

Аннотация

Оглавление

Аннотация	1
Введение.....	4
Определения, обозначения, сокращения	6
1. Обзор существующих методов визуальной оценки положения и формы объектов	7
1.1. Методы анализа двумерных изображений.....	7
1.2. Методы анализа трехмерных облаков точек.....	7
1.3. Интеллектуальные системы технического зрения в робототехнике.	26
1.4. Заключение первой главы	29
2. Модели и алгоритмы нейро-эволюционной системы технического зрения	30
2.1 Архитектура нейро-эволюционной системы технического зрения....	31
2.2. Выбор и настройка нейросетевой модели для оценки положения объектов в кадре.....	32
2.2.1 Принципы построения сверточных нейронных сетей	33
2.2.2 Разработанная архитектура модели	34
2.2.3 Процесс обучения модели.....	35
2.2.4 Оценка модели	36
2.3. Разработка эволюционного алгоритма для оценки формы объектов	37
2.3.1 Описание алгоритма RANSAC.....	38
2.3.2 Описание генетического алгоритма.....	42
2.3.3 Оценка работы эволюционного алгоритма.....	43
2.4 Заключение второй главы	44

3. Разработка программного обеспечения для анализа изображения с бортовой камеры автономного робота.....	45
3.1 Структура программного обеспечения.....	46
3.2 Используемые инструменты	47
3.3 Порядок работы с программным обеспечением.....	48
4. Проведение экспериментальных исследований	49
4.1 Результаты работы эволюционного алгоритма	50
4.2 Результаты работы нейро-эволюционного алгоритма на данных с бортовой камеры	51
4.3 Заключение четвертой главы	52
5. Организационно-экономическая часть	53
Заключение	54
Используемая литература.....	55
Приложения	56

Введение

Системы технического зрения применяются во многих областях робототехники: начиная от мобильных роботов, заканчивая стационарными промышленными манипуляторами. Одна из основных задач систем машинного зрения – анализ среды функционирования робота. Среду функционирования можно представить как в виде двумерного изображения, так и в формате трехмерного облака точек, по которому можно дать качественную характеристику окружающим робота объектов и местности.

Актуальными средствами регистрации облаков точек на данный момент являются камеры глубины с инфракрасным датчиком, стереокамеры и сканирующие лидары. На основе полученных данных можно построить трехмерную карту местности по методу SLAM, однако, это не позволяет дать оценку окружающей среде с точки зрения находящихся в ней объектов. Для полноценного взаимодействия робота с целевыми объектами необходима информация об их форме, габаритных характеристиках, положении и ориентации в пространстве. Также подобные данные требуется для определения положения самого робота, планирования траектории движения и обхода препятствий. Для достижения этой цели следует классифицировать группы точек в облаке таким образом, чтобы можно было описать их поверхностями третьего порядка или заданными формами-примитивами.

Совместив полученную информацию об объектах в поле зрения робота с данными о его положении, можно составить картину о среде функционирования путем нанесения на карту встреченных роботом объектов и препятствий, что позволит производить качественную разведку местности и поиск на ней целевых объектов.

Хорошие результаты при решении задач сегментации объектов в облаке точек показывают нейросети, однако, они требуют больших вычислительных мощностей, что накладывает ограничения на создание полностью автономной и независимой от соединения с удаленным сервером робототехнической системы.

В данной работе будет рассмотрен метод сегментации облаков точек на формы-примитивы, способный работать на маломощных встраиваемых вычислительных системах.

Определения, обозначения, сокращения

1. Обзор существующих методов визуальной оценки положения и формы объектов

Оценка положения и формы объектов в кадре может производиться в двумерном пространстве кадра изображения и в трехмерном облаке точек. Анализ двумерного изображения позволяет определить положение целевого объекта в кадре и форму его видимой области, однако, не позволяет доступно и качественно определить форму его составных частей. С помощью облака точек, составленного по карте глубины, можно дать характеристику формы, положения частей и ориентации объекта.

1.1. Методы анализа двумерных изображений

В системах технического зрения задача сегментации объектов в кадре решается с помощью нескольких подходов:

- Классические методы:
 1. Выделение контуров;
 2. Выделение по цветовой маске;
 3. Использование дескрипторов и детекторов особых точек;
- Нейросетевые методы с использованием сверточных нейронных сетей:
 1. Двухэтапный метод;
 2. Одноэтапный метод.

Выделение контуров

Среди множества алгоритмов поиска контуров на изображении широко используется оператор Кэнни. Алгоритм можно разбить на пять этапов:

1. Сглаживание изображения;

2. Поиск градиентов. Границы отмечаются там, где градиент изображения приобретает максимальное значение;
3. Подавление не-максимумов. Только локальные максимумы отмечаются как границы;
4. Двойная пороговая фильтрация. Потенциальные границы определяются порогами;
5. Трассировка области неоднозначности. Итоговые границы определяются путем подавления всех краев, несвязанных с сильными границами.

Прежде чем начать определять контуры объекта, нужно отфильтровать шумы на изображении. Для этого зачастую применяется размытие по Гауссу. Фильтр Гаусса представляет собой средневзвешенное значение яркостей соседних пикселей в области матрицы ядра с весом, уменьшающимся с расстоянием от центра ядра p .

Дадим определение свертке по функции Гаусса:

$$GC[I]_p = \sum_{q \in S} G_{\sigma}(\|p - q\|) I_q \quad (1)$$

$$G_{\sigma}(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (2)$$

Вес пикселя определяется $G_{\sigma}(\|p - q\|)$, $L2$ нормой матрицы (Евклидовой нормой), которая представляет собой геометрическое расстояние между точками в пространстве, где σ – параметр, определяющий размер окрестности. Сила этого влияния зависит только от пространственного расстояния между пикселями, но не от их значений. Так, яркий пиксель имеет сильное влияние на соседний темный пиксель, хотя значения яркостей этих пикселей сильно отличаются. В результате, края изображения размываются, так как пиксели на неоднородных участках усредняются все вместе.

