Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Кафедра системного программирования

Пономарёва Наталья Александровна

Многовидовая реконструкция плотного облака точек с учётом сегментации изображений

Курсовая работа

Научный руководитель:
Пименов А. А.

Консультант:
Корчёмкин Д. А.

Санкт-Петербург
2019

Оглавление

Введение **3**

1. Постановка задачи **5**

2. Обзор **6**

2.1. Обзор базового алгоритма реконструкции плотного облака точек 6

2.2. Существующие решения: сегментация изображений . . . 9

2.2.1. Pyramid Scene Parsing Network 9

2.2.2. Rethinking Atrous Convolution for Semantic Image Segmentation 11

3. Описание решения **12**

3.1. Принятые технические решения 12

3.2. Сегментация 12

3.3. Описание предлагаемого подхода 13

3.3.1. Фотометрическая согласованность 14

3.3.2. Геометрическая согласованность 14

3.4. Тестирование 15

Заключение **16**

Список литературы **18**

Введение

Многовидовая трёхмерная реконструкция – это процесс получения формы и облика реальных объектов из набора двухмерных изображений. Разреженная реконструкция – это небольшой набор точек в пространстве, а плотная реконструкция – это значительно большее число точек, которые лучше описывают объекты в сцене. На данный момент точность, полнота и эстетическая красота плотных 3D-моделей оставляют желать лучшего, во многом из-за низкого качества и высокого количества отдельно лежащих шумовых фрагментов.

В 3D-реконструкции используются два основополагающих алгоритма – SfM (Structure-from-Motion) для восстановления разреженной модели, и MVS (Multi-View Stereo) для восстановления плотной модели. SfM алгоритм принимает входные изображения и определяет параметры камер и их взаимное расположение, а также восстанавливает разреженное трёхмерное облако точек, которое и называют разреженной реконструкцией. После этого алгоритм MVS используется для уточнения полученной реконструкции, чтобы восстановить плотное облако точек. Наибольшей проблемой всех методов трёхмерной реконструкции является поиск соответствующих пикселей на входных изображениях. Восстановление правильных соответствий является сложным процессом даже в сценах с известной геометрией и освещением. В случайных наборах входных изображений, например, переполненных объектами, важно учитывать многие факторы, такие как неоднородное разрешение и освещение, изменчивость сцены и т.д.

В алгоритме MVS для нахождения соответствий между пикселями используются геометрическая и фотометрическая согласованность, рассчитанные в некоторой окрестности каждого пикселя. Фотометрическая согласованность – это схожесть пикселей по их интенсивности, а геометрическая согласованность – это схожесть окрестностей пикселей по их геометрическим характеристикам. В рамках данной работы высказывается гипотеза, что если пиксель находится на границе некоторого объекта, то в действительности его описывает только та часть

окрестности, что принадлежит этому объекту.

Сегментация изображения – это разбиение изображения на множество покрывающих его областей (сегментов). Сегментация нужна для выделения на изображении контуров объектов. Каждому сегменту должна присвоиться метка, которая характеризует выделенную область по смыслу. Таким образом, если для каждого изображения предпосчитать сегментированное, то не трудно будет для одного пикселя вычислить только ту часть окрестности, что лежит в одном сегменте.

1. Постановка задачи

Целью данной работы является интеграция предпосчитанной сегментации изображений в алгоритм реконструкции плотного облака точек. Для достижения этой цели были поставлены следующие задачи:

- произвести обзор существующих решений сегментации изображений;
- изучить алгоритм реконструкции плотного облака точек;
- добавить в процесс оценки плотного облака точек сегментацию изображений в качестве априорного знания;
- протестировать решение на эталонных наборах данных;
- оценить влияние использования сегментации на качество реконструкции поверхностей около разрывов глубины;
- сравнить эффективность итогового решения с изначальным.

