
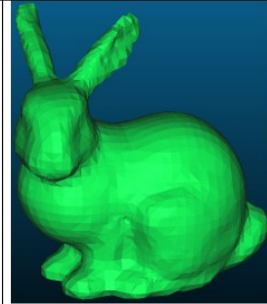
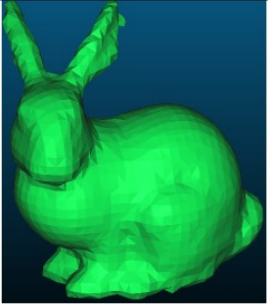
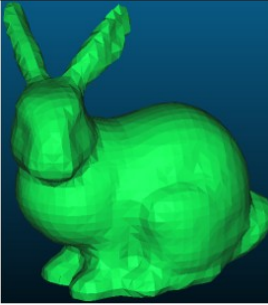
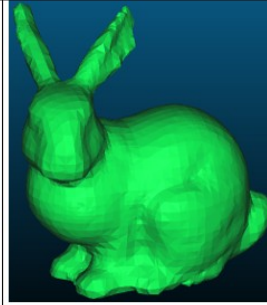
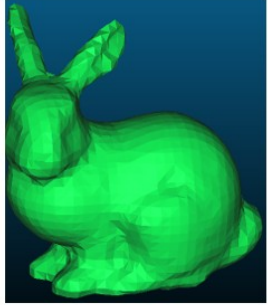
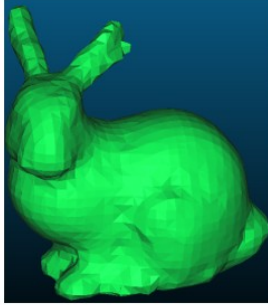
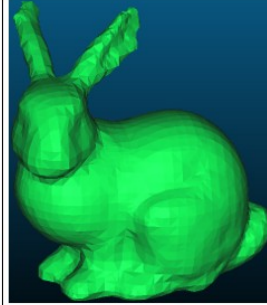
Алгоритмы Ball-Pivoting и Poisson Surface Reconstruction были взяты из библиотеки Open3D [20], а алгоритм Power Crust был взят с официального сайта разработчиков метода [21]. Алгоритм Cloud to Mesh Signed Distances был взят с библиотеки Cloud Compare [18].

Анализ результатов работы алгоритмов

В ходе проведения первого этапа вычислительного эксперимента были получены модели, представленные в табл.

2.

Таблица 2 - Результаты работы алгоритмов на незашумленных данных
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.5>

	Плотность облака точек		
	40%	60%	80%
Алг. 1. Ball-Pivoting			
Алг. 2. Poisson Surface Reconstruction			
Алг. 3. Power Crust			

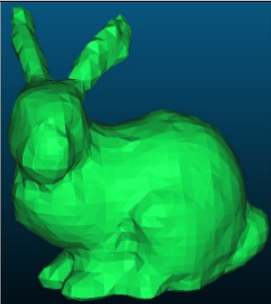
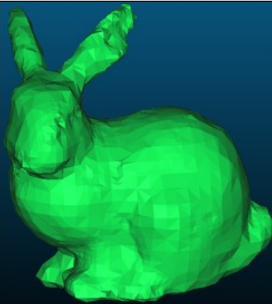
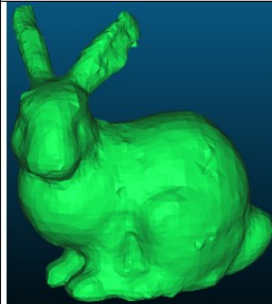
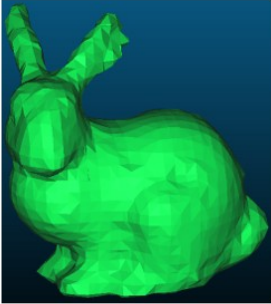
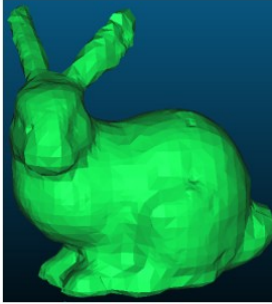
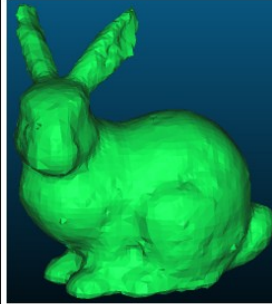
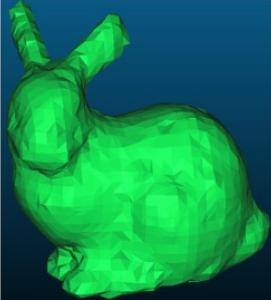
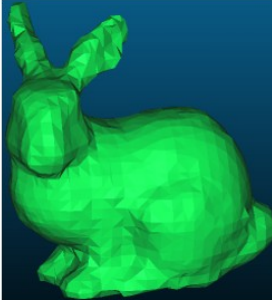
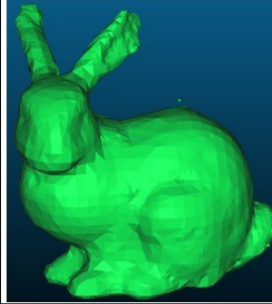
Анализ табл. 2 по строкам дает следующие результаты для всех девяти случаев:

