

УДК 528.8:711.4

DOI 10.33764/2411-1759-2024-29-6-44-52

Разработка алгоритма классификации плотных облаков точек на примере городской застройки

Д. В. Василенко^{1,2✉}

¹ Московский физико-технический институт (Национальный исследовательский университет), г. Москва, Российская Федерация

² АО «Фирма «РАКУРС», г. Москва, Российская Федерация

e-mail: vasilenko.dv@phystech.edu

Аннотация. Данные дистанционного зондирования Земли (ДЗЗ), к которым относятся космическая и аэрофотосъемка, воздушное лазерное сканирование являются ценным источником информации об объектах и местности. В результате обработки таких данных с помощью специализированных программных решений создаются различные информационные продукты: ортофотопланы, цифровые модели рельефа и местности, 3D-модели строений, карты, топопланы и другие материалы, используемые в разных отраслях экономики. Появление новых типов съемочной аппаратуры, увеличение объема данных, качественный и количественный рост потребителей материалов ДЗЗ требует постоянного совершенствования программного обеспечения. Если использование нейронных сетей и классификация растров хорошо изучены в случае двумерных изображений, то классификация трехмерных нерегулярных облаков точек еще плохо исследована. В работе рассмотрены подходы к решению этой задачи – проекционные, воксельные, многослойные перцептроны и свертки точек. Цель работы – разработать алгоритм классификации облаков точек, полученных в результате лазерного сканирования или фотограмметрической обработки аэрофотосъемки. Объект исследования – методы автоматической классификации облаков точек. Предмет – алгоритм применения искусственной нейронной сети (ИНС) к облаку точек.

Ключевые слова: машинное обучение, глубокое обучение, искусственные нейронные сети, лазерное сканирование, аэрофотосъемка, фотограмметрия, облако точек

Для цитирования:

Василенко Д. В. Разработка алгоритма классификации плотных облаков точек на примере городской застройки // Вестник СГУГиТ. – 2024. – Т. 29, № 6. – С. 44-52. – DOI 10.33764/2411-1759-2024-29-6-44-52

Введение

Основная причина успеха ИНС в задачах разметки изображений – применение оператора свертки, который позволяет собирать информацию о локальном контексте, размещенном в двумерной решетке. Важные отличия облака точек от двумерной решетки – разреженность, неупорядоченность и нерегулярность структуры (рис. 1).

Из-за этих особенностей применение оператора свертки невозможно, рассмотрим далее подходы к возможным вариантам решения задачи классификации.

В качестве первого подхода можно рассмотреть применение методов обработки изображений к облаку точек. Для этого необходимо спроецировать облако точек на двумерную решетку и использовать готовые решения, применяемые для обработки изображений. Основные проблемы данного подхода связаны с потерей информации о трехмерной структуре объекта при проецировании на двумерную решетку и краевой эффект (некорректной обработкой перекрытий и областей с переменной плотностью точек). Возможно повышение качества проецирования точек, для этого выбираются плоскости, параллельные граням трех-

мерных объектов, однако результат классификации будет зависеть от точности вычисления нормалей и выбора граней [1–4].

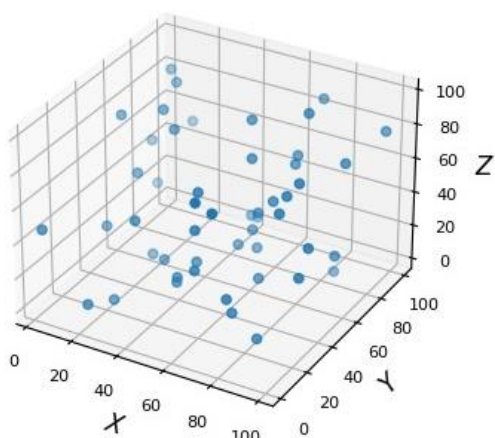


Рис 1. Схема нерегулярного облака точек как в плане, так и по высоте

Альтернативой является использование объемных пикселей (вокселей, VoxEl) – они могут описывать трехмерную структуру с потерей точности и иметь структуру, аналогичную изображению, из-за чего для них можно использовать двумерную свертку либо определить трехмерную. Основным недостатком данного подхода заключается в вычислительной сложности, так как зависимость от объема входных данных становится кубической, в связи с чем они на данный момент непригодны для обработки плотных облаков точек. Облака точек чаще всего описывают только поверхность объекта, большинство вокселей не будут заполнены. Обработка таких вокселей потребует значительного объема памяти, не принося пользы для задач обработки облака точек [5–6].

