

УДК 004.92

Препроцессинг облаков точек в задачах анализа результатов лазерного сканирования объектов капитального строительства

Петросян Артур Владимирович

petrosyanav@student.bmstu.ru

Бурков Павел Викторович

burkovpv@bmstu.ru

МГТУ им. Н.Э. Баумана, Москва, Россия

В статье представлен алгоритм для обработки облаков точек. Рассмотрено применение лазерного сканирования и его результатов в рамках строительного контроля как одного из этапов жизненного цикла объекта капитального строительства. Приведена математическая постановка решения задачи предварительной обработки для будущей валидации объектов капитального строительства в терминах информационного моделирования.

Ключевые слова: объект капитального строительства, жизненный цикл, лазерное сканирование, облако точек, полигональное моделирование

Введение. Лазерное сканирование приобрело актуальность в связи с его использованием в строительной сфере. Помимо проведения сканирования необходима предварительная обработка его результатов для дальнейшего анализа и решения различных задач в рамках строительного контроля.

Валидация объектов капитального строительства на этапе строительного-монтажных работ. Объект капитального строительства требует контроля качества, как и любое изделие, и необходим для соблюдения требований рабочей документации, для успешного ввода в эксплуатацию в достаточно короткие сроки и выявления точных ошибок в процессе строительного-монтажных работ. В соответствии со сводом правил Минстроя РФ, строительный контроль включает в себя проверки [1]: соответствия выполняемых работ при строительстве объектов капитального строительства рабочей документации; требований технических регламентов в целях обеспечения безопасности зданий и сооружений; результатов инженерных изысканий; требований к строительству, реконструкции объекта капитального строительства; разрешений на использование земельных участков.

В качестве измерительных инструментов для проверки, например, конструктивных объемов «допускается применение методов наземного, воздушного лазерного сканирования, аэрофотосъемки, батиметрической съемки с целью мониторинга хода строительных работ».

Одной из целей лазерного сканирования является получение модели пространства, окружающего сканер. Результатом сканирования является облако точек — набор точек в виде координат. Система координат и ее параметры задаются в поставляемом программном обеспечении сканера. Более профессиональные сканеры могут также назначить цвет точки, интенсивность и дру-

гие параметры. На рис. 1 представлена принципиальная схема работы лазерного сканера.



Рис. 1. Принципиальная схема работы лазерного сканера

Под валидацией понимается соответствие цифровой информационной модели (ЦИМ) объекта капитального строительства (ОКС) рабочей документации на основе результатов лазерного сканирования. Сама задача разбивается на подзадачи [2]: проведение лазерного сканирования ОКС; совмещение полученного облака точек с ЦИМ ОКС; вычисление отклонений между точками из облака и моделью; анализ результатов и определения соответствия.

Пример выполнения совмещения точек облака и в специализированной системе автоматизированного проектирования (САПР) совмещение точек облака на модель представлен на рис 2.

