

Т. С. Шевцова¹✉, Н. С. Головачев¹

Разработка программного средства анализа и проектирования озеленения жилых территорий на основе спутниковых данных и моделей застройки

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: shevchova004@gmail.com

Аннотация. В статье представлены основные результаты разработки программного средства для автоматизированного анализа и проектирования озеленения жилых территорий. Для реализации были использованы язык Python с библиотеками геоанализа (OpenCV, NumPy, Rasterio, GeoPandas, Shapely) и среда Unreal Engine 5 для визуализации. Программное средство состоит из двух аналитических модулей и модуля визуализации. Первый модуль по спутниковому снимку автоматически выделяет существующие деревья и формирует таблицу их координат и параметров. Второй модуль с учётом нормативных отступов, зон ограничений и инсоляции генерирует рекомендации по размещению новых деревьев. Третий модуль в Unreal Engine загружает полученные таблицы и отображает существующие и проектируемые деревья на трёхмерной сцене застройки. Приведены результаты тестирования, подтверждающие работоспособность и производительность разработанного решения.

Ключевые слова: озеленение жилых территорий, спутниковые данные, NDVI, Unreal Engine, Python, геоинформационные системы, визуализация, генерация рекомендаций

T. S. Shevtsova¹✉, N. S. Golovachev¹

Development of a software tool for analysis and design of residential area landscaping based on satellite data and building models

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: shevchova004@gmail.com

Abstract. The article presents the main results of the development of a software tool for the automated analysis and design of landscaping in residential areas. Python with geospatial analysis libraries (OpenCV, NumPy, Rasterio, GeoPandas, and Shapely) and Unreal Engine 5 for visualization were used for the implementation. The software consists of two analytical modules and a visualization module. The first module automatically detects existing trees from a satellite image and generates a table containing their coordinates and parameters. The second module generates recommendations for the placement of new trees, taking into account regulatory setback distances, restricted zones, and insolation conditions. The third module, implemented in Unreal Engine, imports the generated tables and visualizes existing and proposed trees within a three-dimensional urban development scene. Testing results confirming the functionality and performance of the developed solution are presented.

Keywords: residential landscaping, satellite data, NDVI, Unreal Engine, Python, geographic information systems, visualization, recommendation generation

Введение

Проблема системного озеленения жилых территорий становится всё более актуальной в условиях урбанизации. Зелёные насаждения улучшают микроклимат, снижают шум, очищают воздух и повышают комфорт проживания. Однако существующие подходы к проектированию озеленения часто либо трудоёмки (ручной обход территории, измерение каждого дерева), либо не обеспечивают наглядности (плоские схемы, таблицы). С развитием спутниковых технологий и систем трёхмерной визуализации появилась возможность автоматизировать процесс инвентаризации существующих деревьев и оптимизировать размещение новых, соблюдая нормативные требования и учитывая экологические факторы.

Цель работы – разработка программного средства, которое по спутниковым данным и модели застройки автоматически формирует таблицу существующих деревьев, генерирует рекомендации по посадке новых с учётом ограничений и предоставляет интерактивную 3D-визуализацию в Unreal Engine [1].

Исследование предметной области

Существующие методы проектирования озеленения можно разделить на три группы:

Традиционное ручное проектирование – выезд специалиста на местность, фотофиксация, нанесение деревьев на план. Недостатки: высокая трудоёмкость, субъективность, невозможность точного учёта инсоляции и затенения от зданий.

ГИС-системы (QGIS, ArcGIS) – позволяют обрабатывать спутниковые данные, вычислять вегетационные индексы, строить карты. Однако они не автоматизируют генерацию рекомендаций по посадке и не предоставляют встроенных средств фотореалистичной трёхмерной визуализации [2, 3].

Архитектурные визуализаторы (Twinmotion, Lumion) – дают качественную 3D-картинку, но растения расставляет пользователь вручную, без опоры на геоаналитику и нормативные ограничения.

Таким образом, существует пробел между аналитическими инструментами и средствами визуализации. Предлагаемое решение заполняет его, объединяя автоматическую обработку спутниковых снимков, генерацию рекомендаций по правилам и визуализацию в современном игровом движке.