В работе PointNet [7] была предложена архитектура нейронной сети, работающая непосредственно с облаком точек. Для этого пришлось отказаться от использования свертки. При помощи многослойного перцептрона и агрегирующей функции выделяется набор глобальных признаков для всего облака точек, после чего многослойный перцептрон применяется к набору признаков каждой точки (собственных и глобальных) для определения класса.

В связи с низкой размерностью глобальных признаков, модель PointNet не может быть использована для разметки наборов данных, описывающих сложную поверхность. В качестве альтернативы может быть выбрана модель PointNet++ [8], использующая иерархичные наборы признаков, для получения информации из локальной окрестности точки, что позволяет извлекать информацию о структуре сложных объектов.

Успех в классификации облаков точек, полученный при использовании полносвязных сетей [9–11], работающих непосредственно с облаком точек, позволил адаптировать оператор свертки к облакам точек, что рассмотрено подробно в работах KPConv [12] и FG-Net [13].

Далее рассмотрим общие компоненты алгоритмов классификации, работающих непосредственно с облаками точек.

Общие компоненты алгоритмов классификации

Каждый фрагмент облака точек или отдельный файл проходят несколько последовательных этапов обработки.

1. Устраняется нерегулярность входных данных путем укрупнения узлов выбранного шага сетки с сохранением барицентров каждого нового образованного узла.

2. Каждому центру узла после генерализации присваивается потенциал, принимающий случайное значение в диапазоне от 0 до 10^{-3} , что дает возможность равномерно обрабатывать все участки облака точек. Потенциал – это показатель того, насколько сильный вклад был внесен точками в окрестности данного центра в результат работы алгоритма.

3. Для быстрого поиска полученные точки сохраняются в KD-деревьях.

4. Формируются «блоки» для обработки. Формирование блоков для обработки выполняется следующим образом.

4.1. Выбор точек. Выбираются все точки, лежащие в определенном радиусе от центра с минимальным потенциалом. Потенциалы всех центров увеличиваются на величину, обратную квадрату расстояния до выбранного центра.

4.2. Добавление шума к координатам и цветам, поворот вокруг оси z на случайный

угол, игнорирование цветовых признаков с задаваемой вероятностью, случайное масштабирование на величину в заданном интервале.

4.3. Расчет зависимых параметров. Строятся матрицы, содержащие индексы соседей каждой точки, и новые уровни детализации блока точек (аналог уровней пирамиды для изображений), образованные генерализацией точек предыдущего уровня.

5. Сформированный блок подается на обработку в ИНС. В случае обучения на этом этапе выполняется подсчет функции потерь и коррекция весов, в случае же классификации производится сохранение результатов для каждой обрабатываемой точки.

Выбор сети и ее описание

В данной работе выбор нейронной сети осуществлялся на основе метрики объеди-

нения над пересечением (IoU) на наборе облаков точек, полученных лазерным сканированием офисных помещений, S3DIS [14]. Задача, описанная в [14], наиболее близка к решаемой в данной работе, так как описывает данные, аналогичные по структуре. Этот набор имеет небольшой объем и простую геометрию объектного состава, из-за чего получил широкое распространение в тестировании разрабатываемых ИНС. Таким образом, в открытом доступе можно найти метрики качества классификации данного набора практически для любой ИНС.

Согласно [15], наилучшие результаты продемонстрировала KPConv, которая и легла в основу разрабатываемого алгоритма.

