КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ
COMPUTER SIMULATION HISTORY

УДК 004.421: 004.925.82 **DOI: 10.18413/2518-1092-2018-3-4-0-1**

Гасилов А.В.
Фролов А.И.

ЭВРИСТИЧЕСКИЙ АЛГОРИТМ СЕГМЕНТАЦИИ ОБЛАКА ТОЧЕК

Орловский государственный университет имени И.С. Тургенева, ул. Комсомольская, д. 95, г. Орел, 302026, Россия

e-mail: gasilov.av@ya.ru

Аннотация
В данной статье рассматривается эвристический алгоритм сегментации облака точек, описывающего предмет интерьера, с целью получения сегментации, близкой к разбиению объекта на функциональные элементы. Данный алгоритм применяется как часть метода улучшения результатов трехмерной реконструкции на основе известной структуры объекта для структуризации облака точек, получаемого на одном из этапов метода. Алгоритм заключается в рекурсивном разбиении облака точек на две части, вплоть до достижения заданного критерия остановки разбиения. В статье рассмотрена структура алгоритма, проанализирована его вычислительная сложность, проведены примеры результатов работы, определено направление дальнейших исследований.
Ключевые слова: облако точек; сегментация.

UDC 004.421: 004.925.82

Gasilov A.V.
Frolov A.I.

HEURISTIC ALGORITHM OF POINT CLOUD SEGMENTATION

Oryol State University named after I.S. Turgenev, 95 Komsomolskaya St., Orel, 302026, Russia

e-mail: gasilov.av@ya.ru

Abstract
The algorithm of point cloud segmentation is proposed in the article. This algorithm aimed to get segmentation similar to segmentation of object by his functional elements in field of interior design. The given algorithm is used as a part of method of dense 3D reconstruction enhancing for point cloud structuration at some point. Main idea of proposed algorithm is recursive division of parent cloud by two parts until exit conditions are met. Algorithm structure is reviewed, computing complexity is analyzed. The field for further investigations of given problems and their solution is suggested.
Keywords: point cloud; segmentation.

ВВЕДЕНИЕ

Построение трехмерных моделей на основе изображений – одна из задач компьютерного зрения, заключающаяся в распознавании образов в поданных на вход алгоритму изображениях и построении на их основе трехмерных моделей.

Задача трехмерной реконструкции уже давно решена в общем виде [1]. Однако, есть пути для улучшения качества существующих технологий по таким параметрам, как точность и полнота реконструкции. Одним из таких путей является применение априорно известной структуры реконструируемого объекта для улучшения полноты реконструкции. В рамках конкретной предметной области, в данном случае – дизайна интерьера, возможен следующий подход:

– Описание оператором структуры объекта;
– Проведение плотной трехмерной реконструкции по снимкам объекта с результатом в виде облака точек;
– Автоматическая сегментация полученного облака точек и получение структуры облака точек в таком же виде, как и структура, описанная оператором;
– Сравнение двух структур и выполнение операций для улучшения полноты реконструкции на основе результатов сравнения.

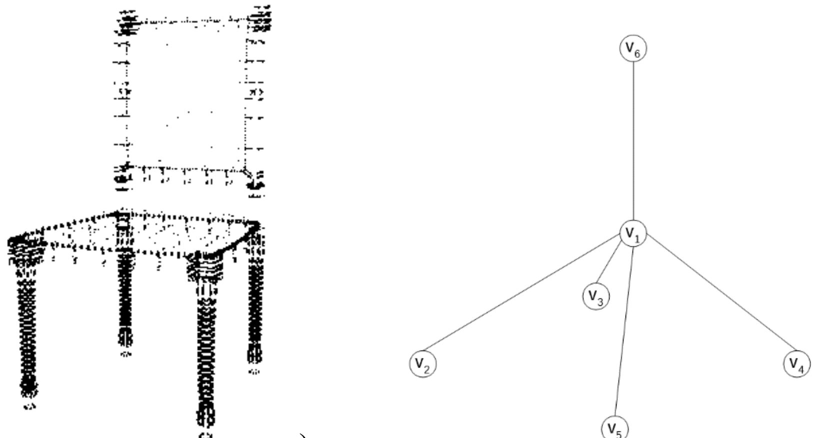
При этом структура объекта описывается оператором на уровне «стул состоит из сиденья, 4 ножек, спинки, имеющих заданные относительные размеры и положения». Таким образом, одним из этапов данного процесса является сегментация реконструированного облака точек с целью получения его структуры в виде похожем на структуру, задаваемую оператором.

