

## ПРЕОБРАЗОВАНИЕ АРХИТЕКТУРНЫХ МОДЕЛЕЙ IFC В GEOJSON И ОПТИМИЗАЦИЯ ИХ РАЗМЕРА ДЛЯ НАВИГАЦИОННЫХ СИСТЕМ



В. С. Попов, студент  
E-mail: po.vit.step@yandex.ru  
ФГБОУ ВО «Калининградский государственный  
технический университет»

Т. В. Снытникова, к. т. н., научный руководитель  
E-mail: tatiana.snitnikova@klgtu  
ФГБОУ ВО «Калининградский государственный  
технический университет»

В статье рассматривается проблема адаптации архитектурных моделей зданий, представленных в формате IFC, для использования в веб-ориентированных навигационных и геоинформационных системах. Формат IFC широко применяется в технологии информационного моделирования зданий, однако из-за сложности структуры, отсутствия географической привязки и значительного объема данных он не пригоден для использования в навигационных приложениях. Предложен подход к преобразованию данных IFC в формат GeoJSON, обеспечивающий совместимость с современными веб-технологиями и интерактивными картографическими сервисами. В исследовании описаны методы оптимизации объема выходных файлов, включающие фильтрацию объектов, удаление избыточных атрибутов, округление координат и упрощение геометрии. Представлены примеры реализации данных операций с использованием инструмента ifc2gis. Результаты исследования показали, что применение предложенных методов позволяет сократить размер итогового GeoJSON-файла до 49 % при сохранении геометрической точности и необходимых атрибутов помещений. В результате повышается скорость загрузки и отображения карт, что делает возможным эффективное использование архитектурных моделей BIM в серверных и клиентских частях навигационных систем.

**Ключевые слова:** IFC, JSON, GeoJSON, BIM, CAD, карты, *indoor*-карты, навигация.

### ВВЕДЕНИЕ

Современные организации все чаще внедряют цифровые решения, направленные на облегчение навигации и повышение эффективности работы с внутренними пространствами зданий. Одним из перспективных направлений цифровизации является создание интерактивных карт помещений, которые позволяют пользователям ориентироваться внутри крупных объектов – от учебных корпусов до бизнес-центров, больниц, торговых и спортивных комплексов. Примером такого подхода может служить Высшая школа экономики, которая для внутренней навигации по своим учебным корпусам разработала приложение «Вышка.Навигация» [9].

Для построения таких систем требуется наличие пространственных данных, описывающих внутреннюю структуру зданий. Самые точные и актуальные сведения содержатся в архитектурных BIM-моделях, обычно хранящихся в формате IFC (Industry Foundation Classes) – международном стандарте [6] для хранения информации о зданиях, поддерживаемом Autodesk Revit, Archicad, Renga и другими CAD-системами.

Однако применение IFC-файлов непосредственно для задач веб-визуализации и интеграции с геоинформационными системами (ГИС) сопряжено с определенными техническими трудностями. Формат IFC обладает обширной структурой, ориентированной на обмен данными между профессиональными CAD-программами, и не оптимизирован для передачи через Интернет, быстрой визуализации в браузере и интеграции с современными веб- или мобильными приложениями.

Формат GeoJSON, напротив, является легким, удобочитаемым и открытым расширением стандартного JSON, предназначенным для хранения геометрических объектов и их свойств. Благодаря широкой поддержке со стороны картографических библиотек (таких как Leaflet [7], OpenLayers [8], MapLibre [3]) и стандартов обмена пространственными данными, GeoJSON фактически стал отраслевым стандартом для представления геоданных в веб-среде.

Поэтому задача преобразования BIM-моделей из формата IFC в GeoJSON востребована для решения задач по цифровой картографии помещений и интеграции внутренней структуры зданий в современные информационные системы. Решение данной задачи предполагает анализ и сопоставление структур обоих форматов, внедрение эффективных инструментов конвертации и оптимизации данных для использования в веб-приложениях и ГИС.

В рамках работы по созданию серверной части навигации вуза были использованы существующие инструменты для преобразования IFC-моделей в GeoJSON, однако они не обеспечивают достаточного уровня оптимизации для задач веб- и мобильной навигации. На их основе разработано улучшенное программное решение, представленное в данной статье, позволяющее получать оптимизированные GeoJSON-файлы.

## ОБЪЕКТ ИССЛЕДОВАНИЯ

Объектом исследования является процесс преобразования архитектурных моделей зданий, созданных в формате IFC, в формат геоданных GeoJSON, совместимый с веб-технологиями и геоинформационными системами.

## ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Цель исследования – разработка и описание методики преобразования данных из формата IFC в GeoJSON, обеспечивающей сохранение геометрической точности, минимальный объем данных и совместимость с ГИС и веб-платформами.

