

PointAtrousGraph: глубоко иерархический кодировщик-декодер с точкой

Плоская свертка для неорганизованных 3D-точек

Лян Пан¹, Чи-Мэн Чу² и Гим Хи Ли³

Аннотация- Движимые успехом кодирования многомасштабной контекстуальной информации для анализа изображений, мы предлагаем наш PointAtrousGraph (PAG) - высокочастотный иерархический кодер-декодер с глубокой перестановкой для эффективного использования многомасштабных граничных объектов в облаках точек. Наш PAG построен на нескольких новых модулях, таких как точечная свертка (PAC), объединение с сохранением границ (EP) и распаковка с сохранением границ (EU). Как и в случае с atrous convolution, наш PAG может эффективно расширять поля восприятия фильтров и, таким образом, эффективно изучать многомасштабные точечные функции. Следуя идее непрерывающихся операций максимального объединения, мы предлагаем наш EP для сохранения критических граничных характеристик во время подвыборки. Соответственно, наши модули EU постепенно восстанавливают пространственную информацию для граничных объектов. Кроме того, мы вводим модули с цепным пропуском подвыборки / повышающей дискретизации, которые непосредственно передают граничные объекты на заключительный этап. В частности, предлагаемые нами вспомогательные функции потерь могут еще больше улучшить нашу производительность. Экспериментальные результаты показывают, что наш PAG превосходит предыдущие современные методы в различных приложениях для семантического восприятия 3D.

Я. Введение

Благодаря эффективности захвата пространственно-локальных соотношений операций свертки, нейронные сети глубокой свертки (CNN) дали впечатляющие результаты для многих задач, основанных на изображениях. Чтобы кодировать многомасштабную контекстуальную информацию, CNNs исследует входящие объекты изображения с помощью фильтров или операций объединения с несколькими скоростями и несколькими эффективными полями обзора [1], такими как Atrous Spatial Pyramid Pooling (ASPP) [2] и модуль объединения пирамид [3], [4]. Обе стратегии эффективно улавливают контекстуальную информацию в нескольких масштабах, следовательно, значительно улучшая возможности CNN. Неорганизованное облако точек - это простое и понятное представление 3D-структур, которое часто применяется в современных приложениях интеллектуальной робототехники, таких как автономное вождение и взаимодействие человека и робота. Однако неупорядоченность и неравномерность 3D-облаков точек делает неприменимой обычную операцию свертки кабеля. Несмотря на то, что недавно были предложены новые ядра фильтрации, были проведены ограниченные исследования по разработке глубоких иерархических архитектур кодировщика-декодера для изучения многомасштабных точечных функций для трехмерного семантического восприятия.

PointNet [5] - это новаторская сеть, не зависящая от перестановок, которая напрямую обрабатывает неупорядоченные облака точек, используя множество симметричных функций. Она учитывает каждый 3D



Рис. 1. Слева обозначен общепринятый метод выбора соседних точек. На рисунке

Очки $\{q_1, q_2, q_3, q_4, q_5\}$, а справа цифра обозначает наши построенный график района $\{q_2, q_4, q_6, q_8, q_{10}\}$ с частотой дискретизации, равной 2. По добавив параметр частоты дискретизации, наш PAC может выполнять операцию свертки в большем поле зрения без увеличения вычислительной нагрузки.

наводите независимо и при этом не обращайте внимания на локальные геометрические данные. PointNet ++ [6] адаптивно объединяет многомасштабные локальные объекты с помощью предлагаемого им уровня абстракции. Существуют почти одинаковые закодированные векторы локальных объектов, если две выбранные контрольные точки имеют общие локальные области. Эти сети, инвариантные к перестановкам [7], [8], которые фиксируют мелкие геометрические структуры из локальных окрестностей, обеспечивают лучшие результаты вывода из облака точек, из чего очевидно, что особенности локальных соседей могут улучшить глубокое обучение в 3D точках. Тем не менее, восприимчивые поля их фильтров ограничены небольшими локальными регионами путем построения локальных графов окрестностей. Совсем недавно PAN [9] представил новый модуль точечной горизонтальной свертки (PAC), который может эффективно расширять поля восприятия фильтров за счет введения частоты дискретизации для эквивалентно малой выборки соседних точечных объектов. PAN может эффективно использовать многомасштабные функции edge за счет использования модулей PAC с разной частотой дискретизации. Однако когда-либо PAN не имеет иерархической архитектуры кодер-декодер, которая становится неэффективной при работе с многомерными объектами с плотной точкой.

В этой статье мы предлагаем PointAtrousGraph (PAG) - иерархический кодер-декодер, не зависящий от глубоких перестановок, для использования многомасштабных локальных геометрических деталей с новым PAC модулем для анализа облака точек. Чтобы решить проблему с графом окрестностей, мы применяем нашу операцию объединения (EP) с сохранением границ для сохранения критических объектов границ во время подвыборки. Следовательно, наш PAG может использовать и иерархически сохранять многомасштабные локальные геометрические детали. Аналогичным образом, наша операция распаковки с сохранением границ (EU) применяется для восстановления пространственной информации разреженных точечных объектов большой размерности. Кроме того, мы представляем адаптированные модули с цепным пропуском подвыборки / повышающей дискретизации для прямого распространения точечных объектов из каждой иерархии. Кроме того, мы предлагаем новые вспомогательные функции потерь, потери при максимальном среднем расхождении (MMD) и потери с глубоким контролем, что еще больше повышает точность наших выводов. Наш PAG также требует меньшего потребления памяти для обучения и более короткого времени обучения, чем большинство существующих сетей, которые в значительной степени зависят от

перекрытые

¹ Лян Пан работает в Центре передовой робототехники Национального университета Сингапура
E-mail: lianp@nus.edu.sg

² Чи-Мэн Чу работает на факультете машиностроения, Национальный университет Сингапура
E-mail: chewcmm@nus.edu.sg

³ Гим Хи Ли из лаборатории компьютерного зрения и восприятия роботов (CVRP) Факультета компьютерных наук Национального университета Сингапура
E-mail: dcs1gh@nus.edu.sg

графики окрестностей для трехмерных точек. Эксперименты показывают, что наш PAG обеспечивает лучшую производительность, чем предыдущие современные методы в различных приложениях облака точек, включая классификацию 3D-объектов, сегментацию частей объекта и семантическую сегментацию 3D .

II. СОПУТСТВУЮЩАЯ РАБОТА

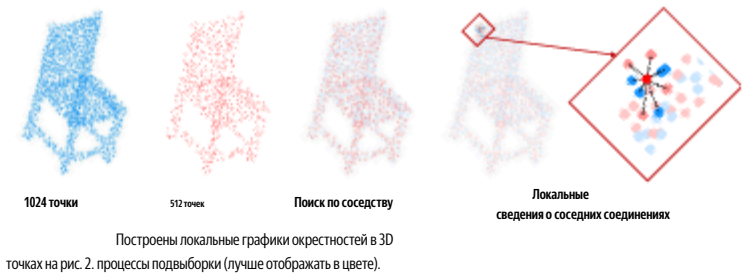
Анализ неорганизованного облака точек. Из-за не-упорядоченности 3D-точек облако точек с N 3D-точками имеет всего N! перестановок в порядке подачи данных. Следовательно, для сети важно поддерживать инвариантность всех возможных перестановок [5]. Новаторская работа PointNet [5] обеспечивает инвариантность к перестановкам за счет применения симметричных функций. Многие последующие работы [6], [10], [7], [8] предлагают более сложные симметричные операции для использования локальных геометрических деталей в трехмерных точках. Семантическая маркировка в облаке точек является более сложной задачей, чем классификация и сегментация частей объекта. SPG [11] и SGPN [12] оба строят суперточечные графики для уточнения результатов их семантической маркировки . PAN [9] предлагает новый модуль PAC для эффективного использования многомасштабных локальных граничных функций. Однако, в отличие от многих сетей для задач семантической маркировки изображений, они [11], [12], [13], [9] не имеют иерархической архитектуры кодировщик-декодер , что ограничивает их производительность.

Иерархический	Кодировщик-декодер.	Глубокий	иерархический
---------------	---------------------	----------	---------------

архитектуры кодировщика-декодера широко и успешно используются для многих задач, основанных на изображениях, таких как оценка позы человека [14], [15], семантическая сегментация [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], оценка оптического потока [26], [27] и обнаружение объекта [28], [29], [30]. Архитектура кодировщика-декодера, модуль "сложенные песочные часы" основан на последовательных этапах объединения и повышения дискретизации, что дает впечатляющие результаты при оценке позы человека [14]. Лин и др. [28] представили функцию pyramid network для обнаружения объектов. Что касается семантических задач сегментации, U-Net [16] и DeconvNet [25] следуют архитектуре симметричного кодировщика-декодера, и они уточняют маски сегментации, используя функции в низкоуровневых слоях. DeepLabV3 + [1] использует преимущества как архитектуры кодировщика-декодера, так и модулей atrous convolution для эффективного изменения полей обзора фильтров для сбора многомасштабной контекстной информации, что обеспечивает новую производительность по последнему слову техники на многих иерархических тестах семантической сегментации. архитектуры кодировщика-декодера содержат: (1) модуль кодировщика, который постепенно снижает разрешение объекта, расширяет поля восприятия фильтров и захватывает более высокую семантическую информацию; (2) модуль декодера, который постепенно восстанавливает пространственную информация [1].

III. МЕТОДЫ

Наш PointAtrousGraph (PAG) ориентирован на изучение многомасштабных функций границ за счет применения глубоко иерархической архитектуры кодировщик- декодер. Чтобы сохранить свойство, не зависящее от перестановок, наш PAG состоит из симметричных функций, таких как общий mpr, максимальное объединение и конкатенация объектов. В частности, модуль PAC применяется в качестве фундаментального



строительный блок, который позволяет эффективно изучать многомасштабные объекты с плотными краями в 3D-точках. Кроме того, мы предлагаем нашу операцию объединения с сохранением границ (EP), которая дает преимущества при построении глубоких иерархических сетей за счет сохранения критических функций границ во время процессов подвыборки. Наша технология EP также расширяет поля восприятия за счет уменьшения плотности трехмерных точечных объектов. Аналогичным образом наша операция распаковки с сохранением границ (EU) постепенно восстанавливает плотность точечных объектов высокой размерности за счет учета пространственного расположения точек в 3D . Мы также напрямую передаем точечные характеристики из разных иерархий на заключительный этап с помощью адаптированных модулей субдискретизации с пропуском / повышающей дискретизацией. Кроме того, мы предлагаем новые вспомогательные потери для дальнейшего повышения нашей точности вывода.