2. Обзор

2.1. Обзор базового алгоритма реконструкции плотного облака точек

В качестве основного алгоритма решено взять «Pixelwise View Selection for Unstructured Multi-View Stereo» [4], как наилучший из открытых алгоритмов. Он опирается на метод [3], который оценивает глубину для каждого пикселя на текущем изображении X^{ref} через набор неструктурированных исходных изображений X^{src}, максимизируя схожесть окрестностей на текущем изображении и гомографически деформированных окрестностей среди всех наблюдаемых исходных изображений. Гомография – это соответствие между точками с окрестности текущего изображения и точками с исходных изображений. Для этого используется GEM алгоритм (generalized expectation maximization) [2]. Этот алгоритм является итеративным методом для оценки максимума функции правдоподобия параметров в статистических моделях, где модель зависит от ненаблюдаемых скрытых переменных. Функция правдоподобия $L(\theta) = logP(z, y \mid \theta)$ показывает, насколько правдоподобно выбранное значение параметра θ при известных наблюдаемых переменных z и ненаблюдаемых y. Каждая итерация алгоритма состоит из двух шагов: на E-шаге (expectation) оценивается распределение ненаблюдаемых переменных, учитывая известные значения для наблюдаемых переменных и текущую оценку параметров, как $P^t(y) = P(y \mid z, \theta^{t-1})$. На M-шаге (maximization) переоцениваются параметры для улучшения текущей функции правдоподобия, но не обязательно ее максимизации (в отличие от EM-алгоритма), в предположении, что найденное на E-шаге распределение верно. То есть новая оценка параметров θ^t выбирается так, чтобы $E_{P(t)}(logP(z, y \mid \theta^t))$ было больше, чем $E_{P(t)}(logP(z, y \mid \theta^{t-1}))$, но не обязательно достигало своего максимума. В рассматриваемом алгоритме оцениваемый параметр – это глубина и нормаль, ненаблюдаемые переменные – индикатор того, что текущая окрестность была без перекрытий видна с камеры, соответствующей одному из исходных изобра-

жений, наблюдаемые переменные – набор исходных изображений.

Для максимизации правдоподобия (на M-шаге) взят алгоритм Patch Match [3], который быстро находит соответствия между маленькими участками изображения. Он определяет некоторую функцию $f : \mathbb{R}^2 \to \mathbb{R}^2$, которая центру каждой окрестности изображения A сопоставляет центр соответствующей ближайшей окрестности на изображении B. Рассматриваемый алгоритм предполагает оценивать кроме глубины еще и нормаль для каждого пикселя, чтобы избежать возникновения ошибок на наклонных поверхностях. Предлагается комбинировать случайные и возмущенные глубины с текущими оптимальными нормалями и наоборот, чтобы увеличить шансы выбрать наилучшее решение. В данном случае под возмущением подразумевается домножение на $1 + \varepsilon$, где ϵ – это некоторая маленькая константа. Это ведет к более высокой скорости сходимости и точности оценок. На рис. 1 приведены примеры изображений с оцененными глубинами и нормалями. На первом рисунке приведено исходное изображение, затем карта глубины, где для наглядности значения глубин интерполированы от ярко-красного до темно-синего, затем карта нормалей, где координаты нормали (x, y, z) кодируются как (r, g, b).

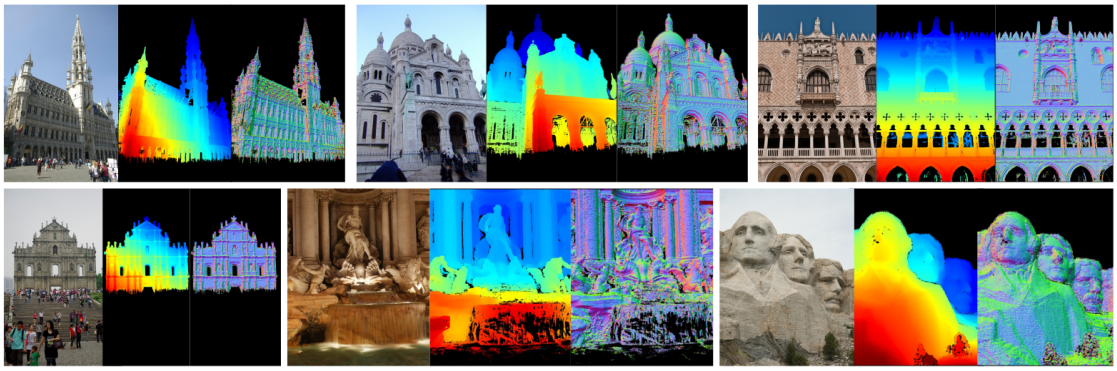

Рис. 1: Примеры изображений [4]