Для достижения поставленной цели были определены следующие задачи:

1. Проанализировать существующие форматы пространственных данных (IFC, JSON, GeoJSON) и определить их применимость для систем внутренней навигации.
2. Исследовать возможности инструмента ifc2gis для конвертации BIM-моделей в ГИС-совместимые форматы.
3. Разработать и реализовать программные скрипты для оптимизации GeoJSON-файлов, включающие фильтрацию, округление координат и упрощение геометрии.
4. Провести сравнительный анализ объема и производительности обработки данных при различных методах оптимизации.
5. Оценить влияние оптимизации на точность геометрии и скорость визуализации карт в веб-приложениях.

## МЕТОДЫ ИССЛЕДОВАНИЯ

Формат IFC (Industry Foundation Classes) [5] является открытым международным стандартом [6] представления данных об элементах зданий и сооружений, используемым в концепции информационного моделирования (BIM).

Файл IFC описывает трехмерную структуру объекта, включая этажи, помещения, стены, двери, окна и иные объекты. Пример BIM модели IFC файла представлен на рисунке 1.

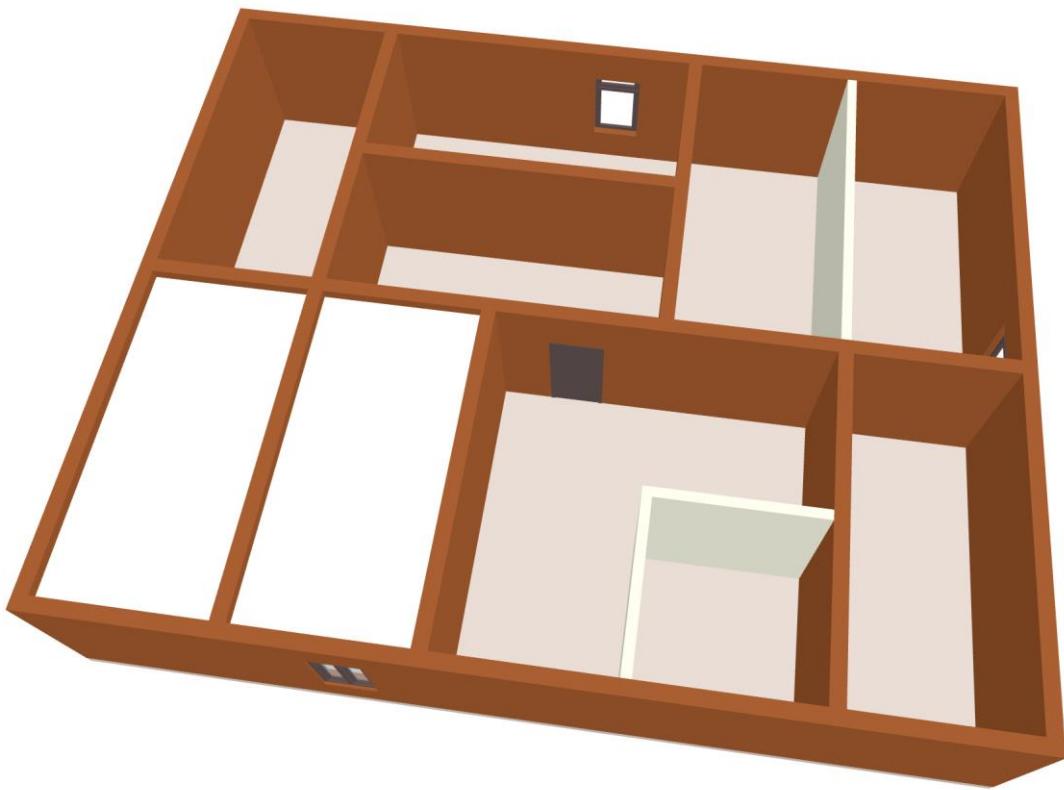


Рисунок 1 – Пример BIM модели IFC файла

Каждый элемент имеет тип (IfcWall, IfcDoor, IfcSpace и др.), пространственные координаты, а также набор свойств – от геометрических параметров до эксплуатационных характеристик.

IFC использует внутреннюю иерархическую структуру, в которой каждый объект связан с другими элементами посредством ссылок. Координаты в файлах IFC, как правило, выражаются в локальной системе отсчета модели и не содержат географической привязки (EPSG).

Кроме того, формат предназначен для хранения полной трехмерной геометрии, что делает его объемным и сложным для прямой интеграции в веб-среду и геоинформационные сервисы.

