

Автоматическая сегментация облаков точек на основе элементов поверхности

A. B. Velizhev, R. V. Shapovalov, D. Potapov, L. Tretiak, A. Konushin

Department of Computational Mathematics and Cybernetics

Moscow State University, Moscow, Russia

avelizhev@graphics.cs.msu.ru, shapovalov@graphics.cs.msu.ru, potapov@graphics.cs.msu.ru, tretiak@graphics.cs.msu.ru, ktosh@graphics.cs.msu.ru

Аннотация

Алгоритмы сегментации результатов лазерного сканирования, работающие с отдельными точками, чувствительны к шуму и требуют значительных вычислительных ресурсов. В последние годы были предложены алгоритмы сегментации, основанные на использовании иерархических деревьев, таких как kd- или окто- деревья. Однако, при построении таких деревьев исходное множество трехмерных точек разбивается с помощью плоскостей, поэтому на сложных поверхностях возникают разрывы. В результате точки принадлежащие непрерывной поверхности попадают в разные листы дерева, что ухудшает результаты сегментации. В данной статье мы предлагаем новую иерархическую структуру данных, названную Seg-Tree, и показываем её эффективность на примере популярного алгоритма разрастающихся регионов. Предлагаемая модификация этого алгоритма более устойчива к шуму и демонстрирует стабильные результаты сегментации при меньшей зависимости от углового порога.

Ключевые слова: лазерное сканирование, облако точек, структура данных, сегментация, элементы поверхности.

1. ВВЕДЕНИЕ

Лазерное сканирование широко используется для детального моделирования трёхмерных сцен. Важным этапом в решении задач построения параметрической модели, классификации и анализе формы объектов является сегментация облака точек. Целью сегментации является разбиение неструктурированного облака точек на не пересекающиеся подмножества точек. Все точки внутри каждого сегмента должны быть близки друг к другу в выбранном пространстве признаков. Типичная задача сегментации - выделение плоскостей в результатах лазерного сканирования, в этом случае все точки внутри каждого сегмента лежат в одной плоскости.

В данной статье мы предлагаем новую иерархическую структуру данных, названную Seg-Tree, и показываем её эффективность на примере задачи сегментации, модифицировав популярный алгоритм разрастающихся регионов [6]. Предлагаемая модификация более устойчива к шуму и демонстрирует стабильные результаты сегментации при меньшей зависимости от углового порога. Seg-Tree создается в процессе специальной процедуры пересегментации и группирует точки в компактные элементы поверхности как для гладких, так и для сильно зашумленных поверхностей. Процедура пересегментации основана на

построении R-дерева (R-Tree) [4] в сочетании с алгоритмом к-средних для разбиения узлов дерева.

Статья организована следующим образом. В разделе 2 приведен обзор существующих иерархических структур и методов автоматической сегментации трехмерных точек. Алгоритм построения Seg-Tree рассмотрен в следующем разделе. В разделе 4 предлагается улучшенная версия алгоритма разрастающихся регионов, основанная на Seg-Tree. Результаты и выводы приведены в последнем разделе.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

В данном разделе мы рассмотрим основные структуры данных для хранения и обработки точек лазерного сканирования, а также приведем обзор существующих алгоритмов сегментации трехмерных точек.

2.1 Иерархические структуры данных

Для обработки и хранения облаков точек наибольшее распространение получили окто- и kd- деревья [5]. Для таких деревьев были разработаны вычислительно эффективные методы поиска k-ближайших соседних точек [7].

В 2006 были предложены деревья элементов поверхности [2], которые позволили представить неструктурированное облако точек в виде набора элементов поверхности. Одним из главных недостатков этого дерева элементов поверхности является некорректное разбиение для некоторых типов поверхности, которое проявляется при работе со сложными и зашумленными облаками точек (например, для естественных сцен). Однако, идея элементов поверхности является на наш взгляд крайне полезной в контексте различных задач обработки облаков точек, например, задачи сегментации.

Другим подходом к построению пространственного индекса для хранения геометрических примитивов является подход на основе R-дерева, предложенный в работе [4]. Данный подход основан на построении иерархии вложенных параллелепипедов. Важным свойством R-деревьев является возможность взаимного пересечения узлов и листьев дерева. Каждый узел может содержать ограниченное число вложенных узлов дерева, а точки хранятся только в листьях дерева. Таким образом, R-деревья представляют собой вариант пространственного индекса, в отличие от окто-деревьев добавление и удаление точек является быстрой операцией.