Данная сеть представляет собой аналог ResNet [16], использующий адаптированный для облаков точек оператор свертки (рис. 2).

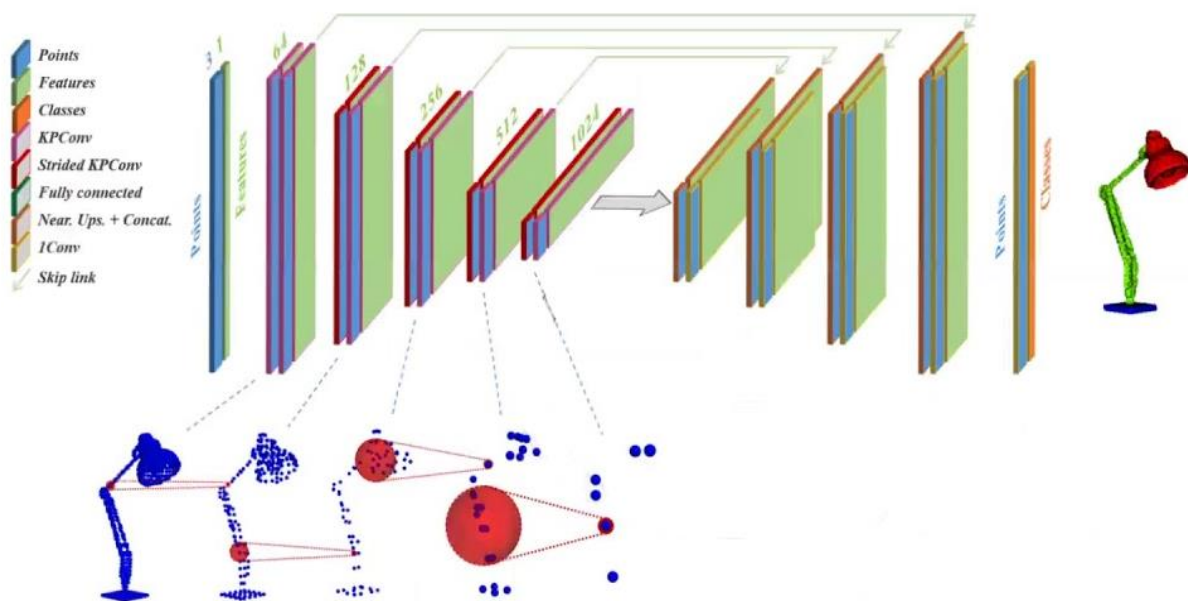


Рис. 2. Схема сети KPConv

Свертка F с ядром g в точке x описывается формулой:

$$(F \cdot g)(x) = \sum_{x_i \in N_x} g(x_i - x) f_i,$$

где x_i и f_i – координаты и признаки i -й точки соответственно; N_x – множество точек, лежащих в определенном радиусе от x .

Пусть $x_k : \{k < K\} \in R^3$ – множество точек ядра свертки, а $W_k : \{k < K\} \in R^3$ – соответствующие веса. Тогда ядро свертки можно определить как

$$g(y_i) = \sum_{k < K} h(y_i, x_k) W_k;$$

$$h(y_i, x_k) = \max \left(0, 1 - \frac{y_i - x_k}{\sigma} \right),$$

где σ – расстояние влияния свертки, выбирается в зависимости от плотности входных данных [12].

Такое определение свертки позволяет эффективно извлекать информацию из окрестности точки на различных уровнях детализации облака точек, используя значительно меньшее количество параметров, чем в полносвязных сетях.

При детальном анализе работы [12] в процессе обучения обнаружен ряд существенных недостатков модели ИНС, а именно:

– чрезмерная зависимость класса от превышения относительно точки с минимальной высотой, что связано с особенностью выборки, которую представляют, в основном, точки с плоским рельефом. Эта зависимость значительно повысила число ошибок в обла-

ках точек со сложным рельефом (при перепаде высот 20 м и выше земная поверхность классифицируется неверно, т. е. иные классы имеют максимум в этой области);

– значительный объем потребления видеопамяти;

– случайный обход контрольной выборки, и, как следствие, недетерминированность метрик;

– случайный обход при классификации, который приводит к сложности определения условия завершения работы и к отсутствию возможности отслеживать прогресс классификации;

– кластеры неправильно классифицированных точек, значительно уступающие по размерам классифицируемому объекту. Например, на рис. 3 изображен пример кластера точек, относящихся к проводам, ошибочно классифицированных как здание. При этом размер данного кластера намного меньше размеров зданий.