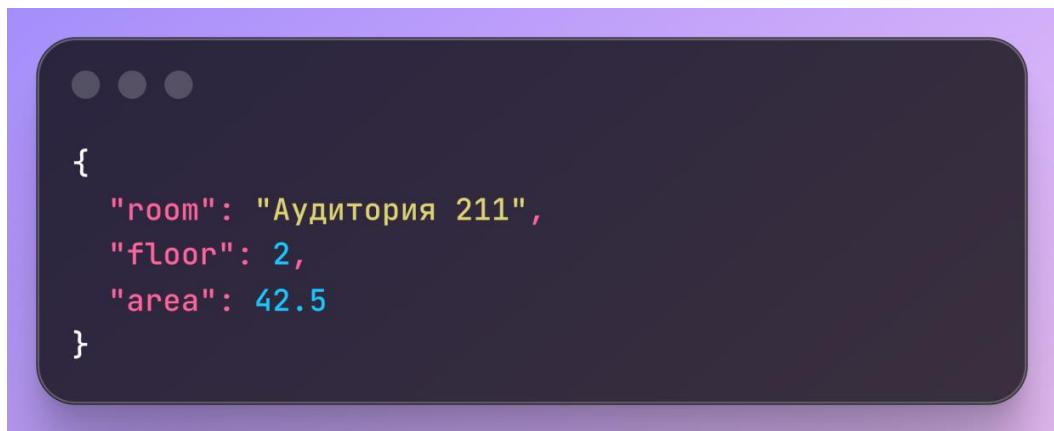
Типовой фрагмент IFC-файла имеет вид, приведенный на рисунке 2.

```
#1820= IFCSpace('36sD45x3D8N05rSde',#41,'Аудитория 211',$$,#1810,$,$,.ELEMENT.,42.5);
#1810= IFCLOCALPLACEMENT(#1770,#1809);
#1809= IFCAXIS2PLACEMENT3D(#30,#31,#32);
```

Рисунок 2 – Фрагмент IFC-файла

Такое представление требует дополнительной обработки и преобразования в более легкий формат, пригодный для веб-визуализации и хранения в базах данных.

Формат JSON (JavaScript Object Notation) представляет собой текстовый способ хранения и передачи структурированных данных. Он является базовым форматом обмена данными между клиентом и сервером в современных веб-приложениях благодаря своей простоте, читаемости и универсальности. Пример структуры JSON-объекта представлен на рисунке 3.



```
{  
  "room": "Аудитория 211",  
  "floor": 2,  
  "area": 42.5  
}
```

Рисунок 3 – Пример JSON объекта

Несмотря на гибкость, базовый JSON не имеет встроенных средств для описания пространственных координат и геометрии. Для этих целей используется специализированное расширение – GeoJSON.

Формат GeoJSON – это подтип JSON, предназначенный для представления географических и геометрических объектов. Он используется для описания точек, линий и полигонов, а также их свойств.

GeoJSON поддерживается всеми современными библиотеками визуализации карт (Leaflet, Mapbox, OpenLayers) и серверными СУБД с пространственными расширениями (PostgreSQL/PostGIS [4]). Пример структуры GeoJSON объекта представлен на рисунке 4.



```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Polygon",  
    "coordinates": [  
      [  
        [  
          [30.1234, 59.9876],  
          [30.124, 59.9878],  
          [30.1242, 59.9875],  
          [30.1234, 59.9876]  
        ]  
      ]  
    ],  
    "properties": {  
      "name": "Аудитория 211",  
      "floor": 2,  
      "area": 42.5  
    }  
  }  
}
```

Рисунок 4 – Пример GeoJSON объекта

В отличие от IFC, формат GeoJSON:

- поддерживает стандартизованные типы геометрии (Point, LineString, Polygon);
- является человеко-читаемым и легко сериализуется в веб-приложениях;
- обеспечивает минимальный объем данных при передаче через API.

Пример отображения данных GeoJSON в MapLibre Native приведен на рисунке 5.

