УДК 001.817

ГИБРИДНЫЙ МЕТОД 3D-РЕКОНСТРУКЦИИ СЦЕН С ИСПОЛЬЗОВАНИЕМ ТРАНСФОРМЕРНОЙ МОДЕЛИ VGGT И ДАННЫХ СЕНСОРА ГЛУБИНЫ В СРЕДЕ ROS 2

Д. И. Елисеев

Белорусский государственный университет,
Минск, Беларусь, danilae007@gmail.com

В данной статье представлен модульный программный комплекс, разработанный в ROS 2, для высоко детальной реконструкции 3D-сцен. Гибридный подход сочетает в себе облака точек из нейронной сети VGGT (полученные из 2D-изображений) и аппаратный датчик глубины для обеспечения метрической точности. Система, состоящая из трех независимых узлов ROS 2, управляемых через сервисы, обеспечивает гибкий сбор данных и эффективное использование ресурсов. Алгоритм Iterative Closest Point (ICP) выравнивает облака точек. Этот конвейер позволяет создавать полные текстурированные 3D-модели, объединяя детали нейронной сети с точностью аппаратных датчиков.

Ключевые слова: 3D-реконструкция; VGGT; ROS 2; облако точек; слияние данных; ICP; компьютерное зрение.

A HYBRID METHOD FOR 3D SCENE RECONSTRUCTION USING THE VGGT TRANSFORMER MODEL AND DEPTH SENSOR DATA IN THE ROS 2 ENVIRONMENT

D. I. Eliseev

Belarusian State University,
Minsk, Belarus, danilae007@gmail.com

This paper introduces a modular software suite, developed in ROS 2, for high-detail 3D scene reconstruction. The hybrid approach combines point clouds from the VGGT neural network (derived from 2D images) and a hardware depth sensor for metric accuracy. The system, comprising three independent ROS 2 nodes controlled via services, offers flexible data acquisition and efficient resource utilization. The Iterative Closest Point (ICP) algorithm aligns the point clouds. This pipeline enables complete, textured 3D models, merging neural network detail with hardware sensor accuracy.

Keywords: 3D reconstruction; VGGT; ROS 2; point cloud; data fusion; ICP; computer vision.

1. Введение

Задача трехмерной реконструкции окружающего пространства является одной из ключевых в областях робототехники, беспилотного транспорта и дополненной реальности. Традиционно для ее решения используются два основных подхода: активные сенсоры (лидары, структурированный подсвет, ToF-камеры) и пассивные методы фотограмметрии. Активные сенсоры обеспечивают высокую метрическую точность, но могут страдать от низкой плотности данных, артефактов на отражающих поверхностях и неспособности реконструировать текстуру. Классическая фотограмметрия (SfM, MVS) позволяет получать плотные текстурированные модели, но требовательна к условиям съемки и вычислительно сложно.

В последние годы активно развиваются методы 3D-реконструкции на основе глубокого обучения, такие как Neural Radiance Fields (NeRF) и трансформерные модели Multi-View Stereo (MVS). Модель VGGT (Vision Geometry-Grounded Transformer) является одним из передовых решений в этой области, способным восстанавливать 3D-геометрию и позы камер из неупорядоченного набора изображений. Однако такие модели часто работают в собственном, неопределенном масштабе и требуют выравнивания с реальными метрическими данными для практического применения.

Целью данной работы является разработка и апробация гибкого программного комплекса в стандартной для робототехники среде ROS 2, который реализует гибридный подход к 3D-реконструкции, объединяя преимущества нейросетевых методов (высокая детализация) и активных сенсоров (метрическая точность).

2. Архитектура и методы

Разработанный программный комплекс построен на микросервисной архитектуре с использованием фреймворка ROS 2 Humble. Пайплайн разделен на три независимых узла, взаимодействие между которыми осуществляется по требованию через вызовы сервисов (рис. 1).

- Узел захвата изображений (snapshot_node). Отвечает за сбор и организацию исходных данных. При первом вызове сервиса /capture_snapshot он создает уникальную директорию сессии с временной меткой и публикует путь к ней в специальный топик для других узлов. При каждом последующем вызове он сохраняет в эту директорию синхронизированный набор из трех кадров (стереопара с инфракрасных сенсоров и цветное изображение).

