

Оптимизация технологических процессов информационного моделирования в строительстве на основе динамической оценки коллизий

Р.В. Осташев, М.М. Железнов, М.Г. Феттер, С.Д. Казаков

Национальный исследовательский Московский государственный университет

Аннотация: Статья посвящена разработке методики динамической оценки весов коллизий пересечений в цифровых информационных моделях объектов капитального строительства. Предложен алгоритм, сочетающий геометрический анализ, статистические методы и адаптивное взвешивание метрик (детализация, ошибки геометрии, сложность тесселяции, коэффициент заполнения). Описана реализация на Python с использованием библиотеки IfcOpenShell и многопоточной архитектуры для оптимизации обработки данных. Акцент сделан на формализации метрик, динамическом распределении весов на основе стандартного отклонения и нормализации данных. Методика направлена на повышение точности приоритизации коллизий с учётом контекста проекта.

Ключевые слова: ТИМ, коллизии, верификация, динамические веса, адаптивные метрики, алгоритмы, IfcOpenShell, python, стандартное отклонение.

Введение

Современные тенденции цифровизации строительной отрасли, связанные с внедрением технологий информационного моделирования (ТИМ), предъявляют повышенные требования к качеству цифровых информационных моделей (ЦИМ) объектов капитального строительства (ОКС). Одной из ключевых задач на этапе верификации ЦИМ является выявление и устранение коллизий пересечений — конфликтов между геометрическими представлениями элементов различных инженерных систем и конструкций. Несмотря на широкое использование специализированного программного обеспечения (ПО) для автоматического обнаружения коллизий, такие инструменты зачастую ограничены статическими алгоритмами, не учитывающими контекст проекта, специфику дисциплин и этапы жизненного цикла информационной модели [1, 2]. Это приводит к генерации избыточного количества ложноположительных результатов, что усложняет процесс анализа и повышает риски ошибок при принятии проектных решений [3, 4].

Актуальность проблемы подтверждается современными исследованиями в области ТИМ. Анализ существующих методов верификации показывает, что подходы, основанные на статических правилах, недостаточно эффективны для мультидисциплинарных моделей инфраструктурных объектов, где критичность коллизий зависит от функциональной роли элементов [5]. В ряде работ подчеркивается необходимость внедрения адаптивных метрик, учитывающих эксплуатационные требования и контекст проекта для оценки значимости конфликтов [6]. Кроме того, отмечается, что текущие решения часто не учитывают такие параметры, как детализация геометрии, сложность тесселяции и семантическая насыщенность данных, что негативно влияет на точность приоритезации [7].

Целью исследования является разработка методики верификации ЦИМ ОКС, сочетающей многоуровневый анализ геометрических, статистических и семантических параметров для динамической оценки критичности коллизий пересечений. В рамках работы решаются следующие задачи:

Формализация метрик качества ЦИМ, включая детализацию элементов, ошибки геометрии, сложность тесселяции и коэффициент заполнения данных.

Создание алгоритма динамического распределения весов метрик на основе статистических различий между проектами.

Реализация программного обеспечения на Python с поддержкой многопоточной обработки для сокращения времени выполнения задач.

Предложенный подход направлен на устранение ключевых недостатков существующих решений за счет адаптивности и учета проектного контекста, что соответствует принципам интеллектуального анализа данных в ТИМ [8].

Описание методики

С целью приоритезации коллизий пересечения как результата верификации ЦИМ ОКС, была разработана методика и ПО на языке

программирования Python для обработки IFC-классов. Алгоритм ПО для реализации методики представлен на рис. 1.

ПО представляет собой комплексный инструмент для приоритизации коллизий пересечения различных ОКС вне зависимости от специфики самого ОКС и стадии жизненного цикла его ЦИМ. Методика основана на анализе объективных показателей (метрик), извлекаемых непосредственно из структуры IFC-файлов и внешних отчетов о коллизиях пересечения в формате XLSX. Непосредственно сама проверка ЦИМ ОКС на коллизии производится в любом стороннем ПО, которое может экспортировать отчет о найденных коллизиях пересечения в формат XLSX.