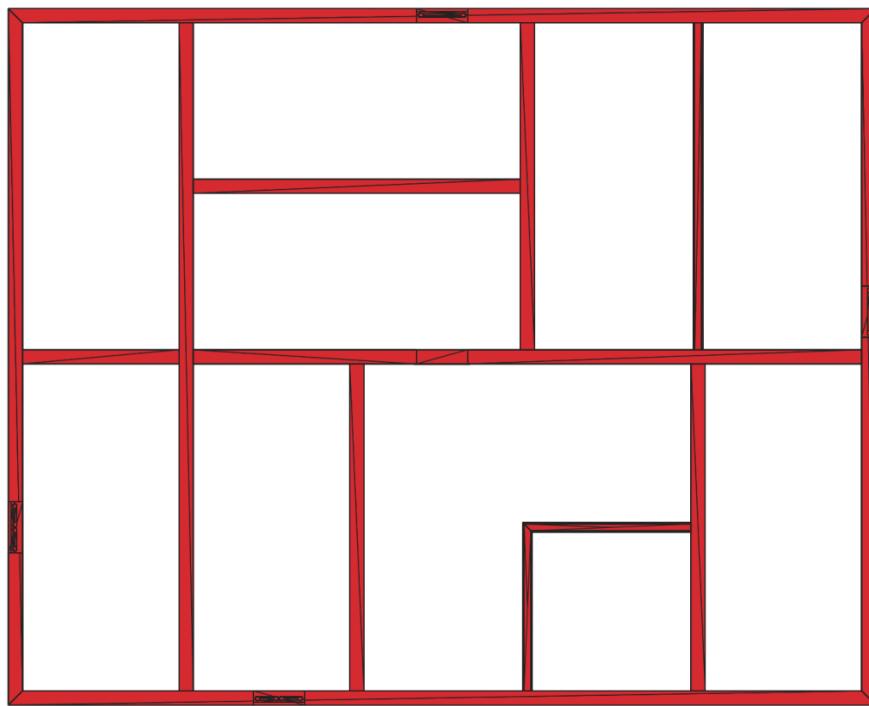


Рисунок 5 – Пример представления GeoJSON файла

Несмотря на богатую структуру данных, использование IFC напрямую в системах навигации имеет ряд ограничений:

- Отсутствие географической привязки. Координаты IFC описываются в локальной системе модели и не соотносятся с глобальной системой координат.
- Избыточность данных. IFC хранит все элементы модели, включая инженерные сети и декоративные объекты, что делает файлы слишком тяжелыми.
- Трехмерная сложность. Большинство навигационных систем используют 2D-проекцию, тогда как IFC хранит 3D-геометрию.
- Неподдерживаемый формат. Веб-карты и клиентские библиотеки не имеют встроенных парсеров IFC.
- Скорость обработки. Размер и структура файлов приводят к высокой нагрузке на сервер при попытке их интерпретации.

Эти ограничения делают необходимым предварительное преобразование IFC в GeoJSON, который является универсальным форматом для веб-визуализации и пространственного анализа.

Для выполнения конвертации архитектурных моделей в формат, пригодный для веб-навигации, в исследовании использовался инструмент ifc2gis [1] – открытая программная система, предназначенная для извлечения геометрических и атрибутивных данных из файлов формата IFC (Industry Foundation Classes) и преобразования их в геоинформационные форматы, включая GeoJSON.

При запуске утилиты ifc2gis пользователь указывает входной файл IFC, желаемый формат вывода и параметры фильтрации. В результате программа выполняет:

- разбор структуры IFC с идентификацией типов объектов (например, IfcSpace, IfcWall, IfcDoor);

- проекцию трехмерной модели в двумерную плоскость, пригодную для визуализации на карте;
- экспорт полученных объектов в формат GeoJSON с сохранением их свойств (названия, уровня, площади, типа помещения);
- формирование выходного файла, содержащего коллекцию объектов типа FeatureCollection.

На рисунке 6 представлена базовая команда для выполнения конвертации.



```
const geojson = await ifc2Geojson(ifcData, crs, (msg) => console.log(msg))
```

Рисунок 6 – Команда для выполнения конвертации ifc2gis

Результирующий файл представляет собой структурированный документ, содержащий все помещения здания в виде полигонов с их свойствами. Однако без фильтрации и оптимизации такой файл может иметь значительный объем (десятки мегабайт) и содержать множество вспомогательных элементов, необходимых для работы с BIM моделью, но требующихся для отображения на карте.