- входные точки остались на своих местах (программные показатели дают среднее расстояние между соответствующими точками в пределах $e-5$);
- модели не имеют овражностей (визуальная оценка);
- модели не имеют пустых полигонов (метод «Ray Casting»).

Таким образом, требования 1-3, указанные в постановке задачи, были выполнены для всех алгоритмов.

Результаты работы алгоритмов на зашумленных данных представлены в табл. 3.

Таблица 3 - Результаты работы алгоритмов на зашумленных данных
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.6>

	Плотность облака точек		
	40 %	60 %	80 %
Алг. 1. Ball-Pivoting			
Алг. 2. Poisson Surface Reconstruction			
Алг. 3. Power Crust			

Анализ табл. 3 по строкам дает следующие результаты для всех девяти моделей:

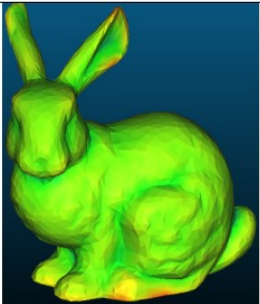
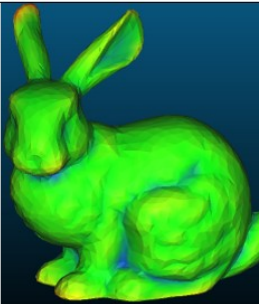
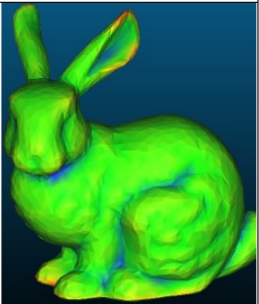
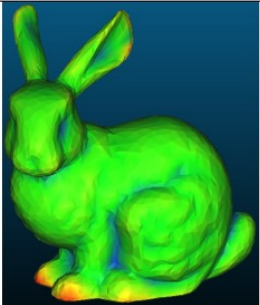
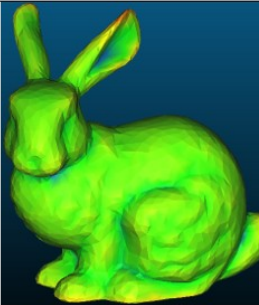
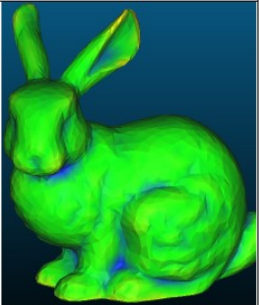
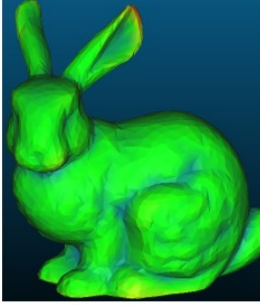
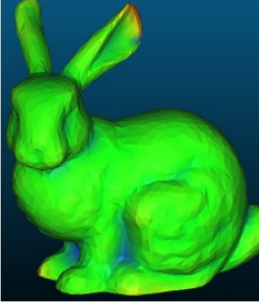
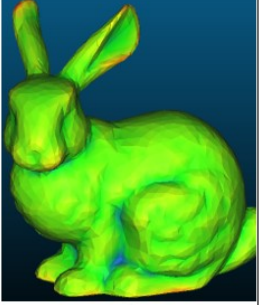
- входные точки остались на своих местах (среднее расстояние между соответствующими точками в пределах $e-3$);
- модели не имеют овражностей (визуальная оценка);
- модели не имеют пустых полигонов (метод «Ray Casting» для поиска разрывов в полигональной сетке).