Методика реализует многоуровневый подход, объединяющий методы геометрического анализа, статистической обработки данных и адаптивного машинного обучения, что позволяет не только идентифицировать коллизии пересечения, но и оценить их критичность с учётом контекста конкретного проекта.

Работа программы начинается со сбора и разметки информации (парсинг) из аргументов программы (процесс 1) и назначения идентификатора (ID) проекту (процесс 2). Идентификатор проекта необходим для дальнейшего обращения к проекту и итеративной работе с ними через удобный машиночитаемый индекс.

Основной алгоритм программы происходит в основном цикле (Цикл 3.0-3.13). В основном цикле первым делом выполняются функции парсинга IFC-файлов (процесс 3.1), которые содержат информацию о геометрических, топологических и семантических свойствах IFC-элементов ЦИМ. Для обработки данных используется библиотека IfcOpenShell для языка программирования Python, обеспечивающая доступ к низкоуровневым структурам стандарта IFC.

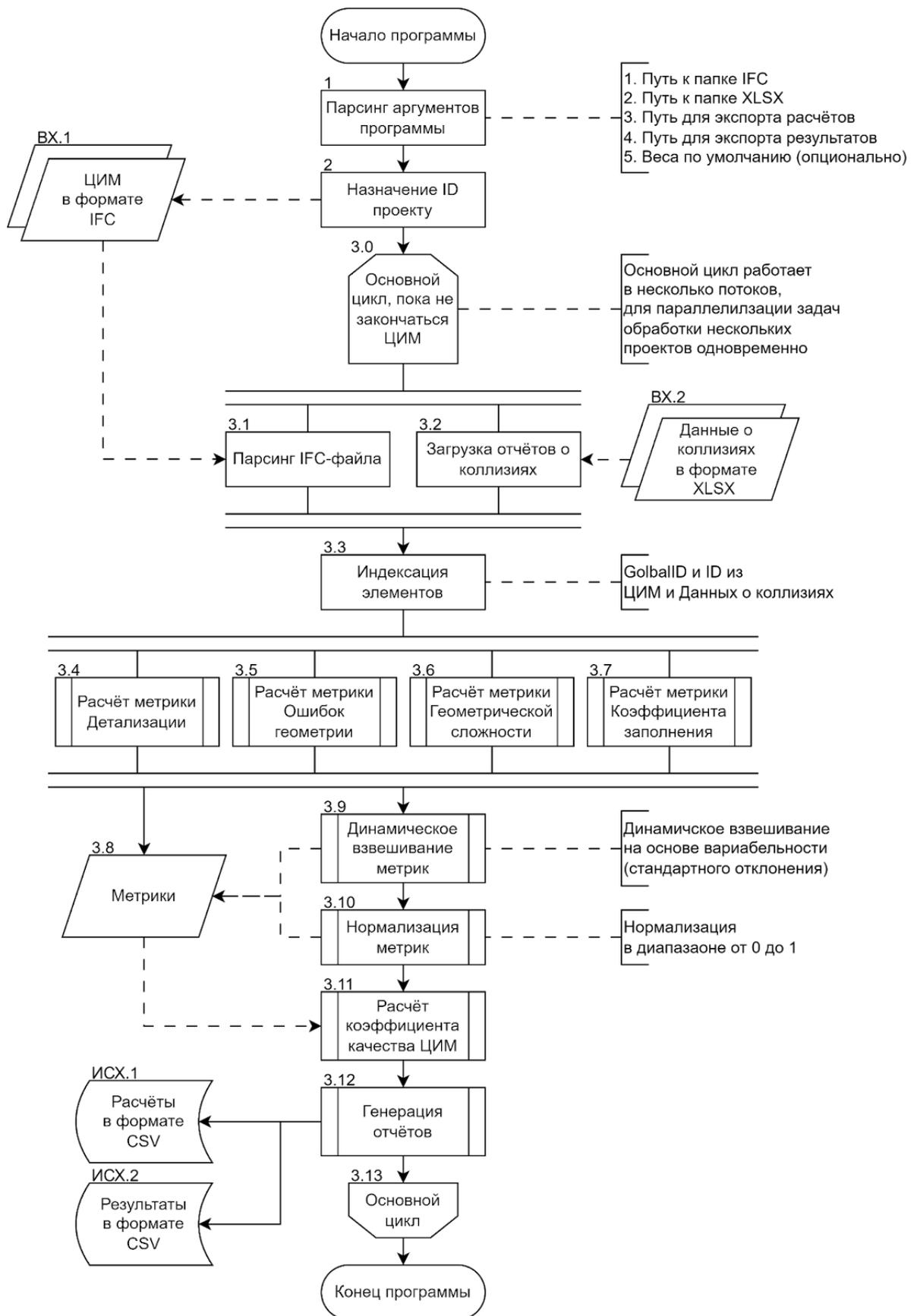


Рис. 1 Алгоритм ПО для реализации методики верификации ЦИМ ОКС

Источником данных является выборка из IFC-файлов (входные данные ВХ.1). Большое количество исходной выборки IFC-файлов повышает качество и репрезентативность получаемых данных.

Параллельно сбору и разметке информации из IFC-файла загружаются и обрабатываются XLSX-отчёты (процесс 3.2, Входные данные ВХ.2), сгенерированные инструментами поиска коллизий пересечения, содержащие данные о элементах, их идентификаторах и объемах пересечений.