В случае, когда число элементов в узле R-дерева превышает заданное, происходит подразбиение узла. Гутман предложил три разных алгоритма разбиения, которые пытаются минимизировать сумму площадей новых узлов (для плоского дерева). Однако, этот критерий не всегда является

удовлетворительным, поэтому в работе [3] было предложено рассматривать задачу разбиения как классическую задачу кластеризации данных. Авторы использовали широко известный алгоритм k-средних для разбиения узлов, назвав полученный пространственный индекс cR-деревом. Применяв такое разбиение к облаку точек, мы получили наборы из групп точек близкие к элементам поверхности. Что позволило использовать данную структуру данных для задачи сегментации (подробнее см. раздел 3).

2.2 Сегментация облаков трехмерных точек

Задача сегментации трехмерных облаков точек является популярной областью исследований в последние годы. В данном обзоре мы сконцентрировали свое внимание на полностью автоматических алгоритмах сегментации. Более детальный обзор алгоритмов сегментации можно найти в работе [1].

Современные лазерные сканеры позволяют получить высокую детальность на поверхности объекта съемки. Полученные наборы точек затем сегментируются на основе анализа геометрических свойств точек. Например, в статье [6] в качестве признаков использовались направление нормали и величина отклонения точки от локальной плоскости. Авторы выполняли сегментацию с помощью алгоритма разрастающихся регионов всего с несколькими порогами, задаваемыми вручную. Предложенный метод позволял получить корректную сегментацию сглаженных поверхностей. Применяв данный алгоритм, к реальным данным мы столкнулись с проблемой возникновения большого числа сегментов, состоящих всего из нескольких точек. Данный эффект проявлялся как на плоских поверхностях при наличии шума, так и на сложных поверхностях (например, барельеф). На практике часто бывает сложно найти баланс между недо- и пересегментацией за счет ручного изменения порогов. Тем не менее, данный алгоритм позволяет выполнять сегментацию произвольных типов поверхности, что является несомненным достоинством этого алгоритма. Ряд работ по автоматической сегментации облаков точек используют kd- и окто- деревья в качестве основы (Wang et al., 2004, Bucksch et al., 2006). В работе (Biosca et al., 2008) предложили использовать для сегментации облака точек алгоритм нечёткой кластеризации k-средних. Однако, указанная методика применима только для сегментации плоскостей.

Подводя итоги, можно выделить несколько главных особенностей текущих алгоритмов сегментации. Во-первых, большинство из ранее предложенных методов выполняют сегментацию только плоских участков. Во-вторых, часть методов используют для сегментации структуру иерархического дерева, в котором хранятся точки. Безусловным достоинством данного подхода является возможность вычислительно эффективной реализации алгоритма сегментации для больших массивов исходных данных. Однако, неудачное разбиение пространства точек, используемыми структурами данных, автоматически переходят и на результаты сегментации. В-третьих, применение методов кластеризации позволяет получить сегментацию устойчивую к шуму. Однако, применение таких методов ограничивается плоскими поверхностями.

3. SEG-TREE

Исследование структур организации данных мы начали с реализации R-дерева [4] и обнаружили, что оно не удовлетворяет нашим требованиям. Гутман успешно решил задачу построения пространственного индекса (R-дерева) активно используются в различных CAD-системах), однако для задачи пересегментации облака точек такой подход не работает и также приводит к разрывам поверхности. У базового алгоритма R-дерева могут быть изменены следующие свойства:

- выбор листа для добавления точки;
- алгоритм разбиения;
- используемая метрика пространства.

Рассмотрим подробнее эти свойства.

3.1 Выбор листа

Главный критерий качества дерева по Гутману является сумма объемов всех листов. Поэтому точка добавляется в тот лист, объем которого увеличится меньше. Иными словами, для добавления точки будет выбран лист с наименьшим значением $minDist$ (Рис. 1)

Такой подход неприемлем для облаков точек, по причине того, что точки часто организованы вдоль отдельных полос, что вызвано техническими особенностями процесса сканирования. В результате в листы дерева попадают точки, принадлежащие одной полосе, что противоречит идее элементов поверхности. Однако, объем таких листов оказывается наименьшим. Решением проблемы может стать рандомизированное добавление точек в дерево, что делает процесс добавления точек не последовательным и на практике не всегда устраняет проблему.