Таким образом, требования 1-3, указанные в постановке задачи, были выполнены для всех алгоритмов. Также все алгоритмы показали способность обрабатывать зашумленное облако точек, что является важным требованием в задаче фотограмметрии.

Теперь оценим качество работы алгоритмов путем наложения построенных моделей на исходную и поиска расстояния между ними. То есть теперь рассмотрим алгоритмы на соответствие требованию 4 из постановки задачи.

Табл. 4 демонстрирует наложение моделей, полученных в результате работы алгоритмов на исходную, и расстояния между ними на незашумленных данных. Расстояния отображены в виде тепловых карт на исходной модели. Карта показывает, на сколько отличается исходная и полученная модели.

Таблица 4 - Расстояния между моделями на незашумленных данных
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.7>

	Плотность облака точек		
	40 %	60 %	80 %
Алг. 1. Ball-Pivoting			
Алг. 2. Poisson Surface Reconstruction			
Алг. 3. Power Crust			

На рис. 4 изображена шкала для расшифровки тепловых карт. Отрицательные значения (синие и зеленые тона) говорят о расположении точек восстановленной модели внутри исходной, а соответственно положительные значения (желтые и красные тона) – о расположении точек снаружи.

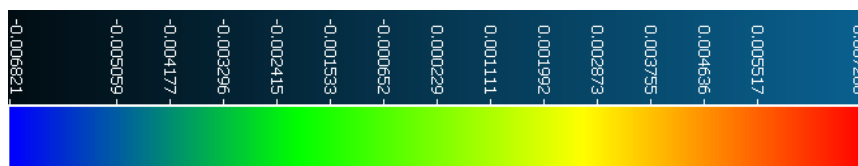


Рисунок 4 - Шкала расшифровки тепловых карт
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.8>

Проанализируем строки и столбцы табл. 4 и выберем алгоритм, который лучше всего описывает исходную модель на разных плотностях. Для этого оценим максимальное по модулю и среднее расстояния между наложенными моделями, которые указаны в табл. 5.

Таблица 5 - Расстояния между моделями на незашумленных данных
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.9>

		Плотность облака точек		
		40%	60%	80%
Алг. 1. Ball-	Ср.	<u>0,00057843</u>	0,00051966	0,00038131

Pivoting	Макс.	<u>0,00664158</u>	0,00767418	<u>0,00402437</u>
Алг. 2. Poisson Surface Reconstruction	Ср.	0,00059747	<u>0,00041903</u>	<u>0,00034862</u>
	Макс.	0,00717853	<u>0,00529197</u>	0,00618712
Алг. 3. Power Crust	Ср.	0,00062632	0,00093134	0,00037360
	Макс.	0,01243361	0,01518252	0,00454011

Примечание: жирным шрифтом выделены наименьшие значения по строкам, а подчеркиванием – в столбцах

Основным показателем в табл. 5 является среднее значение расстояний: чем оно меньше, тем качественнее результат реконструкции.

Показатель максимального расстояния является вспомогательным и может интерпретироваться как наличие «выбросов». Чем меньше данное значение, тем качественнее результат реконструкции.

Исходя из анализа строк табл. 5 можно определить, что наилучший результат по среднему расстоянию достигается при наибольшей плотности (80%), что достаточно очевидно и отражает теоретический анализ алгоритмов (качественная сетка при качественном входе). Также при наибольшей плотности для всех алгоритмов максимальные значения отличаются от средних на один разряд. В то время как на меньших плотностях у алгоритма Power Crust они отличаются на два разряда.

Стоит отметить также высокое среднее значение при плотности 60% у алгоритма Power Crust. Данный факт требует дополнительного анализа и исследования, но не в рамках данной работы.

Исходя из этого, можно сделать вывод, что алгоритм Power Crust уступает двум другим рассматриваемым алгоритмам при работе на малых и средних плотностях.

Теперь проанализируем столбцы табл. 5, то есть работу алгоритмов на одинаковых плотностях:

– на 40% плотности облака точек лучшим алгоритмом оказался Ball-Pivoting. Среднее расстояние составило 0,000578438, а максимальное – 0,00664158.