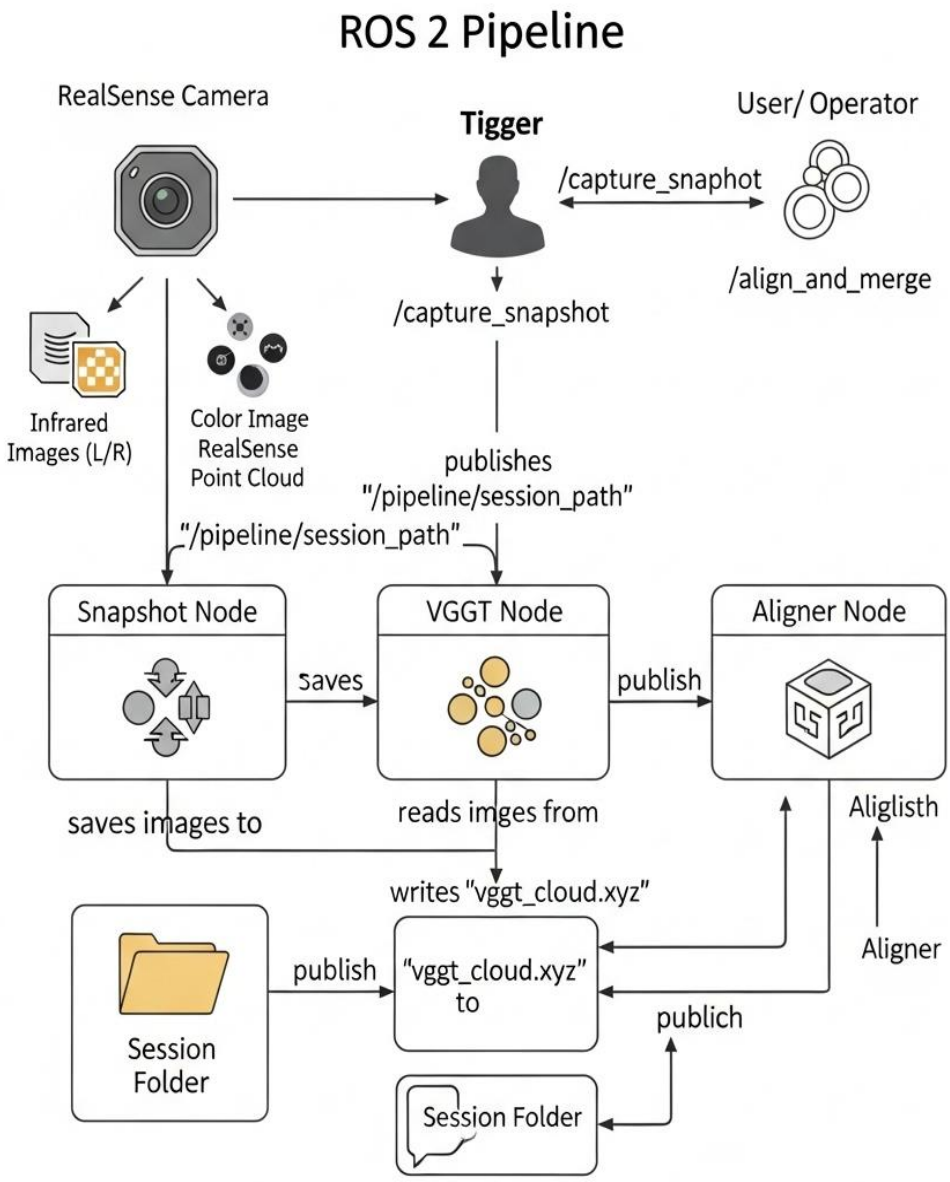

Рис. 1. Архитектурная схема пайплайна 3D-реконструкции

- Узел нейросетевой реконструкции (vggt_node). Является вычислительным ядром системы. По вызову сервиса /reconstruct_vggt он загружает модель VGGT в память GPU, обрабатывает все сохраненные в сессии изображения, генерирует облако точек и позы камер, после чего выгружает модель, освобождая ресурсы.
- Узел совмещения (aligner_node). Отвечает за финальное слияние данных. По вызову сервиса /align_and_merge он получает последнее облако точек от аппаратного сенсора, загружает облако, сгенерированное vggt_node, и выполняет их совмещение.

3. Методы и алгоритмы

Синхронизация и сбор данных. Для одновременного захвата кадров с ИК-сенсоров (/camera/infra1/image_rect_raw, /camera/infra2/image_rect_raw) и цветной камеры (/camera/color/image_raw), которые могут иметь неидеально совпадающие временные метки, используется message_filters.ApproximateTimeSynchronizer. Профиль QoS для подписчиков настраивается на BEST_EFFORT для изображений и RELIABLE для облака точек, что соответствует стандартным настройкам драйвера Intel RealSense в ROS 2.