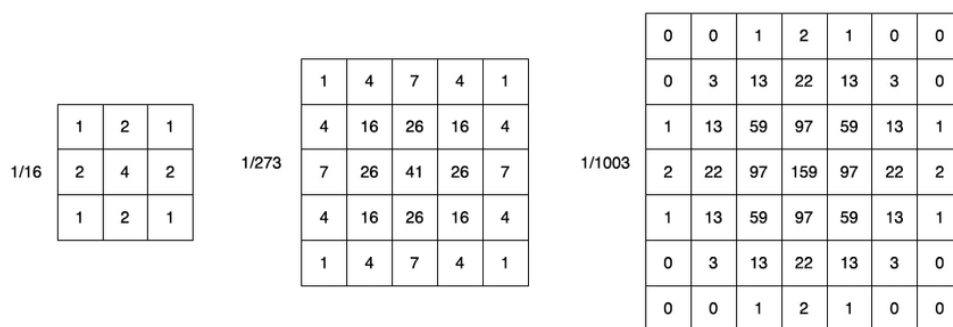


Рисунок 1. Примеры ядер фильтра Гаусса с размерами 3x3, 5x5 и 7x7

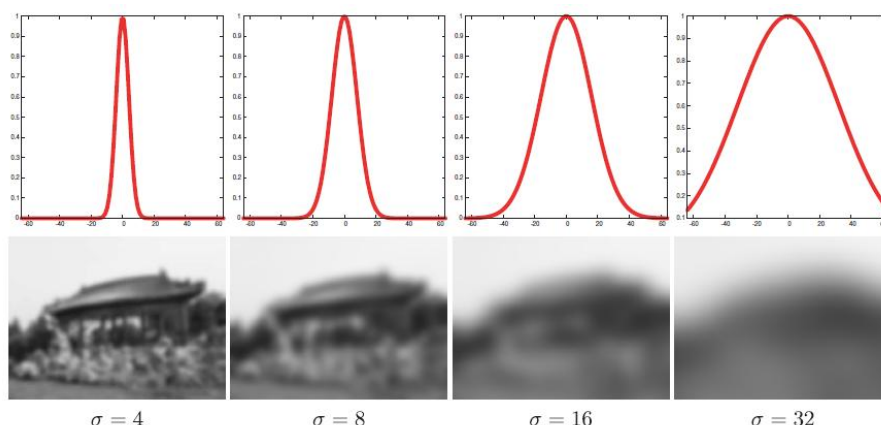


Рисунок 2. Пример Гауссова размытия при разных размерах ядра

Поиск градиента осуществляется следующим образом. Границы отмечаются там, где градиент изображения приобретает максимальное значение. Они могут иметь различное направление, поэтому алгоритм Кэнни использует четыре фильтра для обнаружения горизонтальных, вертикальных и диагональных ребер в размытом изображении.

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctg\left(\frac{G_y}{G_x}\right)$$

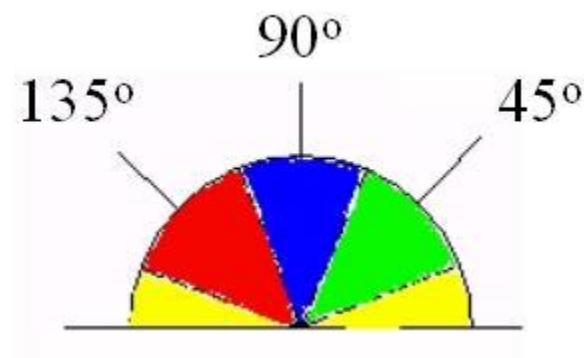


Рисунок 3. Определение направления градиента

Угол направления вектора градиента округляется и может принимать значения 0, 45, 90 и 135 градусов. Направление края, попадающего в каждую цветовую область будет установлено на определенное значение угла.

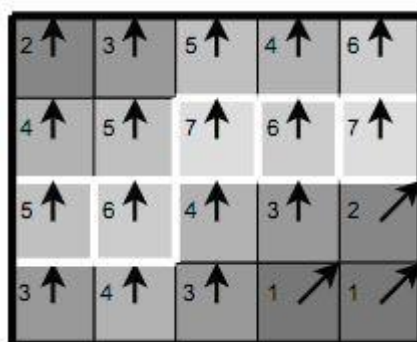


Рисунок 4. Визуализация градиента

Подавление не-максимумов – техника уменьшения толщины границ. Применяется для нахождения областей с резким изменением яркости. Алгоритм для каждого пикселя в градиентах:

1. Сравнение силы границ текущего пикселя с силой границ пикселей в положительных и отрицательных направлениях градиента
2. Если сила границ текущего пикселя больше значения в маске с пикселями с таким же направлением градиента, значение будет сохранено, в противном случае – подавлено.

Сила границ пикселя – производная первого порядка, например, величина градиента.

После подавления не-максимумов, оставшиеся пиксели дают более точное представление границ на изображении. Однако некоторые пиксели остаются из-за шума и цветового разброса. Чтобы отбросить эти выбросы, важно отфильтровать пиксели со слабым значением градиента и сохранить пиксели с сильным значением градиента. Для этого выбираются верхнее и нижнее пороговые значения. Если значение градиента краевого пикселя выше верхнего порогового значения, он помечается как сильный краевой пиксель. Если значение градиента краевого пикселя меньше верхнего порогового значения и больше нижнего значения, он помечается как слабый. Если значение меньше нижнего порога, оно будет подавлено. Пороговые значения определяются эмпирически и будут зависеть от содержимого изображения.