– на 60% плотности облака точек лучшим алгоритмом оказался Poisson Surface Reconstruction. Среднее расстояние составило 0,00041903, а максимальное – 0,00529197.

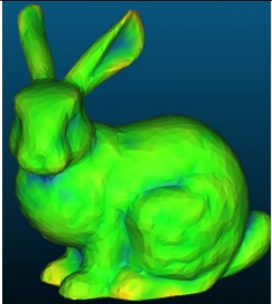
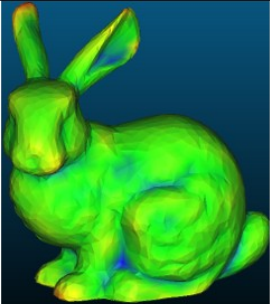
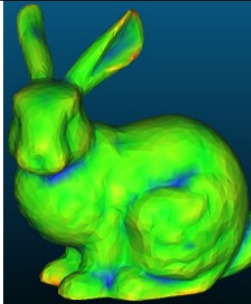
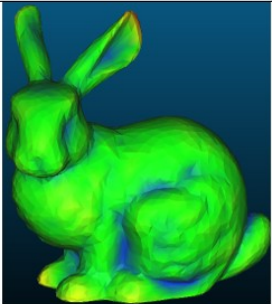
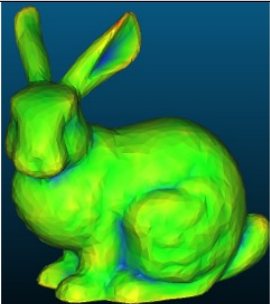
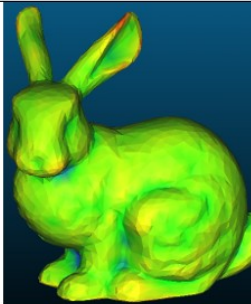
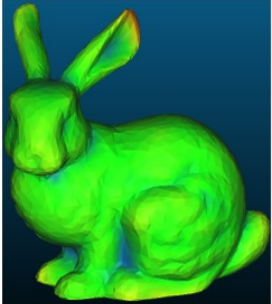
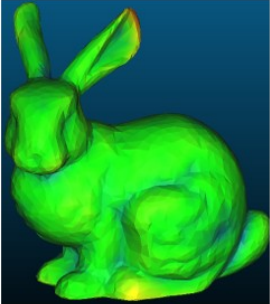
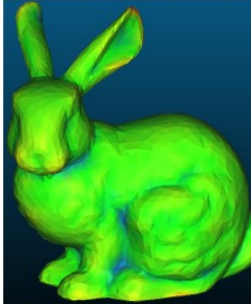
– на 80% плотности облака точек лучшим алгоритмом оказался Poisson Surface Reconstruction. Среднее расстояние составило 0,00034862, а максимальное – 0,00618712.

Подытоживая, можно сказать, что на малых значениях плотности лучшие показатели наблюдаются у алгоритма Ball-Pivoting, на средних – Poisson Surface Reconstruction и на больших значениях все алгоритмы показывают схожий результат. Это отражает теоретический анализ табл. 1. Заметим, что при 40% средние значения у алгоритмов Ball-Pivoting и Poisson Surface Reconstruction (0,00057843 и 0,00059747 соответственно) близки, но максимальное значение меньше у алгоритма Ball-Pivoting (0,00664158 против 0,00717853).

Теперь перейдем к анализу работы алгоритмов восстановления поверхности на зашумленных данных.

Табл. 6 демонстрирует наложение моделей, полученных в результате работы алгоритмов на исходную, и расстояние между ними на зашумленных данных.

Таблица 6 - Расстояния между моделями на зашумленных данных
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.10>

	Плотность облака точек		
	40 %	60 %	80 %
Алг. 1. Ball-Pivoting			
Алг. 2. Poisson Surface Reconstruction			
Алг. 3. Power Crust			

Как и в случае незашумленных данных проанализируем строки и столбцы табл. 6 и выберем алгоритм, который лучше всего описывает исходную модель на разных плотностях. Для этого также оценим максимальное и среднее расстояния по модулю между наложенными моделями, которые указаны в табл. 7.

