



Московский
транспорт

ТЕХНИЧЕСКОЕ ЗАДАНИЕ ЗАДАЧА 5

Редактор лидарных карт для автоматического удаления динамических объектов



1. Контекст и актуальность задачи

Лидарные карты — это высокоточные трёхмерные модели окружающей среды, применяемые в автономном транспорте, навигации и городском планировании. По данным лидара можно с высокой точностью определять своё положение на заранее построенной карте. Для надёжной локализации карта должна быть построена корректно и очищена от временных и подвижных объектов (автомобили, пешеходы, трамваи и т.п.), которые мешают анализу статической среды.

Ваша задача — разработать инструмент для очистки лидарных карт от таких «шумов». Можно использовать нейросети, ручное редактирование или их комбинацию. Решение должно максимально точным и удобным для пользователя.

2. Описание задачи

Участникам необходимо создать ПО, которое:

- **Автоматически** находит и удаляет динамические объекты (автомобили, людей, трамваи) из PCD-файлов.
- **Позволяет вручную корректировать результат:** выделять объекты, расширять и уменьшать области с объектами, добавлять/удалять точки, копировать-вставлять области.
- Поддерживает **3D-визуализацию** с возможностью вращения, приближения.
- Сохраняет результат в **PCD-формате**.

Дополнительные возможности:

- Добавление точек вручную (в плоскости или объёме).
- Поддержка скриптов для автоматизации добавления или удаления точек.
- Интеграция с CloudCompare или другими инструментами.

3. Программно-аппаратные требования

3.1. Аппаратные требования

- Десктоп приложение под Ubuntu.
- Обработка файлов **размером в несколько ГБ**.

3.2. Программные требования

- **Предпочтительный стек для работы с облаками точек:** Python, C++, PCL для работы с облаками точек.
- **Предпочтительный стек если используются нейросети:** PyTorch/TensorFlow.
- **Визуализация:** возможна разработка как отдельного приложения, так и плагина к существующему (например, CloudCompare), так и web решения.

- **Допустимо:** использование open-source ПО и библиотек.
- **Ограничения:** избегать проприетарного ПО из "недружественных" стран.

4. Требования к презентации/демонстрации

- Показать **работу алгоритма** на тестовом PCD-файле.
- Продемонстрировать **интерфейс** (если есть).
- Объяснить, как оценивалось качество очистки облака точек.

5. Требования к сопроводительной документации

- Описание **архитектуры решения**.
- Инструкция по **запуску и использованию**.
- Обоснование выбора **алгоритмов и технологий**.

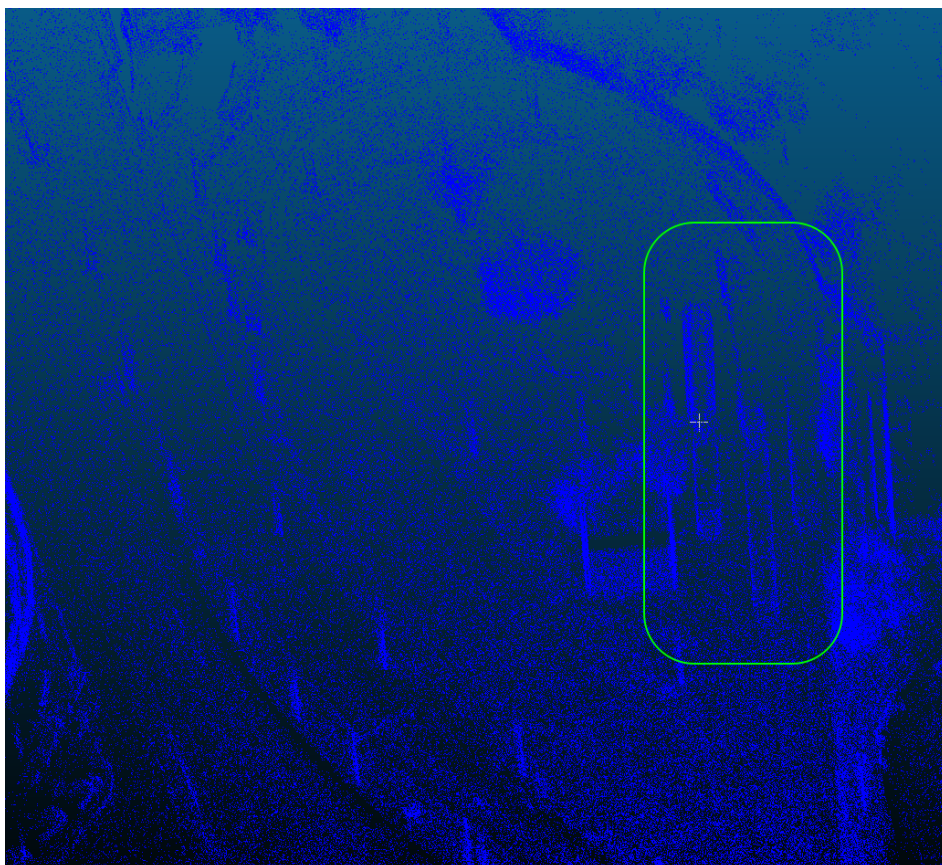
6. Ресурсы

- **Формат данных:** PCD (на вход и выход).
- **Тестовые данные:** предоставляются (с динамическими объектами).