Пиксели с сильными границами обязательно будут задействованы в конечном результате, поскольку они извлекаются из истинных границ изображения. Однако возникает спорный момент с пикселями со слабыми границами, так как они могут быть извлечены и из истинных границ, и из шума. Для достижения более точного результата следует избавиться от пикселей вызванных шумом. Обычно пиксель со слабым значением, полученный от истинных границ будет связан с пикселем с сильным значением, в то время как шумовые характеристики ни с чем не связаны. Чтобы отследить связь, рассматривают слабый пиксель и соединенные с ним соседние 8 пикселей. Если есть хотя бы один сильный пиксель в этом квадрате, слабый пиксель можно сохранить.

В результате работы вышеперечисленных методов можно получить четкие контуры изображения.



Рисунок 5. Контур изображения

Выделение по цветовой маске

Выделение объекта с помощью цветовой маски используется в случае, если целевой объект существенно выделяется на фоне по цвету. Осуществляется путем преобразования изображения из RGB в HSV пространство и выделения в нем нужных значений тона, насыщенности и значения.

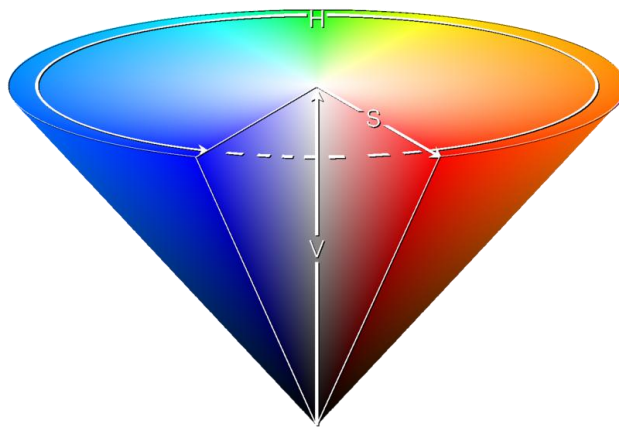


Рисунок 6. Цветовая модель HSV

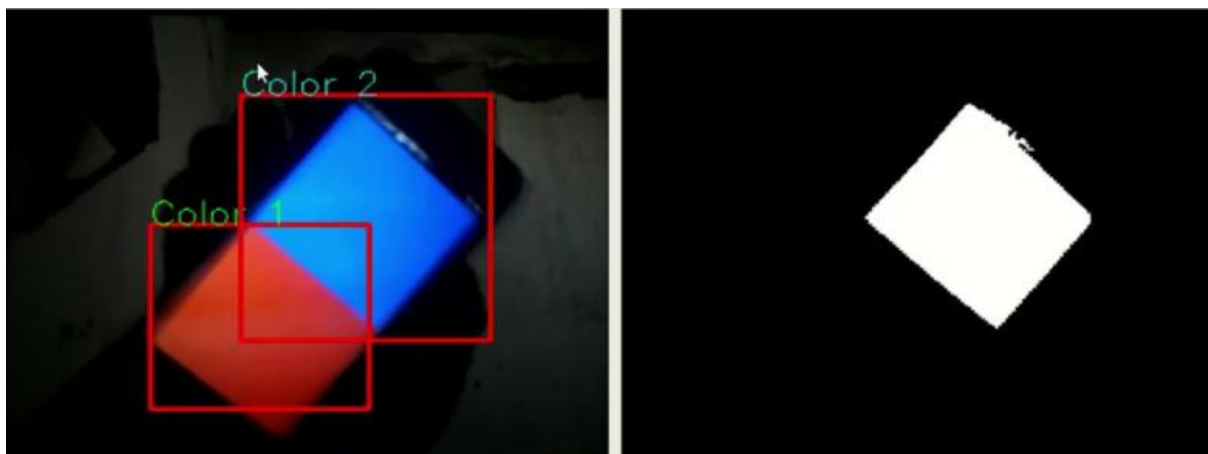


Рисунок 7. Выделение объектов по цветовой маске

Использование дескрипторов и детекторов особых точек

При одинаковой яркости с фоном или перекрытии объект может не иметь четких границ, поэтому до популяризации нейросетевых подходов для поиска объектов широко использовались детекторы признаков и дескрипторы. Детектор особых точек – алгоритм, который выбирает точки на изображении на основе некоторого критерия. Дескриптор – вектор значений, который описывает участок изображения вокруг интересующей точки. Вместе особая точка и ее дескриптор обычно называют локальной особенностью.

Основными критериями нахождения особых точек являются:

- **Отличимость** – особая точка должна явно выделяться на фоне и быть отличимой (уникальной) в своей окрестности;
- **Инвариантность** – определение особой точки должно быть независимо к аффинным преобразованиям;
- **Стабильность** – определение особой точки должно быть устойчиво к шумам и ошибкам;

- Уникальность – кроме локальной отличимости, особая точка должна обладать глобальной уникальностью для улучшения различимости повторяющихся шаблонов;
- Интерпретируемость – особые точки должны определяться так, чтобы их можно было использовать для анализа соответствий и выявления интерпретируемой информации из изображения.

Широкую популярность приобрели такие дескрипторы как SIFT (Scale Invariant Feature Transform), HOG (Histogram of Oriented Gradients), ORB (Oriented FAST and rotated BRIEF).

