

© 2022 г. С.Р. ОРЛОВА (farhulina.sr@edu.spbstu.ru)
(Санкт-Петербургский политехнический университет Петра Великого),
А.В. ЛОПОТА, д-р техн. наук (alopota@rtc.ru)
(Центральный научно-исследовательский и опытно-конструкторский институт
робототехники и технической кибернетики, Санкт-Петербург)

ТРЕХМЕРНОЕ РАСПОЗНАВАНИЕ: ТЕКУЩЕЕ СОСТОЯНИЕ И ТЕНДЕНЦИИ¹

Рассматривается область трехмерного технического зрения и, в частности, трехмерного распознавания. Выделены задачи трехмерного технического зрения, проведены обзоры способов получения и представления трехмерных данных, приложений трехмерного технического зрения. Проведен обзор методов глубокого обучения в задачах трехмерного распознавания. Выявлены основные современные тенденции области. На данный момент предложено большое количество архитектур нейронных сетей, сверточных слоев, операций сэмплирования, пулинга и агрегации, способов представления и обработки входных трехмерных данных. Область находится в стадии активного развития, при этом наибольшее разнообразие методов представлено для облаков точек.

Ключевые слова: трехмерное распознавание, глубокое обучение, компьютерное зрение.

DOI: 10.31857/S000523102204002X, **EDN:** AAALWB

1. Введение

Трехмерное техническое или компьютерное зрение становится все более актуально: технологии и оборудование, необходимые для получения трехмерных представлений, их обработки и визуализации, становятся дешевле и доступнее, вместе с этим растет и заинтересованность в приложениях трехмерного компьютерного зрения. Появление технологий виртуальной и дополненной реальности, рост популярности мобильных роботов и разнообразных сервисных интеллектуальных систем дополнительно стимулируют развитие технического зрения.

Особенно интересна область трехмерного распознавания, поскольку она направлена на решение задачи понимания окружающего мира, являющейся одной из основных проблем технического и компьютерного зрения. Несмотря на то что был достигнут большой прогресс в задачах распознавания изображений, трехмерные модели дают намного более полное представление о

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 20-37-90039).

сцене, объектах в ней и их взаимном расположении. Интуитивно понятно, что распознавание трехмерных представлений, хотя и является более сложной задачей, должно позволить получить более качественные и устойчивые системы, действующие в пространстве или анализирующие его и объекты в нем. И хотя очевидно, что проблема понимания машиной окружающего мира не будет решена в ближайшем будущем, стоит начать формировать фундамент для будущих исследований, развивая область трехмерного распознавания. Кроме того, существует множество менее глобальных задач, которые вполне под силу решить современной науке и технике.

2. Трехмерное техническое зрение

Трехмерное техническое (компьютерное) зрение предполагает работу с трехмерными представлениями существующих или синтезированных объектов и сцен. Можно выделить несколько типичных задач трехмерного технического зрения:

1. *Получение трехмерных представлений существующих объектов и сцен* — трехмерное сканирование. Для этой задачи существует множество методов и аппаратуры, существуют также нейросетевые методы, генерирующие трехмерные представления;
2. *Обработка, фильтрация и анализ трехмерных представлений* может преследовать различные цели: фильтрация шумов и выбросов, коррекции (например, интерполяция или прореживание), совмещение нескольких частичных представлений в одно, приведение к другому виду представления (например, получение полигональной модели из облака точек);
3. *Построение карт и локализация, SLAM* — популярны в мобильной робототехнике;
4. *Визуализация трехмерных представлений* — может быть как вспомогательной (контроль результатов при работе с трехмерными представлениями), так и основной (в приложениях дополненной реальности) задачей;
5. *Распознавание образов по трехмерным представлениям, семантический анализ* — смежное направление, привлекающее различные методы компьютерного зрения, машинного и глубокого обучения. Задачи этой группы направлены на получение из трехмерных представлений информации более высокого уровня абстракции: например, распознавание в облаке точек сцены в виде жилой комнаты с мебелью и людьми. Это необходимо для создания систем, способных понимать окружающий мир и взаимодействовать с ним.

2.1. Представления трехмерных данных

Трехмерные данные, отображающие сцены и объекты реального мира или синтезированные каким-либо методом, построенные дизайнером, могут быть представлены в различных формах. Объединив знания из таких областей,

как техническое зрение и компьютерная графика, можно составить единый список, включающий все наиболее распространенные формы представлений:

1. Облако точек (Point cloud);
2. Карта глубины (Depth map), зачастую рассматривается как 2,5-мерное представление;
3. RGB-D изображение, также рассматривается как 2,5-мерное представление;
4. Коллекция изображений — строго говоря, не является трехмерным представлением. Однако имеется большое количество методов получения облака точек из коллекции изображений (с каналом глубины и без него), включая раскадровку видеоряда;
5. Трехмерные (3D) модели:
 - а. Каркасные модели;
 - б. Граничные (поверхностные) модели;
 - в. Твёрдотельные (Solid) модели:
 - Конструктивная сплошная (блочная) геометрия (КБГ, constructive solid geometry, CSG);
 - Воксельные (Voxel) модели;
 - г. Полигональная сетка (Polygon mesh);
 - д. Параметрические модели;
 - е. Неявные поверхности (Implicit surfaces).

Наиболее популярными представлениями являются облако точек, полигональная сетка и воксельная модель. Карта глубины, RGB-D изображение и коллекция мультиракурсных изображений, строго говоря, не являются трехмерными представлениями, однако чрезвычайно популярны (RGB-D камеры становятся все более доступны) и используются в фотограмметрических методах реконструкции объекта или сцены.

Облако точек — это набор точек в пространстве, представляющий фигуры, объекты, поверхности. Каждая точка представлена трехмерными координатами, также могут быть дополнительные атрибуты, например интенсивность или цвет. Облака точек просто получать и хранить, они позволяют быстро выполнять линейные преобразования в матричном представлении. Также облака точек легко совмещать друг с другом и отображать на плоскость. Большинство методов трехмерного сканирования получают именно облако точек.

Карта глубины — двумерное одноканальное изображение, содержащее информацию о расстоянии от поверхностей сцены до точки (области) съемки.

RGB-D изображение — это двумерное четырехканальное изображение, содержащее в одном канале карту глубины, а в оставшихся трех — цветовую информацию (RGB изображение сцены).

Воксельные (от англ. volumetric pixel — объемный пиксель) модели или *воксельные сетки (voxel grid)* основаны на следующем подходе: все трехмерное пространство представляется в виде равномерной сетки или матрицы, в ячейках которой находятся воксели — трехмерные, объемные пиксели. Положение вокселя обозначается его позицией в матрице, а не координатами

image_id	111
camera_id	1
camera_model	SIMPLE_RADIAL
camera_params	2559,680000, 1536,000000, 1152,000000, -0,020500
qw, qx, qy, qx	0,93735, -0,0312744, -0,338602, 0,0757936
tw, ty, tz	1,5528, 0,900081, 2,90801
dims	3072×2304
num_points2D	12738
num_points3D	1300
num_observations	0
name	P1180201.JPG