Для повышения гибкости и качества результата ifc2gis позволяет отфильтровать экспортируемые объекты. На рисунке 7 представлена команда, выполняющая конвертацию данных с использованием фильтра, удаляющего окна и перекрытия:



```
const toFilter = ["IfcWindow", "IfcSlab"];

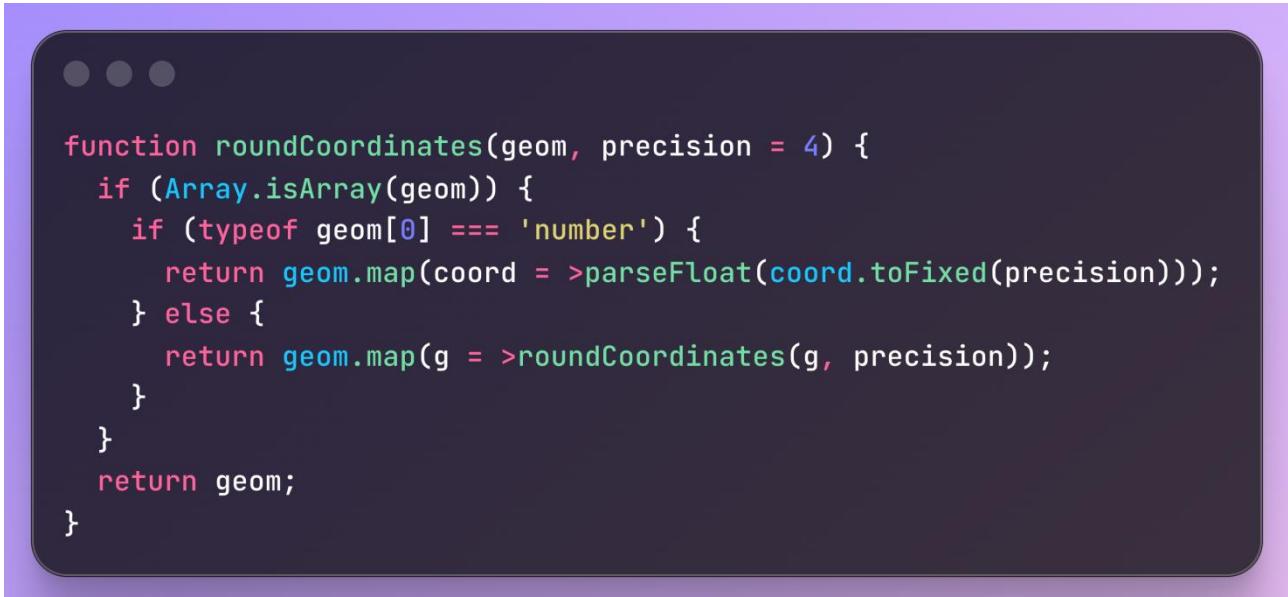
const geojson = await ifc2GeojsonWithFilter(
  ifcData,
  "urn:ogc:def:crs:EPSG::3857",
  toFilter,
  (msg) => console.log(msg)
);
```

Рисунок 7 – Команда для выполнения конвертации ifc2gis с фильтром

Для повышения эффективности хранения и передачи данных после конвертации модели из формата IFC в GeoJSON была реализована система автоматической постобработки, состоящая из нескольких этапов:

- удаления неиспользуемых атрибутов;
- округления координат;
- упрощения геометрии.

Для сокращения объема файла выполняется округление координат с заданной точностью, передающейся в параметрах функции. Реализация функции округления представлена на рисунке 8.



```
function roundCoordinates(geom, precision = 4) {
  if (Array.isArray(geom)) {
    if (typeof geom[0] === 'number') {
      return geom.map(coord => parseFloat(coord.toFixed(precision)));
    } else {
      return geom.map(g => roundCoordinates(g, precision));
    }
  }
  return geom;
}
```

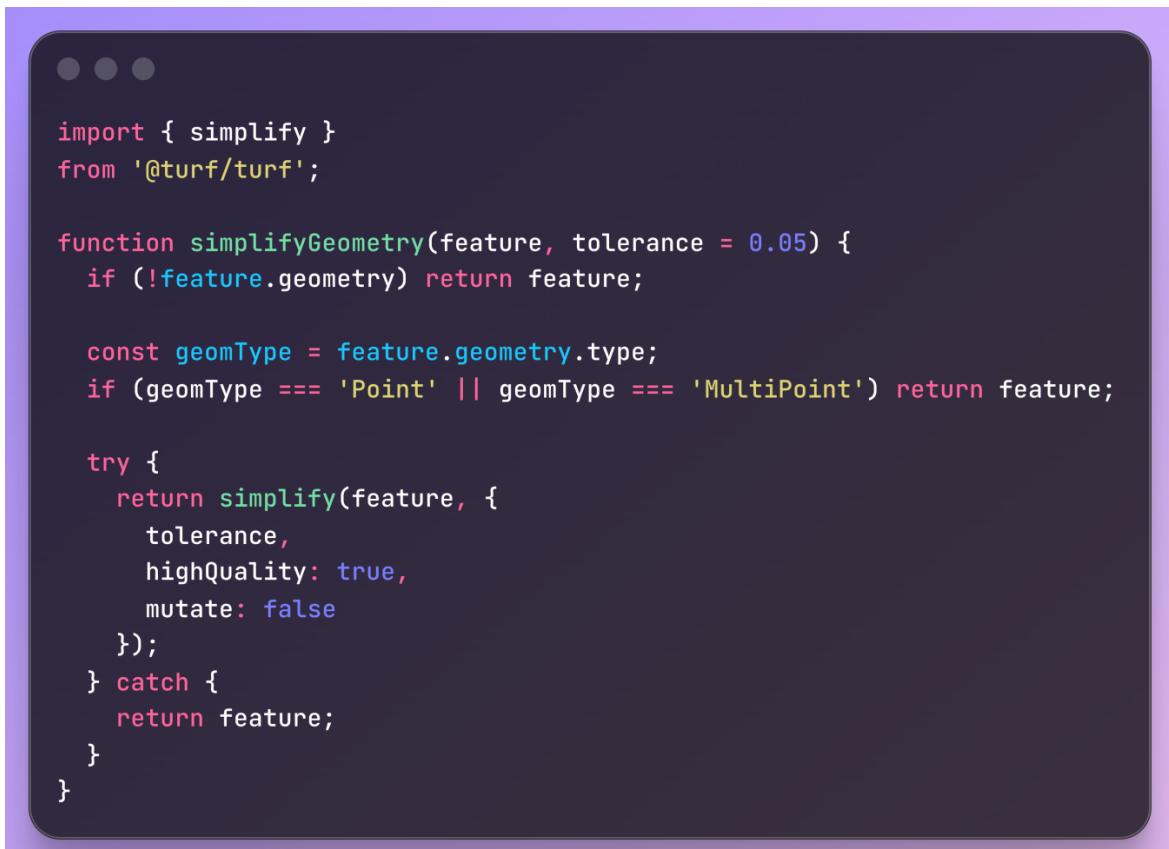
Рисунок 8 – Функция округления координат