Нейросетевая реконструкция (VGGT). Используется модель VGGT-1B с 1 миллиардом параметров. Для управления высокими требованиями к видеопамяти (> 10 ГБ VRAM) реализован механизм загрузки/выгрузки модели по требованию. Входные изображения проходят предобработку: разрешение уменьшается вдвое для экономии памяти, после чего размер приводится к значениям, кратным размеру патча модели (14 пикселей), чтобы избежать ошибок. После реконструкции из модели извлекаются позы камер (матрицы 3x4), которые транслируются в виде TF-фреймов и сохраняются в файл.

Геометрическое выравнивание (ICP). Для совмещения облака точек от VGGT (source) и облака от RealSense (target) применяется алгоритм Iterative Closest Point (ICP), реализованный в библиотеке Open3D (рис. 2, 3). Перед запуском ICP оба облака проходят через процедуру Voxel Downsampling для уменьшения плотности, что повышает скорость и робастность алгоритма. ICP итеративно находит матрицу трансформации, минимизирующую расстояние между соответствующими точками двух облаков. Найденная трансформация применяется к исходному, полноразмерному облаку VGGT.

4. Экспериментальный рабочий процесс и результаты

Рабочий процесс полностью управляется из командной строки, что позволяет гибко контролировать каждый этап. После запуска системы командой ros2 launch vggt_ros_node pipeline.launch.py оператор выполняет последовательность вызовов сервисов:

1) ros2 service call /capture_snapshot std_srvs/srv/Trigger (может вызываться многократно);
2) ros2 service call /reconstruct_vggt std_srvs/srv/Trigger;
3) ros2 service call /align_and_merge std_srvs/srv/Trigger.