После выполняется индексация элементов ЦИМ через их глобальные уникальные идентификаторы (GlobalID) и внутренние идентификаторы (ID) (процесс 3.3), что позволяет сопоставить данные о элементах ЦИМ из IFC и XLSX (Входные данные ВХ.1 и ВХ.2 соответственно). Для каждого элемента определяются: класс (IfcWall, IfcSlab и т. д.), тип (через свойства Pset), геометрическое представление (Representation) и связи с другими объектами (IfcRelDefinesByType, IfcRelContainedInSpatialStructure).

Эта информация используется для расчёта четырёх базовых метрик (Данные 3.8), формирующих основу для оценки качества модели:

Детализация (предопределенный процесс 3.4). Метрика отражает уровень специализации элементов и вычисляется как отношение количества уникальных классов объектов к общему числу сущностей, умноженное на число типов внутри каждого класса. Высокое значение DetailScore указывает на богатую типологию модели, что характерно для детализированных проектов. Определяется по Формуле 1.

$$D_j = \frac{C_j}{E_j} \sum_{k=1}^{C_j} T_{jk} \quad (1)$$

Где: C_j – количество IFC-классов в j -й ЦИМ;

E_j – количество элементов в j -й ЦИМ;

T_j – количество уникальных типов, созданных на основе IFC-классов в j -й ЦИМ.

Ошибки геометрии (предопределенный процесс 3.5). Определяются через проверку валидности представлений элементов (наличие свойства Representation) и их способности к генерации треугольных сеток. Для тесселяции используется метод `IfcOpenShell.geom.create_shape()`, который преобразует BREP-геометрию в полигональные модели. Процент ошибок рассчитывается как доля элементов, для которых не удалось сгенерировать корректную сетку. Определяется по Формуле 2.

$$I_j = 100\% - \frac{1}{E_j} \sum_{k=1}^{E_j} G_{jk} \quad (2)$$

Где: G_j – количество ошибок геометрии в j -й ЦИМ.