В данной статье предлагается эвристический алгоритм решения данной задачи, рассматриваются особенности его реализации, проводится его анализ, строятся предположения о возможных дальнейших его улучшениях.

1. ПОСТАНОВКА ЗАДАЧИ

Задача заключается в приведении входной информации в виде облака точек к виду графа структуры объекта, в котором каждая вершина соответствует некоторому элементу логически цельному элементу объекта, описываемому облаком точек, а ребра будут означать факт соседства элементов графа в пространстве.

Например, разбиение стула, изображенного на рисунке 1а может представлять деление на такие элементы как: «сиденье» - вершина v_1, «ножки» - вершины v_2, v_3, v_4, v_5 и «спинка» - вершина v_6 (рисунок 1б).



а) б)

Рис. 1. а – пример входных данных, б – пример требуемых выходных данных

Fig. 1. a – example input data, b – example of required output sample data

Таким образом можно выделить две подзадачи: получение вершин графа и определение наличия ребер между полученными вершинами.

2. ПОЛУЧЕНИЕ ВЕРШИН ГРАФА

Для получения вершин графа предлагается сегментировать облако точек и сопоставить вершину графа каждому полученному сегменту.

В качестве отправной точки было решено применить подход, рассмотренный в [2], заключающийся в аппроксимации облака точек ограничивающим прямоугольным параллелепипедом минимального объема (MVBB)[3] с последующим рекурсивным разбиением «сверху-вниз» полученного объема на две части с минимизацией целевого функционала на каждом шаге.

Однако, при анализе данного подхода выяснилось, что существуют проблемы, препятствующие его использованию для решения данной задачи. Изначальная цель разработки [2] была задана как «быстрое получение более оптимальной модели для использования ее в алгоритмах координации работы роботизированных манипуляторов». Наиболее важная проблема, которая следует из цели – данный алгоритм оптимизирует разбиение объема с целью аппроксимации его формы в общих чертах, в то время как в данной задаче необходимо сегментировать имеющийся объем на основе семантики его частей.

Коротко алгоритм [2] можно описать следующим образом:

- Вычислить MVBB облака;
- Вычислить наиболее оптимальный разрез полученного MVBB:
    - Спроецировать все точки объема на каждую из сторон MVBB;
    - Провести разрез параллельно границам MVBB через каждую спроецированную точку;
    - Вычислить описывающие прямоугольники для двух наборов точек на плоскости;
    - Из всех возможных разбиений выбрать такое, для которого сумма площадей двух описывающих прямоугольников по отношению к площади исходного описывающего прямоугольника - минимальна.
- Разбить входное облако точек на два по вычисленному разрезу и, если критерий остановки еще не достигнут, запустить алгоритм рекурсивно для обоих потомков.

При этом, критерием остановки является выполнение одного из двух условий:

1) Отношение нового объема после разбиения к объему до разбиения достигло порогового значения:

$$\Theta = \frac{V(C_1)+V(C_2)+V(A^{\backslash P})}{V(P)+V(A^{\backslash P})},$$

где Ɵ – проверяемый параметр, рекомендуемое значение порога: 0.80-0.95;

V(x) – функция, возвращающая объем MVBB облака точек;

P – исходное облако точек, подлежащее разбиению;

C_1, C_2 – облака точек, части полученного разбиения;

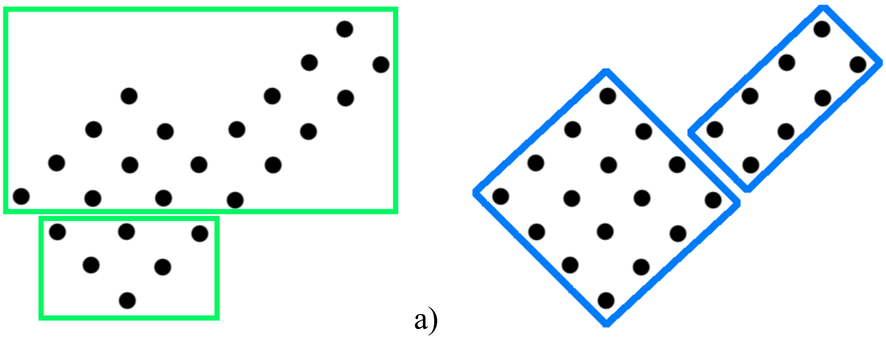
$A^{\backslash P}$ – остальные листья дерева разбиения на текущий момент, за исключением вершины P.