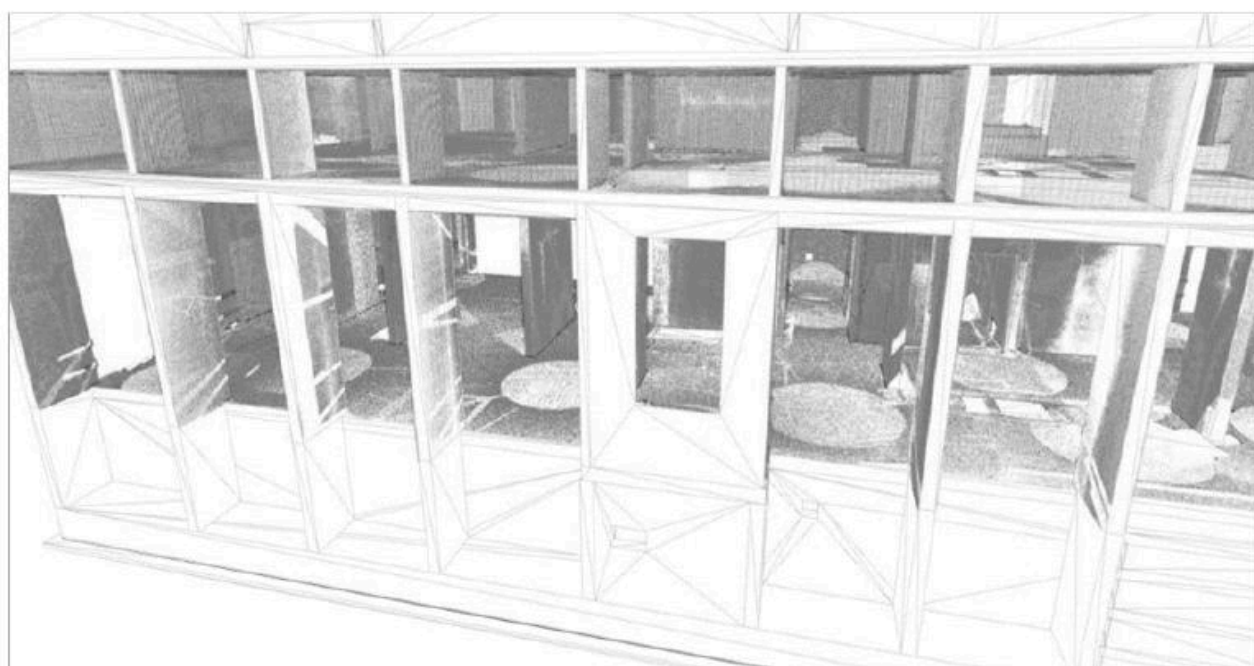


Рис. 2. Результат наложения точек на модель

Проблемы, возникающие перед валидацией. Лазерное сканирование — это наиболее точный подход для выявления конструктивных объемов, однако возникают сложности при недостатке полученной информации. Такой недостаток можно получить при сканировании металлических труб (светоотражающих поверхностей), как видно на рисунке ниже. Лазерный луч не возвращается и данные для анализа потеряны, из-за чего необходимо восстанавливать их для получения более точного результата, что видно на рис. 3.

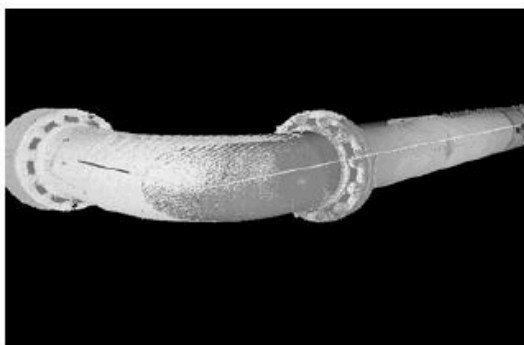


Рис. 3. Результат лазерного сканирования металлической трубы

Такую же проблему можно получить при воздушном сканировании. Несмотря на быстрое время сканирование, ввиду электромагнитных и резких воздействий данные неточные — возникают значительные шумы требующие обработки.

Другой типовой сценарий — при загрузке ЦИМ и облака точек в САПР локальная система координат каждого из объектов разная, как представлено на рис. 4, здесь также требуется предобработка.

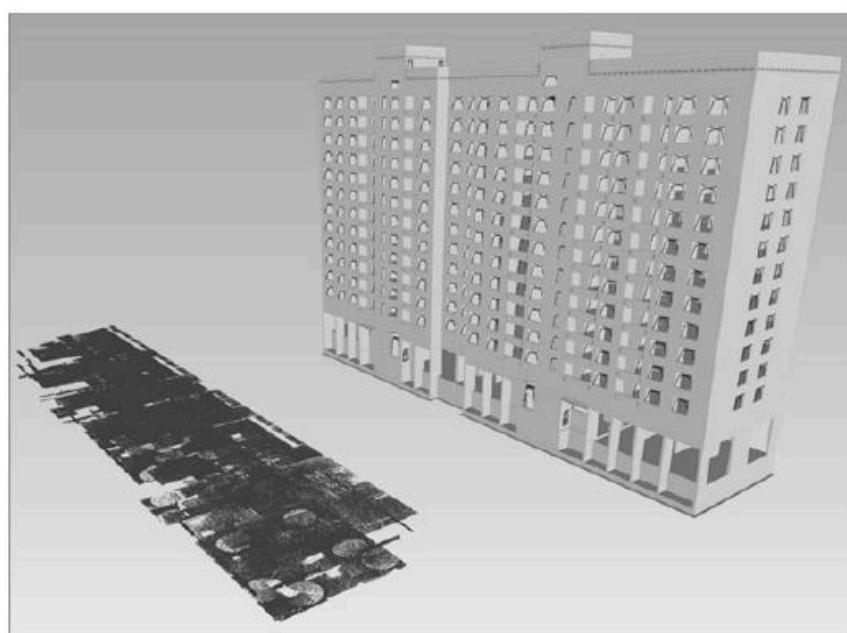


Рис. 4. Неверное сопоставление система координат облака точек этажа ОКС и всей ЦИМ из-за разных СК

В данной статье рассматривается один из вариантов решения проблемы несовместимости систем координат ЦИМ и облака точек.