A. Точечная свертка для плотного точечного объекта

Два типичных метода, ball query и k-ближайших соседей (kNN), применяются для использования локальных геометрических деталей в облаках точек. Однако алгоритм запроса шара, применяемый PointNet ++ [6], всегда выбирает первые #K точек в указанном шаре поиска с предопределенным радиусом, что не может гарантировать, что могут быть выбраны ближайшие точки [31]. Чтобы использовать достаточную локальную контекстуальную информацию в трехмерных точках, обычные сети либо строят графики больших окрестностей (large #K) [7], [8], либо объединяют многомасштабные локальные граничные объекты (large #C_i) [32], [6]. Обе вышеупомянутые стратегии, однако, делают их сети громоздкими и неэффективными.

Напротив, точечная свертка (PAC) [9] может произвольно увеличить свое поле восприятия в плотных точечных объектах без увеличения объема вычислений (малые # K и # C). В отличие от этого, точечная свертка (PAC) [9] может произвольно увеличивать свое поле восприятия в плотных точечных объектах без увеличения объема вычислений (малые # K и # C_i). Вдохновленные технологией atrous convolution [33] для анализа изображений, модули PAC вводят важный параметр частоты дискретизации r для эквивалентно малой выборки соседних точечных объектов. Операция PAC может быть сформулирована следующим образом:

$$x_p = g(H_o(x_p, x_{q1:r}), \dots, H_o(x_p, x_{qk:r})) \quad (1)$$

является особенностью этой точки центроида, где x_p объект (k · r)_o, $x_{qk:r}$ ближайшего соседа точки p, r - частота дискретизации, k - общее количество обысканных соседних точек, g(·) обозначает функцию максимального объединения и $x_p \oplus (x_p, x_{qk:r}) \cdot o^h o^h$ обозначает пограничное ядро o^H (\cdot) - общий уровень mpr и \oplus обозначает объединение объектов

операция. Пример также проиллюстрирован на рис. 1.

B. Объединение с сохранением границ

Объединяющие слои, особенно неперекрывающиеся слои с максимальным объединением , широко используются в CNN [34], [35], которые суммируют результаты соседних групп нейронов в одной и той же карте ядра в доменах изображений [36]. Шерер и др. [37] сообщают, что приращение размера шага перекрывающихся



объединение окон в пул ухудшает их производительность распознавания, поскольку максимумы в перекрывающихся областях окна просто дублируются на следующем слое, а соседние пиксели более коррелированы. Чирсан и др. [38] заменили слои подвыборки неперекрывающимися слоями с максимальным объединением в CNNs [39], что обеспечивает удивительно высокую скорость обучения и лучшую производительность. Были предложены различные методы субдискретизации для 3D-облаков точек [40], [6]. Тем не менее, они либо не суммируют локальные геометрические особенности, либо игнорируют проблему, вызванную перекрывающимися локальными графами окрестностей.

Пример процессов субдискретизации трехмерных точек проиллюстрирован на рис. 2. Размер входного точечного объекта равен 1,024, и после процесса подвыборки выбирается 512 точечных объектов. Если будут распространяться только те особенности "выбранных" точек, то локальные геометрические детали, таким образом, будут упущены из виду. Другая стратегия заключается в распространении граничных объектов с учетом особенностей локальных соседних точек (также показано на рис. 2). Искомые соседние точечные объекты часто состоят как из "выбранных" точечных объектов (показаны розовым цветом), так и из "отброшенных" точечных объектов (показаны синим цветом) во время подвыборки. Две "выбранные" точки центрада часто могут иметь перекрывающиеся графики окрестностей и даже иметь одни и те же локальные соседние точки. Следовательно, если мы распространяем закодированные объекты соседних точек, две соседние "выбранные" точки могут иметь похожие или даже одинаковые объекты. Соответствующая информация о каждой ТРЕХМЕРНОЙ точке исчезает, особенно, когда для построения глубокой иерархической сети для трехмерных точек выполняется несколько операций подвыборки. Мы обозначаем это как задачу графа перекрытой окрестности, показанную на рис. 3.

В связи с этим мы предлагаем наш модуль объединения с сохранением границ (EP), который эффективно фиксирует локальные геометрические детали, сохраняя при этом соответствующие характеристики каждой точки. В соответствии с идеей неперекрывающихся операций максимального объединения, мы кодируем локальные граничные объекты, рассматривая соседние точечные объекты в исходных 1024 точках. Из-за отсутствия регулярных сеток в 3D-облаках точек мы выбираем соседние точечные объекты путем построения графиков окрестностей в метрических пространствах. Чтобы сохранить как отличительные особенности отдельных точек, так и локальные геометрические особенности кромок, наш модуль EP разработан следующим образом:

$$X_p = X_p \oplus g(X_{q_1}, \dots, X_{q_k}), \quad (2)$$

является особенностью выбранного центрада и где X_{q_k} является ли точечные характеристики его k ближайших соседей. Следовательно,,

наша операция EP явно распространяет точечные объекты каждого "выбранного" центрада, а также суммирует его локальные точечные объекты, что согласуется с идеей, лежащей в основе не перекрывающегося максимального объединения для распространения объектов изображения.

С. Распаковка с сохранением границ

Для восстановления пространственной информации для объектов изображения, низкий уровень характеристики изображения из кодера часто применяется для уточнения функции высокого уровня в декодер, особенно, когда

используя симметричную иерархическую архитектуру кодер-декодер [16], [25]. Наш модуль unpooling (EU) с сохранением ребер также учитывает точечные объекты центроидов и их локальные соседние точечные объекты, которые ищутся в метрических пространствах. В отличие от PointNet++, нашему модулю EU не нужно учитывать "координаты d-dim", связанные с каждой точкой:

$$X_p = X_p \oplus \text{Ш}(X_{q_1}^{-1}, \dots, X_{q_k}), \quad (3)$$

где X_p является соответствующей характеристикой, передаваемой от кодировщика посредством пропускного соединения напрямую, $w(\cdot)$ - обратная операция средневзвешенного расстояния, а X_{q_k} является особенностью k -ближайшего соседа точки p в ее предыдущей иерархии.

D. Глубокий иерархический кодер-декодер

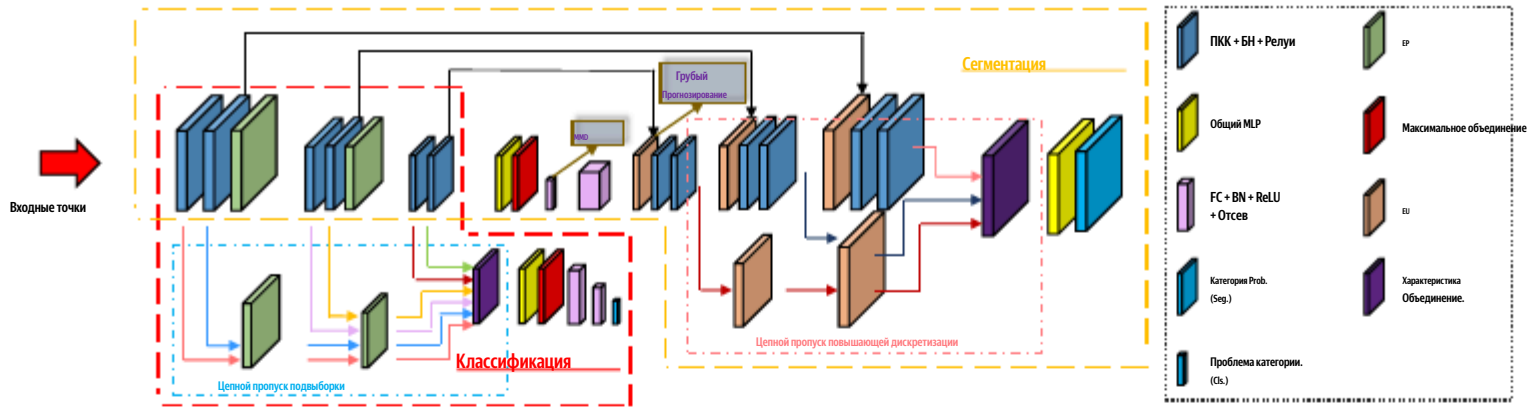
На основе наших модулей PAC, EP и EU мы создаем архитектуру глубокого иерархического кодирования-декодирования pointA- trousGraph (PAG) для изучения многомасштабных функций для задач 3D-классификации точек и сегментации (показано на рис. 4). Та же архитектура кодера применяется в наших сетях классификации и сегментации, которые состоят из трех иерархий, чтобы постепенно уменьшать плотность точечных объектов и, тем временем, расширять области восприятия для изучения более высоких семантических точечных объектов. Внутри каждой иерархии мы последовательно размещаем два слоя PAC с увеличивающейся частотой дискретизации, чтобы постепенно раскрывать более крупные локальные геометрические детали. Наши архитектуры декодеров разработаны по-разному для разных приложений. Кроме того, мы также предлагаем различные модули подключения с пропуском, подвыборку с цепным пропуском и повышающую дискретизацию с цепным пропуском, для классификации и сегментации соответственно.

Сеть классификации.