Важно отобрать только те исходные изображения, которые существенно могут повлиять на оценку. Если же выборка основывается только на результатах разреженной реконструкции, которая дает неполное представление о сцене, то решение может оказаться неоптимальным. Во-первых, пары изображений с нулевым расстоянием между камера-

ми не несут никакой информации про глубину, так как реконструированные точки могут свободно перемещаться по лучу обзора, но при этом имеют высокую схожесть по цвету. Чтобы устранить этот вырожденный случай, предлагается посчитать угол между двумя пересекающимися лучами обзора, как меру стабильности реконструированной точки. Во-вторых, глубина и нормаль накладывают ограничения на исходные изображения. Позиция камеры может находиться только в положительном полупространстве, отсеченным плоскостью, образованной векторами глубины и нормали. К тому же, камера должна быть направлена на вектор, противоположный нормали. Иначе камера не могла бы наблюдать поверхность. Для выполнения этих ограничений предлагается считать смежный угол к направлению обзора камеры лежащим от $[0; \frac{\pi}{2}]$. В-третьих, чтобы как-то сравнивать окрестности изображений, они должны быть примерно одного размера и формы, чтобы избежать недо- и передискретизации. Например, для выбранной на текущем изображении окрестности гомографически деформированная окрестность на исходном изображении может выродиться в точку. Во избежание таких случаев предлагается сравнивать площади окрестностей.

Для оценки фотометрической согласованности предлагается использовать двустороне-взвешенную адаптацию нормированной кросс- кореляции (NCC). Это стандартный метод оценки степени корреляция двух окрестностей, который учитывает удаленность пикселей от центра окрестности, а так же удаленность реконструированных точек в пространстве от центральной, что позволяет избежать ошибок на разрывах глубины. MVS так же страдает от большого числа выбросов из-за шума, неоднозначности и преград. Предлагается интегрировать их фильтрацию в процесс вывода. Так как обычно шумы возникают только при определенном положениии камеры, то имея информацию с нескольких камер, обозревающих одну и ту же точку, можно определить правильное решение. Для этого вводится ошибка отображения, как норма расстояния между текущей точкой и спроецированной на исходное изображение и обратно. Чем она меньше, тем более геометрически согласованны оценённые глубины и нормали.

Итого, GEM алгоритм использует только те исходные изображения, которые существенно могут повлиять на оценку глубины и нормали в текущей окрестности, основываясь на трёх перечисленных критериях, а в качестве метрики схожести окрестностей использует фотометрическую и геометрическую согласованность. Так как оценка глубин и нормалей для всех изображений одновременно не выполнима из-за ограничений по памяти, то алгоритм состоит из 2 стадий. В первой стадии оцениваются начальные глубины и нормали для всех входных изображений. Во второй стадии производится покоординатный спуск для оценки геометрически согласованных нормалей и глубин для каждого изображения X^{ref}.

После этого остается только отфильтровать оставшиеся выбросы и объединить данные в итоговое решение. Для каждого пикселя текущего изображения определяется опорный набор пикселей из исходных изображений, которые удовлетворяют геометрической и фотометрической согласованности. Строится граф, в котором узлы – это пиксели с достаточно большим опорным набором, а ребра – это соответствия между X^{ref} и X^{src}, причём для всех пикселей уже посчитаны глубины и нормали. Пока этот граф не пуст, в нём находится кластер наиболее согласованных пикселей, которые в итоге образуют облако точек.

2.2. Существующие решения: сегментация изображений

2.2.1. Pyramid Scene Parsing Network

Анализ сцены, основанный на семантической сегментации – это фундаментальная тема компьютерного зрения. Его целью является присвоение каждому пикселю на изображении осмысленной категориальной метки, например, стул, машина, дом, а сложность связана с разнообразием сцены и неограниченным количеством близких по смыслу категорий. Большая часть алгоритмов анализа сцены основана на свёрточных нейронных сетях, которые не используют глобальные знания

об изображении. Основные типы возникающих ошибок – это несоответствие классифицированного объекта контексту, ошибка среди выбора из синонимичных понятий, незаметные объекты из-за своего слишком маленького или большого размера.