Поэтому мы предлагаем наш собственный метод выбора листа для добавления точек. Для получения листов, точки в которых принадлежат элементам поверхности, необходимо выбирать лист, центр масс которого находится ближе всего к данной точке. Проверка всех листов дерева требует больших вычислительных затрат, поэтому мы использовали приближенный алгоритм, который сохранял баланс между качеством формы листа и вычислительной сложностью.

Алгоритм состоял в последовательном спуске от корня дерева к его листам. На каждом уровне дерева выбирались некоторые узлы дерева, затем на следующем уровне иерархии выбирались некоторые потомки выбранных ранее узлов. Наконец, на последнем уровне выбирался тот лист, центр масс которого оказывался ближе всего к добавляемой точке. Критерий выбора узлов дерева на промежуточных уровнях состоял в следующем. Если точка попадала внутрь ограничивающего параллелепипеда, то узел добавлялся в список. Для улучшения качества конечного разбиения в список также добавлялись несколько узлов с наименьшими значениями $Dist$ и/или $minDist$. $Dist$ задается расстоянием до центра масс узла, который определяется как среднее между центрами параллелепипедов ближайших потомков узла. Выбор количества узлов для добавления с список позволяет находить баланс между качеством конечной пересегментации и производительностью.

В ходе экспериментов мы остановились на следующем эмпирическом подходе. На каждом уровне мы составляли список ближайших потомков всех выбранных на предыдущем уровне узлов. Сначала в новый список добавлялись все узлы,

у которых $minDist == 0$ (точка оказывается внутри узла). Если таких узлов не оказывалось, то в список добавлялся узел с минимальным значением $minDist$. Затем мы вычисляли $pruneDist$ – расстояние до ближайшего центра масс отобранных ранее узлов. Наконец, в список добавлялись все узлы, $minDist$ которых оказывался меньше, чем $pruneDist$.

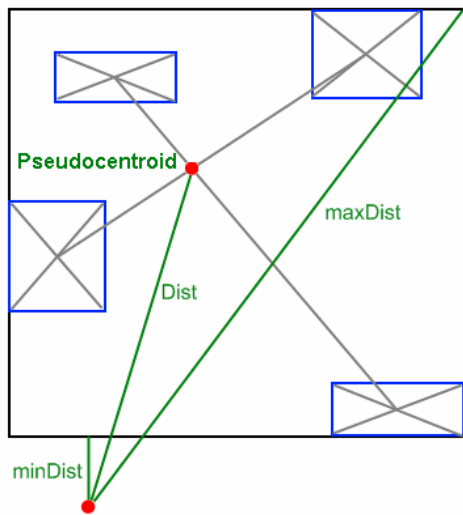


Рисунок 1: Иллюстрация понятий $Dist$, $MinDist$ и $MaxDist$ на примере узла дерева и 4 вложенных потомков.

3.2 Алгоритм разбиения

Гутман предложил три разных алгоритма разбиения, но каждый из них пытался минимизировать сумму объемов новых узлов. Альтернативный подход [3] был основан на рассмотрении задачи разбиения как классической задачи кластеризации данных и использовал для разбиения алгоритм k -средних. Экспериментальные результаты показывают, что использование алгоритма k -средних для подразбиений точек (и даже узлов) позволяет получить листы дерева, являющиеся удачным приближением к элементам поверхности.

3.3 Метрика пространства

Как отмечалось ранее, прямоугольная форма узлов дерева оказывается практичнее, чем сферическая. Однако, использование манхэттенской метрики (расстояния в пространстве L^1) позволяет одновременно сочетать достоинства сферических M -деревьев и прямоугольных R -деревьев.