Методы и технологии

Для реализации программного средства был выбран следующий стек технологий. Основным языком разработки аналитических модулей является Python [4]. Для обработки спутниковых снимков и выделения крон деревьев использовались библиотеки rasterio [5] (чтение GeoTIFF и геопривязка), numpy [6] (математические операции), opencv [7] (морфологическая фильтрация и поиск связанных компонент). Вычисление вегетационного индекса NDVI выполнялось по красному и ближнему инфракрасному каналам. Для работы с векторными данными (граница участка, зоны ограничений, существующие деревья) применялись библиотеки geopandas [8] и shapely [9] – они позволяют строить буферные зоны, выполнять операции вычитания полигонов и проверять вхождение точек.

Конфигурационные параметры (буферы, минимальные расстояния, шаг сетки) вынесены в файл формата YAML и загружаются через `ruyaml` [10]. Все таблицы сохраняются в формате CSV с использованием `pandas` [11].

Визуализация реализована в среде Unreal Engine 5.6. Выбор движка обусловлен фотореалистичным рендерингом, системой динамического освещения Lumen, а также встроенными механизмами уровней детализации (LOD), которые позволяют отображать тысячи деревьев без падения производительности. Загрузка CSV-файлов и расстановка 3D-моделей деревьев выполнены на языке визуального программирования Blueprints. Для каждой строки таблицы производятся проверка валидности данных, конвертация географических координат в мировые координаты сцены и создание актора (Spawn Actor) с соответствующей моделью. Модели деревьев (берёза, дуб, сосна, кустарник) созданы в Blender [12] и имеют по четыре уровня детализации: максимальная детализация на расстоянии до 30 метров, средняя – до 80 метров, низкая – до 200 метров, и спрайт (билборд) – далее 200 метров. Подготовка геоданных (контуры зданий, дорог, зелёных зон) выполнена с помощью аддона BlenderGIS [12], который загружает информацию из открытого источника OpenStreetMap.

Для запуска аналитических модулей и вызова Unreal Engine разработан графический интерфейс на Tkinter [13], предоставляющий три кнопки: «Функция 1» – анализ существующих деревьев, «Функция 2» – генерация рекомендаций, «Функция 3» – открытие сцены в Unreal Engine. Все три модуля обмениваются данными через CSV-файлы, что обеспечивает их независимость и возможность модификации.

Обобщённая структура разработанного программного средства представлена на рис. 1. Схема отражает модульную организацию системы и последовательность обмена данными между аналитическими компонентами, пользовательским интерфейсом и средой визуализации Unreal Engine.

Работа программного средства начинается с запуска проекта пользователем и перехода в главное меню Unreal Engine. Из главного меню доступны три функциональных направления: анализ существующих древесных насаждений по спутниковым данным, генерация рекомендуемых точек размещения новых деревьев и открытие уровня с визуализацией результатов. Первый функциональный блок предназначен для загрузки исходного спутникового изображения и его последующей обработки в Python-модуле 1. В рамках данного этапа выполняются операции предварительной фильтрации изображения, выделения растительности, определения крон деревьев и формирования результирующей таблицы существующих деревьев. Таблица 1 содержит идентификатор объекта, координаты и тип или вид дерева, если такая классификация предусмотрена входными данными или результатами дешифрирования.

Второй функциональный блок реализует генерацию новых точек размещения древесных насаждений. Пользователь задаёт ограничения проектирования, включая минимальные отступы, пустые зоны, области инсоляции, запретные зоны и иные пространственные параметры. Указанные ограничения передаются в Python-модуль 2, который использует результаты анализа существующих дере-

вьев, данные о территории и заданные пользователем параметры. На выходе формируется таблица 2, содержащая рекомендуемые новые деревья с указанием идентификатора, координат, вида и ограничений, повлиявших на выбор каждой точки размещения.

Третий функциональный блок отвечает за визуальное представление результатов в 3D-сцене Unreal Engine. Модуль визуализации считывает данные из таблиц 1 и 2, выполняет преобразование координат в систему координат сцены и отображает соответствующие объекты в составе цифровой модели территории. В результате в единой 3D-сцене отображаются существующие деревья, рекомендуемые новые посадки, застройка и пространственные ограничения. Такая организация обеспечивает разделение задач между аналитическими модулями и модулем визуализации, упрощает отладку отдельных компонентов и позволяет независимо модифицировать алгоритмы обработки спутниковых данных, генерации рекомендаций и отображения объектов в Unreal Engine.