Что означает, что прирост уменьшения объема с последнего разбиения оказался незначительным

2) Количество точек в одной из частей разбиения оказалось меньше порогового значения.

При анализе алгоритма было выдвинуто предложение, что с некоторыми модификациями он может быть применен к данной задаче, при учете особенностей предметной области, главной из которых является то, что предметы интерьера зачастую имеют большое количество прямых углов и прямых линий и многие части объектов могут быть аппроксимированы параллелепипедом не захватывая объем других частей.

Такой предлагаемой модификацией является улучшение способа вычисления оптимального разреза. В текущей вариации разбиение происходит строго вдоль трех сторон MVBB, Таким образом, в приведенном примере (рисунок 2) он найдет далекий от оптимального результат.



а) б)

Рис. 2. а – разрез согласно исходному алгоритму, б – разрез согласно оптимизированному алгоритму

Fig. 2. a – sectional view according to the original algorithm, b – section according to the optimized algorithm

Можно модифицировать алгоритм поиск разреза добавив к каждому просматриваемому разрезу поворот. Таким образом значительно увеличивается качество находимых разрезов, а с учетом специфики предметной области – получаемые разрезы с большой вероятностью будут оптимальны. Тем не менее, подобное улучшение приведет к повышению вычислительной сложности с дополнительным множителем $O(2\pi/k)$, где k – шаг поворота разреза.

Однако, в постановке [2] важным критерием стоит возможность выполнения в реальном времени, в то время как в поставленной задаче такой критерий не является обязательным, в следствие чего появляется возможность повышать вычислительную сложность алгоритма в некоторых разумных пределах.

Кроме того, как вспомогательное средство повышение производительности, ко входным данным применяется фильтрация на основе дискретизации воксельным объемом.

В результате работы модифицированного алгоритма получается дерево разбиения общего объема на несколько подобъемов. Однако, нас интересуют только листья полученного дерева разбиения, поэтому эти листья и будут вершинами результирующего графа структуры объекта.

Исходный алгоритм [2] состоит из таких этапов как построение MVBB, которое имеет вычислительную сложность $O(n \log n + n/\varepsilon^3)$, где n – количество вершин в облаке точек, $0 < \varepsilon \leq 1$ – некоторый параметр алгоритма, а так же вычисления оптимального разреза за $\mathrm{O}(n \log n)$ например, с помощью дерева отрезков[4], что в сумме дает оценку $\mathrm{O}(n \log n + n/\varepsilon^3)$. Модифицированный же алгоритм за счет дополнительной операции будет иметь сложность $\mathrm{O}(n \log n \cdot 2\pi/k + n/\varepsilon^3)$.