При точности  $precision = 4$  обеспечивается ошибка не более 0,1 м, что является приемлемым для задач навигации внутри зданий.

Дополнительно применяется упрощение геометрии с помощью библиотеки Turf.js [2], что уменьшает количество вершин в полигонах.

Метод особенно эффективен для сложных контуров, однако для рассматриваемого здания, состоящего преимущественно из прямоугольных помещений, эффект сокращения оказался незначительным (около 5–7 %).

Реализация функции упрощения геометрии представлена на рисунке 9.



```
import { simplify } from '@turf/turf';

function simplifyGeometry(feature, tolerance = 0.05) {
  if (!feature.geometry) return feature;

  const geomType = feature.geometry.type;
  if (geomType === 'Point' || geomType === 'MultiPoint') return feature;

  try {
    return simplify(feature, {
      tolerance,
      highQuality: true,
      mutate: false
    });
  } catch {
    return feature;
  }
}
```

Рисунок 9 – Функция упрощения геометрии

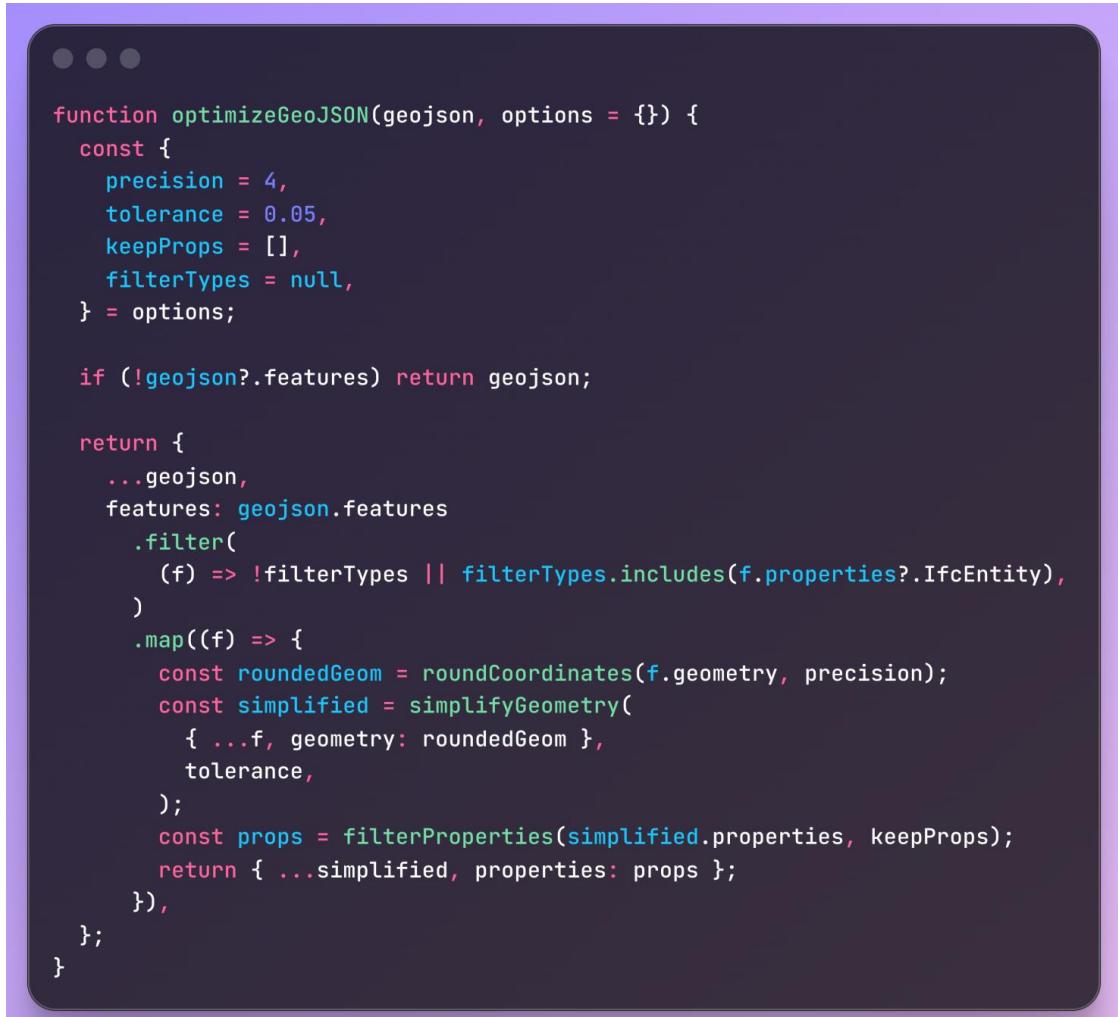
Для уменьшения объема блока properties из каждого объекта сохраняются только необходимые атрибуты (например, Name). Необходимые параметры можно передать массивом в параметры функции. Реализация функции фильтрации атрибутов представлена на рисунке 10.



```
function filterProperties(properties, keepProps = []) {
  if (!properties || keepProps.length === 0) return properties;
  const result = {};
  for (const key of keepProps) {
    if (key in properties) result[key] = properties[key];
  }
  return result;
}
```