Постановка задачи. Цель данной работы — определение значимых плоских граней на основе результатов лазерного сканирования для проведения валидации ОКС. Для достижения этой цели необходимо решить следующие задачи:

- проведение анализа существующих алгоритмов определения значимых плоских граней в облаках точек;
- разработка следующих алгоритмов: группировка смежных полигонов с близкими нормальными, сокращение числа неразмеченных точек вблизи границ обнаруженных плоских граней;
- проведение тестирования моделей на реальных наборах данных.

Формализация общей постановки задачи. Дано: облако точек S . Найти: такие подмножества облаков точек $S_1, S_2, \dots, S_n$, каждое из которых может быть заключено в параллелепипед достаточно малой глубины d_i .

Описание алгоритма. Разберем основные шаги алгоритма. Параллелепипед минимальной глубины d будет называться «мета-плоскостью» — Metaplane. На рис. 5 представление блок-схема алгоритма.

Для построения полигональной сетки используется триангуляция — Advancing Front Surface Reconstruction (на основе триангуляции Делоне) [3]. Для группировки полигонов за основу взят алгоритм Region Growing [4] с учетом весов и двух глобальных условий для формирования группы полигонов: количество полигонов в группе не менее $n \in N$, угол между нормальными полигонов не превышает γ градусов.

В качестве весового коэффициента при сравнении нормалей полигонов считается $w_i = S_i$, где i — порядковый номер полигона. Шаги алгоритма:

- инициализировать список свободных (неразмеченных) полигонов;
- выбрать средний по позиции полигон в списке неразмеченных полигонов;
- выполнить обход графа полигонов в ширину с условием близости углов нормалей не более чем на γ градусов и с учетом веса w_i ;
- если количество полигонов в группе не менее чем n — добавить новую группу в список всех найденных групп, иначе — расформировать группу для дальнейшего рассмотрения;
- исключить сгруппированные полигоны из списка всех неразмеченных полигонов;
- повторить шаги 1–5 до минимально возможного числа неразмеченных полигонов.

Важным отличием от стандартных алгоритмов является сокращение неразмеченных точек — точек, не принадлежащих ни одному подмножеству облака точек. Для их нахождения используется список неразмеченных полигонов, в основе алгоритма лежит метод наименьших квадратов. Шаги алгоритма:

- сформировать плоскости, с минимальной суммой квадратов отклонений от точек в каждой группе;
- вычислить среднеквадратическое отклонение точек в соответствующей плоскости;
- найти неразмеченные точки из списка неразмеченных полигонов:
 - для каждой неразмеченной точки найти плоскость, отклонение от которой будет меньше среднеквадратического отклонения ее точек;
 - включить точку в плоскость;
- провести повторное построение полигональной сетки для каждого полученного итогового подмножества облака точек.