Дескрипторы могут решать задачу сопоставления изображений, что позволяет детектировать и классифицировать объекты в кадре. В общем случае получается схема на Рис.8:



Рисунок 8. Алгоритм сопоставления изображений

Основным этапом при детектировании особых точек является построение пирамиды гауссианов и разностей гауссианов. Разностью гауссианов называют изображение, полученное путем попиксельного вычитания

одного гауссиана исходного изображения из гауссиана с другим радиусом размытия.

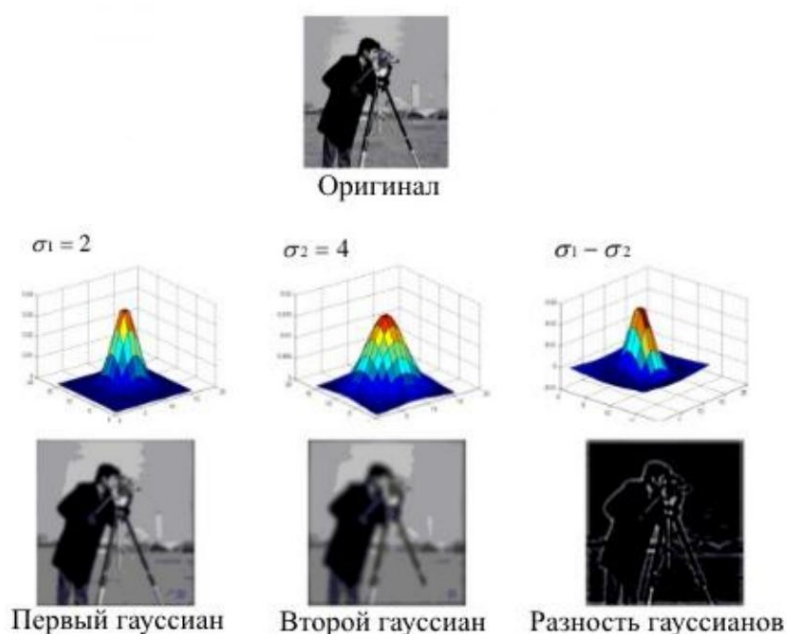


Рисунок 9. Построение разности гауссианов

Масштабируемое пространство изображения — множество различных, сглаженных некоторым фильтром, вариантов исходного изображения. Гауссово масштабируемое пространство является линейным и инвариантным относительно сдвигов, вращений, масштаба, не смещающим локальные экстремумы, и обладает свойством полугрупп. Различная степень размытия изображения гауссовым фильтром может быть принята за исходное изображение, взятое в некотором масштабе.[цитата]

Инвариантность относительно масштаба достигают за счет нахождения ключевых точек для исходного изображения, взятого в разных масштабах. Поэтому строят пирамиду гауссианов: все масштабируемое пространство разбивают на некоторые участки — октавы, причем часть

масштабируемого пространства, занимаемого следующей октавой, в 2 раза больше части, занимаемой предыдущей. При переходе от одной октавы к другой происходит уменьшение размеров изображения в 2 раза. Каждой октаве изображения соответствует бесконечное множество гауссианов изображения. В связи с этим необходимо строить только некоторое их число N с определенным шагом по радиусу размытия. С тем же шагом достраивают два дополнительных гауссиана, таким образом, получают $N + 2$ гауссиана. Масштаб первого изображения следующей октавы равен масштабу изображения предыдущей октавы с номером N . Сглаживая изображение гауссовым фильтром по приведенному алгоритму, получаем пирамиду гауссианов. [цитата]

