Распознавание трехмерных объектов дорожной сцены по данным бортовых камер и лидаров автомобиля

Царин Илья
Научный руководитель - к.т.н. Д.А. Юдин

Актуальность и мотивация

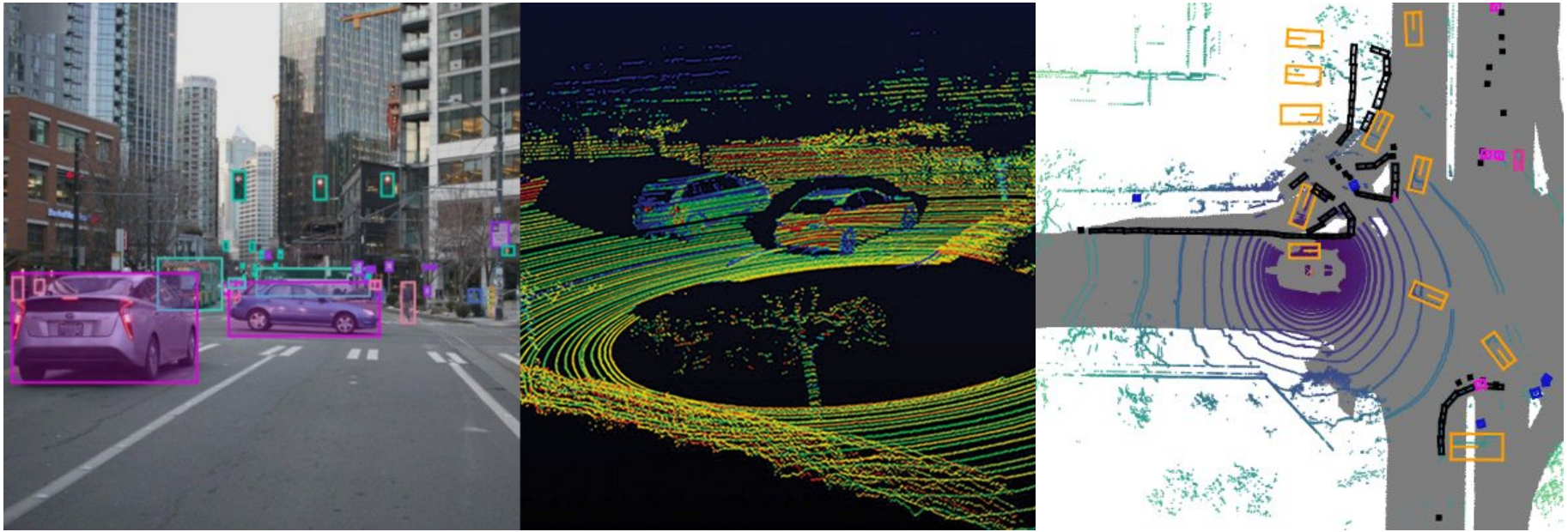
- Беспилотный транспорт и интеллектуальная робототехника становятся все более важными в повседневной жизни
- Важно улучшать алгоритмы распознавания 3D объектов с использованием лидарных данных в реальном времени

Постановка задачи

- Сравнить современные подходы лидарной 3d детекции по скорости и качеству работы
- Выбрать модель с учётом ограничений
- Провести эксперименты по её ускорению
- Реализовать ROS узел для применения модели в настоящем беспилотном автомобиле