Pyramid Scene Parsing Network [5] предполагает помимо стандартной свёрточной нейронной сети использовать специально разработанный глобальный пирамидальный модуль, где каждый уровень формируется операцией субдискретизации или подвыборки из карты признаков. Таким образом, на итоговое предсказание категории влияет как локальная, так и глобальная информация, что приводит к более надёжному результату.

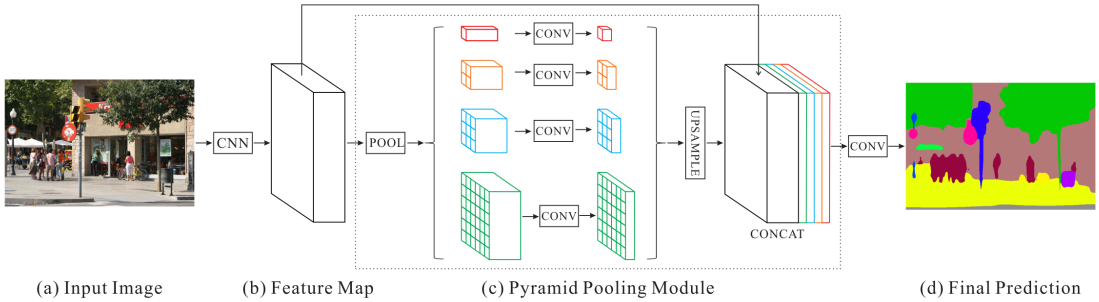

Рис. 2: Архитектура PSPNet [5]

Как показано на рис. 2, на входном изображении запускается обученная ResNet модель [1], чтобы выделить карту признаков. После этого на ней запускается пирамидальный модуль субдискретизации (pooling), чтобы собрать контекстную информацию. Используется 4-ступенчатая пирамида, чьи элементы покрывают определенную часть изображения (все, половину, четверть, малую часть). Затем у более глобальных уровней повышается дискретизация путем билинейной интерполяции, чтобы достичь изначальной размерности карты признаков. В итоге, уровни объединяются, и слой свёртки генерирует результат.

2.2.2. Rethinking Atrous Convolution for Semantic Image Segmentation

Для решения задачи семантической сегментации рассматриваются два подхода на основе глубоких свёрточных нейронных сетей. Первый понижает размерность факторов путём последовательных операций дискретизации или свёрточных шагов, которые и позволяют свёрточной нейронной сети изучать все больше и больше абстрактных представлений факторов. Проблема этого подхода заключается в том, что получается инвариантность относительно локальных трансформаций изображения, что мешает правильному предсказанию, так как пропадает детализированная информация. Поэтому предлагается другой подход на основе расширяющегося свёртывания (atrous convolution) [6]. Он позволяет переориентировать предобученную нейронную сеть, чтобы выделять плотные карты признаков (факторов). Для этого убираются понижающие размерность операции (операции дискретизации) с последних слоев нейронной сети, и повышается дискретизация соответствующих ядер, что соответствует добавлению промежутков между весами по формуле: $y[i] = \sum_k x[i + r * k]w[k]$, где x – это входная карта признаков, w – вес, а r – коэффициент расширяющегося свертывания, который означает добавление $(r - 1)$ нуля. Исходная нейронная сеть соответствует $r = 1$. На рис. 3 приведен пример использования сверточных нейронных сетей с и без расширяющегося свертывания.

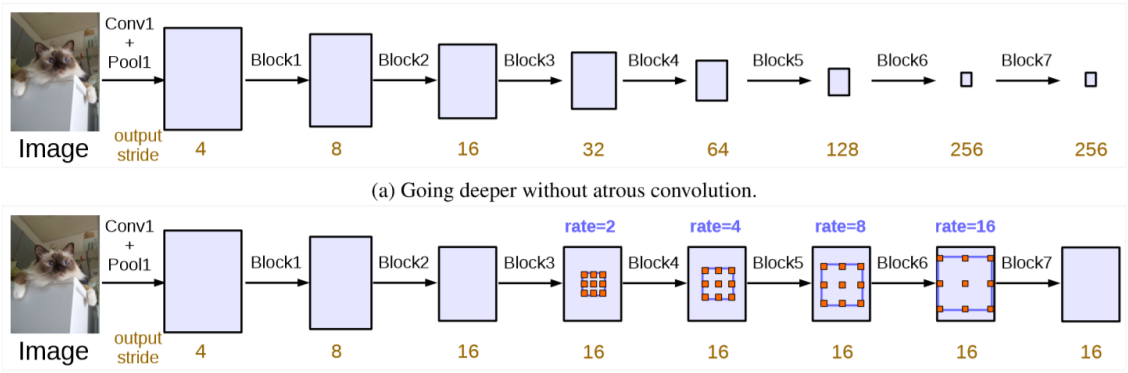

Рис. 3: Каскадные модули с и без расширяющегося свёртывания [6]