Таблица 7 - Максимальное и среднее расстояние между моделями на зашумленных данных

DOI: <https://doi.org/10.23670/IRJ.2023.138.202.11>

		Плотность облака точек		
		40%	60%	80%
Алг. 1. Ball-Pivoting	Ср.	<u>0,00070887</u>	0,00068436	0,00043370
	Макс.	<u>0,00759576</u>	0,00698583	0,00439768
Алг. 2. Poisson Surface Reconstruction	Ср.	0,00075802	<u>0,00054058</u>	0,00038608
	Макс.	0,01061581	<u>0,00509344</u>	0,00454925
Алг. 3. Power Crust	Ср.	0,00108207	0,00076345	0,00042805
	Макс.	0,01367722	0,01226435	0,00633447

Примечание: жирным шрифтом выделены наименьшие значения по строкам, а подчеркиванием – в столбцах

Исходя из анализа строк табл. 7 здесь также наилучший результат по среднему расстоянию достигается при наибольшей плотности (80%), что еще раз отражает теоретический анализ алгоритмов (качественная сетка при качественном входе). Также при наибольшей плотности для всех алгоритмов максимальные значения отличаются от

средних на один разряд. В то время как на меньших плотностях у алгоритма Power Crust значения отличаются на два разряда.

Стоит отметить, что высокого среднего значения при плотности 60% у алгоритма Power Crust здесь не наблюдается. Однако и тут можно сделать вывод о том, что алгоритм Power Crust уступает двум другим рассматриваемым алгоритмам при работе на малых и средних плотностях.

Общий анализ табл. 7 дает следующие результаты:

– на 40% плотности облака точек лучшим алгоритмом оказался Ball-Pivoting. Среднее расстояние составило 0,000708876, а максимальное – 0,00759576.

– на 60% плотности облака точек лучшим алгоритмом оказался Poisson Surface Reconstruction. Среднее расстояние составило 0,00054058, а максимальное – 0,00509344.

– на 80% плотности облака точек лучшим алгоритмом оказался Poisson Surface Reconstruction. Среднее расстояние составило 0,00038608, а максимальное – 0,00454925.

Подытоживая, можно сказать, что на малых значениях плотности лучшие показатели наблюдаются у алгоритма Ball-Pivoting, на средних – Poisson Surface Reconstruction и на больших значениях все алгоритмы показывают схожий результат. Это отражает теоретический анализ табл. 1. То есть выводы получились, как и на незашумленных данных.

Таким образом, исходя из анализа табл. 5 и табл. 7, можно сделать следующие выводы:

1. Все алгоритмы дают качественный результат при качественном входе, то есть на плотности 80%;
2. Сравнительный анализ алгоритмов не зависит от зашумленности данных;
3. На малых плотностях рекомендуется использовать алгоритм Ball-Pivoting;
4. На средних плотностях – Poisson Surface Reconstruction;
5. На высоких плотностях – любой из трех алгоритмов.

И, наконец, сделаем вывод о применимости алгоритмов к задаче фотограмметрии. В задаче фотограмметрии используется облако точек плотности от 70% [10]. Экстремальные случаи – 60%. Все, что ниже – некачественно записанные входные данные. Более того, в фотограмметрических данных могут содержаться шумы. Таким образом, для задачи фотограмметрии, исходя из предложенных выше выводов, идеально подойдут алгоритмы Ball-Pivoting и Poisson Surface Reconstruction.

Заключение

В результате проведено сравнение алгоритмов Ball-Pivoting, Poisson Surface Reconstruction, Power Crust путем проведения вычислительного эксперимента, а также поиск наиболее подходящего алгоритма для задачи фотограмметрии. Суть вычислительного эксперимента заключалась в восстановлении поверхности модели из незашумленного и зашумленного облака точек разной плотности уже существующей модели и последующего анализа получившейся модели.

В результате сделан вывод о том, что лучшим алгоритмом на зашумленном и незашумленном облаке точек плотности 40% оказался алгоритм Ball-Pivoting. А лучшим алгоритмом на зашумленном и незашумленном облаке точек плотности 60% и 80% оказался алгоритм Poisson Surface Reconstruction.

Исходя из известных фактов о входных данных в задаче фотограмметрии, сделан вывод о применимости для работы с ней алгоритмов Ball-Pivoting и Poisson Surface Reconstruction.

Конфликт интересов

Не указан.

Рецензия

Извеков Ю.А., Магнитогорский государственный технический университет имени Г.И. Носова, Магнитогорск, Российская Федерация
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.12>

Conflict of Interest

None declared.