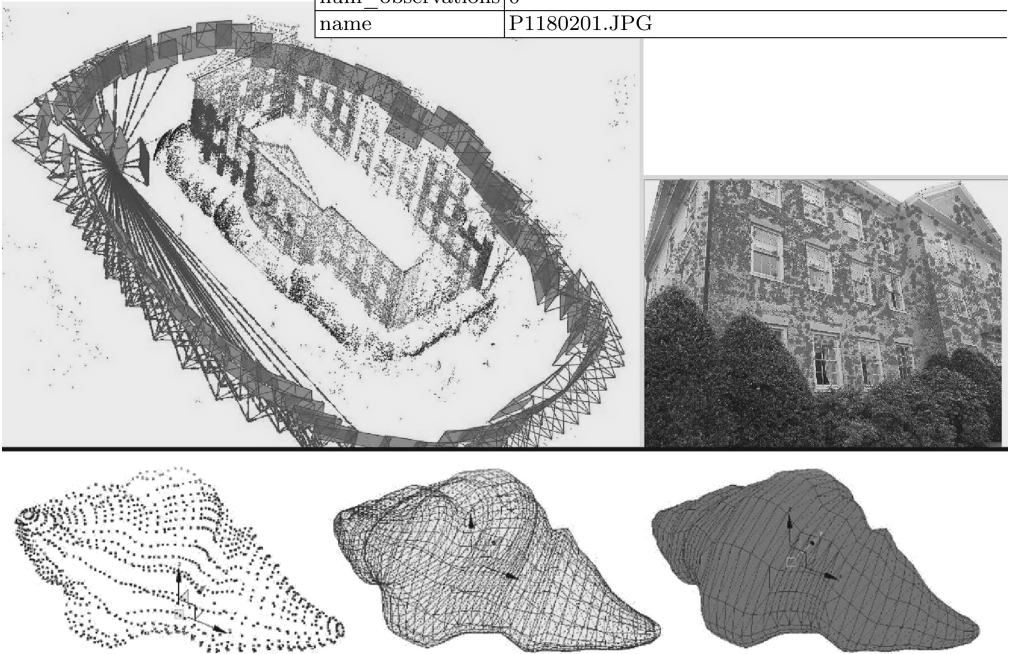


Рис. 1. Сверху слева: Облако точек, построенное в программе COLMAP [1] по 128 изображениям (RGB) одного из зданий университета Северной Каролины в Чапел-Хилл [2]. Показаны вычисленные положения камеры. Выбрано одно положение (показано более темным цветом). Сверху справа: Изображение, соответствующее выбранному положению, и его параметры. Отмечены точки (пиксели), использованные при построении облака. Внизу: Трехмерная модель круассана, основанная на полигональной сетке. Слева показаны вершины, по центру — ребра, справа — грани модели. Изображение взято из [3].

в пространстве. Воксель может содержать множество атрибутов, например цвет или материал. Воксельные модели довольно часто используются в медицине в задачах анализа.

Модели, основанные на полигональной сетке (polygon mesh), как и граничные, состоят из вершин, ребер и граней.

Но в отличие от граничных моделей грани полигональной сетки всегда плоские, т.е. являются полигонами. На текущий момент модели, основанные на треугольной полигональной сетке, являются наиболее популярной формой представления трехмерных данных в области компьютерной графики и в задачах интерактивной визуализации.

2.2. Приложения трехмерного распознавания

В настоящий момент с повышением доступности сенсорной, вычислительной техники и программных средств, способных выполнять трехмерное сканирование и реконструкцию, имеется широкое поле возможных приложений трехмерного технического зрения и, в частности, распознавания.

В медицине методы глубокого обучения используются для сегментации трехмерных снимков и обнаружения и оценки положения, размеров и формы каких-либо элементов — органов, опухолей и т.д. При этом часто используется воксельное представление, так как многие методы сканирования предполагают получение набора снимков при движении сканера в одном направлении с заданным шагом. Возможны приложения, помогающие оценить позу, сутулость пациента и характер его движений.

В геоматике трехмерное распознавание может быть полезно в задачах разметки местности и поиска каких-либо объектов в массивах трехмерных данных. Генерация трехмерных объектов может быть востребована в задачах разработки и дизайна изделий.

В приложениях дополненной реальности часто используются двумерные изображения, — это обусловлено необходимостью использования таких приложений на мобильных устройствах с малыми вычислительными ресурсами и объемом памяти, однако в некоторых задачах может потребоваться использование трехмерных представлений — например в приложениях дополненной реальности для медицины, обеспечивающих качественную визуализацию исследуемых областей, выполнение сегментации, детектирования каких-либо элементов и позволяющих взаимодействовать с трехмерной моделью. Возможно применение таких технологий и на производстве.

В области автономного вождения методы глубокого обучения применяются в таких задачах анализа сцены, как обнаружение объектов, сегментация сцены.

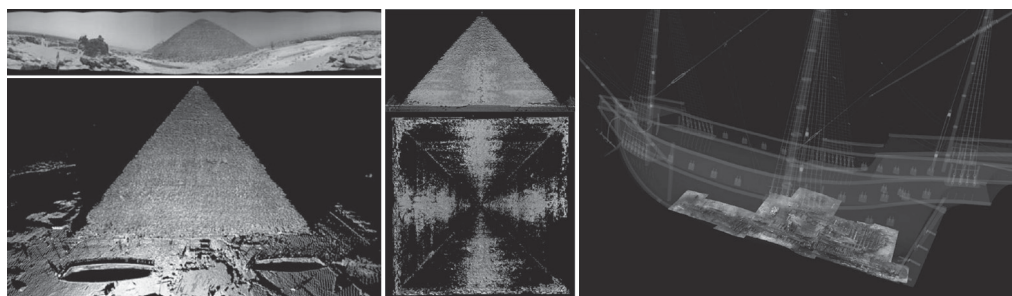


Рис. 2. Слева: Скан восточной стороны пирамиды Хеопса, сверху показана двумерная панорама [4]. По центру: Обнаруженная аномалия, сверху показана восточная сторона пирамиды с цветовым (и яркостным) кодированием отклонения поверхности ската от плоскости, снизу показано расположение аномалий на виде сверху [4]. Справа: Положение трехмерной модели исследованной на раскопках части дна судна XVI в. относительно модели судна целиком [5].

В области мобильных и сервисных роботов, сервисных систем может быть множество разнообразных приложений трехмерного распознавания, включающих всевозможные задачи: сегментация сцены, сегментация частей объектов, обнаружение и локализация объектов и ключевых точек, классификация, оценка позы и т.д. В данной области и задачи, и окружение могут быть очень вариативны, также часто предполагается взаимодействие с человеком, поэтому вариант использования именно трехмерного представления сцены или объектов может быть наиболее выгоден, поскольку дает максимально полную информацию о сцене.

3. Задачи и методы трехмерного распознавания

Главной проблемой компьютерного зрения является проблема понимания машиной окружающего мира. Человек, даже смотря не на какую-либо реальную сцену, а на ее изображение, представляет ее в трехмерном виде, оценивает положение объектов друг относительно друга, пространство, доступное для действий и т.д. При этом ряд исследований показывает, что люди воспринимают трехмерную сцену не в виде абсолютной карты глубины, а по относительному расположению поверхностей [6]. Предполагается, что человек строит одну или несколько трехмерных моделей сцены и использует их для различных задач: оценки расстояния, поиска предметов, построения маршрута. Также большую роль играет понимание контекста. Ирвинг Бидерман, ученый в области нейронаук и зрения, сформулировал пять ограничений, которым должна следовать хорошо организованная сцена: ограничение опоры, взаимного позиционирования, вероятности появления, положения и размера. Эксперименты показали, что люди в среднем дольше ищут предмет на изображении сцены, если он нарушает хотя бы одно из ограничений Бидермана [7]. Таким образом, люди распознают предметы в привычном контексте быстрее.

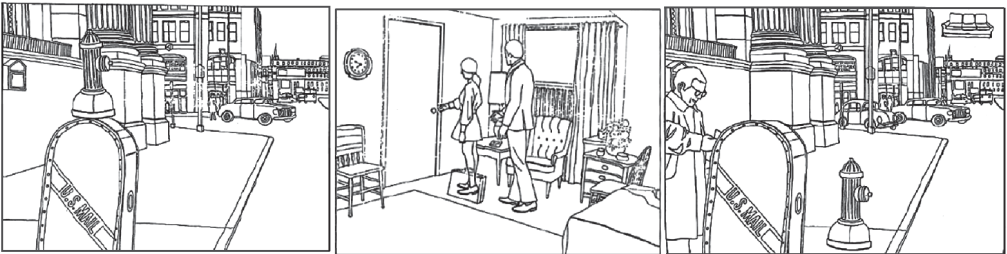


Рис. 3. Примеры нарушений ограничений Бидермана. Слева: Пожарный гидрант нарушает ограничения положения. По центру: Нарушение взаимного расположения (портфель и ноги). Справа: Тройное нарушение: диван нарушает ограничения опоры, вероятности появления и размера. Изображения взяты из [7].

Наш жизненный опыт, все наши имеющиеся знания о мире помогают в распознавании. Мы знаем, что трава растет из почвы и бывает в щелях между плиток, но она не может расти из цельного цементного пола. Мы понимаем, что такое отражение в зеркалах или других поверхностях и картины на стенах. Мы знаем, что обычно маленькие дети не гуляют без сопровождения взрослых.