Наша сеть классификации (en)-

закрывается красными пунктирными линиями на рис. 4) направлен на кодирование глобального вектора точечных объектов путем использования многомасштабных локальных геометрических деталей в облаке точек. Основным потоком является наш сетевой кодировщик, который решает многомасштабную контекстную проблему сбора информации путем иерархического применения множества модулей PAC. В дополнение к основному потоку, который последовательно распространяет объекты, мы также предлагаем модуль подвыборки с цепным пропуском (заключен синими пунктирными линиями на рис. 4). Модуль последовательного пропуска подвыборки постепенно передает функции каждой иерархии на заключительный этап. В каждой иерархии мы выбираем один и тот же набор центроидных точек с соответствующим модулем EP для построения локальных графов окрестностей. Однако, мы распространяем особенности соседей только в цепочке, пропускаем операции подвыборки, что отличается от модуля EP. После объединения иерархических точечных объектов из обоих потоков глобальный объект получается путем применения глобального максимального объединения. После этого используются два полностью соединенных слоя (FC) для получения окончательных результатов классификации.

Сеть сегментации. Задачу сегментации можно представить как классификацию по точкам. Следовательно, наша сегментационная сеть (обведена золотистыми пунктирными линиями на рис. 4) плотно изучает многомасштабные граничные объекты для каждой входной 3D-точки. Соответственно, мы предлагаем иерархический декодер для постепенного восстановления плотности точечных объектов высокой размерности. Наш декодер для сегментации имеет аналогичную архитектуру, что и наш кодировщик, который также имеет три иерархии. Аналогично, мы создаем два уровня PAC с уменьшением частоты дискретизации до



Наша архитектура PointAtrousGraph (PAG) (лучше видна в цвете). Наши сети классификации и сегментации имеют одинаковую конструкцию, рис. 4. архитектура кодировщика. Наша сеть классификации обведена красными пунктирными линиями, а наша сеть сегментации обведена золотистыми пунктирными линиями. Модуль подвыборки с цепным пропуском применяется в нашей сети классификации, которая обведена синими пунктирными линиями. Аналогично, наш модуль последовательного пропуска с повышением дискретизации окружен розовыми пунктирными линиями в нашей сети сегментации.

постепенно агрегируйте характеристики для каждой точки. В частности, мы применяем регрессированные глобальные функции, что в значительной степени повышает нашу производительность. Мы также используем модуль chained skip up-выборки для последовательного увеличения выборки точечных объектов каждой иерархии (обведен розовыми пунктирными линиями на рис. 4). На каждом этапе мы применяем ту же стратегию повышения дискретизации с модулем EU с теми же построенными локальными графами соседства. В отличие от модуля EU, мы не объединяем функции центраида. Наконец, мы объединяем все иерархические объекты для каждой 3D-точки, чтобы выполнить предварительные точечные прогнозы.

Е. Вспомогательные функции потерь

Мы также вводим вспомогательные потери, максимальную среднюю дисперсию стерансу (MMD) и потери с глубоким контролем, которые в основном применяются для задач сегментации.

Критерий MMD [41], который обычно используется в вариационных архитектурах автокодирования, выполняется над нашими встроенными глобальными точечными объектами. Потеря MMD количественно определяет сходство между двумя распределениями путем сравнения всех их моментов. Применяя трюк ядра, потеря MMD определяется как:

$$L_{mmd} = \frac{1}{2} \left(\mathbb{E}_{q(z), q(z)} [k(z, z)] + \mathbb{E}_{p(z), p(z)} [k(z, z)] - 2\mathbb{E}_{q(z), p(z)} [k(z, z)] \right) \quad (4)$$

≥ 0 , $q(z)$ обозначает наш встроенный объект, где L_{mmd} распределение и $p(z)$ обозначают предыдущее гауссово распределение (мы

используем $\mu = 0, \sigma = 1, 0$). $L_{mmd} = 0$ тогда и только тогда, когда $q = p$.

Аналогично с сетями сегментации на основе изображений [3], [42], мы добавляем потери с глубоким контролем (кросс-энтропийные) L_{ds} на первом этапе нашего сетевого декодера сегментации. Наряду с потерей главной ветви (кросс-энтропии) $L_{главная}$, функции вспомогательных потерь также проходят через все предыдущие уровни. Следовательно, мы обучаем нашу сеть сегментации, минимизируя следующую функцию совместных потерь:

$$L_{BCE}^{master} + w_{mmd} L_{mmd} + w_{ds} L_{ds} \quad (5)$$

где w_{mmd} и w_{ds} предназначены для уравнивания гирь соответствующих вспомогательных потерь.

Ф. Обсуждение

Многие сети [5], [8], [40] которые учитывают свойство инвариантности к перестановкам, ориентированы на проектирование узлов свертки для неупорядоченных трехмерных точек. Предыдущие исследования [6], [7], [8], [43], [10] показывают эффективность учета локальных геометрических

Подробные сведения. Однако было уделено ограниченное внимание разработке архитектур кодировщика-декодера с глубокой иерархией для захвата многомасштабных локальных контекстов. Напротив, наш PAG - глубоко иерархический кодер-декодер, адаптивно изучающий многомасштабные граничные объекты, изменяя поле зрения для фильтров на каждом уровне. Мотивирован успехом использования многомасштабной контекстуальной информации в операциях по изучению функций изображения [17], [33], [44], [25], [3], такие как масштабная свертка, неперекрывающееся максимальное объединение и распаковка, мы применяем новые модули, включая PAC, EP и EU, для нашего многомасштабного изучения граничных объектов в неорганизованных 3D точках. Наш модуль PAC эффективно расширяет свое поле восприятия объектов с плотной точечностью без увеличения параметров обучения. Благодаря нашим модулям EP наш PAG способен распространять объекты большой размерности, сохраняя при этом локальные граничные объекты и соответствующую информацию о каждой точке. Чтобы восстановить плотность точечных объектов, мы применяем операции интерполяции, рассматривая особенности этих пространственных соседних точек..... Предлагаемые вспомогательные функции потерь еще больше повышают производительность нашей сети. Более того, наша иерархическая архитектура состоит из нескольких операций подвыборки, что значительно сокращает объем вычислений и, таким образом, в значительной степени повышает эффективность.

IV. ЭКСПЕРИМЕНТЫ

Наш PAG оценивается в трех задачах анализа облака точек, включая классификацию форм, сегментацию частей объекта и семантическую сегментацию. Без дополнительных процессов наши PAG превосходят предыдущие современные методы.

A. Подробности реализации

Наши сети реализованы с помощью TensorFlow [45] на графическом процессоре NVIDIA GTX1080Ti. Мы сообщаем о наших результатах с учетом различных входных данных и стратегий обучения для достижения справедливых сравнений при классификации фигур. Для задач сегментации мы следуем тем же настройкам обучения и оценки, что и в [5]. Для повышения эффективности мы фиксируем $\# K = 10$ для всех наших графиков окрестностей. Мы применяем алгоритм выборки в самой удаленной точке (FPS) [6] для точек подвыборки, рассматривая их метрики (статические) или особенности (динамические). Наш код и модели находятся в открытом доступе на веб-сайте проекта¹.

¹ <https://github.com/paul007pl/PointAtrousGraph>

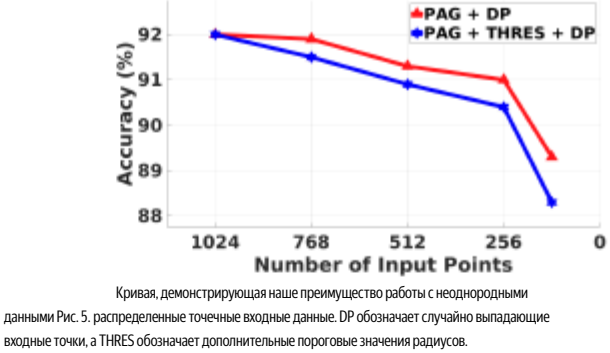
ТАБЛИЦА I			
S H APE CL УТОЧНЯЕТ РЕЗУЛЬТАТЫ ПО N		40 [46] ЭТА МОДЕЛЬ	
	Методы	Ввод	О памяти.
По оси вверх вращение	PointNet [5]	1,024 очка	99,2 2,4 Гб
	PointNet++ [6]	1,024 очка	100,7 11,1 Гб
	Wang(41-spec-ср)	1,024 очка	11,5
	[47]	4,000 очка	11,7
	MRTNet(kd-дерево) [48] DGCNN [7]	1,024 очка	122,2 8,9 Гб
	PAN [9]	1,024 очка	122,2 9,2 Гб
	ПАГ (наш)	1,024 очка	11,9
	PointNet++ [6]	5,000 очка + n	12,4 11,3 Гб
	SpiderCNN [8]	1,024 очка + n	12,1
	Wang(41-spec-ср) [47]	2,048 очка + n	12,6 9,2 Гб
Вертикаль вращение	PAN [9]	1,024 очка + n	12,7 2,8 Гб
	PAG (наш)	1,024 очка + n	11,0
	KCNet [10]	1,024 очка	12,5
	Ацмон и др. [49]	1,024 очка	10,9
	PointCNN+ [50]	1,024 очка	12,6
	SO-Net [43]	2,048 очка	13,1 9,2 Гб
	A-CNN [31]	1,024 очка	13,1 2,4 Гб
	PAN [9]	1,024 очка	13,4
	PAG (наш)	1,024 очка	13,4
	SO-Net [43]	5000 очков + n	13,8
Без	ПАНОРАМИРОВАНИЕ	5000 очков + n	11,4 часа 8,7 часа 4,4 часа
	[9] PAG (наш)	5000 очков + n	8,8 4,5 ч 8,7 ч 4,4 ч 11,4
		5000 очков + n	

V. Классификация форм

Мы оцениваем производительность нашей сети для классификации форм на основе набора данных ModelNet40 [46]. ModelNet40 бенчмарк содержит 13 834 модели САПР из 40 категорий, и разделен на обучающий (9 843 модели) и тестовый набор (2468 моделей). Большинство 3D-моделей из ModelNet40 предварительно выровнены по общему направлению вверх и по горизонтали. Чтобы лучше аппроксимировать сценарии в реальных приложениях, предварительно выровненное направление можно игнорировать, применяя случайные повороты вверх по оси. Для объективного сравнения, мы сообщаем о производительности нашей классификационной сети в обоих условиях обучения, приведенных в таблице I. Столбец "OA" показывает общую точность классификации (в процентах). Столбцы "Память" и "Время" обозначают требуемое для обучения потребление памяти и время соответственно. Согласно таблице I, наш PAG обеспечивает наилучшую точность классификации для всех параметров обучения. Кроме того, наши ПАГ требует меньший объем памяти и более короткий срок для подготовки, чем большинство существующих сетей.