3. Описание решения

3.1. Принятые технические решения

В связи с тем, что предполагается интегрировать сегментацию в существующую реализацию [1] статьи [4], то и язык разработки выбран аналогичный – C++. Так же, поскольку трехмерная реконструкция большого числа изображений – трудоемкий процесс, используются ресурсы видеокарты для ускорения расчетов, а конкретно технология CUDA. Для сегментации был выбран алгоритм [6], поскольку в данной работе не так важно определить семантику выделенного сегмента, как его точные границы. На рисунке 4 приведены примеры сегментированных изображений из набора данных для 3D реконструкции.

Рис. 4: Пример релевантных сегментированных изображений

3.2. Сегментация

В процессе работы выяснилось, что наборы данных для 3D - реконструкции обладают некоторыми особенностями. Во-первых, большинство изображений в большом разрешении, то есть большого размера, например, 6048×4032, тогда как наборы данных для сегментации не

[1] https://github.com/colmap/colmap

содержат изображений больше, чем 1024×1024. Поэтому нейронная сеть, обученная на маленьких картинках не справлялась с релевантными нам данными. Было решено разбивать исходное большое изображение на несколько маленьких, сегментировать их отдельно, а затем склеивать. Во-вторых, большинство изображений – темные и неконтрастные, что мешает правильному выделению объектов. Поэтому было решено применить алгоритм адаптивного контрастирования изображения (CLAHE). Этот метод увеличивает общий контраст изображения при помощи выравнивания гистограммы, что позволяет эффективно распределять наиболее часто используемые значения интенсивности. В итоге, на рисунке 5 представлен итоговый результат для сегментированных изображений

Рис. 5: Итоговый результат для релевантных сегментированных изображений

3.3. Описание предлагаемого подхода

Ранее на видеокарту загружались только текущее и исходные изображение, после чего параллельно производился покоординатный спуск, чтобы оценить геометрически согласованные глубины и нормали. Теперь предлагается при инициализации предпосчитать сегментированные изображения при помощи описанного выше способа. Далее на ви-

деокарту загрузить изображения и соответствующие им сегментированные. При оценке фотометрической и геометрической согласованности предлагается определять сегмент, которому принадлежит текущий пиксель, «занулять» все пиксели из окрестности, которые не принадлежат этому сегменту, после чего пересекать получившиеся окрестности на X^{ref} и X^{src}. Для полученных окрестностей вычисление фотометрической и геометрической согласованности производится аналогично существующему решению.

3.3.1. Фотометрическая согласованность

Для оценки фотометрической согласованности используется двустороневзвешенная адаптация NCC.

$$p_l^m = \frac{cov_w(w_l, w_l^m)}{\sqrt{cov_w(w_l, w_l) cov_w(w_l^m, w_l^m)}} \tag{1}$$

Степень схожести считается по формуле 1 между окрестностью на текущем изображении w_l с центром в x_l и окрестностью на исходном изображении w_l^m с центром в x_l^m. Где взвешенная корреляция считается, как $cov_w(x, y) = E_w(x - E_w(x))E_w(y - E_w(y))$, а взвешеное среднее $E_w(x) = \frac{\sum_{i \in A} w_i x_i}{\sum_{i \in A} w_i}$. Вес считается по формуле, которая зависит от текущего и центрального пикселей. Пусть теперь s_l – это сегмент, которому принадлежит пиксель x_l, а $A(s_l)$ – множество всех пикселей окрестности, принадлежащих сегменту s_l. Для $A(s_l^m)$ определим аналогично, только на изображении X^{src}. Тогда множество $A := \{x : x \in A(s_l), H(x) \in A(s_l^m)\}$, где H(x) – это соответствующий x пиксель на изображении X^{src}.