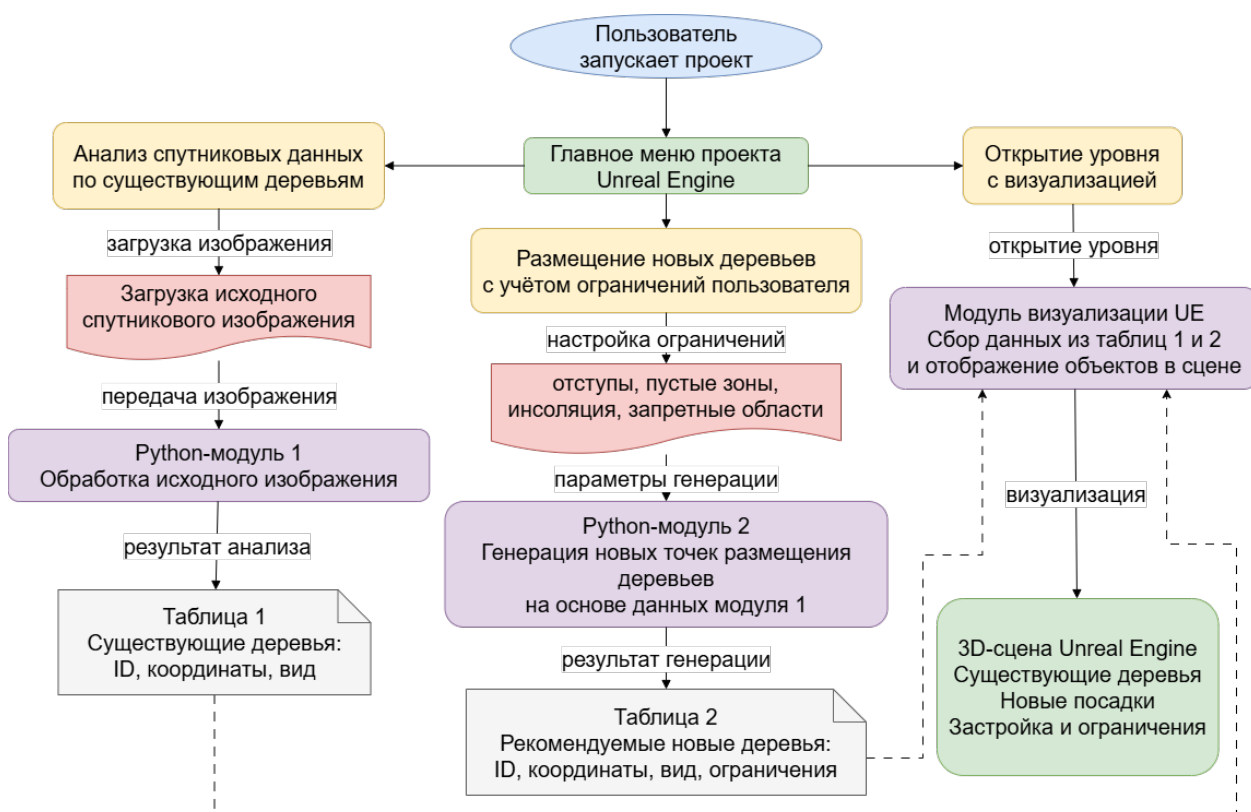


Рис. 1. Концептуальная схема функционирования системы

Результаты

В результате работы было создано программное средство, реализующее три основные функции. Первая функция – анализ спутниковых данных – по загруженному GeoTIFF вычисляет NDVI, выделяет кроны, формирует файл `existing_trees.csv` (рис. 2).