Пока неясно, как передать машине весь этот опыт. Однако на данном этапе развития компьютерного зрения идея использовать максимально полное, т.е. трехмерное визуальное представление сцены, кажется целесообразной. Помимо решения некоторых прикладных задач, актуальных сейчас или в ближайшем будущем, развитие трехмерных методов распознавания может оказаться необходимым шагом к решению проблемы понимания машиной окружающего мира.

К наиболее популярным задачам трехмерного распознавания на данный момент относятся:

- выделение ключевых точек и оценка положения (позиции, ракурса), нахождение соответствия между трехмерными объектами;
- классификация объектов и форм, экстракция признаков;
- генерация трехмерных объектов, форм (включая восстановление участков);
- распознавание и генерация лиц и тел людей, животных;
- обнаружение объектов;
- сегментация объектов или составных частей объектов.

Тем не менее при рассмотрении методов глубокого обучения сложно провести четкое разделение по задачам. В большинстве случаев ключевым элементом для решения любой задачи является нейронная сеть-экстрактор признаков, которая является основой модели и к которой добавляются различные модули и ветви, использующие полученные признаки для решения целевых задач — классификации, сегментации и других. Поэтому целесообразнее рассматривать область трехмерного распознавания, не привязываясь к задачам.

3.1. Методы глубокого обучения в задачах трехмерного распознавания

В публикации [8] рассматриваются методы глубокого обучения для распознавания облаков точек с 2015 по 2020 г. Авторы выделили несколько подходов и привели примеры популярных алгоритмов (рис. 4).

Авторы выделяют три свойства, присущих облакам точек: нерегулярность (неравномерное распределение плотностей точек в объеме облака), неструктурированность (отсутствие какой-либо сетки, на которой лежат точки), неупорядоченность (точки хранятся в каком-либо списке, и порядок их расположения в этом списке не имеет значения). Несколько лет назад эти свойства не позволяли использовать методы глубокого обучения и особенно — сверточные нейронные сети, предназначенные для упорядоченных, регулярных и структурированных данных, с облаками точек, поэтому в ранних исследованиях облака точек конвертировали в регулярные и структурированные

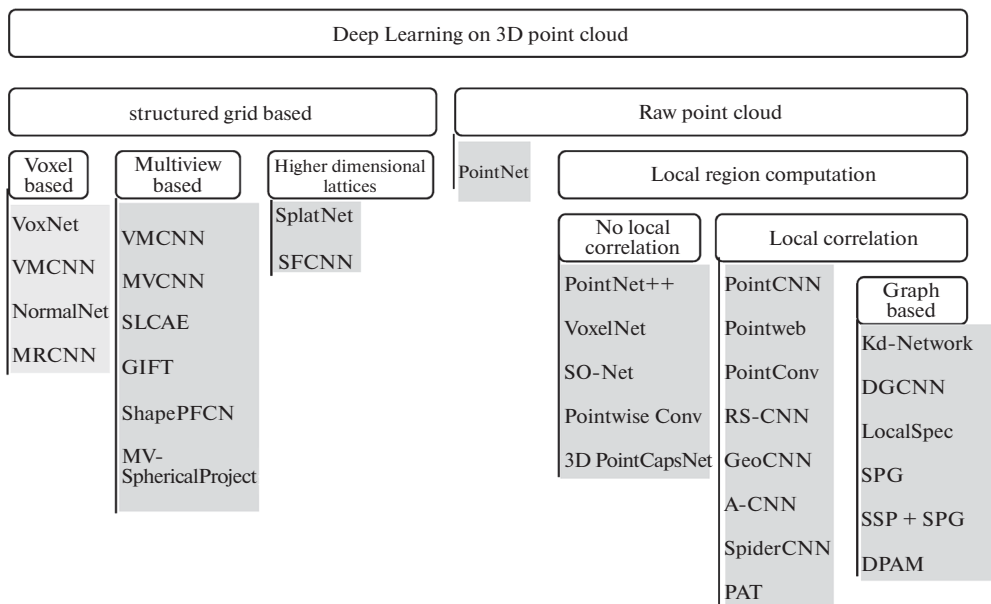


Рис. 4. В публикации [8] методы сначала разделяются на группы по типу данных (трехмерных представлений), для которых они предназначены, затем авторы выделяют более узкие группы методов на основе подходов, которые они используют.

представления. Популярны следующие подходы: вокселизация, набор видов с различных ракурсов, решетка более высокой размерности.

Вокселизация предполагает конвертирование облака точек в воксельную сетку фиксированного размера. В таком случае можно построить нейронную сеть, использующую трехмерные сверточные фильтры. Большое число публикаций [9–14] использует данный подход, выполняя вокселизацию на этапе предобработки. 3D ShapeNets [14] формирует воксельную сетку размером $30 \times 30 \times 30$ вокселей; воксель имеет значение 1, если является частью объекта, и 0, если находится снаружи объекта. Таким образом, объект можно представить бинарным тензором. Однако проблемой воксельных представлений является разреженность — когда объект занимает лишь малую часть воксельной сетки, большая часть вычислений производится над пустыми областями, что крайне неэффективно. Более поздние публикации [15, 16] предлагают решение этой проблемы путем использования октодеревьев — часто используемой для разделения трехмерного пространства древовидной структуры данных, в которой каждый воксель, являющийся кубом, может содержать ровно восемь потомков-вокселей меньшего размера. Таким образом, области, требующие более детальной обработки, могут быть помещены в более мелкие ячейки-воксели.

Использование вместо облака точек набора изображений, полученных с разных ракурсов, имеет ряд преимуществ: такой подход позволяет использовать все техники, применяемые для распознавания двумерных изображе-

ний, а сами изображения не имеют артефактов (разрешение воксельной сетки, как правило, довольно мало, что приводит к искажению оригинальной формы объекта). Первой публикацией в этом направлении является MultiviewCNN [17], в которой предлагалось получать признаковое описание для каждого изображения объекта с использованием сверточной сети, затем агрегировать все признаковые описания операцией пулинга и применять еще одну сверточную нейронную сеть для получения итогового вектора признаков для классификации. Ряд работ продолжил направление [11, 18–23], используя все достижения глубокого обучения в обработке изображений, однако на данный момент направление теряет актуальность в связи с развитием методов, способных работать с трехмерными представлениями напрямую. Тем не менее в 2020 г. была представлена работа [24], предлагающая алгоритм сегментации полигональных сеток (мешей), основанный на мультиракурсном подходе, для задачи семантической сегментации. Примечательно, что авторы предлагают использовать не оригинальные кадры, полученные при фотографировании объекта или сцены с различных ракурсов, но синтетические изображения с повышенным полем зрения, рендеры, полученные из построенной трехмерной сцены. Авторы утверждают, что представленный алгоритм, принимая на вход приблизительно 12 синтетических изображений, достигает точности сегментации выше, чем другие существующие мультиракурсные методы при использовании более тысячи изображений. При этом виртуальные ракурсы, используемые при получении таких изображений, могут быть невозможными, например из-за стен при скрытии этих стен.

Идея подхода с решеткой более высокой размерности заключается в том, чтобы преобразовать облако точек к структурированному представлению, устойчивому к перестановкам, например к перестановочному многограннику. Нейронная сеть SPLATNet (Sparse Lattice Networks for Point Cloud Processing) [25] принимает на вход облако точек и предсказывает класс для каждой точки, т.е. выполняет семантическую сегментацию. Сеть строится на основе BCL (bilateral convolutional layer) — билатеральных сверточных слоев. Принцип BCL заключается в преобразовании входных признаков, соответствующих точкам, к перестановочному многограннику и последующему выполнению операции свертки с ядром, также являющимся перестановочным многогранником. Метод SFCNN [26] адаптивно проецирует входное облако точек на дискретизированную сферу, после чего применяет ряд сверточных операций для извлечения локальных и глобальных признаков, которые затем агрегируются для получения итогового признакового представления. Сеть может использоваться для классификации и сегментации.