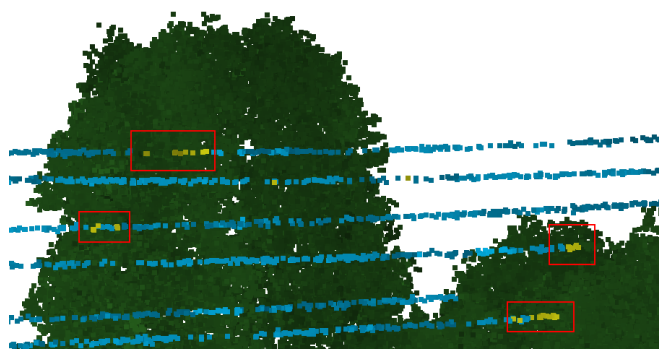


Рис. 3. Кластеры точек, ошибочно классифицированных как здание, показаны красными прямоугольниками

Все указанные проблемы приводили к нестабильному качеству обучения при повторных запусках, к неконтролируемому времени обучения и классификации.

Повышение надежности классификации

Для повешения качества и надежности классификации необходимо провести ряд существенных логических изменений при обработке сети.

1. Генерация и выбора центров блоков для инференса и валидации. Центры генерируются путем прореживания точек с шагом

$\frac{d}{\sqrt{n}}$, где d – размер блока (форма блока была изменена с шара на куб); n – минимальное число итераций на каждую точку. Блоки, описываемые данными точками, проходят обработку по одной итерации на каждую. Это позволяет устранить критическую секцию из многопоточной части, благодаря чему формирование блоков полностью выполняется параллельно. Количество блоков теперь определено заранее, что дает возможность отслеживать прогресс обработки. При обработке группы файлов реализованы чтение новых и запись классифицированных файлов. Это

позволило устранить случайность при подсчете метрик, ввести осмысленные условия выхода и повысить производительность в случае многопоточной обработки.

2. Выбор процессора. Нейросетевая обработка данных может выполняться как на центральном, так и на графическом процессоре. Графический процессор значительно более производительный для данной задачи, но ему доступен намного меньший объем оперативной памяти. Для достижения наилучшей производительности без потери качества в программе реализован автоматический выбор процессора для блока точек.

В процессе обработки блока точек рассчитывается необходимый объем памяти для его обработки. Наибольшее потребление памяти достигается на первом блоке resnet. В этом блоке используются тензоры следующих размеров:

- координаты соседних точек $n_0 \cdot n_1 \cdot 3$;
- расстояния до ядер свертки $n_0 \cdot n_1 \cdot n_{kp}$;
- спроецированные на ядра свертки признаки $n_0 \cdot n_{kp} \cdot n_{dim}$;

- выходные признаки ядер свертки

$$n_0 \cdot n_{kp} \cdot n_{dim};$$

- выходные точки с признаками

$$n_0 \cdot (3 + n_{dim}),$$

где n_0 – число точек; n_1 – число соседей; n_{kp} – число ядер свертки; n_{dim} – число выходных признаков.

По рассчитанному объему подбирается наиболее приоритетное устройство, имеющее необходимый объем памяти. Благодаря этому программа эффективно использует доступные процессоры.

3. Удаление высоты из признаков. В работе [12] в качестве признака использовалась относительная высота над уровнем пола офиса, а при обработке реальных данных физическая поверхность Земли выглядит сложнее. Поэтому использование высоты в качестве признака при обучении ухудшает качество классификации.