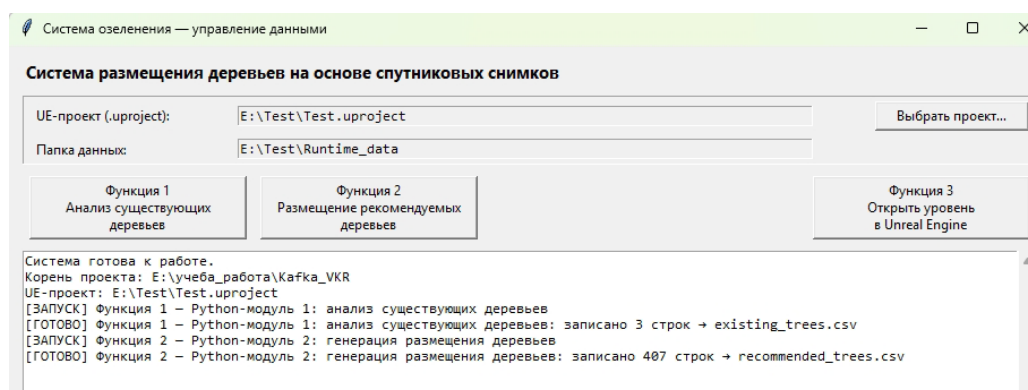


Рис. 2. Интерфейс мобильного приложения

На тестовом квартале площадью 0,5 км² обнаружено 187 существующих деревьев. Вторая функция – генерация рекомендаций – с учётом нормативов (отступ от зданий 5 м, от дорог 1 м, от существующих деревьев 4 м) и зон инсоляции сгенерировала 407 рекомендованных мест посадки. Все точки находятся внутри границы участка и не пересекают запретные зоны. Третья функция – визуализация в Unreal Engine – обеспечивает стабильную частоту кадров 58–72 FPS на конфигурации i5-12450H / RTX 3050 Ti / 16 ГБ при отображении 1000 деревьев; благодаря использованию LOD нагрузка на GPU снижена до 65 %.

Проведено тестирование: 10 модульных тестов (pytest), 3 интеграционных, 4 проверки валидации выходных данных – все 17 тестов пройдены. Выходные CSV-файлы полностью совместимы с Unreal Engine, координаты переведены в сантиметры (масштаб ×100). На данный момент все данные пользователя хранятся в памяти мобильного телефона. Использование таблиц для экспорта позволит пользователю упростить взаимодействие с лечащим врачом-эндокринологом, которому необходимо регулярно следить за состоянием пациента. Обучающие карточки позволят, в свою очередь, лучше разобраться как с функционалом мобильного приложения, так и, частично, с проблемными вопросами, с которым может столкнуться пользователь.

Заключение

Разработанное программное средство позволяет автоматизировать трудоёмкий процесс инвентаризации существующих деревьев и проектирования новых посадок. Оно объединяет возможности геоаналитики (NDVI, буферные зоны, сетка кандидатов) и фотореалистичной 3D-визуализации, доступной для неспециалистов. Решение может быть использовано ландшафтными архитекторами, студентами профильных направлений и муниципальными службами. В перспективе планируется добавить нейросетевое распознавание видов деревьев [14], интеграцию с BIM-моделями и поддержку виртуальной реальности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Epic Games. Unreal Engine 5 Documentation. – URL: <https://docs.unrealengine.com> (дата обращения: 10.05.2026).

2. Коновалова, Н. В. ГИС в ландшафтном планировании / Н. В. Коновалова. – Новосибирск: СГУГиТ, 2019. – 180 с.
3. Jensen, J. R. Remote Sensing of the Environment: An Earth Resource Perspective / J. R. Jensen. – Pearson, 2016. – 656 p.
4. Python Software Foundation. Python Language Reference. – URL: <https://docs.python.org> (дата обращения: 10.05.2026).
5. Rasterio Documentation. – URL: <https://rasterio.readthedocs.io> (дата обращения: 10.05.2026).
6. NumPy Documentation. – URL: <https://numpy.org/doc> (дата обращения: 10.05.2026).
7. OpenCV. Image Processing. – URL: <https://opencv.org> (дата обращения: 10.05.2026).
8. GeoPandas Documentation. – URL: <https://geopandas.org> (дата обращения: 10.05.2026).
9. Shapely Documentation. – URL: <https://shapely.readthedocs.io> (дата обращения: 10.05.2026).
10. PyYAML Documentation. – URL: <https://pyyaml.org> (дата обращения: 10.05.2026).
11. Pandas Documentation. – URL: <https://pandas.pydata.org> (дата обращения: 10.05.2026).
12. Blender Foundation. BlenderGIS Add-on. – URL: <https://github.com/domlysz/BlenderGIS> (дата обращения: 10.05.2026).
13. Tkinter Documentation. Python standard library. – URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 10.05.2026).
14. Рыбалкин, Д. А. Методы автоматизированного выделения древесной растительности по спутниковым снимкам / Д. А. Рыбалкин // Геопрофи. – 2024. – № 3. – С. 34–41.

© Т. С. Шевцова, Н. С. Головачев, 2026