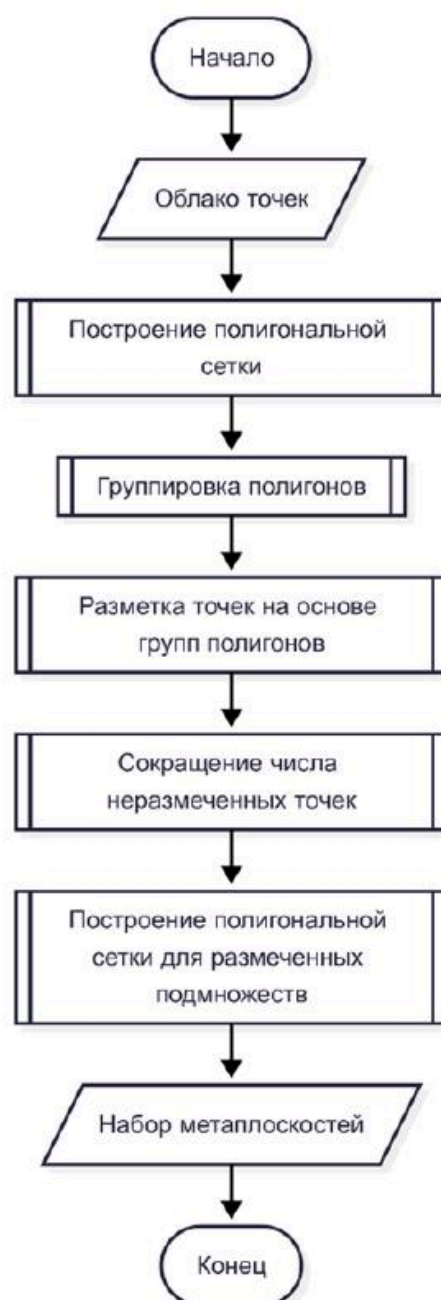


Рис. 5. Блок-схема алгоритма

Реализация алгоритма. Разработанная программа выполнена с использованием языка программирования C++ и библиотеки CGAL, принимает на вход три аргумента: путь к облаку точек; выходной путь к трехмерной модели с найденными группами полигонов; выходной путь к трехмерной модели с сокращенным количеством неразмеченных точек.

Был проведен запуск на тестовом наборе на рис. 6.

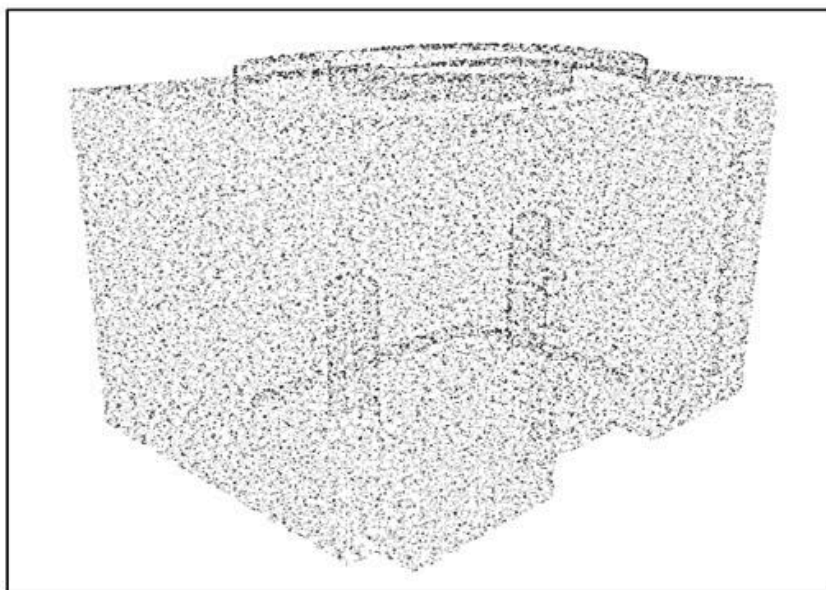


Рис. 6. Тестовое облако точек

Результат построения полигональной сетки и группировки полигонов оставляет неразмеченные полигоны или даже точки, поскольку заданный угол недостаточен для явного определения принадлежности к группам полигонов (полностью черные) как видно на рис. 7.