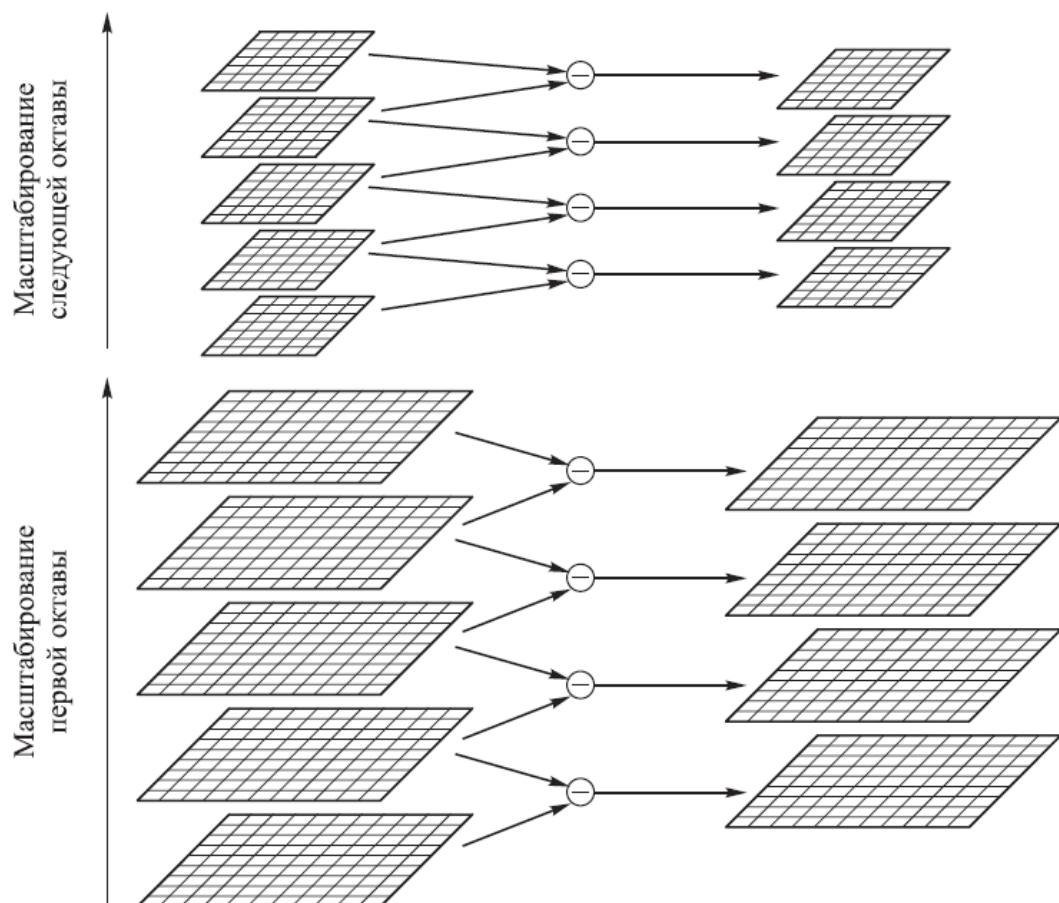


Рисунок 10. Построение разности гауссианов

После построения пирамид определяют является ли точка ключевой, а также определяют ее ориентацию. Точка является ключевой, если она представляет собой локальный экстремум разности гауссианов. Далее создается дескриптор ключевой точки.



Рисунок 11. Пример получения дескриптора



Рисунок 12. Пример работы дескриптора SIFT

Вышеперечисленные подходы успешно могут справляться с некоторыми задачами поиска объектов, однако, в значительной мере уступают в производительности нейросетевым подходам.

Нейросетевые методы с использованием сверточных нейронных сетей

С популяризацией в 2012 году сверточных нейронных сетей [Krizhevsky] в задачах классификации изображений и детекции объектов стали применять различные модели на их основе.

В общем случае детекция сводится к перебору ограничивающих рамок целевого объекта на изображении с порогом уверенности в правдивости классификации. За счет того что рамки могут быть различных размеров и иметь различные координаты, полный перебор становится длительным и неэффективным методом. Для уменьшения количества этих рамок выделяют два подхода:

1. Двухэтапный метод – на первом этапе с помощью селективного поиска или слоя нейронной сети выделяются регионы интереса, с некоторой вероятностью содержащие внутри целевые объекты. На втором этапе регионы рассматриваются классификатором с точки зрения принадлежности исходным классам и производится уточнение ограничивающих рамок.
2. Одноэтапный метод – метод, не использующий отдельный алгоритм для генерации регионов интереса, а вместо этого предсказывающий координаты определенного количества ограничивающих рамок с различными параметрами, такими как степень уверенности результатов классификации и корректируя в дальнейшем их положение.

Двухэтапные методы

Двухэтапные методы используются в таких известных моделях как R-CNN/Fast/Faster R-CNN/Mask R-CNN.

В качестве примера рассмотрим R-CNN (Region-based Convolutional Neural Network). Вместо использования для поиска сегментов скользящие окна фиксированного размера, на первом шаге алгоритм пытается найти селективным поиском "регионы" – прямоугольные области различных размеров, которые, могут содержать целевой объект. Размеры найденных регионов подаются на вход сверточной нейронной сети. На последнем этапе вектора признаков регионов анализируются с помощью метода опорных векторов, проводится классификация регионов.

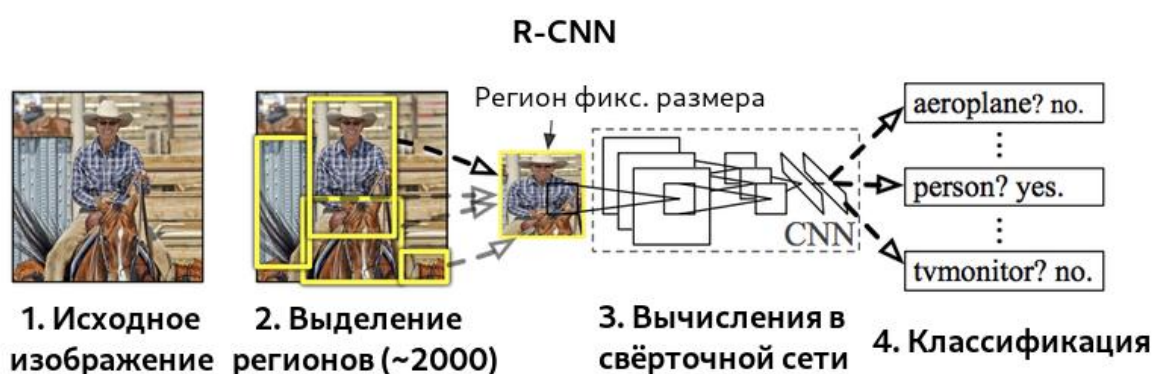


Рисунок 13. Схема работы R-CNN

Одноэтапные методы

По сравнению с двухэтапными методами, одноэтапные имеют более высокое быстродействие. Рассмотрим принцип работы на примере SSD.

Модель SSD использует принцип пирамидальной иерархии выходов сверточной нейросети для эффективного обнаружения объектов различных размеров. Изображение последовательно подается на уменьшающиеся в размерах слои свёрточной сети. Выход из последнего слоя каждой размерности участвует в принятии решения по детекции объектов, таким образом, складывается "пирамидальная характеристика" изображения. Это дает возможность обнаруживать объекты различных размеров, так как размерность выходов первых слоёв сильно коррелирует с ограничивающими рамками для крупных объектов, а последних – для небольших. Таким образом, крупные размеры могут быть обнаружены на более высоком уровне, а маленькие объекты – на низких уровнях. Как и в других моделях, функция потерь обеспечивает совместный вклад как в локализацию, так в классификацию.