В 2017 г. была представлена первая нейронная сеть для распознавания облака точек — PointNet [27], работающая с облаками точек без какой-либо их предобработки. Авторы выделяют три наиболее важных подхода, реализованных в архитектуре PointNet:

1. Симметрическая функция для неупорядоченного входа, обеспечивающая инвариантность к перестановкам во входном облаке точек. Функция реализована в виде многослойного перцептрона со слоем подвыборки по максимальному значению (max pooling);

2. Агрегация локальной и глобальной информации. В ветви сегментации авторы совмещают выходной вектор глобальных признаков с набором векторов признаков для каждой точки облака и получают выходной вектор принадлежности для каждой точки, применяя набор многослойных перцептронов, что и является семантической сегментацией. Такой подход позволил авторам успешно применять PointNet для задач сегментации объектов в сцене и сегментации частей объектов;
3. Сеть выравнивания, обеспечивающая инвариантность к геометрическим трансформациям облака путем приведения облака точек к канонической форме перед извлечением признаков. Авторы использовали малую нейронную сеть T-Net, которая предсказывает аффинное преобразование, необходимое для приведения облака точек к стандартной форме. Сеть T-Net встроена в архитектуру PointNet, таким образом, предсказанная трансформация сразу же применяется к облаку точек. Такой же подход использован для приведения к канонической форме не только облака точек, но и признаков, полученных из разных облаков точек.

Однако в архитектуру PointNet не была заложена возможность находить локальные структуры, учитывать локальные зависимости между точками. Дальнейшие исследования были направлены на решение этой проблемы.

Метод PointNet++ [28] группирует локальные регионы и применяет на таких группах PointNet, повторяя эту процедуру иерархическим образом. Нейронная сеть VoxelNet [29] для обнаружения объектов принимает на вход облако точек и преобразует его в воксельную сетку, т.е. точки группируются по вокселям. Для каждой такой группы-вокселя вычисляется признаковое представление, которое затем подается на вход сети предложения регионов для получения обнаружений. В SO-Net [30] для уменьшения размерности начального облака точек строится самоорганизующаяся карта Кохонена с заданным числом узлов, генерирующая узловые точки-центроиды. Затем для каждого такого центроида выбирается заданное число точек-соседей, которые подаются на вход сети-энкодера, на выходе которой получается глобальный вектор признаков для облака точек. В публикации [31] предлагается точечная свертка (pointwise convolution). Для каждой точки облака определяются соседние точки, попадающие в трехмерное сверточное окно, которые, подобно окружающим пикселям, подвергаются операции свертки вместе с рассматриваемой точкой. Так строится полносверточная нейронная сеть, имеющая на выходе вектор признаков для каждой точки. Работа [32] представляет 3DPointCapsNet, трехмерный автоэнкодер, учитывающий геометрические отношения между частями объектов, что позволяет улучшить способности сети к обучению и генерализации. Скрытое пространство, строящееся с применением капсульной динамической маршрутизации [33], параметризуется так называемыми скрытыми капсулами (latent capsules) — набором признаков заданного размера.

Часть работ не только группируют точки, но и исследуют корреляцию между точками в локальных группах. Интуитивное допущение заключается в

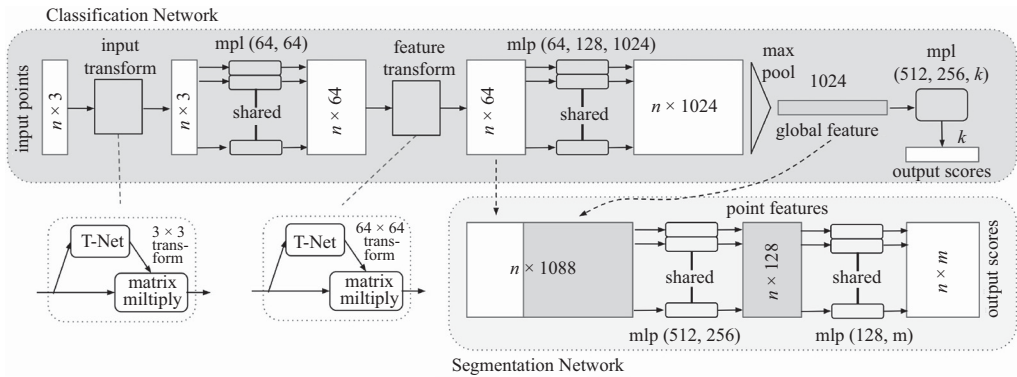


Рис. 5. Архитектура нейронной сети PointNet [27]. На вход подаются n точек, представленных тремя координатами. В верхнем ряду показана основная сеть-классификатор, а в нижнем ряду показана дополнительная ветвь для семантической сегментации. Обозначение *mpl* подразумевает многослойный перцептрон (multi-layer perceptron), в скобках указаны размеры слоев.

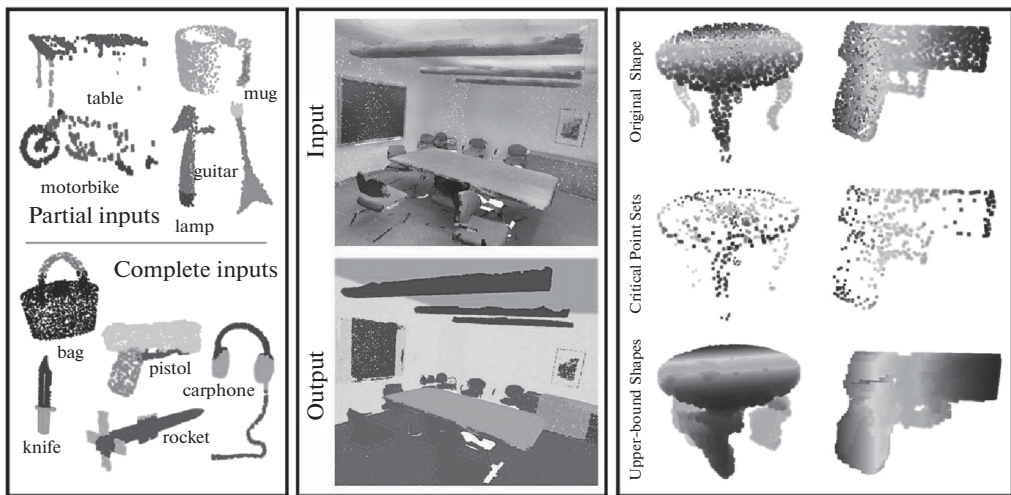


Рис. 6. Результаты PointNet. Слева: Примеры классификации цельных объектов и их частей. По центру: Пример семантической сегментации сцены. Справа: Для каждой формы объекта (верхний ряд) PointNet определяет набор ключевых точек (средний ряд) так, что любое облако точек, находящееся между набором ключевых точек и предельной формой (нижний ряд), имеет один и тот же вектор глобальных признаков.

том, что точки не могут существовать в изоляции, их скопления отображают какие-либо формы, которые можно анализировать. В [34] авторы замечают, что операция свертки, выполненная напрямую над группой точек, не учитывает взаиморасположения точек, формы их скученности, однако зависима от порядка точек, который в облаках точек не имеет никакого значения. В ка-

честве решения этой проблемы предлагается использовать χ -преобразование над координатами точек, реализованное в виде многослойного перцептрона и позволяющее одновременно взвесить и выполнить перестановку входных точек. Объединяя χ -преобразование с последующей операцией свертки, получается χ -Conv, базовый блок предлагаемой нейронной сети PointCNN. Работа [35] представляет нейронную сеть Pointweb. Здесь предлагается новый обучаемый блок AFA (Adaptive Feature Adjustment), получающий всевозможные пары точек в локальной группе и вычисляющий влияние точек друг на друга, формируя векторы признаков, агрегирующих свойства всей группы точек.