3.4 Seg-Tree

Наконец, мы решили выбрать следующие параметры R -дерева, соответствующие нашей структуре данных Seg-Tree. Для добавления точек мы используем адаптивный алгоритм выбора листов дерева, описанный в разделе 3.1, который одновременно учитывает $minDist$ и $Dist$. Мы использовали алгоритм кластеризации k -средних для подразбиения узлов и листов дерева. Мы допускали от 1 до 64 точек в листах и от 4 до 16 элементов в узлах дерева. Мы использовали манхэттенскую метрику внутри процедур добавления и разбиения. На рисунке 2 представлен результат построения Seg-Tree.

4. СЕГМЕНТАЦИЯ

Рассмотрим развитие алгоритма разрастающихся регионов [6] адаптированного для использования Seg-Tree.. Процедура сегментации выполняется в два этапа. Сначала выполняется грубая сегментация, в которой алгоритм [6] применяется не к отдельным точкам, а к целым листьям Seg-Tree. Для каждого листа независимо оценивается направление нормали только по внутренним точкам листа. Результат сегментации на этом шаге приведен на рисунке 3б.

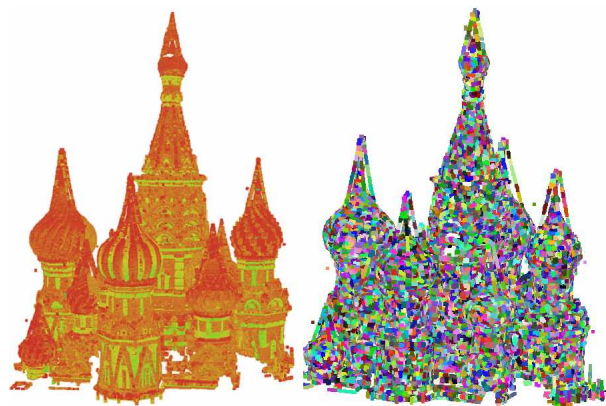


Рисунок 2: Структура Seg-Tree.

Элементы поверхности, расположенные на границах сегментов, могут содержать одновременно точки, принадлежащие разным поверхностям, например, на стыке двух плоскостей. Поэтому на втором этапе происходит уточнение границ сегментов уже на уровне отдельных точек. Для выделения границ выявляются все листы дерева, среди k -ближайших соседей которых есть листы, принадлежащие другому сегменту. Для всех точек, содержащихся в граничных листах, применяется традиционный алгоритм разрастающихся регионов, работающий уже на уровне отдельных точек. Небольшое различие в алгоритме состоит в том, что среди k -ближайших соседних точек граничных листов могут оказаться уже отсегментированные точки, лежащие в неграничных листах. В случае, когда в сегмент добавляется такая отсегментированная точка, мы выполняем слияние сегментов (Рис. 3с).. Такой подход позволяет получить корректную сегментацию на границах сегментов.

5. РЕЗУЛЬТАТЫ

Мы выполнили сравнение предлагаемого алгоритма с вариантом the [6]. Сначала мы выполнили построение Seg-Tree непосредственно из облака точек и получили пересегментированное иерархическое представление в виде элементов поверхности. Затем мы выполнили полностью автоматическую сегментацию с помощью алгоритма, описанного в предыдущем разделе, и провели сравнение результатов. Предлагаемый алгоритм обычно производит аналогичную сегментацию, однако, число небольших сегментов оказывается заметно меньшим. Мы также сравнивали результаты при различных значениях углового порога и получали более стабильные результаты в отличие от существующего алгоритма (Рис. 4.).

Другим способом снижения числа маленьких сегментов является увеличение числа ближайших точек при вычислении направлений нормали, однако, это одновременно приводит к сглаживанию поверхности точек и множество различных сегментов сливаются.

6. ВЫВОДЫ

В данной работе предложена новая схема организации трехмерных точечных данных в специальное иерархическое дерево, листья которого приближают элементарные участки поверхности. В случае некорректной поверхности, например, точки на реальных деревьях, листья дерева представляют собой локально связанные группы точек.

Затем мы применили Seg-Tree для выполнения сегментации с помощью алгоритма разрастающихся регионов. Предлагаемый метод сегментации объединяет в себе свойства части предложенных ранее методов:

- использование структуры организации данных для сегментации;
- применение методов кластеризации;
- использование метода разрастающихся регионов, позволяющего сегментировать разные типы поверхностей.