Исследование абляции. Мы сообщаем о результатах исследования абляции в таблице II. Без применения наших модулей PAC точность классификации снизилась из-за недостаточной контекстуальной информации..... В отличие от предыдущих работ [7], [8], установка числа выбранных ближайших соседей равным 20 не повышает точность нашей классификации. Кроме того, установка меньшего количества выбранных ближайших соседей снижает потребность в памяти для обучения, а также повышает эффективность обучения. Кроме того, предлагаемый модуль последовательной выборки с пропуском (CSS) может дополнительно улучшить производительность классификации. В наших операциях EP, если мы распространяем только "центроидные" или "соседние" объекты, точность нашей классификации значительно снизится.

Устойчивость к изменению плотности выборки. В реальных приложениях облака точек часто захватываются частично и, таким образом, становятся нерегулярными и неполными. Наш PAG в значительной степени полагается на kNN, который не может гарантировать фиксированный масштаб для используемых геометрических деталей, особенно для неравномерно распределенных



Кривая, демонстрирующая наше преимущество работы с неоднородными данными Рис. 5. распределенные точечные входные данные. DP обозначает случайно выпадающие входные точки, а THRES обозначает дополнительные пороговые значения радиусов.

ТАБЛИЦА II				
A СНИЖЕНИЕ УДЕЛЬНОГО ВЕСА КАТАЛИЗАТОРА В ИОННОЙ СЕТИ WO R K				
K	PAC	Центроид / Соседи	CSS	OA
10 20	*	оба		92.0 91.9
10		оба		92.2
10		оба	*	91.7
10		центроид		92.0
10		соседи		92.7

3D точки. С этой целью мы модифицируем модуль PAC,

применяя два радиуса поиска для привязки выбранных соседних

точек в указанном поле поиска (r

max

радиусы поиска обозначают расстояние между точкой центроида

и соседней с ней точкой. В соответствии с [6], мы случайным образом отбрасываем точки (DP), чтобы имитировать неоднородные

3D-точкам.

Объект-часть с. сегментация

Объект-часть сегментация есть продемонстрирован о в

набор данных ShapeNet-part [51], который содержит 16,881 фигура, представленная в виде отдельных облаков точек из 16 категорий с примечаниями для каждой точки. Следуя предыдущей работе [5], мы разделили набор данных на обучающий (14 034 объекта) и тестовый (2847 объектов) наборы. Мы применяем метрику среднего пересечения над объединением (IoU) для оценки нашего метода в каждой точке. Для каждого экземпляра мы получаем среднее значение IoU путем усреднения IOU для всех типов деталей в соответствующей категории объектов. Общий объем IoU экземпляра ("pIoU") вычисляется путем усреднения IOU по всем протестированным экземплярам. IoU по категориям вычисляется как среднее значение для всех экземпляров в соответствующей категории. Среднее значение IoU по категории ("mpIoU"), таким образом, вычисляется как среднее значение всех IOU по категории. В таблице III наша модель представляет наилучшее общее среднее значение IoU на уровне 86,4%. Результаты качественной сегментации показаны на рис. 6.

Исследование абляции. В таблице IV приведены результаты для абла- ции изучения объекта-часть сегментация задач. Результаты



На фиг. 6. Качественные результаты в части решения задачи сегментации.

ТАБЛИЦА III

S ПРИМЕР ДЛЯ СРАВНЕНИЯ:		S		[51] N/APEET		S3DIS [52], и	
Метод		Схема		k-кратный S3DIS		Площадь-5	
		mIoU	pIoU	OA	mIoU	S3DIS OA	mIoU
SGPN- [12]		82.8	85.8	80.8	-	-	-
RSNet- [13]		81.4	84.9	-	50.4	56.5	-
SPG- [11]		-	-	85.5	62.1	86.4	58.0
SegCloud-		-	-	-	-	-	48.9
PointCNN- [53]		84.6	86.1	88.1	65.4	85.9	57.3
PCCN [54]	[50]	-	-	-	-	-	58.3
PointNet [5]		-	-	-	-	-	-
PointNet++ [6]		80.4	83.7	78.5	47.6	-	41.1
SO-Net [43]		81.9	85.1	-	-	-	-
SPLATNET _{3D}		81.0	84.9	-	-	-	-
	[55]	82.0	84.6	-	-	-	-
Ацмон и др. al		81.8	85.1	-	-	-	-
[49] SpiderCNN		82.4	85.3	-	-	-	-
[8] DGCNN [7]		82.3	85.2	84.1	56.1	-	-
PAN [9]		82.6	85.7	85.9	61.4	-	-
A-CNN [31]		84.0	86.1	87.3	62.9	-	-
PAG (наш)		84.0	86.4	88.1	65.9	86.8	59.3

проиллюстрируйте, что предлагаемый модуль последовательной дискретизации с пропуском (CSU) и извлеченные глобальные функции полезны для повышения точности сегментации. Предлагаемый нами модуль PAC еще больше повышает точность сегментации. Установка числа выбранных ближайших соседей равным 20 не повышает точность нашей сегментации, но в значительной степени увеличивает потребление обучающей памяти. В наших операциях EP, игнорирование либо объектов центроидной точки, либо объектов соседних точек может привести к снижению точности. Добавляя глобальные точечные функции (GF) и вспомогательные функции потерь (Aux.L), наша сеть сегментации может давать лучшие результаты.

D. Семантическая сегментация

Мы оцениваем нашу модель на основе крупномасштабной 3D-семантической сегментации (см. рис. 7) в наборе данных Stanford 3D Indoor Semantic (S3DIS) [52]. Этот набор данных содержит облака точек 3D RGB, состоящие из 271 комнаты из 3 разных зданий, разделенных на 6 областей. Чтобы добиться справедливого сравнения, мы применяем ту же настройку стратегии обучения с PointNet [5]. Мы случайным образом отбираем 4096 баллов в каждой блоке для обучения, и все баллы используются для тестирования. Результаты семантической сегментации (k-кратность и площадь-5) представлены в таблице III. Эти методы [50], [12], [13], [11], [56] (обозначаются)получайте несправедливые преимущества, применяя определенные процедуры предварительной и / или постобработки, такие как графики суперточек или рекуррентные нейронные сети. Более того, они не поддерживают важное свойство, не зависящее от перестановок, что делает их результаты ненадежными и чувствительными к порядку подачи точек . Без дополнительных процессов наш PAG, инвариантный к перестановкам, обеспечивает наилучшие результаты по OA (88,1% и 86,8%) и mIoU (65,9% и 59,3%).

V. АНАЛИЗ И НАПРАВЛЕНИЯ НА БУДУЩЕЕ Свойство перестановочной

инвариантности. Предлагаемый нами мод-

ule состоят из симметричных функций, таких как общий mlp, максимальное объединение и объединение функций. Следовательно, наша PAG - сеть, не зависящая от перестановок.

Выборка соседних точек. Шаровой запрос, представленный PointNet ++, является еще одной стратегией выборки соседних точек. Однако PointNet ++ выбирает только первые # K точек, найденных в радиусе, что не может гарантировать наличие ближайших точек

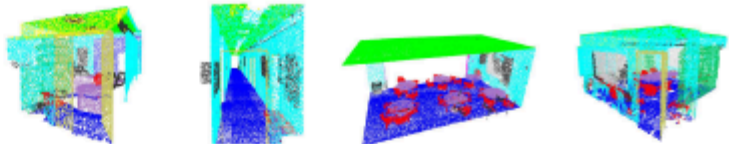


Рис. 7. Качественные результаты по задаче семантической сегментации.

ТАБЛИЦА IV

A КОНТРОЛЬНАЯ ТОЧКА В ПОИСКОВОЙ СЕТИ WO						-
R K Центроид PAC / соседи CSU, GF Aux.L pIoU						
к						
10	10			*	-	85.3
20		*	оба обоих		-	85.7
10			обоих центроидных	*	-	85.7
10					-	85.6
10					-	85.8
10			соседей		-	85.7
10					-	86.1
10			оба		-	86.4

могут быть выбраны. Кроме того, PointNet ++ (MSG) выполняет поиск множества соседних точек по 3 раза (например, #K={16,32,128}) , а затем объединяет все используемые функции вместе (например, # C_i =64+128+128) в каждой иерархии, что делает его вычислительно дорогостоящим [6].. Напротив, наш модуль PAC эквивалентно редко выполняет выборку соседних точек в пространствах объектов с параметром частоты дискретизации "r". В нашем PAG мы последовательно размещаем модули PAC последовательно для объединения многомасштабных граничных объектов без увеличения выбранных соседних точек (# K) и размера точечного объекта (# C_i).

Иерархическая архитектура кодировщика-декодера. Вдохновленные успехом кодирования обширной контекстной информации об изображении в различных масштабах для приложений на основе изображений, мы фокусируемся на передаче локальных геометрических деталей большого масштаба в 3D точках с помощью иерархической архитектуры кодер-декодер. Наш PAG содержит: (1) модуль кодирования, который постепенно уменьшает плотность точечных объектов, расширяет поле зрения фильтров и изучает более высокие значения семантических границ; (2) модуль декодера, который постепенно восстанавливает плотность для точечных объектов большой размерности. Глубоко иерархические CNN обычно используют неперекрывающиеся операции максимального объединения, которые уменьшают разрешение объектов изображения при суммировании локально-пространственной контекстуальной информации. Чтобы решить проблему графа перекрытых окрестностей, мы предлагаем нашу операцию EP, которая объединяет как "центроидные", так и "соседние" объекты для иерархического распространения объектов с учетом границ . В наших исследованиях абляции мы наблюдаем, что "соседние" объекты выигрывают в генерации глобальных объектов для задач классификации. Функции "центроида" полезны для распространения соответствующей информации о каждой отдельной точке, что приводит к лучшим результатам сегментации. Следуя операциям объединения пространственных пирамид [57], [58], мы также можем выполнять наши операции EP в нескольких масштабах для будущих исследований.