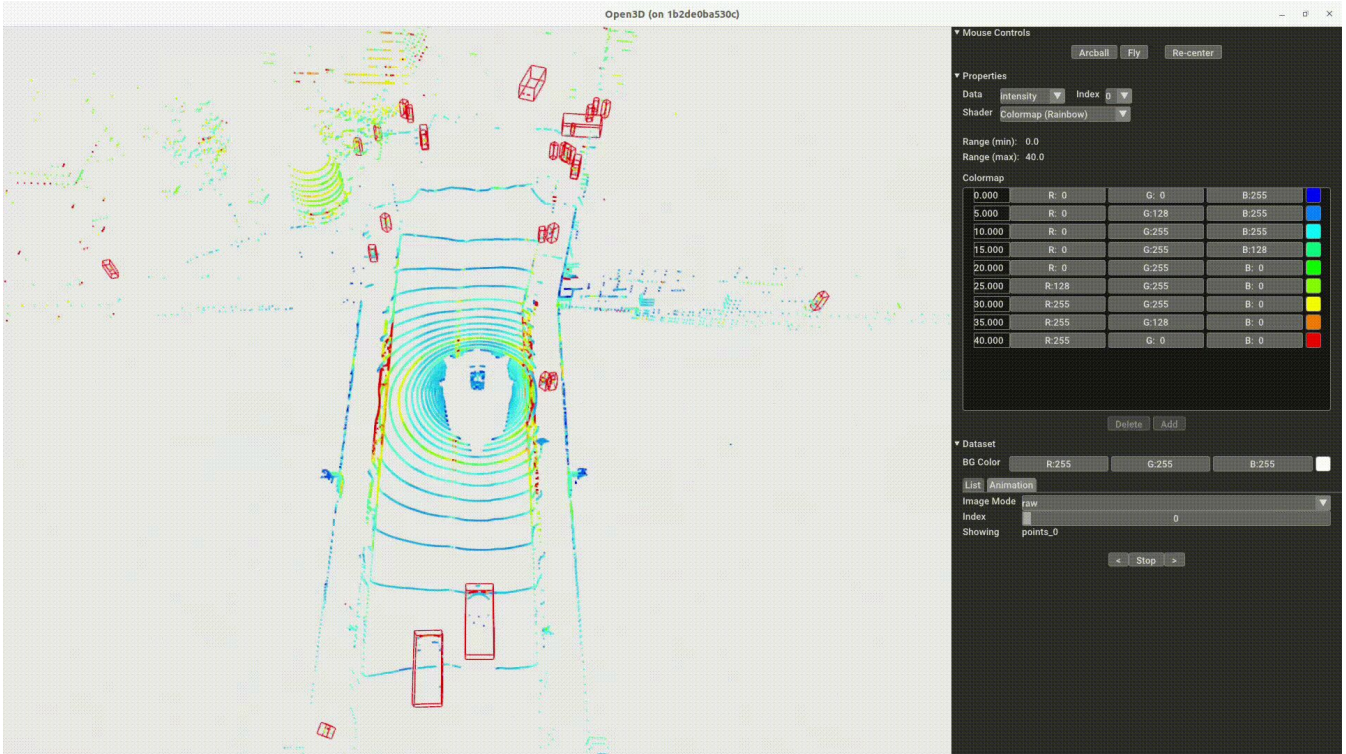
3D детекция на лидарных данных

- Есть данные лидаров по периметру автомобиля, хотим в режиме реального времени выделять объекты в bounding box’ы
- Ограничения: 100 мс на инференс на RTX 2080Ti



Результаты

- Написал код для визуализации работы моделей



Результаты

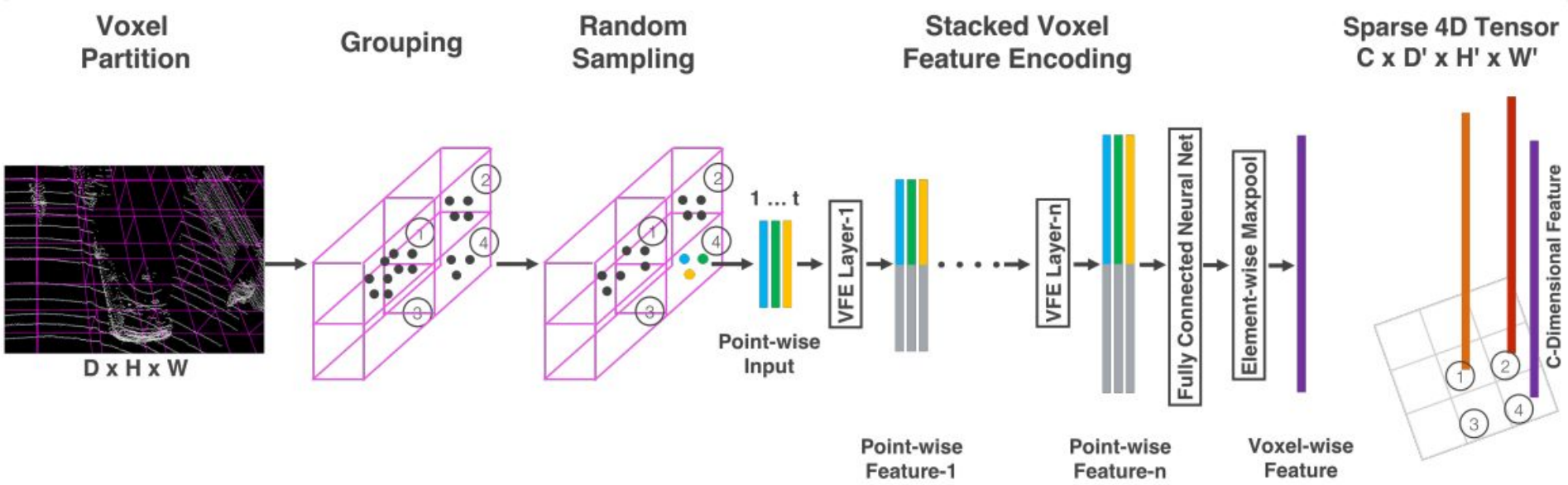
- Сравнил различные архитектуры по качеству и скорости работы
- Решил выбрать VoxelNeXt основной моделью
  - лучшее качество при наших ограничениях - 60.53 mAP на nuScenes при 77ms на RTX3060Ti