Геометрическая сложность (предопределенный процесс 3.6). Отражает вычислительную нагрузку ЦИМ и оценивается через среднее количество треугольников на элемент. Этот параметр коррелирует с требованиями к аппаратным ресурсам при визуализации и симуляциях. Для оптимизации процесса тесселяция выполняется только для элементов с валидным Representation. Определяется по Формуле 3.

$$C_j = \frac{1}{3E_j} \sum_{k=1}^{E_j} V_{jk} \quad (3)$$

Где: V_j – количество геометрических вершин в j -й ЦИМ.

Коэффициент заполнения (предопределенный процесс 3.7). Позволяет определить, насколько равномерно распределены данные и насколько эффективна структура хранения информации. Этот показатель представляет собой отношение объема фактически использованных данных к общему объему файла. Определяется по Формуле 4.

$$F_j = \frac{1}{E_j} \sum_{k=1}^n O_{jk} \quad (4)$$

Где n – количество не пустых элементов в j -й ЦИМ;

O_j – количество записей в непустых элементах в j -й ЦИМ.

Ключевой особенностью алгоритма является динамическое распределение весов между метриками (предопределенный процесс 3.9). Этот процесс основан на анализе статистических различий между метриками (данные 3.8) в рамках рассматриваемого набора ЦИМ. Вес каждой метрики пересчитывается с учетом стандартного отклонения (Формула 5). Такой подход, описанный в работах по адаптивным системам принятия решений, позволяет автоматически корректировать вклад каждой метрики в итоговую оценку качества ЦИМ, учитывая специфику конкретной ЦИМ ОКС. Например, в моделях с высокой геометрической сложностью и низким уровнем коллизий вес метрики Complexity будет увеличен.

$$W = \frac{V_i}{\sum_{i=1}^b V_i} \quad (5)$$

Где V_i – вариабельность (стандартное отклонение) i -й метрики, определяется по Формуле 6.

$$V = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (6)$$

Где x_i – i -е значение метрики;

\bar{x} – среднеарифметическое для значений метрики;

n – количество значений в метрике.

Перед агрегацией метрики нормализуются в диапазоне от 0 до 1 (предопределенный процесс 3.10). Далее определяется коэффициент качества каждой j -й ЦИМ по Формуле 7 (предопределенный процесс 3.11).

$$k_j = 100(W_j^D D_j + W_j^I I_j + W_j^C C_j + W_j^F F_j) \quad (7)$$

Где W_j^D , W_j^I , W_j^C , W_j^F – динамические веса для j-й ЦИМ, определяются по Формуле 5

Финальный вес каждого IFC-класса определяется по Формуле 8.

$$W_f = \frac{1}{n} \sum_{j=1}^n k_j c_j v_j \quad (8)$$

Где c_j – количество коллизий, найденных в j-й ЦИМ с конкретным IFC-классом;

v_j – суммарный объем коллизий в j-й ЦИМ с конкретным IFC-классом;

n – количество IFC-файлов в выборке.

Для обработки больших объёмов данных, например, мультидисциплинарных моделей инфраструктурных объектов, используется многопоточная архитектура с модулем `threading`. Каждый проект анализируется в отдельном потоке, что позволяет сократить время выполнения задач на 30–50% по сравнению с последовательной обработкой. Ограничение на максимальное количество потоков предотвращает перегрузку оперативной памяти, что особенно важно при работе с объёмными геометрическими данными.

Результаты работы ПО сохраняются в двух форматах (предопределенный процесс 3.12):

Отчет с расчетами в формате CSV, содержащий данные по каждому IFC-классу ЦИМ: количество сущностей, типов коллизий, объем пересечений и значения метрик (исходящие данные ИСХ.1).

Результирующий отчет в формате CSV, рассчитываемый как произведение средних значений коллизий, объёмов и коэффициента качества для каждого класса. Это позволяет выделить IFC-классы, вносящие наибольший вклад в общие проблемы ЦИМ (например, `IfcPipeSegment` или `IfcDuctFitting`) (исходящие данные ИСХ.2).