VI. ЗАКЛЮЧЕНИЕ

Мы предложили новую архитектуру PointAtrousGraph (PAG) для захвата многомасштабных локальных геометрических деталей для изучения иерархических точечных объектов. Многие новые модули предлагаются и затем оцениваются в наших экспериментах, что значительно повышает точность наших оценок при выполнении задач классификации и сегментации. В частности, наша сеть потребляет гораздо меньший объем обучающей памяти, чем предыдущие современные модели. Следовательно, наши архитектуры могут быть удобно применены для более сложных задач, связанных с крупномасштабными 3D-сценами.

Дополнительный материал

А. Обзор

Наш PointAtrousGraph (PAG) ориентирован на разработку архитектуры кодировщика-декодера, не зависящей от глубоких перестановок, для изучения функций иерархических границ для неорганизованного анализа облака точек. Следовательно, мы подчеркиваем и развиваем эти наиболее связанные и важные факторы в нашем основном документе:

анализ неорганизованного облака точек; 2) 1) инвариантность иерархическая структура кодер-декодер перестановка-облака точек. В этих документах представлены дополнительные обсуждения, анализ и качественные результаты.

В разделе В мы обсуждаем и сравниваем другие методы и / или представления для организации 3D-сцен. В разделе С мы раскрываем преимущества и недостатки для двух категорий методов глубокого обучения для анализа неорганизованного облака точек : 1) использование симметричных функций; 2) изучение канонических порядков. Подробности и исследования абляции наших сетевых архитектур представлены в разделе D, который также раскрывает преимущества и эффективность изучения многомасштабных локальных геометрических деталей как с помощью наших операций PAC, так и с сохранением границ объединения. В разделе E представлен более подробный анализ нашей точечной атрофической свертки (PAC). После этого мы проиллюстрируем интуицию и эффективность применения поиска по k ближайшим соседям в пространствах объектов и метрических пространствах в разделе F. В конце более качественные результаты визуализируются в разделе G.

В. Организованные 3D-структуры по сравнению с Неорганизованные облака точек

Помимо представления облака точек, 3D-сцены также могут быть представлены объемными сетками и масштабируемыми структурами индексации.

1) Объемная сетка: много работ [59], [60], [46], [61], [62] используйте объемные сетки для упорядочивания облаков точек для глубокого обучения. VoxNet [59] применяет 3D CNN к объемной сетке занятости, но дорогостоящие требования к вычислениям и памяти ограничивают его производительность. VoxelNet [61] кодирует облако точек как описательное объемное представление, и затем оно соединяется с сетью предложений регионов для генерации обнаружений. PointGrid [62] включает трехмерные точки внутри каждой ячейки объемной сетки. Это позволяет сети изучать локальные объекты 3D-облака точек более высокого порядка при меньшем объеме памяти. SSCN [63] вводит под-многообразные разреженные свертки, которые поддерживают разреженность вокселизированных входных данных на многих уровнях путем фиксации местоположения активных сайтов.

2) Масштабируемая структура индексирования: Масштабируемые структуры индексирования, например, kd-tree и octree, используются во многих существующих работах [64], [65], [66]. OctNet [64] иерархически разбивает трехмерное пространство на набор несбалансированных восьмиричных деревьев, что приводит к значительному сокращению вычислительных ресурсов и ресурсов памяти. O-CNN [65] сохраняет информацию об октантах и функциях CNN в графической памяти путем разработки структуры данных octree. Операции 3D CNN выполняются только над теми октантами, которые занимают поверхности 3D-формы. Его время вычислений и требования к памяти растут вчетверо с увеличением глубины проектируемого октодеревья. Kd-Net [66] использует структуру kd-дерева для построения вычислительного графа

с общими параметрами для изучения. Аналогично CNNs, Kd-Net вычисляет последовательность иерархических представлений в режиме прямой связи снизу вверх.

3) Сравнение и обсуждение:

а) Сравнение с представлением в объемной сетке

.: Основные преимущества использования объемных сеток заключаются в следующем: 1) объемная сетка является наиболее простым и понятным методом организации нерегулярных 3D-изображений ; 2) объемные сетки являются регулярными и равномерно распределены; 3) сетка занятий или поле расстояний могут быть использованы в качестве средства представления данных с помощью объемной сетки.

Однако очевидны и его недостатки: 1) объемная сетка rс печально известна своей объемной памятью и вычислительной неэффективностью; 2) объемная сетка плохо масштабируется для плотных 3D-данных, что не позволяет использовать богатую и детализированную геометрию исходных 3D-данных; 3) результаты вывода могут быть чувствительны к размеру объемной сетки, а также к разрешению 3D-сетки.

б) Сравнение с представлением масштабируемой структуры индексирования : Из-за неэффективности организации 3D точек с помощью объемных сеток для организации 3D облаков точек используются масштабируемые структуры индексации, такие как kd-tree и octree. Он формирует вычислительный граф для применения 3D свертки уровень за уровнем, где ядра свертки часто совместно используют обучаемые параметры. Следовательно, последовательность иерархических представлений может быть вычислена по принципу "снизу вверх" для опеки [62]. Самое главное преимущество применения этих конструкций является возможность использовать разреженности 3D-данных, и, таким образом, адаптивно распределять computa- онных ресурсов памяти и в отношении данных плотности. Однако эти структуры данных более сложны, что делает реализацию сетей очень сложной задачей.

с) Обсуждение.: Облако точек - это наиболее понятное представление 3D-структур. Проектирование сетей непосредственно на 3D-точках может быть намного более гибким и удобным по сравнению с применением двух описанных выше методов организации. Кроме того, наблюдаемые 3D-сцены в реальных приложениях обычно распределены нерегулярно и непредсказуемо, что может привести к очень ненадежному пониманию, если они организованы обычными структурами. Основной трудностью применения представления облака точек является отсутствие трехмерного упорядочения точек, а также неравномерное распределение. Чтобы смягчить это ограничение, предлагаются две категории сетей. Методы, относящиеся к первой категории, следуют за новаторской сетью PointNet [5], которая применяет симметричные функции для поддержания важного свойства, не зависящего от перестановок. Другая категория пытается изучить канонический порядок в 3D точках, а репрезентативной сетью является PointCNN [50]. Мы собираемся подробно обсудить и сравнить две категории методов в следующем разделе.

С. Изучение канонического порядка по сравнению с Использованием симметричных функций

Из-за неупорядоченности 3D-точек облако точек с N 3D-точек имеет всего N ! перестановок в данных

ТАБЛИЦА V

НЭТВОРК ДВ А РЕЗУЛЬТАТЫ АНАЛИЗА И					ТОЧНОСТЬ ОЦЕНЕНО			
ПРОТИВНОМ СЛУЧАЕ Конфигурация сети					Иерархия К	Частота субдискретизации	Согласно	
([32, 1],[32, 2])	[64, 2]	[128, 2]	[256, 4]	[512, 8]	1	10	Нет	92.4
	([32, 1],[32, 1])	[64, 2]	[128, 4]	[256, 8]	1	10	Нет	92.3
	[32, 2]	[128, 2]	[256, 2]	[512, 2]	5	10	2	92.2
	([32, 2])	2)[64, 2],[64, 2]	[256, 2]	[512, 2]	5	5	2	92.1
	[32, 1]	[64, 1],[64, 1],[64, 1]	[256, 2]	[512, 2]	5	10	2	92.2
	2)[128, 2],[128, 2]	[128, 1],[128, 2]	[256, 1],[256, 2]		4	10	2	92.3
	2)[32, 1],[64, 2]	[128, 1],[256, 2]			2	10	2	92.3
	([64, 1],[64, 2])	[128, 1],[128, 2]			2	10	2	92.1
	([64, 1],[64, 2])	[256, 1],[256, 2]			2	10	2	92.4
	([64, 1],[64, 2])	[256, 1],[256, 2]			2	10	4	92.3
	([32, 1],[32, 2])	[64, 1],[64, 2]	[128, 1],[128, 2]		3	10	2	92.1
	([64, 2],[64, 4])	[128, 2],[128, 4]	[256, 2],[256, 4]		3	5	2	92.0
	([64, 1],[64, 2])	[128, 1],[128, 2]	[256, 1],[256, 2]		3	10	2	92.3
	[32, 1],[32, 2]	[64, 1],[64, 2]	[128, 1],[128, 2]		3	5	4	91.5
	([64, 2],[64, 4])	[128, 2],[128, 4]	[256, 2],[256, 4]		3	10	4	91.8
([64, 1],[64, 2])	[256, 1],[256, 2]	[512, 1],[512, 2]		3	10	4	91.7	
([64, 1],[64, 2])	[128, 1],[128, 2]	[256, 1],[256, 2]		3	10	4	92.7	

порядок подачи. Мы надеемся, что результаты нашего вывода могут быть стабильными, надежными, а также инвариантными к порядку ввода баллов. Таким образом, предлагаются две стратегии: 1) сохранение инвариантности к перестановкам за счет использования только симметричных функций; 2) изучение канонического порядка. PointNet [5] и его последующие работы [6], [8], [7], [10], [43] поддерживают свойство инвариантности к перестановкам, применяя только симметричные функции, которое эффективно применяется и проверяется многими приложениями. Принадлежит к этой категории, все операции в нашем PAG симметричны, что делает наш PAG архитектурой, не зависящей от перестановок.