4. Моделирование рельефа. Для лучшего переноса обучения на области со сложным рельефом добавлен новый этап формирования блока, который выполняется после поворота, масштабирования и добавления шума к координатам точек. Цель данного этапа – сгенерировать новый блок с рельефом, отличным от исходного. Для этого с задаваемой вероятностью (по умолчанию 80 %) к точкам добавляется вертикальное смещение, описываемое суммой трех низкочастотных синусоид, направленных вдоль осей координат:

$$\sum_i^N a_{xi} \cos(f_{xi} \cdot x + p_{xi}) + \sum_i^N a_{yi} \cos(f_{yi} \cdot y + p_{yi}),$$

где f – равномерно распределенная на промежутке $\left\{ \frac{2\pi}{750}, \frac{2\pi}{150} \right\}$ случайная величина;

p – равномерно распределенная на промежутке $\{0, 2\pi\}$ случайная величина;

a – нормально распределенная случайная величина, $M(a) = 0$, $\sigma(a) = 5$.

Данная формула представляет собой описание трех членов ряда Фурье, что позволяет генерировать достаточно сложные функции, при этом ограничивая их амплитуду и спектр. Параметры распределений подобраны для генерации рельефа, наиболее приближенному к реальному. Ограничение минимального периода в 150 м позволяет избежать необходимости применения отдельного смещения для точек зданий, что было бы сложной задачей, учитывая отсутствие разметки отдельных объектов.

5. Приращение потенциалов. В исходной реализации потенциалы точек увеличивались на

$$\frac{1}{\left(1 - \frac{r^2}{R^2}\right)},$$

где r – расстояние от точки до центра блока; R – радиус блока. Это позволяло добиться равномерной обработки всех участков облака точек. Однако большая часть облака точек малоинформативна для обучения ИНС (например, содержит только плоскую часть земной поверхности). Для компенсации неинформативности таких областей добавлен множитель

к приращению потенциалов, равный значению функции потерь для данного блока. Таким образом, регионы с высокой ошибкой обрабатываются чаще, что должно компенсировать неравномерную представленность объектов в обучающей выборке.

6. Сглаживание. Для устранения кластеров неправильно классифицированных точек добавлен алгоритм постобработки путем сглаживания вероятностей. Для этого используется матрица соседей точек, определяемая на этапе формирования блока. Далее к вектору вероятностей каждой точки прибавляются вероятности ее соседей с весами, обратными евклидовому расстоянию между векторами признаков на выходе предпоследнего слоя ИНС.

Внесенные изменения были протестированы на ряде примеров с предварительно выполненной ручной разметкой для оценки величины повышения качества классификации.

Результаты

Для оценки качества классификации были обучены модели на наборе данных SensatUrban [17] с гиперпараметрами из исходной реализации. В данном наборе фотограмметрические данные получены с низкой высоты и, следовательно, с высокой степенью подробности (2 см на пиксель изображения). Общее количество точек составляет 2 162 млн.

После обучения подсчитывались метрики точности и объединения над пересечением на наборе данных STPL3D [18] (за исключением синтезированной части). От SensatUrban данный набор отличает наличие областей с холмистым рельефом, а также другой характер строений и растительности.

Таким образом, были получены количественные оценки качества классификации на наборе, отличном от обучающей выборки.

Изменение точности и объединения над пересечением для каждого класса при последовательном внесении модификаций изображено на рис. 4.

Анализируя графики, можно заметить, что каждая последующая модификация повышала качество классификации. Таким образом, после применения всех модификаций точ-

ность возросла на 9 %, а объединение над пересечением – на 7 %.

Снижена избыточность обработки облака точек при инференсе: предложенный метод обработал облако точек с избыточностью 14 %; исходный метод за то же число итераций избыточно обработал 39 % точек, а 12 % осталось необработанными (рис. 5). Разработанный алгоритм обрабатывает облако точек быстрее и за заранее известное число итераций.