Рисунок 10 – Функция фильтрации атрибутов

Итоговая функция оптимизации GeoJSON, включающая фильтрацию, округление, упрощение и очистку свойств, представлена на рисунке 11.



```
function optimizeGeoJSON(geojson, options = {}) {
  const {
    precision = 4,
    tolerance = 0.05,
    keepProps = [],
    filterTypes = null,
  } = options;

  if (!geojson?.features) return geojson;

  return {
    ...geojson,
    features: geojson.features
      .filter(
        (f) => !filterTypes || filterTypes.includes(f.properties?.IfcEntity),
      )
      .map((f) => {
        const roundedGeom = roundCoordinates(f.geometry, precision);
        const simplified = simplifyGeometry(
          { ...f, geometry: roundedGeom },
          tolerance,
        );
        const props = filterProperties(simplified.properties, keepProps);
        return { ...simplified, properties: props };
      }),
  };
}
```

Рисунок 11 – Комплексная функция для оптимизации GeoJSON

## РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Для оценки влияния различных методов оптимизации использовалась тестовая BIM-модель этажа учебного корпуса в формате IFC объемом 12,3 МБ. Исходный файл содержал 842 объекта, включая помещения, стены, двери, окна и вспомогательные объекты. После конвертации получен GeoJSON-файл, размер файла составил 8,4 МБ. На основе данного GeoJSON-файла оценивалось влияние методов оптимизации.

В качестве основных показателей использовался итоговый размер файла и процентное изменение относительно исходного результата. Итоги эксперимента представлены в таблице.

Таблица – Результаты оптимизаций размера GeoJSON файла

Метод оптимизации	Суть метода	Размер после оптимизации	Изменение, %	Примечание
Базовый экспорт (без параметров)	Все объекты модели	8.4 МБ	—	Исходный результат
Фильтрация (--filter IfcSpace,IfcDoor)	Исключены пли-ты, окна	6.1 МБ	-27 %	Хороший эффект, если есть избыточные объекты
Удаление второстепенных атрибутов (Оставляем только Name)	Убраны метадан-ные	7.5 МБ	-10 %	Без потери информа-тивности
Округление координат (Параметр precision = 4)	Сокращены лиш-ние десятичные разряды	5.3 МБ	-37 %	Потеря точности < 0.1 м
Упрощение геометрии (Параметр tolerance = 0.05)	Уменьшено ко-личество вершин	7.2 МБ	-14 %	Небольшой эффект из-за простой геометрии помещений
Комбинированная оптимизация	Все методы одновременно	4.3 МБ	-49 %	Наилучшее соотноше-ние объема и точности

## ОБСУЖДЕНИЕ

Результаты эксперимента показывают, что наибольшее сокращение достигается за счет округления координат и фильтрации избыточных объектов.