Недавно выпущенный PointCNN [50], целью которого является изучение χ -преобразования входных точек для перестановки самих себя в канонические порядки. В идеале, их вывод результаты могут быть инвариантны к порядку входных точек с помощью этого χ -преобразования. Однако их изученные χ -преобразования далеки от идеала, особенно с аспектом эквивалентности перестановок. Другими словами, им не удалось решить проблему обучения каноническому упорядочиванию с помощью своих χ -преобразований. Хотя они предоставили самые современные результаты по различным тестам, их результаты могут зависеть и быть чувствительными к порядку подачи 3D-точек. Кроме того, они не предоставляют ковариационный анализ, который важен и необходим для анализа влияния порядка подачи точек на точность вывода. Кроме того, PointCNN готовят свои собственные наборы данных для обучения и тестирования, а также применяют множество специфических стратегий во время обучения. Все эти операции предоставляют PointCNN дополнительные и несправедливые преимущества. Следовательно, результаты, предоставленные PointCNN, не могут быть напрямую сопоставлены с теми сетями, не зависящими от перестановок, которые следуют идее, лежащей в основе PointNet. Как работа на ранней стадии, строгое понимание PointCNN до сих пор остается открытой и нерешенной проблемой.

D. Сведения об архитектуре сети.

Наши сети классификации и сегментации используют одну и ту же архитектуру кодировщика. Конфигурация кодера, которая обеспечивает наилучшие результаты, называется Encoder([64, 1], [64, 2]; [128, 1], [128, 2]; [256, 1], [256, 2], гдe [C, r] обозначает размер функционального канала C и частоту дискретизации r соответственно. Частота субдискретизации между каждыми двумя иерархиями установлена равной 4.

Например, если входные данные содержат 1024 точки, то после обработки EP будет выбрано 256 точек.

После кодирования мы добавляем общий слой mlp для проецирования каждого точечного объекта в многомерное пространство объектов. Уровни с полной связью (FC) в нашей сети классификации имеют следующий размер объекта: 512, 256 и количество общих классов в оцениваемом наборе данных. Что касается нашей сегментации сети, мы задаем слоям FC следующий размер объекта: 512 и 1024. Декодер нашей сети сегментации имеет аналогичную архитектуру с нашим кодировщиком, но все конфигурации устанавливаются обратно пропорционально: Декодер([256, 2], [256, 1], [128, 2], [128, 1], [64, 2], [64, 1]). Аналогично, частота дискретизации с повышением также установлена равной 4. Каждый слой в подсэмплингах с цепным пропуском слоев с повышением дискретизации имеет ту же конфигурацию, что и соответствующий слой в основном потоке.

Конфигурация нашего декодера для сети классификации выбирается и фиксируется заранее, наш декодер для сети сегментации следует настройкам нашего сетевого кодера. Поэтому важно найти наилучшую конфигурацию для нашего сетевого кодера. Мы оценили несколько различных архитектур декодеров в задаче классификации, как указано в таблице V. В наших экспериментах мы наблюдаем, что идея изучения пограничных признаков в разных масштабах восприимчивых полей действительно улучшает нашу точность вывода, а также скорость сходимости. Только применение модулей PAC (иерархия равна 1) может повысить точность вывода, но обучение может быть неэффективным, особенно при работе с плотными точечными объектами высокой размерности. Применение иерархических архитектур без модулей PAC может привести к повышению эффективности, но к ограниченным областям восприятия в каждой сетевой иерархии. Таким образом, наши модули PAC и EP приносят пользу друг другу в улучшении возможностей обучения работе с локальными геометрическими деталями разного масштаба. С одной стороны, PAC может увеличивать поле зрения фильтров без уменьшения плотности точечных объектов. С другой стороны, EP увеличивает восприимчивые поля, а также эффективность сети. Время от времени могут наблюдаться некоторые колебания в эффективности обучения. Однако выбранная нами конфигурация часто обеспечивает относительно стабильные результаты, особенно при применении стратегии обучения без ротации.

Время и сложность памяти.

В таблице VI приводится общее значение

ТАБЛИЦА VI

м Время ОЖИДАНИЯ И ВРЕМЯ ДЕЙСТВИЯ

Метод	#Параметры (М)	Fwd. (мс)
PointNet (ванильный)	0.8	11.5
[5] PointNet [5]	3.5	29.3
PointNet++ (msg) [6]	1.7	98.1
DGCNN [7]	1.8	173.7
SpiderCNN (трехслойный)	3.2	275.4
[8] PAG (наш)	1.8	109.8

количество параметров, которые необходимо обучить, и среднее время вывода для нескольких репрезентативных сетей. По сравнению с другими сетями [7], [8], которые в значительной степени полагаются на графики ближайших окрестностей, наша модель требует меньшего времени для единого прямого процесса. Е. Анализ точечной свертки

Наша точечная свертка основана на свертке atrous [33] в области изображений. Обычная свертка ограничена в локальных регионах из-за применения ядер преобразования 3×3 . Чтобы увеличить восприимчивые поля ядер свертки без увеличения общего объема вычислений, в обычном ядре свертки вводится частота дискретизации $г$ для выбора и выборки объектов соседнего изображения. Наиболее важным преимуществом atrous convolution является то, что оно эффективно расширяет поле зрения для каждого операционного ядра без ущерба для плотности объектов, которая оценивается многими приложениями на основе изображений, особенно в задачах семантической сегментации [33], [2], [1], [67].

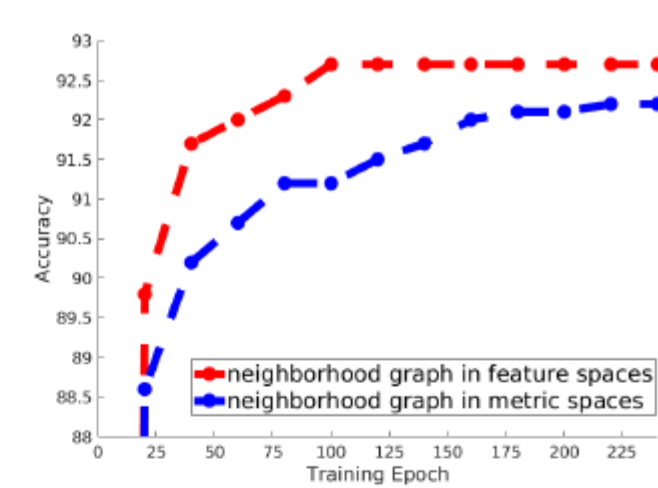
Принимая во внимание успех arous convolution в областях изображений, мы расширяем идею arous convolution для анализа облаков точек. В аналогичном духе мы применяем частоту выборки $г$ к эквивалентно разреженной выборке и выбираем соседние точки для использования и агрегирования контекстуальной информации из более крупного локального региона соседства. Следовательно, мы вводим нашу точечную свертку (РАС). Наш РАС позволяет нашей сети использовать локальную контекстную информацию в 3D-точках с гибкими масштабами. Однако это может быть неэффективно, когда мы распространяем точечные объекты на пространство многомерных объектов. В этой работе мы также предлагаем нашу операцию объединения (ЕР) с сохранением границ, которая также способна расширять восприимчивые поля для наших ядер свертки. Тем не менее, плотность точечных объектов уменьшается после операции ЕР. В разделе

наши эксперименты, мы видим, что обе операции приносят пользу. друг друга. Дополнительные результаты для различных сетевых архитектур подробно описаны в разделе VI-D.

Ф. Сочетание поиска по К ближайшим соседям в пространствах объектов и метрических пространствах

В нашей архитектуре PAG мы используем соседние точечные объекты в пространствах объектов и метрических пространствах. Наш модуль РАС применяется для изучения граничных объектов путем рассмотрения локальных соседних точечных объектов, выбранных в пространствах объектов. В наших экспериментах мы наблюдаем, что построение графиков окрестностей в пространствах объектов для извлечения граничных объектов может привести к лучшим результатам вывода, а также к более высокой скорости сходимости во время обучения, чем построение графиков окрестностей в метрических пространствах, как показано на рис. 8.

Однако, что касается построения глубоких иерархических сетей-работает, предлагаемый пул с сохранением границ (ЕР) и edge-



Точность проверки (показать наилучшую) для выбора РАС соседних точек
Рис. 8. в пространствах объектов и метрических пространствах во время обучения.

сохраненное развертывание (ЕU) позволяет выбрать соседние точки в метрических пространствах. Интуитивно понятные принципы проектирования модулей ЕР и ЕU основаны на операциях объединения и распаковки в доменах изображений. Эти операции для анализа изображений суммируют / восстанавливают локальную контекстуальную информацию с помощью регулярных шаблонов изображений. Из-за отсутствия регулярных сеток в 3D-точках локальные геометрические детали могут быть суммированы / восстановлены путем построения вместо этого графиков локальной окрестности в метрических пространствах. Более того, построенные графики окрестностей в метрических пространствах статичны, тогда как графики окрестностей в пространствах объектов динамичны. Соответственно, поддержание стабильной иерархической структуры может способствовать сближению общей сетевой структуры, особенно функционирования ЕС, что является второй причиной. И последнее, но не менее важное: в основной статье мы обсуждаем и анализируем важную проблему - проблему перекрытого графа окрестностей. Если мы выбираем соседние точечные объекты в пространствах объектов, может часто возникать проблема перекрытия графа окрестностей, что также отклоняется от нашего первоначального намерения - зафиксировать характерные локальные геометрические детали.

G. Более качественные результаты

В этом разделе представлены более качественные результаты.