Review

Izvekov Y.A., G.I. Nosov Magnitogorsk State Technical University, Magnitogorsk, Russian Federation
DOI: <https://doi.org/10.23670/IRJ.2023.138.202.12>

Список литературы / References

1. Wöhler Ch. 3D Computer Vision: Efficient Methods and Applications / Ch. Wöhler. — Springer London, 2013. — 382 p.
2. Thrun S. Probabilistic Robotics / S. Thrun, W. Burgard, D. Fox. — Cambridge: MIT Press, 2005. — 647 p.
3. Abramczyk H. Introduction to Laser Spectroscopy / H. Abramczyk. — Oxford: Elsevier Science & Technology, 2005. — 429 p.
4. Меженин А.В. Моделирование виртуальной среды в технологиях кибервизуализации и виртуального присутствия / А.В. Меженин, В.В. Извозчикова, Д.И. Бурлов // Кибернетика и программирование. — 2019. — № 4. — С. 26-35.
5. Hughes J.F. Computer Graphics: Principles and Practice / J.F. Hughes, A. van Dam, M. McGuire [et al.]. — Addison-Wesley, 2013. — 1264 p.
6. Bernardini F. The Ball-Pivoting Algorithm for Surface Reconstruction / F. Bernardini, J. Mittleman, H. Rushmeier [et al.] // LIDAR Widgets. — URL: https://lidarwidgets.com/samples/bpa_tvsg.pdf (accessed: 06.04.2023).
7. Kazhdan M. Poisson Surface Reconstruction / M. Kazhdan, M. Bolitho, H. Hoppe // hhoppe.com. — URL: <https://hhoppe.com/poissonrecon.pdf> (accessed: 06.04.2023).

8. The Power Crust Algorithm for Surface Reconstruction // StudyLib. — URL: <https://studylib.net/doc/5715208/the-power-crust-algorithm-for-surface-reconstruction> (accessed: 06.04.2023).
9. Photogrammetry: Step-by-Step Guide and Software Comparison // FormLabs. — URL: <https://formlabs.com/blog/photogrammetry-guide-and-software-comparison/> (accessed: 06.04.2023).
10. Классификация плотного облака точек // Geoscan. — 2022. — URL: <https://geoscan.freshdesk.com/support/solutions/articles/35000152262-Классификация-плотного-облака-точек> (дата обращения: 06.04.2023).
11. Расчет плотности облака точек // Русские блоги. — URL: <https://russianblogs.com/article/4516817067/> (дата обращения: 06.04.2023).
12. Estimating Surface Normals in a PointCloud // Point Cloud Library. — URL: https://pointclouds.org/documentation/tutorials/normal_estimation.html (accessed: 06.04.2023).
13. Уравнение Пуассона // Fandom. — URL: https://science.fandom.com/ru/wiki/Уравнение_Пуассона (дата обращения: 06.04.2023).
14. Signed Distance Fields // Render Diagrams. — 2017. — URL: <https://renderdiagrams.org/2017/12/28/signed-distance-fields/> (accessed: 06.04.2023).
15. Marching Cubes Surface Reconstruction Algorithm // ProgrammerSought. — URL: <https://www.programmersought.com/article/50924997821/> (accessed: 06.04.2023).
16. de Berg M. 3D Computer Vision: Efficient Methods and Applications / M. de Berg, O. Cheong, M. van Kreveld [et al.]. — Springer, 2008. — 380 p.
17. Stanford Bunny // Sketchfab. — URL: <https://sketchfab.com/3d-models/stanford-bunny-43f266d6cd6e4c6888b9943557528c0f> (accessed: 06.04.2023).
18. Signed cloud to mesh distance // CloudCompare. — URL: <https://www.cloudcompare.org/forum/viewtopic.php?t=1478> (accessed: 06.04.2023).
19. Ray-Casting Tutorial For Game Development And Other Purposes // permadi.com. — URL: <https://permadi.com/1996/05/ray-casting-tutorial-table-of-contents/> (accessed: 06.04.2023).
20. Open3D. — URL: <http://www.open3d.org/> (accessed: 06.04.2023).
21. Amenta N. The Power Crust / N. Amenta, S. Choi, R.K. Kolluri // UC Davis Computer Science. — URL: <https://web.cs.ucdavis.edu/~amenta/pubs/sm.pdf> (accessed: 06.04.2023).