Предложено большое количество работ, исследующих различные варианты сверточных операций, учитывающих взаиморасположение точек и формы, ими образуемые. В работе [36] предлагается вариант сверточной операции для облака точек, PointConv. В отличие от обычных сверток с дискретными ядрами (весами) PointConv предлагает обучаемую непрерывную функцию, позволяющую получить вес для точки по ее положению. Кроме этого, предполагается взвешивать точки оценками их обратных плотностей — таким образом, вклад отдельных точек, составляющих плотную скученность, не будет слишком велик. В нейронных сетях RS-CNN [37] и Geo-CNN [38] выполняется выборка равномерно распределенных точек из облака, каждая из которых становится центроидом сферической области-соседства, однако рассматриваются не сами точки соседства, а их положения относительно центроида (представленные в виде векторов). Особенностью сверточной операции GeoConv, из которой строится Geo-CNN, является проекция этих векторов на оси, т.е. формирование ортогонального базиса. Признаки вычисляются по каждому направлению, после чего агрегируются с учетом углов между вектором и осями в единый итоговый вектор признаков точки локального региона. Затем признаки точек локального региона агрегируются с весами, зависящими от положения точек относительно центроида и радиуса региона. Радиус локальных регионов растет по ходу сети. В [39] предлагаются кольцевые свертки, имеющие сферическую рецептивную область с концентрической сферической полостью внутри (при проекции на плоскость принимающие форму кольца). Авторы замечают, что в современных нейросетевых архитектурах для устойчивости к изменению масштаба обычно объединяются разномасштабные признаки, что приводит к перекрытию рецептивных полей и многократному попаданию признаков некоторых точек в расчет итогового вектора признаков. Кольцевая свертка может состоять из нескольких концентрических колец — таким образом, можно получить разномасштабные признаки, не переиспользуя одни и те же точки. Нейронная сеть SpiderCNN [40] для классификации и сегментации построена на основе сверточных слоев SpiderConv. SpiderConv использует параметризованное семейство фильтров для выполнения операции свертки. Фильтры являются композицией двух функций: простой ступенчатой функции для захвата грубой геометрии и разложения Тейлора 3-го порядка для захвата более сложных структур.

Работа [41] представляет DPRNet, нейронную сеть для классификации и сегментации, использующую разреженные свертки и остаточные соединения, а вместо нормализации батча использует самонормализующуюся функцию активации SELU (Scaled Exponential Linear Units) [42].

Работа [43] предлагает PATs (Point Attention Transformers) — нейронную сеть-трансформер, предназначенную для работы с облаками точек. Ключевыми операциями PAT являются Group Shuffle Attention (GSA) и Gumbel Subset Sampling (GSS). GSA является параметрически эффективной операцией самовнимания для тренировки понимания отношений между точками. GSS служит альтернативой операции сэмплирования по дальним точкам (Farthest Point Sampling, FPS) и имеет несколько преимуществ: инвариантность к перестановкам, дифференцируемость, лучшая устойчивость к выбросам. PAT предполагает возможность классификации и сегментации. В обоих случаях результирующий вектор или набор векторов принадлежности к классам вычисляется с использованием многослойных перцептронов. В [44] также предлагаются операции сэмплирования с обучаемыми параметрами. Авторы замечают, что несмотря на качественные результаты, получаемые при использовании иерархических архитектур, сэмплирование и группировка точек в таких методах производится в исходном Евклидовом пространстве фиксированным способом, что может приводить к снижению робастности. Для решения этой проблемы в работе предлагается модуль DPAM (Dynamic points agglomeration module). Модуль строится на основе набора многослойных перцептронов и формирует обучаемую матрицу агломерации (agglomeration matrix), которая при умножении на облако точек или набор признаков снижает их размерность — т.е. является комплексом операций группировки и подвыборки (цулинга).

В [45] предложен метод PointHop для классификации облака точек, обладающий низкой вычислительной сложностью и обучающийся без учителя. Метод представляет собой каскад одноименных модулей. Признаки со всех модулей агрегируются в единый вектор для классификации методом опорных векторов и случайного леса. Модуль PointHop формирует для каждой точки облака локальный регион из k ближайших соседей, затем разбивает регион на восемь областей-октантов, вычисляет признаки и конкатенирует их с применением преобразования Saab (вариации метода главных компонент) [46]. Работа получила продолжение в методе PointHop++ [47]. Авторы оптимизировали модель, снизив число ее параметров, приведя ее к древовидной форме и реализовав выбор отличительных признаков по энтропийному критерию. Наконец, в 2021 г. была представлена модель R-PointHop [48] для нахождения жесткой трансформации (поворот и смещение) между двумя облаками точек, обеспечивающей оптимальное прилегание. Все методы группы PointHop не являются методами глубокого обучения и используют обучение без учителя, что дает возможность использовать их на мобильных и мало-мощных вычислительных платформах. Еще одним методом, использующим обучение без учителя, является метод, представленный в [49]. Предлагаемая нейронная сеть извлекает из облака ключевые (структурные) точки и позво-

ляет решать задачи сопоставления форм облаков точек и переноса сегментационной разметки и потенциально — задачи реконструкции и дополнения формы.

Подход, основанный на графах, рассматривает облако точек как граф, узлами которого являются точки. Ребра графа отображают корреляцию между точками. Работа [50] 2017 г. представляет нейронную сеть *Kd-network*, представляющую облако точек в виде *k-d-дерева* (*k*-мерного дерева), листьями которого являются точки облака (их координаты), а векторное представление каждого нелистового узла рассчитывается как нелинейная функция от аффинного преобразования с обучаемыми параметрами, примененного к двум узлам-соседям рассматриваемого нелистового узла. После обработки корневого узла получается вектор признаков всего облака точек. Работа [51] представляет модуль *EdgeConv*, генерирующий вектор признаков для точки с учетом ее соседства, представляющего направленный граф. В модуле попарно вычисляются ребра между точкой-центроидом и ее соседями с применением полносвязного слоя, называемые ребровыми признаками (*edge features*). Выход модуля *EdgeConv* получается путем применения операции пулинга к ребровым признакам. Особенностью такого подхода является то, что ребровые признаки зависят не только от взаимного расположения точек, но и от ребровых признаков, полученных с предыдущего слоя *EdgeConv*. Таким образом, используемые графы являются динамическими. В работе [52] предлагаются спектральные свертки, собственная операция подвыборки (пулинга) и нейросетевая архитектура для классификации и сегментации, построенная на их основе. В модуле спектральной свертки точки соседства являются узлами графа, а длины ребер вычисляются как расстояния между точками в пространстве признаков и формируют матрицу смежности, по которой вычисляется спектр графа. Далее выполняются преобразование Фурье для графа, спектральная фильтрация и, наконец, обратное преобразование Фурье. Модуль рекурсивного кластерного пулинга итеративно выполняет спектральную кластеризацию и пулинг таким образом, чтобы каждый кластер участвовал в формировании итогового набора векторов признаков. Авторы предполагают, что, в отличие от популярного максимального пулинга, такой способ позволит лучше сохранить полезную информацию. Работа [53] предлагает модель *Point2Node* для семантической сегментации, ключевой особенностью которой является возможность использовать корреляцию не только точек-соседей, но и любых точек модели. Граф строится из многомерных узлов, получаемых из входных точек облака с использованием свертки *X-Conv*, предложенных в [34]. Далее модуль динамической корреляции узлов *DNC* (*Dynamic Node Correlation*) представляет каждый узел в трех вариантах — с учетом автокорреляции (между каналами вектора признаков узла), локальной (между узлами локальной группы) и нелокальной корреляции (между взаимодальними узлами). Используется адаптивный обучаемый модуль агрегации разноуровневых признаков. Работа [54] предлагает разделять облако точек на так называемые суперточки, *superpoints* — простые семантически однородные геометрические формы. Для каждой суперточки формируется вектор признаков с использованием модели *PointNet* [27], и полученные векторы

формируют граф Superpoint Graph (SPG). Такой подход позволяет значительно снижать размер входного облака точек, сохраняя важные признаки, и может быть использован на очень крупных облаках точек. В последующей работе [55] рассматривается случай, при котором геометрически гомогенный регион не является семантически гомогенным, и предлагается дополнительно сегментировать суперточки на более мелкие регионы. В работе [56] предлагается новая сверточная операция Graph Attention Convolution (GAC), динамически определяющая вес каждой точки в локальной группе. Это позволяет фокусироваться на более важных элементах локальной группы и исключать элементы, относящиеся к другому семантическому классу. Нейронная сеть 3D Graph Convolution Network (3D-GCN), предложенная в [57], использует сверточные модули, в которых ядро, являющееся графом и представляющее набор точек локальной группы, имеет не только настраиваемые в процессе обучения веса, но и форму, т.е. положение самих точек. Косинусное расстояние, используемое для определения схожести входных данных и сверточного ядра, не зависит от длины векторов и обеспечивает инвариантность к масштабу.