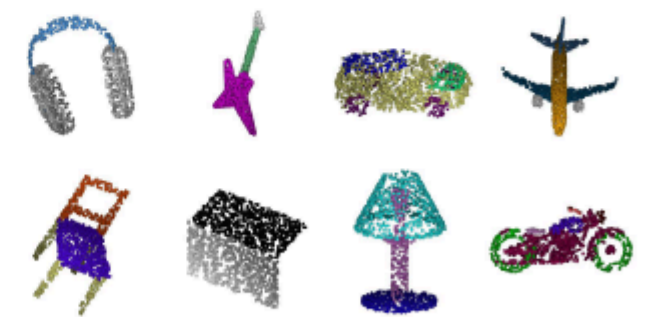
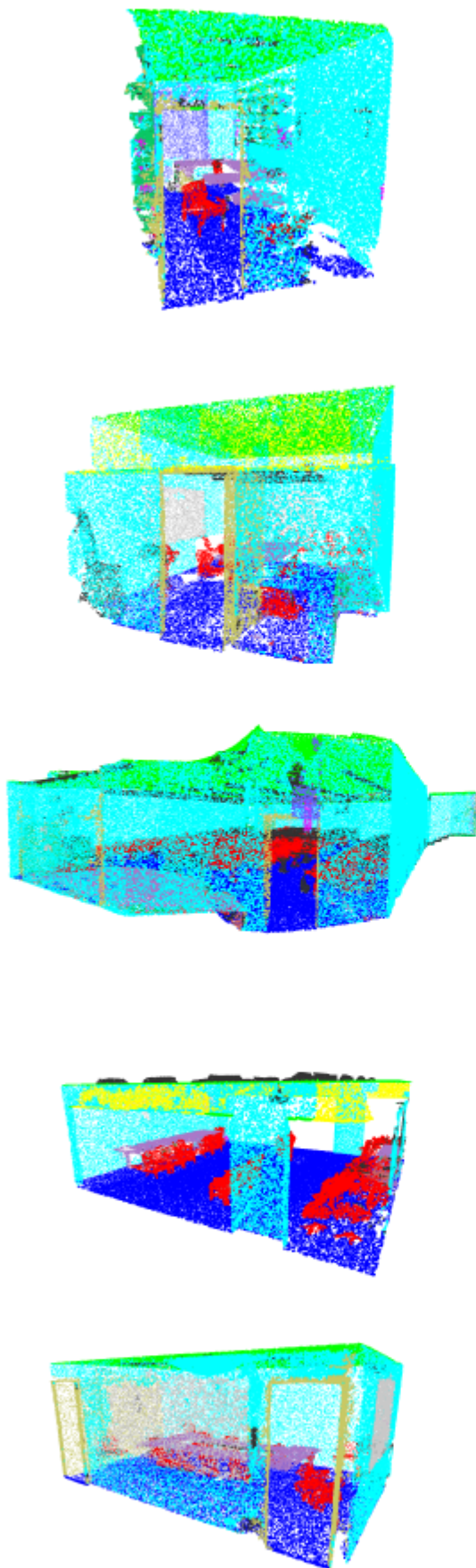


Рис. 9. Более качественные результаты в ShapeNet [51]

Р ССЫЛКИ

[1] Л.-К. Чен, Ю. Чжу, Г. Папандреу, Ф. Шрофф и Х. Адам, "Кодировщик-декодер с гибкой разделяемой сверткой для семантической сегментации изображений", в Трудях Европейской конференции по компьютерному зрению (ECCV), 2018, стр. 801-818.



Более качественные результаты для 3D-семантической сегментации на S3DISFig. 10. набор данных [52].

[2] Л.-К. Чен, Г. Папандреу, Ф. Шрофф и Х. Адам, "Переосмысление сложная свертка для семантической сегментации изображений", препринт arXiv arXiv: 1706.05587, 2017. [3] Х. Чжао, Дж. Ши, Х. Ци, Х. Ван и Дж. Цзя, "Анализ сцены пирамиды".

сеть", в Трудях конференции IEEE по компьютерному зрению и распознаванию образов, 2017, стр. 2881-2890. [4] К. Хэ, Х. Чжан, С. Жэнь и Дж. Сун, "Объединение пространственных пирамид в глубоком

сверточные сети для визуального распознавания", IEEE transactions on анализ шаблонов и машинный интеллект, том 37, № 9, стр. 1904-1916, 2015. [5] К. Р. Ци, Х. Су, К. Мо и Л. Дж. Гибас, "Pointnet: глубокое обучение".

о наборах точек для 3d-классификации и сегментации", Proc. Компьютерное зрение и распознавание образов (CVPR), IEEE, том 1, № 2, стр. 4, 2017. [6] К. Р. Ци, Л. Йи, Х. Су и Л. Дж. Гибас, "Pointnet ++: глубокая иерархия".

изучение химических признаков на множествах точек в метрическом пространстве", в "Достижения в системах обработки нейронной информации", 2017, стр. 5099-5108. [7] Ю. Ван, Ю. Сун, З. Лю, С. Э. Сарма, М. М. Бронштейн и Дж. М.

Соломон, "Динамический график спл для обучения в облаках точек", arXiv препринт arXiv: 1801.07829, 2018. [8] Ю. Сюй, Т. Фан, М. Сюй, Л. Цзэн и Ю. Цяо, "Spiderscnn: глубокое обучение".

о наборах точек с параметризованными сверточными фильтрами", препринт arXiv, arXiv: 1803.11527, 2018.

[9] Л. Пан, П. Ван и К. М. Чу, "Pointatrousnet: точечная сеть

свертка для анализа облака точек", Письма, 2019.

[10] Ю. Шен, К. Фенг, Ю. Янг и Д. Тянь, "Локальный анализ облака точек".

структуры с помощью корреляции ядра и объединения графов", в Трудях Конференции IEEE по компьютерному зрению и распознаванию образов, том 4, 2018. [11] Л. Ландриу и М. Симоновский, "Крупномасштабное облако точек se-

мантическая сегментация с помощью arXiv: 1711.09869, 2017.

графики сверточек",

arXiv

препринт

[12] У. Ван, Р. Ю, К. Хуан и У. Нейман, "Sgrp: сходство

сеть групповых предложений для сегментации экземпляров 3d-облака точек", в Материалах конференции IEEE по компьютерному зрению и распознаванию образов, 2018, стр. 2569-2578.

[13] Вопрос Хуанг, У. Ван и У. Нейман, "Сети рекуррентных срезов

для 3D-сегментации облаков точек", в Трудях IEEE Конференция по компьютерному зрению и распознаванию образов, 2018, стр. 2626-2635. [14] А. Ньюэлл, К. Янг и Дж. Дэн, "Сети с многоуровневыми песочными часами для

оценка позы человека" на Европейской конференции по компьютерному зрению. Спрингер, 2016, с. 483-499. [15]

Г. Павлакос, Х. Чжоу, К. Г. Дерпанис и К. Данилидис, "Трубый-

точное прогнозирование объема для позы человека в формате 3D с одним изображением", в Материалах конференции IEEE по компьютерному зрению и распознаванию образов, 2017, стр. 7025-7034.

[16] О. Роннебергер, П. Фишер и Т. Брокс, "U-net: Сверточная

сети для сегментации биомедицинских изображений" в рамках Международной конференции по вычислению медицинских изображений и компьютерному вмешательству. "Сети для сегментации биомедицинских изображений". Спрингер, 2015, с. 234-241. [17] В. Бадринараянан, А. Кендалл и Р. Чиполла, "Сегнет: глубокая

архитектура сверточного кодировщика-декодера для сегментации изображений", IEEE transactions on pattern analysis and machine intelligence, том 39, № 12, стр. 2481-2495, 2017. [18]

Т. Полен, А. Херманс, М. Матиас и Б. Лейбе, "Полное разрешение

остаточные сети для семантической сегментации в уличных сценах", в Материалах конференции IEEE по компьютерному зрению и распознаванию образов, 2017, стр. 4151-4160. [19]

Г. Лин, А. Милан, К. Шен и И. Рид, "Refinenet: многоканальный

сети уточнения для семантической сегментации высокого разрешения", в Материалах конференции IEEE по компьютерному зрению и распознаванию образов, 2017, стр. 1925-1934. [20]

С. Пэн, Х. Чжан, Г. Ю, Г. Ло и Дж. Сун, "Большое ядро имеет значение".

улучшить семантическую сегментацию с помощью глобальной сверточной сети", в Материалах конференции IEEE по компьютерному зрению и распознаванию образов, 2017, стр. 4353-4361. [21]

М. Амирул Ислам, М. Рочан, Н. Д. Брюс и Ю. Ван, "Закрытый канал-

сеть обратного уточнения для плотной маркировки изображений", в Трудях Конференции IEEE по компьютерному зрению и распознаванию образов, 2017, стр. 3751-3759. [22]

З. Война, В. Феррари, С. Гвадаррама, Н. Зильберман, Л.-К. Чен,

А. Фатхи и Дж. Уиджлингс, "Дьявол в декодере", arXiv: 1707.05847, 2017.

Препринт arXiv

[23] Дж. Фу, Дж. Лю, Ю. Ван, Дж. Чжоу, К. Ван и Х. Лу, "Сложенный декон-
объемная сеть для семантической сегментации", IEEE
Transactions on Обработка изображений, 2019. [24] З.
Чжан, Х. Чжан, К. Пэн, Х. Сюэ и Дж. Сун, "Exfuse: улучшение

объединение функций для семантической сегментации," в
Материалы конференции
Европейская конференция по компьютерному зрению
(ECCV), 2018, стр. 269- 284. [25] Х. Но, С.
Хонг и Б. Хан, "Обучающая сеть деконволюции для
семантическая сегментация", на
Материалы международного IEEE
конференции по компьютерному зрению, 2015, стр. 1520-1528.

[26] А. Досовицкий, П. Фишер, Э. Ильг, П. Хауссер, К. Хазирбас, В. Голков,
П. Ван Дер Смагт, Д. Кремерс и Т. Брокс, "FlowNet: изучение оптического
потока с помощью сверточных сетей", в Труды IEEE международная
конференция по компьютерному зрению, 2015, стр. 2758-2766.

[27] Э. Илг, Н. Майер, Т. Сайкия, М. Кеупер, А. Досовицкий и Т. Брокс,
"FlowNet 2.0: эволюция оценки оптического потока с помощью
deep net- работает", в Материалах конференции IEEE по
компьютерному зрению и распознаванию образов, 2017, стр. 2462-2470.