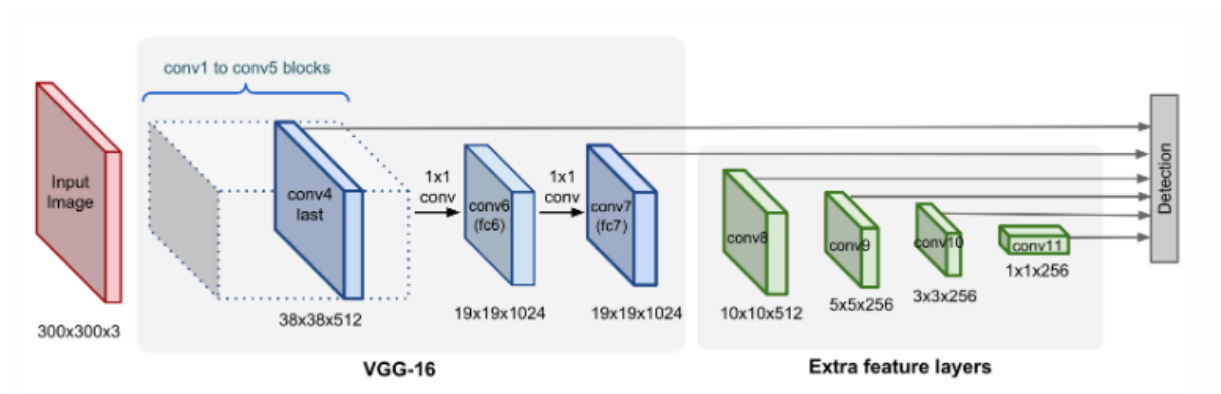


Рисунок 14. Архитектура нейронной сети для алгоритма SSD

Как показал обзор актуальных методов поиска объектов на изображении, для соблюдения баланса между эффективностью и быстродействием в поставленной задаче оптимальнее всего использовать алгоритм SSD.

1.2. Методы анализа трехмерных облаков точек

Развитие методов анализа и сегментации облаков точек началось с методов использования эвристических алгоритмов. Основным методом, который используется в некоторых задачах и до сих пор – алгоритм RANSAC и различные его модификации. В статье [schnabel] представлен способ сегментации и аппроксимации сложного облака точек с помощью данного алгоритма путем поиска в нем форм-примитивов плоскости, цилиндра, сферы и конуса.



Рисунок 15. Результат работы алгоритма RANSAC для аппроксимации облака точек примитивами

Алгоритм отличается хорошей устойчивостью к шумам и выбросам, а также способен сравнительно быстро работать в большом облаке точек и способен работать с формами различной степени замкнутости. Облако точек составленное по карте глубины не является полноценным результатом сканирования местности и объекта, а значит трехмерное облако ограничено ракурсом и углом обзора камеры, то есть задние части объектов не видны.

Еще один алгоритм основанный на RANSAC представлен в статье [baysac буряты]. Его отличием является возможность одновременного поиска нескольких примитивов и вероятностный подбор точек в окрестности

формы. Это позволяет выбирать точки для построения гипотезы примитива не вслепую, что ускоряет работу алгоритма и точность найденных гипотез.