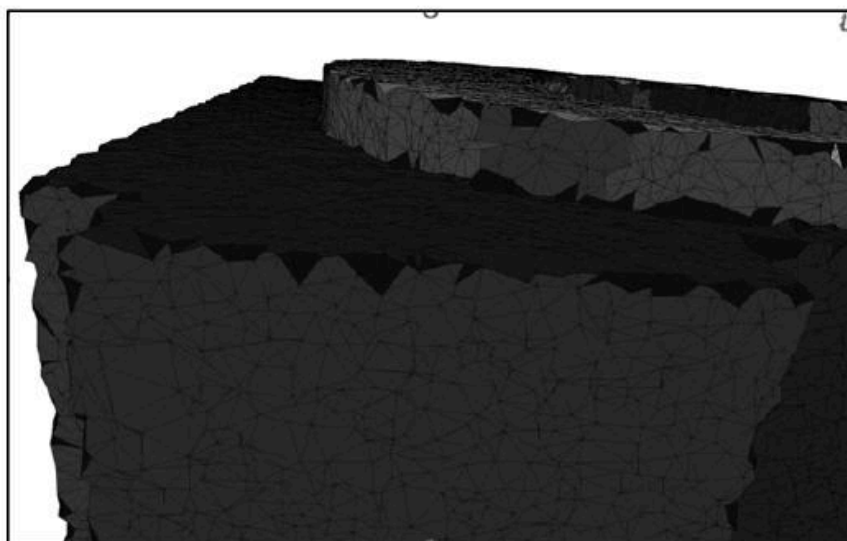


Рис. 7. Результат группировки полигонов

Используя анализ отклонения точек от усредненной плоскости, удалось сократить неразмеченные точки до нуля, однако количество неразмеченных полигонов осталось, что видно на рис. 8.

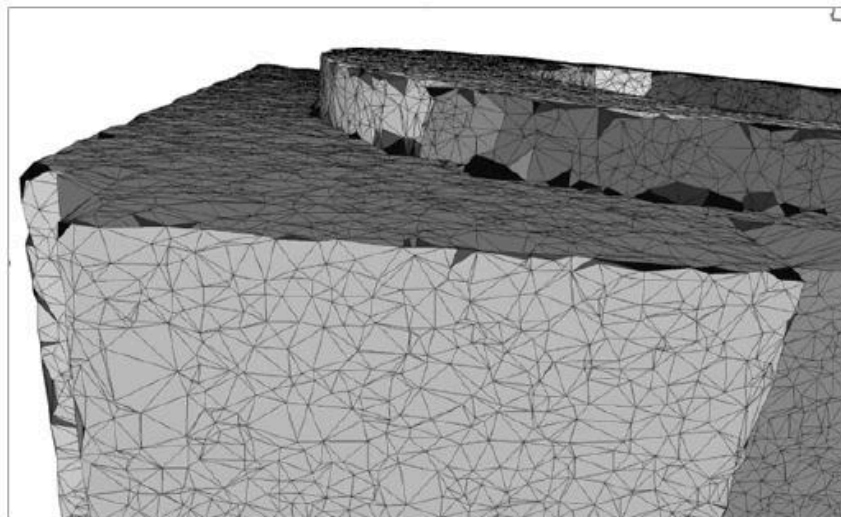


Рис. 8. Результат сокращения неразмеченных точек

Заключение. Изучены принцип и задача валидации ОКС на основе результатов лазерного сканирования. Спроектировано техническое решение, связанное с препроцессингом облаков точек — определения плоских граней. Выделен недостаток в виде неполной разметки полигонов.

Литература

- [1] СП 543.1325800.2024. Строительный контроль при строительстве, реконструкции, капитальном ремонте объектов капитального строительства. Утвержден приказом Минстроя России от 27 декабря 2024 г. № 950/пр.
- [2] Abreu N., Pinto A., Matos A., Pires M. Procedural Point Cloud Modelling in Scan-to-BIM and Scan-vs-BIM Applications: A Review. ISPRS International Journal of Geo-Information, 2023, № 7, art. no. 260.
- [3] CGAL. CGAL Manual. URL: <https://doc.cgal.org/latest/Manual/index.html> (accessed 15.09.2025).
- [4] GSoC 2018 — Generalized Region Growing. URL: https://docs.google.com/document/d/1J9VncBENYJ48Uuj1nYwPJDBZsOt_R4EgQEE0xIKUXj8/edit?tab=t (accessed 22.04.2025).
- [5] Cohen-Steiner D., Da F. A Greedy Delaunay Based Surface Reconstruction Algorithm. The Visual Computer, 2004, vol. 20 (1), pp. 4–16. <https://doi.org/10.1007/s00371-003-0217-z>

Point cloud preprocessing of laser scan results in civil engineering tasks analysis

Petrosyan Artur Vladimirovich

petrosyanav@student.bmstu.ru

Burkov Pavel Viktorovich (scientific advisor)

burkovpv@bmstu.ru

BMSTU, Moscow, Russia

This article presents an algorithm for processing point clouds. The application of laser scanning and its results in construction supervision as one of the stages of a capital construction project's life cycle is discussed. A mathematical formulation of the preprocessing problem for future validation of capital construction projects is presented using information modeling.

Keywords: *construction project, construction project lifecycle, laser scan, point cloud, points registration, surface reconstruction*