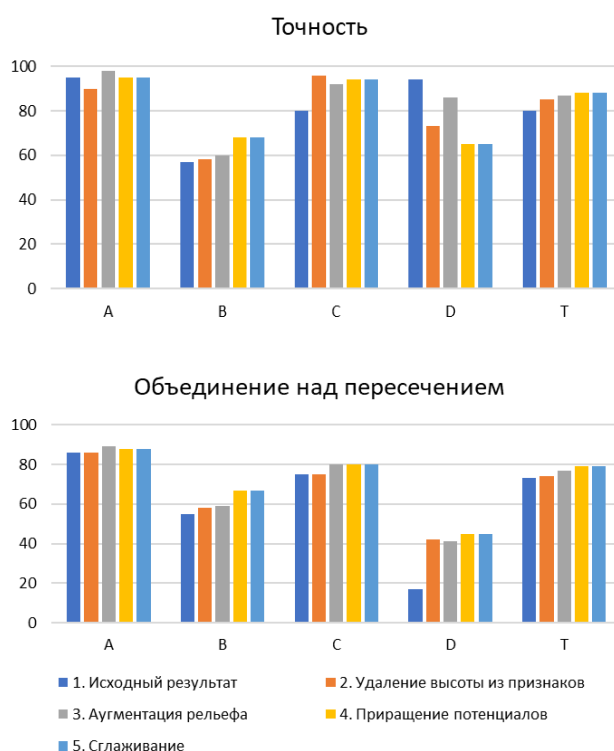


Рис. 4. Метрики качества классификации при различных модификациях.

Обозначения классов:

A – точки на земле; B – растительность; C – здания; D – автомобили; T – среднее взвешенное значение метрик

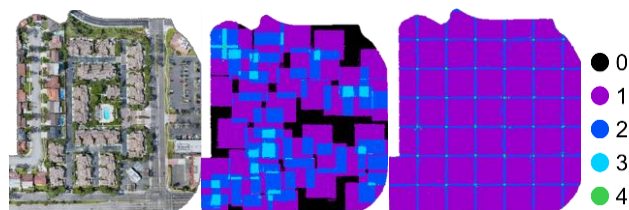


Рис. 5. Количество итераций на точку (слева направо: цветное облако точек, число итераций исходного алгоритма, число итераций предложенного алгоритма)

Заклучение

В работе рассмотрены разные алгоритмы классификации облаков точек, полученных в результате лазерного сканирования или при фотограмметрической обработке аэросъемки. Выбрана оптимальная нейросеть [12], модифи-