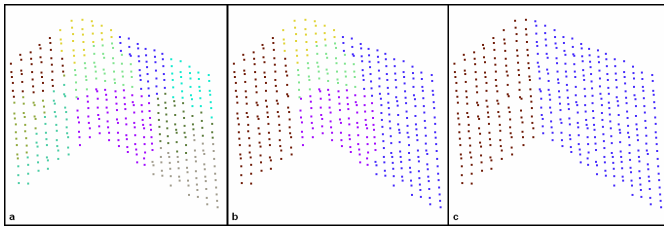


Рисунок 3: Сегментация с помощью листьев Seg-Tree.

а) Исходные листья Seg-Tree, б) Результат сегментации только по листьям дерева в) Результаты уточнения сегментации на уровне отдельных точек.

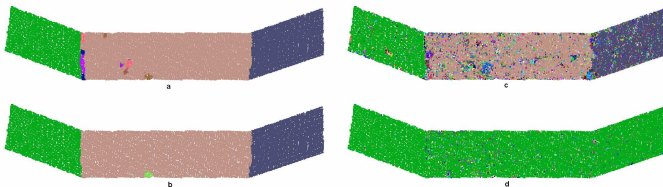


Рисунок 4: Сравнение результатов сегментации при различных угловых порогах.

а), б) – сегментация с помощью листьев Seg-tree, угловой порог 5° и 10° соответственно. в), д), -сегментация только по отдельным точкам при аналогичных значениях углового порога.

Предложенный подход менее чувствителен к шуму в исходных данных, чем [6], за счет анализа элементов поверхности, а не отдельных точек. Традиционный алгоритм производит большое число отдельных маленьких сегментов, а для сегментации на основе Seg-Tree этот эффект проявляется значительно реже. Наконец, предложенный алгоритм позволяет получить более стабильные результаты при различных значениях углового порога.

7. ЛИТЕРАТУРА

- [1] Biosca, J., Lerma, J., 2008. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 63(1), Pages 84-98
- [2] Boubekeur, T., Heidrich, W., Granier, X., Schlick, C., 2006. *Volume-Surface Trees*. Computer Graphics Forum (Proceedings of EUROGRAPHICS 2006), 25(3), pp. 399-406
- [3] Brakatsoulas S., Pfoser D., Theodoridis Y., 2002. *Revisiting R-tree construction principles*. Proceedings of the 6th East European Conference on Advances in Databases and Information Systems, pp. 149-162.
- [4] Guttman A., 1984. *R-trees: a dynamic index structure for spatial searching*. Proceedings of the 1984 ACM SIGMOD international conference on Management of data, pp. 47-57.
- [5] Jackins, C., Tanimoto S., 1983. *Quad-trees, oct-trees, and k-trees. A generalized approach to recursive decomposition of Euclidean space*, IEEE Transactions, Pattern Analysis and Machine Intelligence, vol. PAMI-5, pp. 533-539
- [6] Rabbani T., van den Heuvel, F., Vosselmann, G., 2006. *Segmentation of point clouds using smoothness constraint*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(5), pp. 248-253
- [7] Sankaranarayanan, J., Samet, H., Varshney, A., 2007. *A fast all nearest neighbor algorithm for applications involving large point-clouds*. Computers and Graphics, 31(2), pp. 157-174

8. БЛАГОДАРНОСТИ

Работа выполнена при поддержке фонда «Научный потенциал», грант №182.

Robust LIDAR data segmentation using compact point clusters

Abstract

Segmentation algorithms, which work with individual points, are sensitive to measurement noise and are computational-intensive. In recent years several segmentation algorithms based on hierarchical structure, like octree or kd-tree, were proposed. However, such plane-based partitioning procedures cannot produce correct results for various complex scenes with holes and occlusions. Continuous surfaces are often split and neighbor points are inserted into different tree nodes that degrade the segmentation result. In this paper we introduce a new hierarchical tree, named Seg-Tree, and show its effectiveness by applying to the popular segmentation algorithm region grow. Our version of this algorithm is robust to the noise and produce stable segmentation with lesser dependency of angular threshold. Seg-Tree is created using special over-segmentation procedure that produces compact leaves both for smooth and heavily noisy point sample surfaces. The over-segmentation procedure is performed using R-Tree construction with k-means clustering technique.