В 2021 г. была представлена работа [58], в которой предлагалось группировать точки не в локальные группы, а по кривым, т.е. цепочкам точек, складывающихся в ломаную линию. При использовании локальных групп схожее распределение точек в группах приводит к получению схожих признаков, что не позволяет разделять регионы. Кривые же хорошо отображают геометрическую форму, позволяя строить более различимые признаки. Модуль построения кривой (криволинейной агрегации) является обучаемым — каждая следующая точка выбирается как точка из локальной группы, обладающая наибольшей численной оценкой, полученной с использованием многослойных перцептронов. Нейронная сеть CurveNet, использующая такой подход, позволяет выполнить классификацию и сегментацию облака точек.

В 2019 г. был представлен первый нейросетевой алгоритм, работающий с треугольными полигональными сетками (polygon mesh, меш) — MeshNet [59]. Входная полигональная сетка рассматривалась как набор элементов-граней. Каждая грань описывалась следующим набором параметров: координаты центра грани, векторы вершин грани, проведенные из центра грани, нормальный вектор единичной длины, индексы сопряженных граней. Параметры обрабатываются в блоках structural descriptor (структурный дескриптор) и spatial descriptor (пространственный дескриптор), получающих первичные структурные и пространственные признаки соответственно. Признаки затем проходят через сверточные блоки для полигональных сеток (блоки mesh convolution). Наконец, путем агрегирования и подвыборки признаков получается итоговый глобальный вектор признаков, который может быть использован для различных задач, в том числе классификации.

В [60] предлагается трансформировать меш в трехмерную сетку, содержащую в ячейках векторы кривизны, и подавать эту сетку на вход представленной сверточной нейронной сети CurveNet (не путать с CurveNet из [58]). LaplacianNet [61] (the Laplacian Encoding and Pooling Network) принимает в

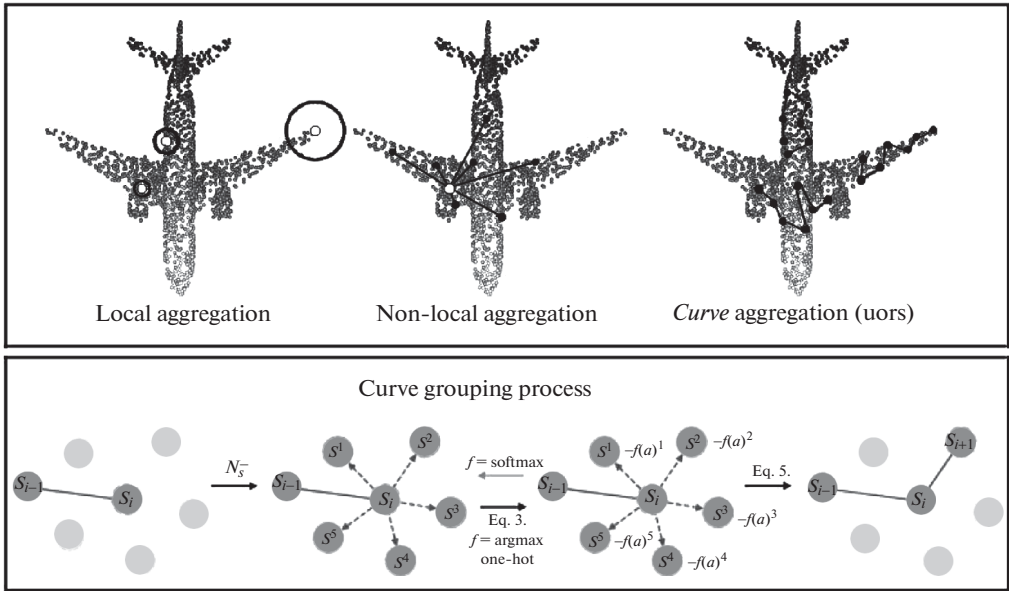


Рис. 7. Сверху: Способы группировки: локальные группы, нелокальная агрегация и криволинейная агрегация. Внизу: Иллюстрация процесса построения кривой в работе [58].

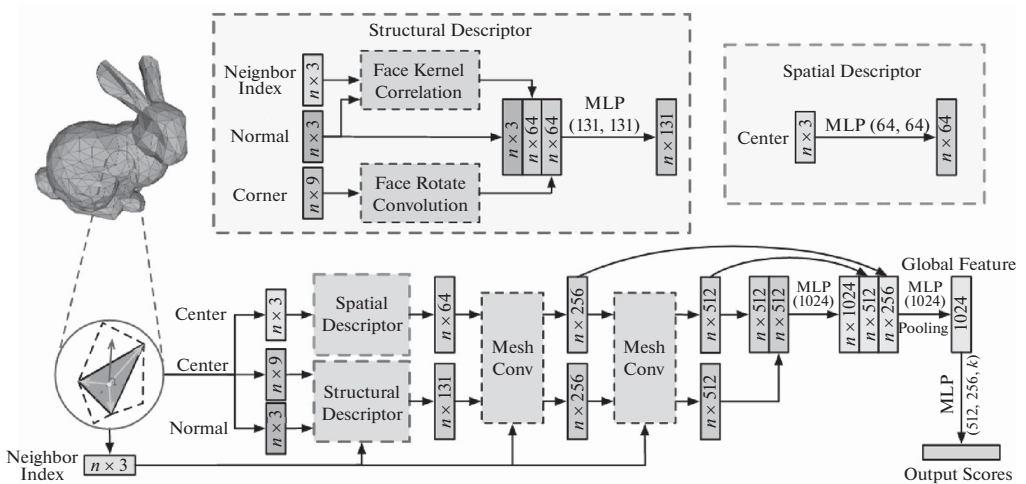


Рис. 8. Архитектура MeshNet [59].

качестве входных данных вершины и нормали полигональной сетки, а также собственные векторы и кластеры, полученные методом спектральной кластеризации. Нейронная сеть состоит из двух блоков МРВ (Mesh Pooling Block) для анализа данных в различном разрешении и формирования признаков для каждой вершины, после чего эти признаки используются в сети сегментации

или классификации в зависимости от задачи. Начальное формирование кластеров (аналог суперпикселей для изображений) в качестве локальных групп и варьирование размера кластеров от мелкого (детального) до крупного (грубого) позволяет обрабатывать полигональные сетки иерархически. Нейронная сеть MeshWalker [62] принимает на вход цепочку-путь, составленную случайным образом из вершин полигональной сетки объемом 40% от их общего числа. Сеть содержит рекуррентные слои для агрегации информации по всему пути. Преимуществом MeshWalker является возможность получения качественного результата даже при малом объеме обучающих данных. Работа [63] предлагает сверточный слой LRFConv (Local reference Frame Convolution) с пропускным соединением. В LRFConv для каждой обрабатываемой точки трехмерной модели, представленной ее координатами и вектором нормали, определяется группа точек-соседей по геодезическим дистанциям. Координаты соседей пересчитываются относительно точки-центроида, вместе с дистанциями поступают на вход многослойным перцептронам для приведения к требуемой размерности, конкатенируются с признаками предыдущего слоя и поступают на вход непрерывной свертки.

Авторы работы [64] замечают, что современные подходы к сегментации трехмерных форм (объектов) зависят от качества и объема размеченных наборов обучающих данных, и представляют алгоритм ROSS, требующий лишь один размеченный пример для обучения. ROSS обучается переносить разметку с примера на схожие входные данные.

Также происходит развитие и в направлении генеративных методов. TM-Net [65] является сетью для генерации текстурированных полигональных сеток и состоит из трех основных частей: TextureVAE, PixelSNAIL и PartVAE. Каждая из частей строится на основе двух вариационных автоэнкодеров. PartVAE генерирует геометрию объекта, TextureVAE — текстуру, PixelSNAIL обеспечивает соответствие результирующей текстуры сгенерированной геометрии.