цированная для работы с рассматриваемыми облаками точек. Сравнительный анализ результатов классификации облаков точек до и после модификации алгоритма подтвердил повышение производительности до 10 раз и точности классификации на 9 %, что является критичным на фотограмметрическом производстве.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Zhang J., Zhao X., Chen Z., Lu Z. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. // *IEEE Access* – 2019. – Vol. 7. – P. 179118-179133. – DOI: 10.1109/ACCESS.2019.2958671.
2. Guo Y., Wang H., Hu Q. et al. Deep Learning for 3D Point Clouds: A Survey // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2020. – Vol. 43. – P. 4338–4364. – DOI 10.1109/tpami.2020.3005434.
3. Орлова С. Р., Лопота А. В. Трехмерное распознавание: текущее состояние и тенденции // *Автоматика и телемеханика*. – 2022. – Вып 4. – С. 5–26. – DOI 10.31857/S000523102204002X.
4. Jhaldiyal A., Chaudhary N. Semantic segmentation of 3D LiDAR data using deep learning: a review of projection-based methods // *Applied Intelligence*. – 2022. – Vol. 53. – P. 6844–6855.
5. Liu Z., Tang H., Lin Y., Han S. Point-Voxel CNN for Efficient 3D Deep Learning // *Advances in Neural Information Processing Systems*. – 2019. – P. 963–973. – DOI 10.48550/arXiv.1907.03739.
6. Riegler G., Ulusoy A. O., Geiger A. OctNet: Learning Deep 3D Representations at High Resolutions // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. – 2017. – P. 6620–6629. – DOI 10.1109/CVPR.2017.701.
7. Qi C. R., Su H., Mo K., Guibas L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. – 2016. – P. 77–85. – DOI 10.1109/cvpr.2017.16.
8. Qi C. R., Yi L., Mo K., Guibas L. J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space // *Advances in Neural Information Processing Systems*. – 2017. – P. 5099–5108. – DOI 10.48550/arXiv.1706.02413.
9. Hu Q., Yang B., Xie L. et al. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. – 2020. – P. 11105–11114. – DOI 10.1109/cvpr42600.2020.01112.
10. Qi C., Li Y., Peng H. et al. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies // *Advances in Neural Information Processing Systems*. – 2022. – P. 23192–23204. – DOI 10.48550/arXiv.2206.04670.
11. Desai A., Parikh S., Kumari S., Raman S. PointResNet: Residual Network for 3D Point Cloud Segmentation and Classification // *arXiv preprint*. – [2022]. – arXiv: 2211.11040. – DOI 10.48550/arXiv.2211.11040.
12. Thomas H., Qi C. R., Deschard J.-E. et al. KPConv: Flexible and Deformable Convolution for Point Clouds // *Proceedings of the IEEE International Conference on Computer Vision*. – 2019. – P. 6410–6419. – DOI 10.1109/iccv.2019.00651.
13. Liu K., Gao Z., Lin F., Chen B.M. FG-Net: A Fast and Accurate Framework for Large-Scale LiDAR Point Cloud Understanding // *IEEE Transactions on Cybernetics*. – 2022. – Vol. 53. – P. 553–564. – DOI 10.48550/arXiv.2012.09439.
14. Armeni I., Sener O., Zamir A.R. et al. 3D Semantic Parsing of Large-Scale Indoor Spaces // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. – 2016. – P. 1534–1543. – DOI 10.1109/CVPR.2016.170.
15. Pan Y., Braun A., Borrmann A., Brilakis I. 3D deep learning enhanced void-growing approach in creating geometric digital twins of buildings // *Proceedings of the Institution of Civil Engineers: Smart Infrastructure and Construction*. – 2023. – Vol. 176. – P. 24–40. – DOI 10.1680/jsmic.21.00035.

16. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2016. – P. 770-778. – DOI 10.1109/cvpr.2016.90.

17. Hu Q., Yang B., Khalid S. et al. SensatUrban: Learning Semantics from Urban-Scale Photogrammetric Point Clouds // International Journal of Computer Vision. – 2022. – P. 1–28. – DOI 10.1007/s11263-021-01554-9.

18. Chen M., Hu Q., Yu Z. et al. STPLS3D: A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset // arXiv preprint. – 2022. – arXiv: 2203.09065

Об авторах

Дмитрий Владимирович Василенко – аспирант, программист-разработчик.

Получено 28.06.2024

© Д. В. Василенко, 2024

Development of an algorithm for urban scale point cloud classification

D. V. Vasilenko^{1,2}✉

¹ Moscow Institute of Physics and Technology (National Research University), Moscow, Russian Federation

² JSC Racurs, Moscow, Russian Federation

e-mail: vasilenko.dv@phystech.edu

Abstract. Earth remote sensing data, including space and aerial photography, airborne laser scanning are a valuable source of information about objects and terrain. As a result of processing such data with the help of specialized software solutions various information products are created: ortho-photos, digital models of relief and terrain, 3D models of buildings, maps, topo-plans, and other materials used in different sectors of economy. Appearance of new types of imaging equipment, increase of data volume, qualitative and quantitative growth of remote sensing materials consumers requires constant improvement of software. If the use of neural networks and image classification is well studied in the case of two-dimensional images, the classification of three-dimensional irregular point clouds is still poorly investigated. This paper provides review of the main approaches to this task - projective, voxel, multilayer perceptrons and point convolutions. The aim of the work is to develop an algorithm for classification of point clouds obtained as a result of laser scanning or photogrammetric processing of aerial survey. Object of research - methods of automatic classification of point clouds. Subject - an algorithm for applying an artificial neural network to a point cloud.