3.3.2. Геометрическая согласованность

Для оценки геометрической согласованности используется ошибка прямого-обратного проецирования. Точке x_a на текущем изображении X^{ref} сопоставляется точка в трехмерном мире на основании текущих оценённых нормали и глубины, которая потом проецируется на исходное изображение X^{src}. И обратно, полученная точка на исходном изображении переходит в трехмерный мир на основании оценённой глуби-

ны, а затем проецируется на текущее изображение в точку x_b. Расстояние между полученными точками x_a и x_b определяет искомую величину ошибки. Теперь, в том случае, если эти две точки лежат в разных сегментах на соответствующем сегментированном изображении, то величина ошибки равна некому наперед заданному большому значению, то есть определяет неправильность такого расположения.

3.4. Тестирование

Алгоритм был протестирован на наборах данных из ETH3D Benchmark [2], здесь приведены результаты для датасета «kicker». В качестве метрик для оценки качества были взяты три следующие метрики. Полнота – это доля истинных значений, которая ближе к реконструкции, чем пороговое значение расстояния (в процентах). Точность – это доля реконструкции, которая ближе к истинным значениям, чем пороговое значение расстояния (в процентах). F1-score - это гармоническое среднее точности и полноты, используемое для общей оценки метода. Все метрики рассчитываются в зависимости от порогового расстояния. В таблицах 1, 2, 3 приведены результаты, рассчитанные для 6 разных пороговых расстояний.

Таблица 1: Результаты исходного алгоритма

	0.01	0.02	0.05	0.1	0.2	0.5
Полнота	0.196012	0.270566	0.402766	0.545028	0.73063	0.935438
Точность	0.922381	0.964014	0.987058	0.992411	0.994564	0.996181
F1-scores	0.323317	0.42254	0.572092	0.703627	0.842408	0.964854

[2]https://www.eth3d.net/datasets

Таблица 2: Результаты алгоритма с исходной сегментацией

	0.01	0.02	0.05	0.1	0.2	0.5
Полнота	0.156865	0.214676	0.321279	0.438149	0.601273	0.802415
Точность	0.921377	0.965165	0.989506	0.994607	0.996093	0.997378
F1-scores	0.268088	0.35123	0.485064	0.608319	0.749889	0.889337

Таблица 3: Результаты алгоритма с улучшенной сегментацией

	0.01	0.02	0.05	0.1	0.2	0.5
Полнота	0.167473	0.230167	0.345057	0.469382	0.644189	0.860092
Точность	0.919143	0.964215	0.989203	0.994518	0.996074	0.997343
F1-scores	0.283323	0.371624	0.511642	0.637761	0.782387	0.923647

Заключение

В ходе работы были выполнены следующие задачи:

- произведен обзор существующих решений сегментации изображений;
- изучен алгоритм реконструкции плотного облака точек;
- улучшена сегментация на релевантных наборах данных;
- добавлена сегментация изображений в качестве априорного знания в процесс оценки фотометрической согласованности.
- добавлена сегментация изображений в качестве априорного знания в процесс оценки геометрической согласованности;
- протестировано решение на эталонных наборах данных;
- оценено влияние использования сегментации на качество реконструкции поверхностей около разрывов глубины;

В результате сравнения эффективности итогового решения с изначальным был сделан вывод, что на данный момент улучшение точности не стоит затраченных ресурсов. В дальнейшем возможно движение в сторону улучшения сегментации и, соответственно, результатов.

Список литературы

[1] Deep residual learning for image recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 770–778.

[2] Neal Radford M, Hinton Geoffrey E. A view of the EM algorithm that justifies incremental, sparse, and other variants // Learning in graphical models. — 1998. — P. 355–368.

[3] Patchmatch based joint view selection and depthmap estimation / Enliang Zheng, Enrique Dunn, Vladimir Jojic, Jan-Michael Frahm // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2014. — P. 1510–1517.

[4] Pixelwise view selection for unstructured multi-view stereo / Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, Marc Pollefeys // European Conference on Computer Vision. — 2016. — P. 501–518.

[5] Pyramid scene parsing network / Hengshuang Zhao, Jianping Shi, Xiaojuan Qi et al. // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 2881–2890.

[6] Rethinking atrous convolution for semantic image segmentation / Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam // arXiv preprint arXiv:1706.05587. — 2017.