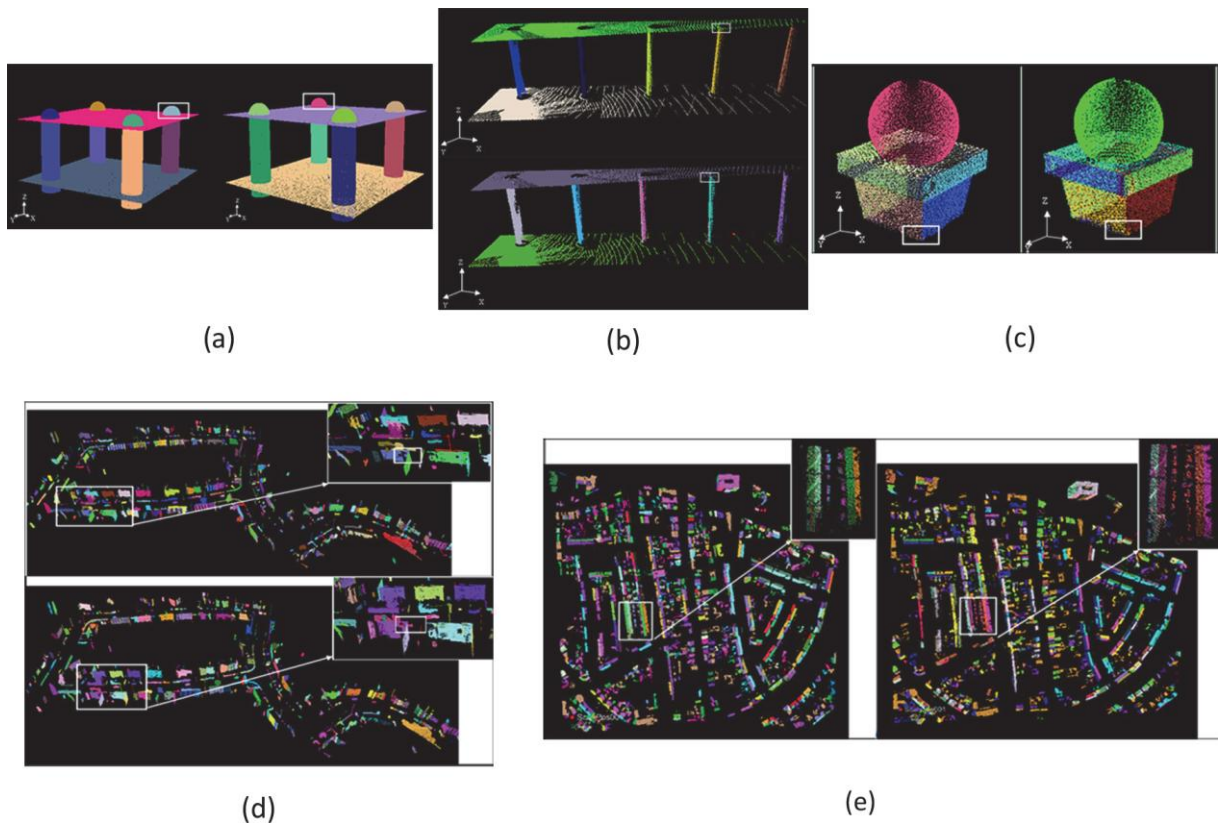


Рисунок 16. Результат работы MultyBaySAC

Современные способы сегментации облаков точек также бывают основаны на нейросетях. Самые популярные и эффективные – PointNet и SPFN (Supervised Primitive Fitting Network), основанный на PointNet.

Типичные сверточные архитектуры для перераспределений весов и оптимизации ядра требуют регулярных форматов входных данных, таких как сетки изображений или трехмерные воксели. Так как облака точек имеют нестандартный формат, обычно их преобразуют в трехмерные воксельные сетки, наборы изображений или проекции на координатные плоскости прежде чем подавать на вход нейронной сети. Однако такие

преобразования делают входные данные слишком объемными – появляются артефакты квантизации, которые могут скрыть естественную инвариантность данных. По этой причине было решено в качестве входных данных использовать само облако точек. Выходом нейросети является класс для всего входа или его сегмента.

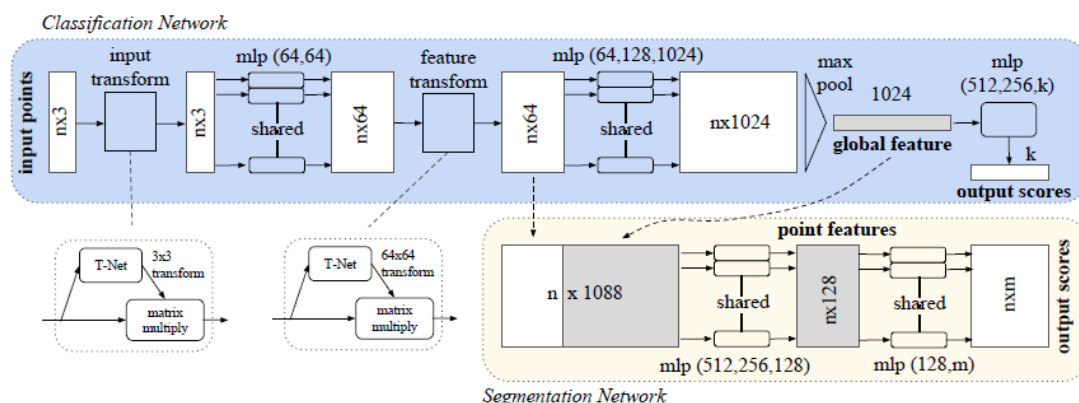


Рисунок 17. Архитектура PointNet

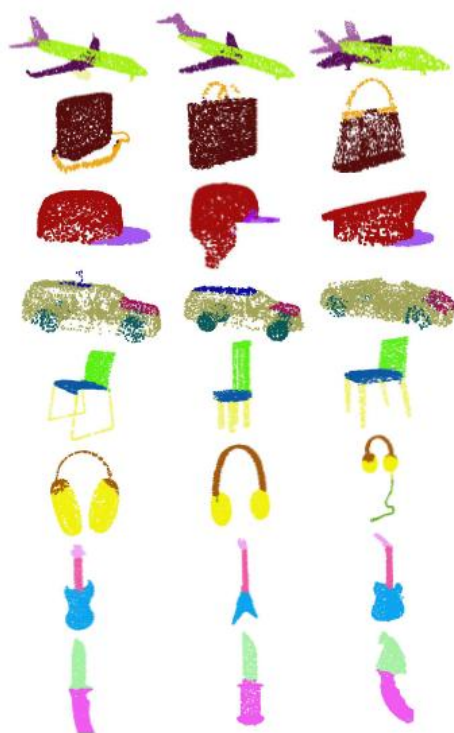


Рисунок 18. Результат работы PointNet при сегментации частей

Как показано на Рис.18, модель успешно справляется с задачей сегментации частей, а также может производить семантическую классификацию найденных частей.

Основанная на PointNet модель SPFN одновременно решает задачу сегментации и классификации методом разделения облака точек на примитивы. В набор входят плоскость, сфера, цилиндр и конус. В сравнении с RANSAC этот метод показывает гораздо более точные результаты, что делает его использование более выгодным в некоторых задачах.