Keywords: machine learning, deep learning, algorithm architecture, laser scanning, aerial photography, photogrammetry, point cloud

REFERENCE

1. Zhang, J., Zhao, X., Chen, Z., & Lu, Z. (2019). A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access*, 7, 179118-179133. DOI 10.1109/ACCESS.2019.2958671.

2. Guo, Y., Wang, H., Hu, Q., & et al. (2020). Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 4338–4364. DOI 10.1109/tpami.2020.3005434.

3. Orlova, S. R., Lopota, A. V (2022). Three-dimensional recognition: current state and trends *Автоматика и телемеханика. [Automation and Remote Control]*, 4, 5–26 DOI 10.31857/S000523102204002X [in Russian].

4. Jhaldiyal, A., & Chaudhary, N. (2022). Semantic segmentation of 3D LiDAR data using deep learning: a review of projection-based methods. *Applied Intelligence*, 53, 6844–6855.
5. Liu, Z., Tang, H., Lin, Y., & Han, S. (2019). Point-Voxel CNN for Efficient 3D Deep Learning. *Advances in Neural Information Processing Systems*, 963–973. DOI 10.48550/arXiv.1907.03739.
6. Riegler, G., Ulusoy, A. O., & Geiger, A. (2017). OctNet: Learning Deep 3D Representations at High Resolutions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6620–6629. DOI 10.1109/CVPR.2017.701.
7. Qi, C. R., Su, H., Mo, K., & Guibas, L.J. (2016). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 77–85. DOI 10.1109/cvpr.2017.16.
8. Qi, C. R., Yi, L., Mo, K., & Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 5099–5108. DOI 10.48550/arXiv.1706.02413.
9. Hu, Q., Yang, B., Xie, L., & et al. (2020). RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11105–11114. DOI 10.1109/cvpr42600.2020.01112.
10. Qi, C., Li, Y., Peng, H., & et al. (2022). PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. *Advances in Neural Information Processing Systems* pp. 23192–23204. DOI 10.48550/arXiv.2206.04670.
11. Desai, A., Parikh, S., Kumari, S., & Raman, S. (2022). PointResNet: Residual Network for 3D Point Cloud Segmentation and Classification. *arXiv preprint. arXiv: 2211.11040*. DOI 10.48550/arXiv.2211.11040.
12. Thomas, H., Qi, C. R., Deschud, J.-E., & et al. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 6410–6419. DOI 10.1109/iccv.2019.00651.
13. Liu, K., Gao, Z., Lin, F., & Chen, B.M. (2022). FG-Net: A Fast and Accurate Framework for Large-Scale LiDAR Point Cloud Understanding. *IEEE Transactions on Cybernetics*. 53, 553–564. DOI 10.48550/arXiv.2012.09439.
14. Armeni, I., Sener, O., Zamir, A. R., & et al. (2016). 3D Semantic Parsing of Large-Scale Indoor Spaces. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1534–1543. DOI 10.1109/CVPR.2016.170.
15. Pan, Y., Braun, A., Borrmann, A., & Brilakis, I. (2023). 3D deep learning enhanced void-growing approach in creating geometric digital twins of buildings. In *Proceedings of the Institution of Civil Engineers: Smart Infrastructure and Construction*, 176, 24–40. DOI 10.1680/jsmic.21.00035.
16. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 770–778. DOI 10.1109/cvpr.2016.90.
17. Hu, Q., Yang, B., Khalid, S., & et al. (2022). SensatUrban: Learning Semantics from Urban-Scale Photogrammetric Point Clouds. *International Journal of Computer Vision*, 1–28. DOI 10.1007/s11263-021-01554-9.
18. Chen, M., Hu, Q., Yu, Z., & et al. (2022). STPLS3D: A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset. *arXiv preprint. arXiv: 2203.09065*.

Author details

Dmitry V. Vasilenko – Ph. D. Student, Programmer-developer.

Received 28.06.2024

© D. V. Vasilenko, 2024