4. Заключение

Несмотря на значительное развитие методов глубокого обучения, в том числе для двумерных изображений, в области трехмерного распознавания их применение было затруднено до недавнего времени. Облака точек, достаточно легко получаемые с различных датчиков или набора изображений объекта или сцены, используются во многих системах технического зрения, однако их нерегулярность, неструктурированность и неупорядоченность вносили сложности в разработку нейросетевых решений. Ранние методы предполагали конвертацию облаков точек в регулярные и структурированные представления: воксельные сетки, решетки более высокой размерности, набор изображений с различных ракурсов.

В 2017 г. была представлена PointNet — первая нейронная сеть, работающая с облаками точек напрямую и решающая задачи классификации и сегментации облака точек. Однако нейронная сеть не учитывала локальные

зависимости между точками, и дальнейшие исследования области были направлены на решение этой проблемы. В настоящий момент имеется большое количество подходов и техник, с успехом примененных в области трехмерного распознавания облаков точек: рассмотрение облака точек как графа, супер-точки (superpoints), сверточные ядра с настраиваемыми весами и формой, использование локального ортогонального базиса, обучаемые непрерывные функции в качестве сверточных ядер и т.д.

С развитием методов глубокого обучения для облаков точек методы, использующие воксельные представления, стали менее популярны. Причиной этого служат высокие требования к вычислительным ресурсам: из-за разреженности воксельной сетки большая часть операций происходит в пустых областях, в результате чего разрешение воксельной сетки сильно ограничено и не позволяет качественно представить крупные сцены или объекты со множеством мелких деталей. Однако воксельные представления остаются популярными в области медицинских задач, связанных с распознаванием радиологических снимков, из-за особенностей сканирования.

Методы, использующие наборы изображений с различных ракурсов для трехмерного распознавания (как правило, сегментации), также теряют популярность и, как правило, проигрывают в точности методам, использующим другие типы представлений. Тем не менее недавняя работа [24] показала, что данное направление имеет перспективы, однако предложенный алгоритм предполагал использование не оригинальных изображений, а рендеров, выполненных по построенной трехмерной модели.

В 2019 г. были впервые представлены нейросетевые алгоритмы, работающие с полигональными сетками (мешами). Преимуществами мешей являются наличие информации о поверхности и связность элементов. Данное направление позволяет как использовать достижения в распознавании облаков точек, рассматривая меш как набор вершин сетки с дополнительной информацией: векторами нормалей, сопряженными вершинами и т.д., так и применить новые подходы: формировать сетку векторов кривизны или цепочку-путь по мешу в качестве входных данных, использовать геодезические дистанции и др. На данный момент не удалось заметить какие-либо устоявшиеся практики в этом направлении.

Таким образом, современное состояние области трехмерного распознавания позволяет работать с любыми популярными трехмерными представлениями — облаком точек, воксельной сеткой, полигональной сеткой, а также с набором мультиракурсных изображений, однако наибольший прогресс и активность прослеживаются в исследованиях, касающихся облаков точек.

СПИСОК ЛИТЕРАТУРЫ

1. COLMAP Project Page on Github.io – Main Page. [Электронный ресурс] URL: <https://colmap.github.io> (дата обр. 03.08.2021).
2. COLMAP Project Page on Github.io – Datasets. [Электронный ресурс] URL: <https://colmap.github.io/datasets.html> (дата обр. 03.08.2021).

3. *Pérez P., Iván R.* Blurring the Boundaries Between Real and Artificial in Architecture and Urban Design through the Use Artificial Intelligence: PhD thesis. Universidade da Coruna, 2017. P. 1–300.
4. *Neubauer W., Doneus M., Studnicka N., Riegl J.* Combined High Resolution Laser Scanning and Photogrammetrical Documentation of the Pyramids at Giza // CIPA XX Int. Sympos. Citeseer. 2005. P. 470–475.
5. *McCarthy J.K., Benjamin J., Winton T., van Duivenvoorde W.* 3D Recording and Interpretation for Maritime Archaeology. Springer Nature, 2019.
6. *Hoiem D., Savarese S.* Representations and Techniques for 3D Object Recognition and Scene Interpretation // Synthesis Lectures on Artificial Intelligence and Machine Learning. 2011. V. 5. No. 5. P. 1–169.
7. *Biederman I.* On the Semantics of a Glance at a Scene // Perceptual Organization. 1981. V. 213. P. 253.
8. *Bello S.A., Yu S., Wang C., Adam J.M., Li J.* Deep Learning on 3D Point Clouds // Remote Sensing. 2020. V. 12. No. 11. P. 1729.
9. *Maturana D., Scherer S.* 3D Convolutional Neural Networks for Landing Zone Detection from Lidar // IEEE ICRA. IEEE. 2015. P. 3471–3478.
10. *Maturana D., Scherer S.* Voxnet: a 3D Convolutional Neural Network for Real-Time Object Recognition // IEEE/RSJ IROS. IEEE. 2015. P. 922–928.
11. *Qi C.R., Su H., Nießner M., Dai A., Yan M., Guibas L.J.* Volumetric and Multi-View CNNs for Object Classification on 3D Data // Proc. CVPR. 2016. P. 5648–5656.
12. *Wang C., Cheng M., Sohel F., Bennamoun M., Li J.* NormalNet: a Voxel-Based CNN for 3D Object Classification and Retrieval // Neurocomputing. 2019. V. 323. P. 139–147.
13. *Ghadai S., Lee X., Balu A., Sarkar S., Krishnamurthy A.* Multi-Resolution 3D Convolutional Neural Networks for Object Recognition // arXiv preprint: 1805.12254. 2018.
14. *Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X., Xiao J.* 3D ShapeNets: a Deep Representation for Volumetric Shapes // Proc. CVPR. 2015. V. 1912–1920.
15. *Riegler G., Osman Ulusoy A., Geiger A.* Octnet: Learning Deep 3D Representations at High Resolutions // Proc. CVPR. 2017. P. 3577–3586.
16. *Tatarchenko M., Dosovitskiy A., Brox T.* Octree Generating Networks: Efficient Convolutional Architectures for High-Resolution 3D Outputs // Proc. IEEE Int. Conf. on Computer Vision. 2017. P. 2088–2096.
17. *Su Hang, Maji S., Kalogerakis E., Learned-Miller E.* Multi-View Convolutional Neural Networks for 3D Shape Recognition // Proc. IEEE Int. Conf. on Computer Vision. 2015. P. 945–953.
18. *Leng B., Guo S., Zhang X., Xiong Z.* 3D Object Retrieval with Stacked Local Convolutional Autoencoder // Signal Processing. 2015. V. 112. P. 119–128.
19. *Bai S., Bai X., Zhou Z., Zhang Z., Jan Latecki L.* Gift: a Real-Time and Scalable 3D Shape Search Engine // Proc. CVPR. 2016. P. 5023–5032.
20. *Kalogerakis E., Averkiou M., Maji S., Chaudhuri S.* 3D Shape Segmentation with Projective Convolutional Networks // Proc. CVPR. 2017. P. 3779–3788.
21. *Cao Z., Huang Q., Karthik R.* 3D Object Classification via Spherical Projections // 2017 Int. Conf. on 3D Vision (3DV). IEEE. 2017. P. 566–574.