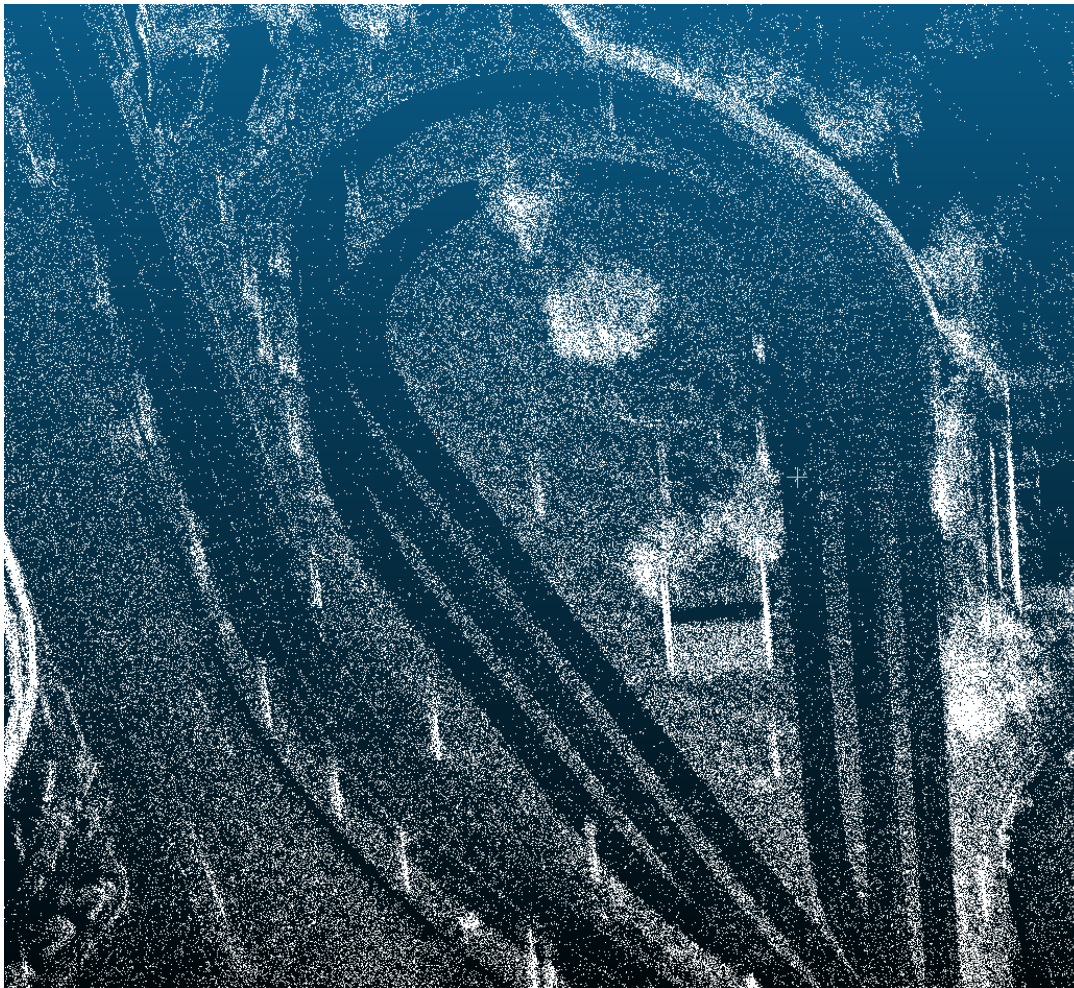
7. Метрика качества

Визуальная оценка.

Пример облака точек с динамическими объектами



Пример облака точек после удаления динамических объектов



8. Критерии оценки

Критерий	Вес
Автоматический поиск и удаления объектов	45%
3D визуализация	10%
Создание инструментов для ручного редактирования	35%
Презентация	10%

Что будет круто?

- Хорошая точность автоматического удаления.
- Удобный интерфейс для ручного редактирования.

- Подсвечивание спорных областей для ручной правки.

Готовы к нестандартным решениям — предлагайте свои идеи!

9. Приложение

К заданию прилагается два файла как ориентир ожидаемого результата, чтобы командам было проще понять, какой эффект требуется достичь и как его можно существенно улучшить:

- **points.pcd** — исходная карта после склеивания лидарных снимков вдоль маршрута движения со всеми артефактами динамики;
- **processed_points.pcd** — базовая («лёгкая») обработка, в которой удалена часть динамики только на дорожной сети (автомобильные и трамвайные пути).

Применённый алгоритм обработки имеет ряд недостатков:

- обрабатываются только дороги;
- возможны ложные удаления (перерезанные бордюры, разметка) и неполные удаления (части автомобилей остаются);
- удаляется значительная часть точек, принадлежащих дорожной поверхности.

Желаем командам удачи в поиске более эффективных решений!