Метод упрощения геометрии оказывает минимальное влияние на размер файла (около 5–6 %), что объясняется простой формой помещений: большинство аудиторий и коридоров представляют собой прямоугольники с четырьмя вершинами, не требующие редукции полигональной сложности.

Сокращение объема GeoJSON-файла напрямую влияет на производительность серверной и клиентской части системы. При загрузке неотфильтрованного файла (8,4 МБ) время рендеринга в мобильном клиентском приложении на основе библиотеки MapLibre [3] составляло 2,1 с, тогда как после оптимизации (4.3 МБ) – в среднем 1,2 с.

Визуальный анализ подтвердил сохранение корректной геометрии и пространственного расположения помещений после всех этапов оптимизации: при упрощении геометрии с допуском 0,05 м форма помещений не претерпела видимых искажений, что подтверждает достаточность выбранных параметров для зданий с простой архитектурой.

## ЗАКЛЮЧЕНИЕ

Проведенное исследование показало, что формат IFC, несмотря на богатое информационное наполнение, не предназначен для использования в геоинформационных и навигационных системах. Его структура изначально ориентирована на детальное трехмерное моделирование и хранение инженерных данных, что приводит к избыточности при передаче в веб-среду.

Преобразование IFC в формат GeoJSON позволяет адаптировать архитектурные модели к потребностям интерактивных карт и систем внутренней навигации.

Анализ эффективности различных методов оптимизации подтвердил, что:

- фильтрация типов объектов и округление координат являются ключевыми мерами для уменьшения объема данных;
- упрощение геометрии оказывает незначительное влияние при работе с архитектурными планами простой формы (здания с прямолинейной планировкой);
- итоговый GeoJSON-файл сохраняет всю функциональную информацию, необходимую для построения навигации: геометрию помещений, этажность и площади.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. IFC to GeoJSON Converter – ifc2geojson [Электронный ресурс]. – URL: <https://github.com/abdoulayediak/ifc2geojson> (дата обращения: 10.09.2025).
2. Turf.js документация [Электронный ресурс]. – URL: <https://turfjs.org/docs/api/simplify> (дата обращения: 12.09.2025).
3. MapLibre Native Documentation [Электронный ресурс]. – URL: <https://maplibre.org> (дата обращения: 10.09.2025).
4. PostGIS документация [Электронный ресурс]. – URL: <https://postgis.net> (дата обращения: 12.09.2025).
5. IFC Industry Foundation Classes [Электронный ресурс]. – URL: <https://timmcginley.github.io/41934/Concepts/IFC.html> (дата обращения: 12.09.2025).
6. ISO 16739-1:2024 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries [Электронный ресурс]. – URL: <https://www.iso.org/standard/84123.html> (дата обращения: 10.09.2025).
7. Leaflet Documentation [Электронный ресурс]. – URL: <https://github.com/Leaflet/Leaflet> (дата обращения: 10.09.2025).
8. OpenLayers Documentation [Электронный ресурс]. – URL: <https://github.com/openlayers/openlayers> (дата обращения: 10.09.2025).
9. Вышка.Навигация [Электронный ресурс]. – URL: <https://play.google.com/store/apps/details?id=ru.hse.navigation> (дата обращения: 10.09.2025).

## CONVERTING IFC ARCHITECTURAL MODELS TO GEOJSON AND OPTIMIZING THEIR SIZE FOR NAVIGATION SYSTEMS

V. S. Popov, student  
E-mail: [po.vit.step@yandex.ru](mailto:po.vit.step@yandex.ru)  
Kaliningrad State Technical University

Scientific supervisor: Candidate of Technical Sciences  
T. V. Snytnikova

The article discusses the problem of adapting architectural models of buildings presented in the IFC format for the use in web-based navigation and geoinformation systems. The IFC format is widely used in building information modeling technology, however, due to the complexity of the structure, lack of geographical reference and significant amount of data, it is not suitable for navigation applications. An approach to converting IFC data into the GeoJSON format has been proposed, which ensures compatibility with modern web technologies and interactive cartographic services. The study describes methods for optimizing the volume of output files, including filtering objects, removing redundant attributes, rounding coordinates, and simplifying geometry. Examples of the implementation of these operations using the ifc2gis tool are presented. The results of the study shows that the application of the proposed methods makes it possible to reduce the size of the final GeoJSON file by up to 49 % while maintaining geometric accuracy and the necessary attributes of

the premises. As a result, the speed of loading and displaying maps increases, which makes it possible to effectively use BIM architectural models in the server and client parts of navigation systems.

**Keywords:** *IFC, JSON, GeoJSON, BIM, CAD, maps, indoor maps, navigation.*