	mATE	mASE	mAOE	mAVE	mAAE	mAP	NDS
PointPillar-MultiHead	33.87	26.00	32.07	28.74	20.15	44.63	58.23
SECOND-MultiHead (CBGS)	31.15	25.51	26.64	26.26	20.46	50.59	62.29
CenterPoint-PointPillar	31.13	26.04	42.92	23.90	19.14	50.03	60.70
CenterPoint (voxel_size=0.1)	30.11	25.55	38.28	21.94	18.87	56.03	64.54
CenterPoint (voxel_size=0.075)	28.80	25.43	37.27	21.55	18.24	59.22	66.48
VoxelNeXt (voxel_size=0.075)	30.11	25.23	40.57	21.69	18.56	60.53	66.65

checkpoint name	mean inference time	std inference time	основано на
cbgs_voxel0075_voxelnext	76.65	22.44	voxelnext
cbgs_voxel0075_voxelnext_doubleflip	254.25	50.96	voxelnext
cbgs_dyn_pp_centerpoint	34.88	12.76	pointpillar
cbgs_pillar0075_res2d_centerpoint	82.11	12.76	pillarnet
cbgs_pp_multihead	31.97	7.41	pointpillar
cbgs_second_multihead	50.32	12.71	second
cbgs_voxel0075_res3d_centerpoint	74.71	15.99	voxelnet + centerpoint?
cbgs_voxel0075_voxelnext2d	96.91	21.80	voxelnext

VoxelNeXt - VFE

Voxel Feature Extractor -> Sparse Backbone -> Sparse Head



VoxelNeXt - Backbone

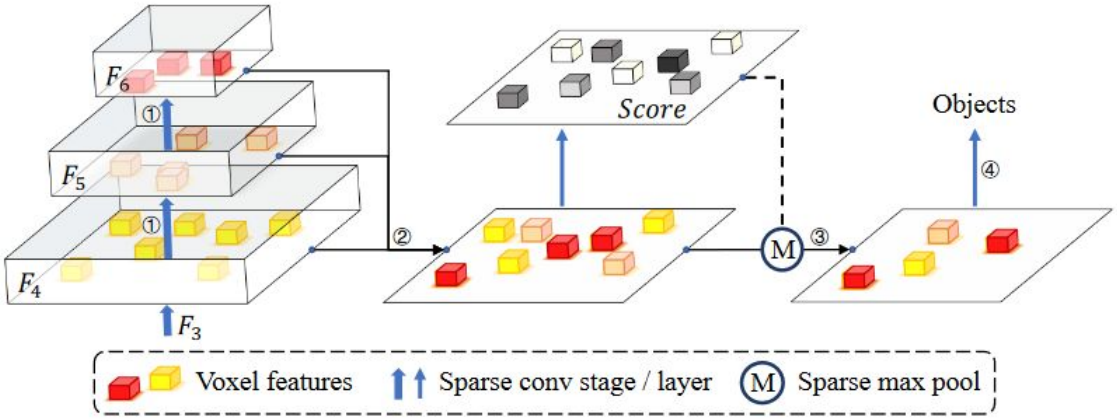
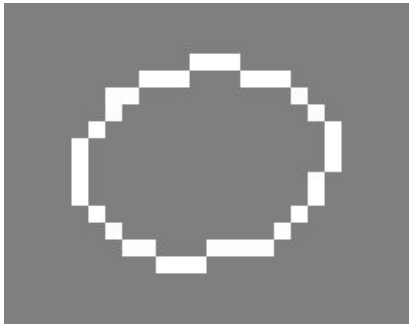
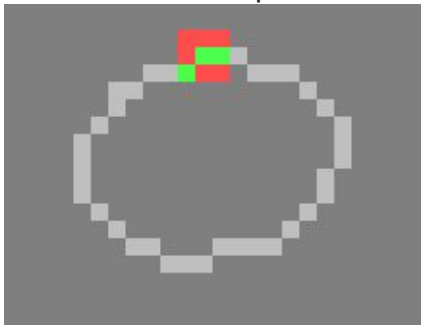