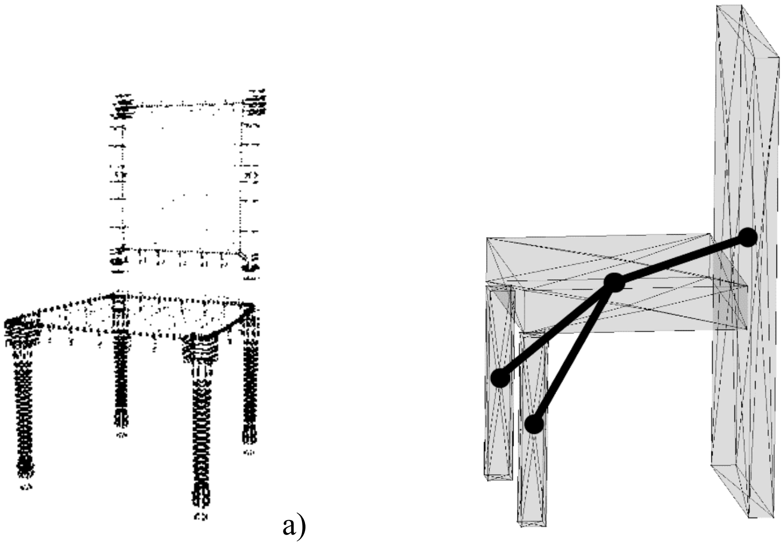
3. НАХОЖДЕНИЕ РЕБЕР ГРАФА

Исходя из определения ребра как «факта соседства описываемых объектов в пространстве», можно получать ребра путем анализа декартова расстояния между описывающими кубами соответствующих вершин: в случае если данное расстояние равно нулю, либо значительно меньше некоторого порогового значения (например, зависящего от размера описывающих кубов вершин), то считается что ребро между двумя данными вершинами присутствует в графе, иначе – нет.

Задача нахождения расстояния между двумя параллелепипедами решается аналитически и имеет сложность $O(1)$ для одной пары параллелепипедов. Так как необходимо проверить существование каждого ребра в графе, вычислительная сложность данного этапа достигает $O(m^2)$, где m – количество вершин полученного графа.

4. РЕЗУЛЬТАТЫ РАБОТЫ АЛГОРИТМА

Пример работы предложенного алгоритма представлен на рисунке 3.



а) б)

Рис. 3. пример работы алгоритма: а – исходные данные, б – результат работы алгоритма
Fig. 3. example of the algorithm: a – the original data, b – the result of the algorithm

Как можно заметить, в результате, в данном примере спинка стула и задние его ножки оказались одним единым объектом. Это можно объяснить тем фактом, что на каждом этапе каждый объем бьется ровно на две части, не учитывая тот факт, что одна из частей может быть заполнена у краев и абсолютно не иметь точек в центре объема. Решение данной проблемы является одним из возможных направлений дальнейших исследований.

В виду того, что в рамках задачи $m \ll n$, в оценке вычислительной сложности алгоритма в целом, параметром m можно пренебречь. Таким образом, финальная оценка сложности алгоритма составляет $\mathrm{O}(n\ \log n \cdot 2\pi/k + n/\varepsilon^3)$.

ВЫВОДЫ

В ходе данной статьи был рассмотрен эвристический алгоритм сегментации облака точек в рамках заданной предметной области. Рассмотрен возможный подходы к реализации поставленных подзадач. На основе проведенного анализа исходного алгоритма был предложен и реализован способ его улучшения. Перспективным направлением исследования является поиск других возможных путей улучшения качестве сегментации, например, учет возможности разбиения элемента на три различных подобъема.

Список литературы

1. Bradski G.R., Kaehler A. Learning OpenCV. – O'Reilly Media, 2008. – 556 p.
2. Huebner K., Ruthotto S., Kragic D. Minimum volume bounding box decomposition for shape approximation in robot grasping / In IEEE International Conference on Robotics and Automation, 2008. – P. 1628-1633.
3. Barequet G. and Har-Peled S. Efficiently. Approximating the minimum-volume bounding box of a point set in three dimensions. – J. Algorithms, 2001. – Vol. 38(1). – P. 91-109
4. Алгоритм построения дерева отрезков [Электронный ресурс]. – URL: http://e-maxx.ru/algo/segment_tree (Дата обращения 05.09.2018)

References

1. Bradski G.R., Kaehler A. Learning OpenCV. – O'Reilly Media, 2008. – 556 p.
2. Huebner K., Ruthotto S., Kragic D. Minimum volume bounding box decomposition for shape approximation in robot grasping / In IEEE International Conference on Robotics and Automation, 2008. – P. 1628-1633.

3. Barequet G. and Har-Peled S. Efficiently. Approximating the minimum-volume bounding box of a point set in three dimensions. – J. Algorithms, 2001. – Vol. 38(1). – P.91-109
4. Algorithm for constructing a tree of segments [Electronic resource]. – URL: http:// e-maxx.ru/algo/segment_tree (date access 05.09.2018)

Гасилов Артур Владимирович, аспирант
Фролов Алексей Иванович, кандидат технических наук, доцент, зав. кафедрой программной инженерии

Gasilov Artur Vladimirovich, graduate student
Frolov Alexey Ivanovich, Candidate of Technical Sciences, Associate Professor, Head Department of Software Engineering