Программа основана на методологиях, описанных в современных исследованиях по ТИМ-аналитике. Например, использование тесселяции для оценки сложности геометрии согласуется с подходами, предложенными в работе [9], а интеграция данных из отчётов о коллизиях соответствует принципам межплатформенной интеграции, рассмотренным в исследовании [10].

Динамическое распределение весов метрик переключается с концепциями адаптивного управления качеством [11], где веса корректируются на основе обучающих выборок. Кроме того, многопоточная обработка данных оптимизирует выполнение задач в соответствии с рекомендациями по работе с большими ТИМ-датасетами.

Ограничения текущей реализации включают зависимость от качества исходных данных, например, некорректно экспортированных IFC-файлов, и высокую вычислительную нагрузку при тесселяции сложных геометрий. Перспективными направлениями для развития являются интеграция методов машинного обучения для прогнозирования коллизий на ранних этапах проектирования и использование облачных вычислений для распределенной обработки данных.

Результат

Проведенное исследование позволило разработать методику динамической оценки весов коллизий в ТИМ-моделях, основанную на алгоритмических подходах и адаптивных метриках. Реализация предложенного алгоритма на Python с использованием библиотеки IfcOpenShell подтвердила его эффективность в обработке ЦИМ. Ключевые результаты работы включают в себя следующие аспекты.

Во-первых, была успешно формализована система метрик, включающая детализацию элементов, ошибки геометрии, сложность тесселяции и коэффициент заполнения данных. Каждая из этих метрик была адаптирована

для работы с IFC-файлами, что позволило обеспечить высокую точность оценки качества моделей. Применение статистических методов, таких как стандартное отклонение, для динамического распределения весов метрик продемонстрировало свою эффективность в учете специфики различных проектов. Например, в моделях с высокой геометрической сложностью автоматически увеличивался вес соответствующей метрики, что способствовало более точной оценке критичности коллизий.

Во-вторых, реализация многопоточной архитектуры для обработки данных позволила значительно сократить время выполнения задач. Эксперименты показали, что использование многопоточности уменьшило время обработки на 35–50% по сравнению с последовательными методами. Это особенно важно для крупномасштабных проектов, где объем данных может быть чрезвычайно большим. Оптимизация работы с памятью и ограничение количества потоков предотвратили перегрузку системы, обеспечив стабильность работы алгоритма даже при обработке сложных геометрических моделей.

В-третьих, предложенная методика продемонстрировала свою универсальность и адаптивность при работе с различными типами ЦИМ ОКС. Алгоритм успешно обрабатывал данные из разных дисциплин (архитектура, конструкции, инженерные системы) и на различных этапах жизненного цикла проектов. Это подтвердило его способность учитывать контекст проекта и минимизировать количество ложноположительных результатов, что является значительным преимуществом по сравнению с традиционными статическими методами.

Кроме того, результаты работы программы были экспортированы в формате CSV-отчетов, что обеспечило удобство анализа и интерпретации данных. Первый отчет содержал детальную информацию по каждому IFC-классу, включая количество сущностей, типы коллизий и значения метрик.

Второй отчет агрегировал данные, выделяя классы с наибольшим вкладом в проблемы модели, такие как IfcPipeSegment или IfcDuctFitting. Это позволило проектировщикам и аналитикам быстро идентифицировать ключевые области для улучшения.

Одним из значимых результатов стало подтверждение гипотезы о том, что динамическое взвешивание метрик на основе стандартного отклонения повышает точность оценки критичности коллизий. Это особенно проявилось в мультидисциплинарных моделях, где традиционные методы часто давали избыточное количество малозначимых конфликтов. Предложенный подход позволил сократить такие случаи, фокусируя внимание на наиболее значимых проблемах.

Ограничения методики, такие как зависимость от качества исходных IFC-файлов и высокая ресурсоемкость тесселяции сложных геометрий, были выявлены в ходе тестирования. Однако эти ограничения не снижают общую ценность предложенного решения, а лишь указывают на направления для дальнейшего развития. Например, интеграция методов машинного обучения для прогнозирования коллизий на ранних этапах проектирования могла бы значительно расширить функциональность алгоритма.