Список литературы на английском языке / References in English

1. Wöhler Ch. 3D Computer Vision: Efficient Methods and Applications / Ch. Wöhler. — Springer London, 2013. — 382 p.
2. Thrun S. Probabilistic Robotics / S. Thrun, W. Burgard, D. Fox. — Cambridge: MIT Press, 2005. — 647 p.
3. Abramczyk H. Introduction to Laser Spectroscopy / H. Abramczyk. — Oxford: Elsevier Science & Technology, 2005. — 429 p.
4. Mezhenin A.V. Modelirovanie virtual'noj sredy v tehnologijah kibervizualizacii i virtual'nogo prisutstvija [Modeling of the virtual environment in cybervisualization and virtual presence technologies] / A.V. Mezhenin, V.V. Izvozchikova, D.I. Burlov // Kibernetika i programmirovaniye [Cybernetics and programming]. — 2019. — № 4. — P. 26-35. [in Russian]
5. Hughes J.F. Computer Graphics: Principles and Practice / J.F. Hughes, A. van Dam, M. McGuire [et al.]. — Addison-Wesley, 2013. — 1264 p.
6. Bernardini F. The Ball-Pivoting Algorithm for Surface Reconstruction / F. Bernardini, J. Mittleman, H. Rushmeier [et al.] // LIDAR Widgets. — URL: https://lidarwidgets.com/samples/bpa_tvsg.pdf (accessed: 06.04.2023).
7. Kazhdan M. Poisson Surface Reconstruction / M. Kazhdan, M. Bolitho, H. Hoppe // hhoppe.com. — URL: <https://hhoppe.com/poissonrecon.pdf> (accessed: 06.04.2023).
8. The Power Crust Algorithm for Surface Reconstruction // StudyLib. — URL: <https://studylib.net/doc/5715208/the-power-crust-algorithm-for-surface-reconstruction> (accessed: 06.04.2023).
9. Photogrammetry: Step-by-Step Guide and Software Comparison // FormLabs. — URL: <https://formlabs.com/blog/photogrammetry-guide-and-software-comparison/> (accessed: 06.04.2023).
10. Klassifikacija plotnogo oblaka toчек [Classification of a Dense Point Cloud] // Geoscan. — URL: <https://geoscan.freshdesk.com/support/solutions/articles/35000152262-Klassifikacija-plotnogo-oblaka-toчек> (accessed: 06.04.2023). [in Russian]
11. Raschet plotnosti oblaka toчек [Calculation of Point Cloud Density] // RussianBlogs. — URL: <https://russianblogs.com/article/4516817067/> (accessed: 06.04.2023). [in Russian]
12. Estimating Surface Normals in a PointCloud // Point Cloud Library. — URL: https://pointclouds.org/documentation/tutorials/normal_estimation.html (accessed: 06.04.2023).
13. Uravnenie Puassona [Poisson Equation] // Fandom. — URL: https://science.fandom.com/ru/wiki/Уравнение_Пуассона (accessed: 06.04.2023). [in Russian]
14. Signed Distance Fields // Render Diagrams. — 2017. — URL: <https://renderdiagrams.org/2017/12/28/signed-distance-fields/> (accessed: 06.04.2023).
15. Marching Cubes Surface Reconstruction Algorithm // ProgrammerSought. — URL: <https://www.programmersought.com/article/50924997821/> (accessed: 06.04.2023).
16. de Berg M. 3D Computer Vision: Efficient Methods and Applications / M. de Berg, O. Cheong, M. van Kreveld [et al.]. — Springer, 2008. — 380 p.
17. Stanford Bunny // Sketchfab. — URL: <https://sketchfab.com/3d-models/stanford-bunny-43f266d6cd6e4c6888b9943557528c0f> (accessed: 06.04.2023).

18. Signed cloud to mesh distance // CloudCompare. — URL: <https://www.cloudcompare.org/forum/viewtopic.php?t=1478> (accessed: 06.04.2023).
19. Ray-Casting Tutorial For Game Development And Other Purposes // permadi.com. — URL: <https://permadi.com/1996/05/ray-casting-tutorial-table-of-contents/> (accessed: 06.04.2023).
20. Open3D. — URL: <http://www.open3d.org/> (accessed: 06.04.2023).
21. Amenta N. The Power Crust / N. Amenta, S. Choi, R.K. Kolluri // UC Davis Computer Science. — URL: <https://web.cs.ucdavis.edu/~amenta/pubs/sm.pdf> (accessed: 06.04.2023).