Figure 4. Detailed structure of VoxelNeXt framework. Circled numbers in the figure correspond to the paragraphs in Sections 3.1 and 3.2. 1 - Additional down-samplings. 2 - Sparse height compression. 3 - Voxel selection. 4 - Box regression. We omit the generation of F_1, F_2, and F_3 here for the simplicity sake.

With regular 3x3 convolutions, the set of active (non-zero) sites grows rapidly:

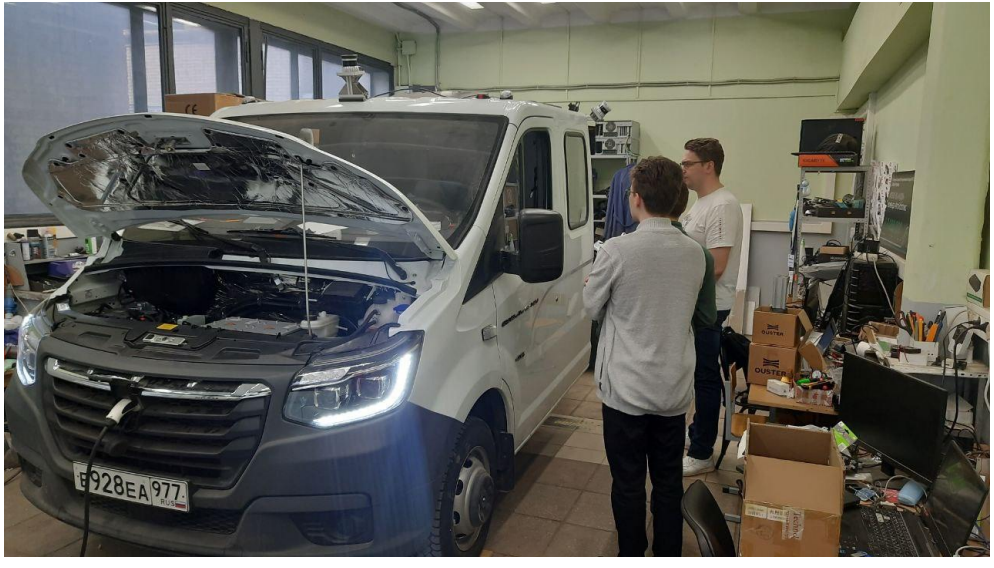


With Submanifold Sparse Convolutions, the set of active sites is unchanged and hence there is no computational overhead:



Результаты

- Пытался ускорять VoxelNeXt через TensorRT и ONNX
  - Submanifold Sparse Convolutions по умолчанию в этих фреймворках не поддерживаются, поэтому за адекватное время это сделать не получилось :(
- Сконвертировал в FP16 и добился ускорения инференса 77ms -> 67ms
- Написал ROS узел для инференса модели на настоящем автомобиле



Основные результаты

- Сравнил существующие подходы в лидарной 3d детекции
- Добился ускорения VoxelNeXt на 13% за счёт конвертации в FP16
- Написал ROS узел для realtime детекции в реальных условиях

Вклад

- Показаны практические аспекты применения и ускорения VoxelNeXt
- Сделан шаг в сторону улучшения алгоритмов надежного распознавания дорожной сцены, которые важны в беспилотном транспорте