22. Zhang L., Sun J., Zheng Q. 3D Point Cloud Recognition Based on a Multi-View Convolutional Neural Network // *Sensors*. 2018. V. 18. No. 11. P. 3681.
23. Kanezaki A., Matsushita Y., Nishida Y. RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints // *Proc. CVPR*. 2018. P. 5010–5019.
24. Kundu A., Yin X., Fathi A., Ross D., Brewington B., Funkhouser T., Pantofaru C. Virtual Multi-View Fusion for 3D Semantic Segmentation // *Eur. Conf. on Computer Vision (ECCV)*. Springer. 2020. P. 518–535.
25. Su H., Jampani V., Sun D., Maji S., Kalogerakis E., Yang M.-H., Kautz J. Splatnet: Sparse Lattice Networks for Point Cloud Processing // *Proc. CVPR*. 2018. P. 2530–2539.
26. Rao Y., Lu J., Zhou J. Spherical Fractal Convolutional Neural Networks for Point Cloud Recognition // *Proc. CVPR*. 2019. P. 452–460.
27. Qi C.R., Su H., Mo K., Guibas L.J. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation // *Proc. CVPR*. 2017. P. 652–660.
28. Qi C.R., Yi L., Su H., Guibas L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space // *arXiv preprint: 1706.02413*. 2017.
29. Zhou Y., Tuzel O. Voxynet: End-to-End Learning for Point Cloud Based 3D Object Detection // *Proc. CVPR*. 2018. P. 4490–4499.
30. Li J., Chen B.M., Lee G.H. So-Net: Self-Organizing Network for Point Cloud Analysis // *Proc. CVPR*. 2018. P. 9397–9406.
31. Hua B.-S., Tran M.-K., Yeung S.-K. Pointwise Convolutional Neural Networks // *Proc. CVPR*. 2018. P. 984–993.
32. Zhao Y., Birdal T., Deng H., Tombari F. 3D Point Capsule Networks // *Proc. CVPR*. 2019. P. 1009–1018.
33. Sabour S., Frosst N., Hinton G.E. Dynamic Routing Between Capsules // *arXiv preprint:1710.09829*. 2017.
34. Li Y., Bu R., Sun M., Wu W., Di X., Chen B. PointCNN: Convolution on χ -Transformed Points // *arXiv preprint:1801.07791*. 2018.
35. Zhao H., Jiang L., Fu C.-W., Jia J. Pointweb: Enhancing Local Neighborhood Features for Point Cloud Processing // *Proc. CVPR*. 2019. P. 5565–5573.
36. Wu W., Qi Z., Fuxin L. PointConv: Deep Convolutional Networks on 3D Point Clouds // *Proc. CVPR*. 2019. P. 9621–9630.
37. Liu Y., Fan B., Xiang S., Pan C. Relation-Shape Convolutional Neural Network for Point Cloud Analysis // *Proc. CVPR*. 2019. P. 8895–8904.
38. Lan S., Yu R., Yu G., Davis L.S. Modeling Local Geometric Structure of 3D Point Clouds Using Geo-CNN // *Proc. CVPR*. 2019. P. 998–1008.
39. Komarichev A., Zhong Z., Hua J. A-CNN: Annularly Convolutional Neural Networks on Point Clouds // *Proc. CVPR*. 2019. P. 7421–7430.
40. Xu Y., Fan T., Xu M., Zeng L., Qiao Y. Spidercnn: Deep Learning on Point Sets with Parameterized Convolutional Filters // *Proc. ECCV*. 2018. P. 87–102.
41. Arshad S., Shahzad M., Riaz Q., Fraz M.M. DPRNet: Deep 3D Point Based Residual Network for Semantic Segmentation and Classification of 3D Point Clouds // *IEEE Access*. 2019. V. 7. P. 68892–68904.
42. Klambauer G., Unterthiner T., Mayr A., Hochreiter S. Self-Normalizing Neural Networks // *arXiv preprint: 1706.02515*. 2017.

43. *Yang J., Zhang Q., Ni B., Li L., Liu J., Zhou M., Tian Q.* Modeling Point Clouds With Self-Attention and Gumbel Subset Sampling // Proc. CVPR. 2019. P. 3323–3332.
44. *Liu J., Ni B., Li C., Yang J., Tian Q.* Dynamic Points Agglomeration for Hierarchical Point Sets Learning // Proc. CVPR. 2019. P. 7546–7555.
45. *Zhang M., You H., Kadam P., Liu S., Kuo C.-C.J.* PointHop: an Explainable Machine Learning Method for Point Cloud Classification // IEEE Trans. Multimedia. 2020. V. 22. No. 7. P. 1744–1755.
46. *Kuo C.-C.J., Zhang M., Li S., Duan J., Chen Y.* Interpretable Convolutional Neural Networks via Feedforward Design // J. Visual Communication and Image Representation. 2019. V. 60. P. 346–359.
47. *Zhang M., Wang Y., Kadam P., Liu S., Kuo C.-C.J.* PointHop++: a Lightweight Learning Model on Point Sets for 3D Classification // IEEE Int. Conf. on Image Processing (ICIP). IEEE. 2020. P. 3319–3323.
48. *Kadam P., Zhang M., Liu S., Kuo C.-C.J.* R-PointHop: a Green, Accurate and Unsupervised Point Cloud Registration Method // arXiv preprint: 2103.08129. 2021.
49. *Chen N., Liu L., Cui Z., Chen R., Ceylan D., Tu C., Wang W.* Unsupervised Learning of Intrinsic Structural Representation Points // Proc. CVPR. 2020. P. 9121–9130.
50. *Klokov R., Lempitsky V.* Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models // Proc. IEEE Int. Conf. on Computer Vision. 2017. P. 863–872.
51. *Wang Y., Sun Y., Liu Z., Sarma S.E., Bronstein M.M. Solomon J.M.* Dynamic Graph CNN for Learning on Point Clouds // ACM Trans. Graphics (TOG). 2019. V. 38. No. 5. P. 1–12.
52. *Wang C., Samari B., Siddiqi K.* Local Spectral Graph Convolution for Point Set Feature Learning // Proc. ECCV. 2018. P. 52–66.
53. *Han W., Wen C., Wang C., Li X., Li Q.* Point2Node: Correlation Learning of Dynamic-Node for Point Cloud Feature Modeling // Proc. AAAI Conf. on Artificial Intelligence. V. 34. 2020. P. 10925–10932.
54. *Landrieu L., Simonovsky M.* Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs // Proc. CVPR. 2018. P. 4558–4567.
55. *Landrieu L., Boussaha M.* Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning // Proc. CVPR. 2019. P. 7440–7449.
56. *Wang L., Huang Y., Hou Y., Zhang S., Shan J.* Graph Attention Convolution for Point Cloud Semantic Segmentation // Proc. CVPR. 2019. P. 10296–10305.
57. *Lin Z.-H., Huang S.-Y., Wang Y.-C.F.* Convolution in the Cloud: Learning Deformable Kernels in 3D Graph Convolution Networks for Point Cloud Analysis // Proc. CVPR. 2020. P. 1800–1809.
58. *Xiang T., Zhang C., Song Y., Yu J., Cai W.* Walk in the Cloud: Learning Curves for Point Clouds Shape Analysis // arXiv preprint: 2105.01288. 2021.
59. *Feng Y., Feng Y., You H., Zhao X., Gao Y.* MeshNet: Mesh Neural Network for 3D Shape Representation // Proc. AAAI Conf. on Artificial Intelligence. V. 33. 2019. P. 8279–8286.
60. *Muzahid A., Wan W., Sohel F., Wu L., Hou L.* CurveNet: Curvature-Based Multitask Learning Deep Networks for 3D Object Recognition // IEEE/CAA J. Automatica Sinica. 2020. V. 8. No. 6. P. 1177–1187.

61. *Qiao Y.-L., Gao L., Rosin P., Lai Y.-K., Chen X.* Learning on 3D Meshes with Laplacian Encoding and Pooling // IEEE Trans. Visualization and Computer Graphics. 2020.
62. *Lahav A., Tal A.* MeshWalker: Deep Mesh Understanding by Random Walks // ACM Trans. Graphics (TOG). 2020. V. 39. No. 6. P. 1–13.
63. *Yang Z., Litany O., Birdal T., Sridhar S., Guibas L.* Continuous Geodesic Convolutions for Learning on 3D Shapes // Proc. IEEE/CVF Winter Conf. on Appl. of Computer Vision. 2021. P. 134–144.
64. *Yuan S., Fang Y.* Ross: Robust Learning of One-Shot 3D Shape Segmentation // Proc. IEEE/CVF Winter Conf. on Appl. of Computer Vision. 2020. P. 1961–1969.
65. *Gao L., Wu T., Yuan Y.-J., Lin M.-X., Lai Y.-K., Zhang H.* TM-Net: Deep Generative Networks for Textured Meshes // arXiv preprint: 2010.06217. 2020

Статья представлена к публикации членом редколлегии Д.В. Виноградовым.

Поступила в редакцию 09.08.2021

После доработки 27.12.2021

Принята к публикации 30.12.2021