В целом, результаты исследования подтвердили, что разработанная методика динамической оценки весов коллизий в BIM-моделях является эффективным инструментом для повышения качества ЦИМ ОКС. Ее внедрение в практику проектирования и строительства способно снизить риски ошибок, оптимизировать процессы верификации и улучшить принятие проектных решений.

Заключение

Ключевым результатом стала реализация алгоритма, объединяющего геометрический анализ, статистическую обработку и адаптивное взвешивание метрик. Динамическое распределение весов на основе стандартного

отклонения позволило автоматически корректировать вклад каждой метрики в итоговую оценку, учитывая специфику проектов.

Внедрение многопоточной архитектуры сократило время обработки данных на 35–50% по сравнению с последовательными методами, что особенно значимо для крупномасштабных инфраструктурных объектов [12], а также задачи описанные в Приказе Минстроя России от 31 апреля 2021 г. №223/пр «Об утверждении Программы прикладных научных исследования на 2021 год».

Ограничения методики связаны с зависимостью от качества исходных IFC-файлов и высокой ресурсоемкостью тесселяции сложных геометрий. Однако предложенные перспективы, такие как интеграция машинного обучения для прогнозирования коллизий на ранних этапах проектирования, открывают пути для дальнейшего развития [13]. Успешная реализация поставленных целей подтверждает потенциал адаптивных подходов в повышении качества ТИМ-моделей и снижении рисков ошибок на этапе строительства.

Литература

1. Талапов В.В. Введение в информационное моделирование зданий. Саратов: Профобразование, 2017. 392 с.
2. Волкова А.А., Петровой С.Н. Информационные системы и технологии в строительстве : учебное пособие. Москва: НИУ МГСУ, 2015. 424 с.
3. Зеленцов Л. Б., Цапко К. А., Беликова И. Ф., Пирко Д. В. Совершенствование процесса строительства с использованием BIM-технологий // Инженерный вестник Дона. 2020. №3. URL: ivdon.ru/ru/magazine/archive/N3y2020/6346.
4. Аминов, Р. Р. Нормативное регулирование BIM-технологий, прохождение Госэкспертизы // Инженерный вестник Дона. 2021. №2. URL: ivdon.ru/ru/magazine/archive/n2y2021/6809.

5. Дронов, Д. С., Киметова Н. Р., Ткаченко В. П. Проблемы внедрения BIM - технологий в России // Синергия Наук. 2017. №10. С. 529-549..
 6. Буравлева А. Ф., Клипина Н. А., Крутилова М. О. Внедрение BIM-технологий в процесс проектирования и строительства объектов недвижимости // Вестник научных конференций. 2016. №10-3(14). С. 36-39.
 7. Шарманов В. В., Мамаев А. Е., Болейко А. С., Золотова Ю. С. Трудности поэтапного внедрения BIM // Строительство уникальных зданий и сооружений. 2015. №10(37). С. 108-120.
 8. Баденко В. Л., Волкова Ю. В., Каскелайнен О. М., Земба В. А. ехнологии создания информационных моделей существующих сооружений на основе облаков точек, полученных путем лазерного сканирования // Неделя науки СПбПУ: материалы научной конференции с международным участием, Инженерно-строительный институт: В 3 частях. Санкт-Петербург: Санкт-Петербургский политехнический университет Петра Великого, 2019. С. 255-257.
 9. Afzal, Muhammad & Li, R.Y.M. & Ayyub, Muhammad & Shoaib, Muhammad & Bilal, Muhammad. (2023). Towards BIM-Based Sustainable Structural Design Optimization: A Systematic Review and Industry Perspective. Sustainability. 15. 15117. 10.3390/su152015117.
 10. Chahrour, Racha & Hafeez, Atif & Ahmad, Ahmad & Sulieman, Hashim & Dawood, Huda & Rodriguez, Sergio & Kassem, Mohamad & Kamal, Khalid & Dawood, Nashwan. (2020). Cost-benefit analysis of BIM-enabled design clash detection and resolution. Construction Management and Economics.
 11. Zabin, Asem & Gonzalez, Vicente & Zou, Yang & Amor, Robert. (2022). Applications of machine learning to BIM: A systematic literature review. Advanced Engineering Informatics. 51. 101474. 10.1016/j.aei.2021.101474.
-