[28] Т.-Я. Лин, П. Доллар, Р. Гиршик, К. Хе, Б. Харихаран и С. Белони,
"Сети пирамид объектов для обнаружения объектов", в
Материалы конференции

Конференция IEEE по компьютерному зрению и
распознаванию образов, 2017, стр. 2117-2125. [29] А. Шривастава,
Р. Суктханкар, Дж. Малик и А. Гупта, "За пределами
пропускать соединения: модуляция сверху вниз для
обнаружения объектов", препринт arXiv: 1612.06851, 2016.
arXiv

[30] К.-Ю. Фу, В. Лю, А. Ранга, А. Тяги и А. К. Берг, "Dssd: De-
сверточный детектор с одним выстрелом", препринт
arXiv arXiv: 1701.06659, 2017. [31] А. Комаричев,
З. Чжун и Дж. Хуа, "А-спл: кольцевая свертка-
традиционные нейронные сети на облаках точек", в
Труды IEEE Конференция по компьютерному зрению и
распознаванию образов, 2019, стр. 7421-7430.

[32] М. Цзян, Ю. Ву и К. Лу, "Pointsift: сетевой мод, подобный sift-
ule для семантической сегментации 3d-облака точек",
препринт arXiv arXiv: 1807.00652, 2018. [33] Л.-К.
Чен, Г. Папандреу, И. Кожинос, К. Мерфи и А. Л. Юилле,
"Deerlab: семантическая сегментация изображений с помощью
глубоких сверточных сетей, горизонтальной свертки и полностью
связанных CRF", IEEE transactions on анализ шаблонов и
машинный интеллект, том 40, № 4, стр. 834- 848, 2018.

[34] К. Симонян и А. Зиссерман, "Очень глубокие сверточные сети
для крупномасштабного распознавания изображений", препринт
arXiv arXiv: 1409.1556, 2014. [35] К. Хэ, Х. Чжан, С. Жэнь
и Дж. Сун, "Глубокое остаточное обучение для создания образа".
распознавание", в Труды конференции IEEE по компьютерному
зрению и распознаванию образов, 2016, стр. 770-778. [36] А.
Крижевский, И. Суцкевер и Г. Э. Хинтон, "Классификация ImageNet

с глубокими сверточными нейронными сетями", в "Достижениях в
области нейронных систем обработки информации", 2012, стр. 1097-1105.

[37] Д. Шерер, А. Мюллер и С. Бенке, "Оценка объединения ор-
преобразования в сверточных архитектурах для распознавания
объектов", на Международной конференции по искусственным
нейронным сетям. Спрингер, 2010, стр. 92-101.

[38] Д. К. Чирезан, У. Мейер, Дж. Маски, Л. М. Гамбарделла и Дж. Шмид-
хубер, "Гибкие, высокопроизводительные сверточные нейронные сети
для классификации изображений", на Двадцать Второй
Международной совместной конференции по искусственному интеллекту, 2011.

[39] Ю. Лекун, Л. Боттоу, Ю. Бенджио, П. Хаффнер и др., "Основанный на градиенте
обучение, применяемое к распознаванию документов", Труды
IEEE, том 86, № 11, стр. 2278-2324, 1998. [40] Ф. Грок,
П. Вишоллек и Х. Ленц, "Гибкая свертка (глубокое изучение-
ing beyond grid-worlds)", препринт arXiv arXiv: 1803.07289, 2018.

[41] С. Чжао, Дж. Сонг и С. Эрмон, "Infovae: максимизация информации".
вариационные автокодеры", препринт arXiv arXiv: 1706.02262, 2017.

[42] Х. Чжао, Ю. Чжан, С. Лю, Дж. Ши, К. Чендж Лой, Д. Лин и Дж. Цзя,
"Psanet: точечная сеть пространственного внимания для
анализа сцен", в Материалах Европейской конференции по
компьютерному зрению (ECCV), 2018, стр. 267-283.

[43] Дж. Ли, Б. М. Чен и Г. Х. Ли, "So-net: самоорганизующаяся сеть".

для анализа облака точек", в Труды конференции IEEE по
компьютерному зрению и распознаванию образов, 2018, стр. 9397-9406.

[44] Дж. Лонг, Э. Шелхамер и Т. Даррелл, "Полностью сверточные сети
для семантической сегментации", в компьютерном зрении
и распознавани образов, 2015, стр. 3431-3440. Материалы конференции IEEE по

[45] М. Абади, П. Бархэм, Дж. Чен, З. Чен, А. Дэвис, Дж. Дин, М. Девин,

С. Гемават, Г. Ирвинг, М. Айсард и др., "Tensorflow: система для крупномасштабного
машинного обучения", в, OSDIvol. 16, 2016, стр.
265-283. [46] Z. Бу, С. Сонг, А. Хоспа. Ф. Ю, Л. Чжан, Х. Тан и Дж. Сю,

"3d shapenets: глубокое представление объемных форм", в
Трудах конференции IEEE по компьютерному зрению и
распознаванию образов , 2015, стр. 1912-1920. [47] С. Ван, Б.
Самари и К. Сиддики, "Локальная свертка спектральных графиков

для изучения объектов с набором точек", препринт arXiv arXiv: 1803.05827, 2018.
[48] М. Гаделья, Р. Ван и С. Маджи, "Древовидные сети с несколькими решениями для
обработка 3d-облака точек", препринт arXiv arXiv: 1807.03520, 2018.

[49] М. Ацмон, Х. Марон и Ю. Липман, "Точечный сверточный нейронный
сети от операторов расширения", препринт
arXiv arXiv: 1803.10091, 2018. [50] Ю. Ли,
Р. Бу, М. Сун и Б. Чен, "Pointcnn", препринт arXiv

arXiv: 1801.07791, 2018.

[51] Л. И. В. Г. Ким, Д. Джейлан, И. Шен, М. Янь, Х. Су, К. Лу, К. Хуан,
А. Шеффер, Л. Гибас и др., "Масштабируемый активный
фреймворк для региона" аннотации в коллекциях 3d-фигур", ACM Transactions
on Graphics (TOG), том 35, № 6, стр. 210, 2016. [52]
И. Армени, О. Сенер, А. Р. Замир, Х. Цзян, И. Брилакис, М. Фишер,

и С. Савареze, "3d-семантический анализ крупномасштабных внутренних
пространств", в Материалах конференции IEEE по компьютерному
зрению и распознаванию образов , 2016, стр. 1534-1543.

[53] Л. Чапми, К. Чой, И. Армени, Дж. Гвак и С. Савареze, "Segcloud:

Семантическая сегментация 3d-облаков точек" в 3D Vision (3DV), 2017
Международная конференция по. IEEE, 2017, стр. 537-547.

[54] С. Ван, С. Суо, В.-К. Ма, А. Покровский и Р. Уртасун, "Глубокий

параметрические непрерывные сверточные нейронные
сети", в Труды Конференции IEEE по компьютерному зрению и
распознаванию образов, 2018, стр. 2589-2597. [55]
Х. Су, В. Джампани, Д. Сан, С. Маджи, Э. Калогеракис, М.-Х. Янг,

и Дж. Кауцц, "Splatnet: сети с разреженной решеткой для
обработки облаков точек.", в материалах конференции IEEE по компьютерному
зрению и распознаванию образов, 2018, стр. 2530-2539. [56]
Ф. Энгельманн, Т. Контоджианни, А. Херманс и Б. Лейбе, "Исследуя

пространственный контекст для трехмерной семантической сегментации
облаков точек", в Материалах конференции IEEE по компьютерному
зрению и распознаванию образов , 2017, стр. 716-724. [57] К. Грауман
и Т. Даррелл, "Ядра соответствия пирамиде: дискриминационный

классификация с помощью наборов признаков изображения (версия 2)", 2006.

[58] С. Лазебник, К. Шмид и Дж. Понсе, "За пределами набора функций":
Сопоставление пространственных пирамид для распознавания категорий природных
сцен", на конференции компьютерного общества IEEE 2006 по компьютерному
зрению и распознаванию образов (CVPR'06), том. 2. IEEE, 2006, стр.
2169-2178. [59] Д. Матурана и С. Шерер, "Voxnet: трехмерный сверточный нейронный

сеть для распознавания объектов в реальном времени" в
интеллектуальных роботах и системах (IROS), 2015 Международная
конференция IEEE/RSJ по. IEEE, 2015, стр. 922-928.
[60] К. Ци, Х. Су, М. Ниснер, А. Дай, М. Ян и Л. Дж. Гибас,

"Объемные и многовидовые CNN для классификации объектов по
3D-данным", в Материалах конференции IEEE по компьютерному зрению
и распознаванию образов , 2016, стр. 5648-5656. [61] Ю.
Чжоу и О. Тузель, "Voxelnet: комплексное обучение для облака точек

обнаружение 3d-объектов на основе", препринт arXiv arXiv:
1711.06396, 2017. [62] Т. Ле и Ю. Дуан, "Pointgrid: глубокая сеть для 3d-формы под-
standing", в Труды Конференции IEEE по компьютерному зрению
и распознаванию образов, 2018, стр. 9204-9214.
[63] Б. Грэм, М. Энгельке и Л. ван дер Маатен, "3d semantic

сегментация с помощью подмногобразных разреженных
сверточных сетей", arXiv препринт arXiv: 1711.10275, 2017. [64]
Г. Риглер, А. О. Улуосой и А. Гейгер, "Octnet: глубокое изучение

3d-представления с высоким разрешением" в
Материалы IEEE
Конференция по компьютерному зрению и распознаванию
образов, том 3, 2017. [65] П.С. Ван, Ю. Лю, Ю.-Х. Го, К.-Ю. Сун и Х. Тонг, "O-спл:
Сверточные нейронные сети на основе Octree для анализа 3d-формы",
ACM Transactions on Graphics (TOG), том 36, № 4, стр. 72, 2017.
[66] Р. Клоков и В. Лемпицкий, "Побег из ячеек: глубокие kd-сети

для распознавания 3d-моделей облака точек" в области
компьютерного зрения (ICCV), Международная конференция IEEE 2017
по. IEEE, 2017, стр. 863-872. [67] Ф. Ю. и В. Колтун,
"Многомасштабная агрегация контекста с помощью расширенных
свертки", препринт arXiv arXiv: 1511.07122, 2015.