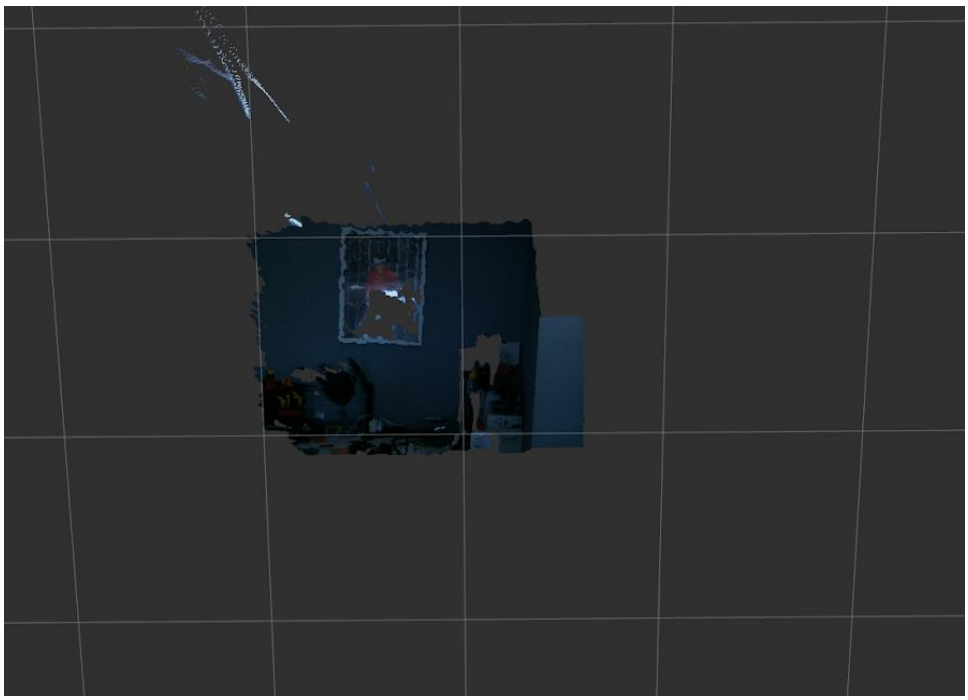

Рис. 2. Сравнение облаков точек: RealSens

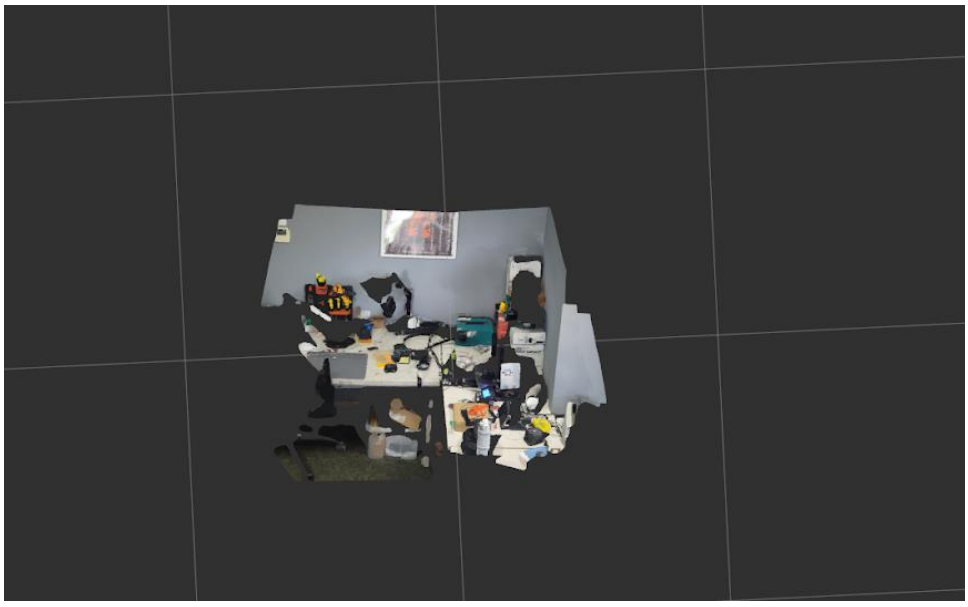

Рис. 3. Сравнение облаков точек: VGGT

Тестирование производительности пайплайна проводилось на наборе из 4 цветных изображений, снятых камерой Intel RealSense D415. Измерения проводились на системе с GPU Nvidia RTX 3080 (12 ГБ). В табл. 1–2 представлены данные о потреблении видеопамяти (VRAM) на этапе работы узла vggt_node и детальный хронометраж всех этапов выполнения пайплайна.

Таблица 1

Потребление видеопамяти на этапе реконструкции VGGT

Метрика	Значение, МБ
Загруженная модель	4798,53
Модель + данные (4 изображения)	4808,62
Пиковое потребление при инференсе	8397,60

Примечание. Потребление видеопамяти на этапе реконструкции VGGT

Таблица 2

Хронометраж выполнения пайплайна (4 изображения)

Этап	Время, с	Доля, %
1.1 Загрузка модели	12,35	64,1
1.2 Загрузка данных	0,06	0,3
1.3 Предобработка изображений	0,01	0
1.4 Инференс	1,29	6,7
1.5 Постобработка и сохранение	2,12	11
2.1 Загрузка данных для выравнивания	0,31	1,6
2.2 Генерация облака из карты глубины	1,02	5,3
2.3 Выравнивание (PnP)	0,14	0,7
2.4 Слияние облаков и сохранение	1,67	8,7
ИТОГО	19,26	100

Примечание. Хронометраж выполнения пайплайна (4 изображения)

Данные подтверждают, что сама модель VGGT-1B является основным потребителем ресурсов (~4,8 ГБ). Важно отметить, что пиковое потребление памяти во время прямого прохода (инференса) почти вдвое выше, что делает невозможным использование модели на GPU с объемом памяти менее 10–12 ГБ при работе с изображениями высокого разрешения.

Анализ хронометража показывает, что загрузка модели в память (12,35 с) является доминирующей операцией, занимая 64,1% от общего времени выполнения. Сами вычисления (инференс 1,29 с, выравнивание 0,14 с) происходят значительно быстрее. Этот результат полностью оправдывает выбранную архитектуру с динамической загрузкой и выгрузкой модели, так как позволяет высвобождать значительные ресурсы GPU в промежутках между операциями реконструкции.

Библиографические ссылки

1. VGGT: Vision Geometry-Grounded Transformer for Large-Scale 3D Reconstruction / J. Wang [et al.] // GitHub. URL: https://github.com/facebookresearch/vggt (date of access: 09.09.2025).

2. *Zhou Q. Y.* Open3D: A Modern Library for 3D Data Processing // arXiv preprint arXiv:1801.09847. 2018.

3. ROS 2 Documentation. URL: https://docs.ros.org/humble (date of access: 09.09.2025).