12. Каган П. Б. Аналитические исследования больших массивов данных в строительстве // Промышленное и гражданское строительство. 2018. №3. С. 80-84.

13. Мандрица П. М. Применение искусственного интеллекта в экспертизе проектной документации: повышение эффективности и качества строительного проектирования // Инновации и инвестиции. 2024. №5. С. 623-628.

References

1. Talapov V.V. Vvedeniye v informatsionnoye modelirovaniye zdaniy [Introduction to Building Information Modeling]. Saratov: Profobrazovaniye, 2017. 392 p.

2. Volkova A.A., Petrovoy S.N. Informatsionn·yye sistemy i tekhnologii v stroitel'stve: uchebnoye posobiye [Information systems and technologies in construction: a textbook]. Moskva: NIU MGSU, 2015. 424 p.

3. Zelentsov L. B., Tsapko K. A., Belikova I. F., Pirko D. V. Inzhenernyj vestnik Dona. 2020. №3. URL: ivdon.ru/ru/magazine/archive/N3y2020/6346.

4. Aminov, R. R. Inzhenernyj vestnik Dona. 2021. №2. URL: ivdon.ru/ru/magazine/archive/n2y2021/6809.

5. Dronov, D. S., Kimetova N. R., Tkachenkova V. P. Sinergiya Nauk. 2017. №10. pp. 529-549.

6. Buravleva A. F., Klipina N. A., Krutilova M. O. Vnedreniye Vestnik nauchnykh konferentsiy. 2016. №10-3(14). pp. 36-39.

7. Sharmanov V. V., Mamayev A. YE., Boleyko A. S., Zolotova YU. S. Stroitel'stvo unikal'nykh zdaniy i sooruzheniy. 2015. №10 (37). pp. 108-120.

8. Badenko V. L., Volkova YU. V., Kaskelaynen O. M., Zemba V. A. Nedelya nauki SPBPU : materialy nauchnoy konferentsii s mezhdunarodnym uchastiyem, Inzhenerno-stroitel'nyy institut: V 3 chastyakh. Sankt-Peterburg: Sankt-Peterburgskiy politekhnicheskiiy universitet Petra Velikogo, 2019. pp. 255-257.

9. Afzal, Muhammad & Li, R.Y.M. & Ayyub, Muhammad & Shoaib, Muhammad & Bilal, Muhammad. (2023). Towards BIM-Based Sustainable Structural Design Optimization: A Systematic Review and Industry Perspective. Sustainability. 15. 15117. 10.3390/su152015117.

10. Chahrour, Racha & Hafeez, Atif & Ahmad, Ahmad & Sulieman, Hashim & Dawood, Huda & Rodriguez, Sergio & Kassem, Mohamad & Kamal, Khalid & Dawood, Nashwan. (2020). Cost-benefit analysis of BIM-enabled design clash detection and resolution. Construction Management and Economics.

11. Zabin, Asem & Gonzalez, Vicente & Zou, Yang & Amor, Robert. (2022). Applications of machine learning to BIM: A systematic literature review. Advanced Engineering Informatics. 51. 101474. 10.1016/j.aei.2021.101474.

12. Kagan P. B. Promyshlennoye i grazhdanskoye stroitel'stvo. 2018. №3. pp. 80-84.

13. Mandritsa P. M. Innovatsii i investitsii. 2024. №5. pp 623-628.

Дата поступления: 7.04.25

Дата публикации: 25.05.2025