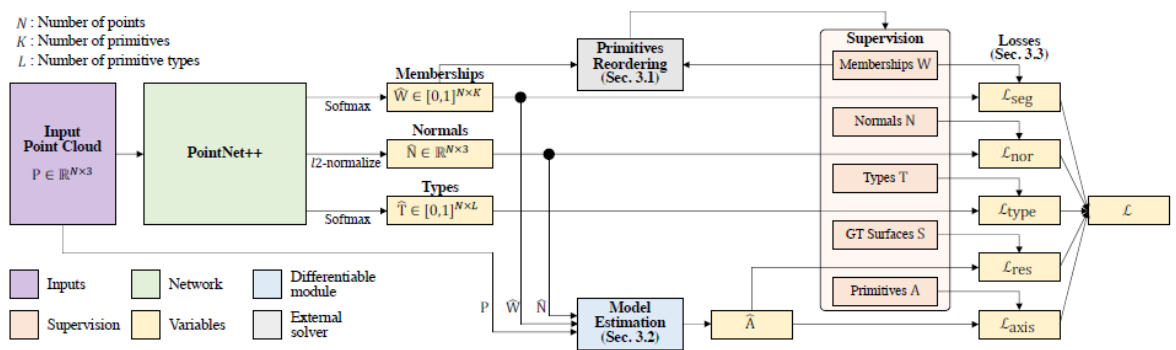


Рисунок 19. Архитектура SPFN

Ind	Method	Seg. (Mean IoU)	Primitive Type (%)
1	Eff. RANSAC [28]+J	43.68	52.92
2	Eff. RANSAC [28]*+J*	56.07	43.90
3	Eff. RANSAC [28]+J*	45.90	46.99
4	Eff. RANSAC [28]+J*+ \hat{W}	69.91	60.56
5	Eff. RANSAC [28]+J*+ \hat{W} + \hat{t}	60.68	92.76
6	Eff. RANSAC [28]+ \hat{N} + \hat{W} + \hat{t}	60.56	93.13
7	DPPN (Sec. 4.4)	44.05	51.33
8	SPFN- \mathcal{L}_{seg}	41.61	92.40
9	SPFN- \mathcal{L}_{norm} +J*	71.18	95.44
10	SPFN- \mathcal{L}_{res}	72.70	96.66
11	SPFN- \mathcal{L}_{axis}	77.31	96.47
12	SPFN ($\hat{t} \rightarrow \text{Est.}$)	75.71	95.95
13	SPFN	77.14	96.93

Рисунок 20. Метрика IoU в сравнении с RANSAC

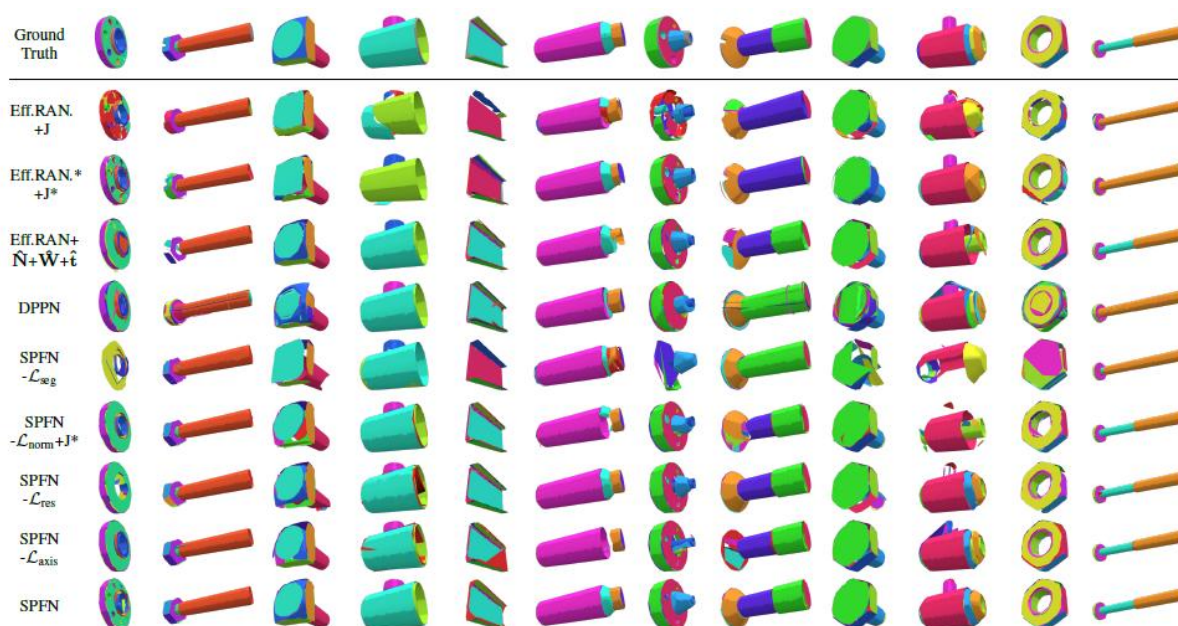


Рисунок 21. Результаты аппроксимации в сравнении с RANSAC

В сравнении с SPFN PointNet эффективнее работает в больших облаках точек и способен оперировать на целой сцене, в то время как PointNet хорош для сегментации и классификации трехмерных сканов объектов.

1.3. Интеллектуальные системы технического зрения в робототехнике

Система технического зрения в робототехнике — одна из важнейших частей всего программно-аппаратного комплекса робота, так как именно она позволяет наладить взаимодействие робота и окружающей среды.

В качестве аппаратных средств используются как стандартные цифровые RGB-камеры, стереокамеры и камеры с инфракрасным датчиком глубины. Также для достижения определенных целей, таких как устранение бликов и фильтрации цвета в промышленных решениях используют различные светофильтры и поляризаторы. Также с использованием камер глубины можно построить облако точек окружающей робота местности, что, например, позволяет промышленным манипуляторам определять способ

захвата целевого объекта, а мобильным просчитывать траекторию и способ перемещения, учитывая особенности рельефа.

Интеллектуальные системы технического зрения в мобильной робототехнике чаще всего применяются с использованием RGB-камер и нейросетевых моделей анализа изображений. Ярким примером является использование таких систем в беспилотных автомобилях, летательных аппаратах и космических аппаратах.

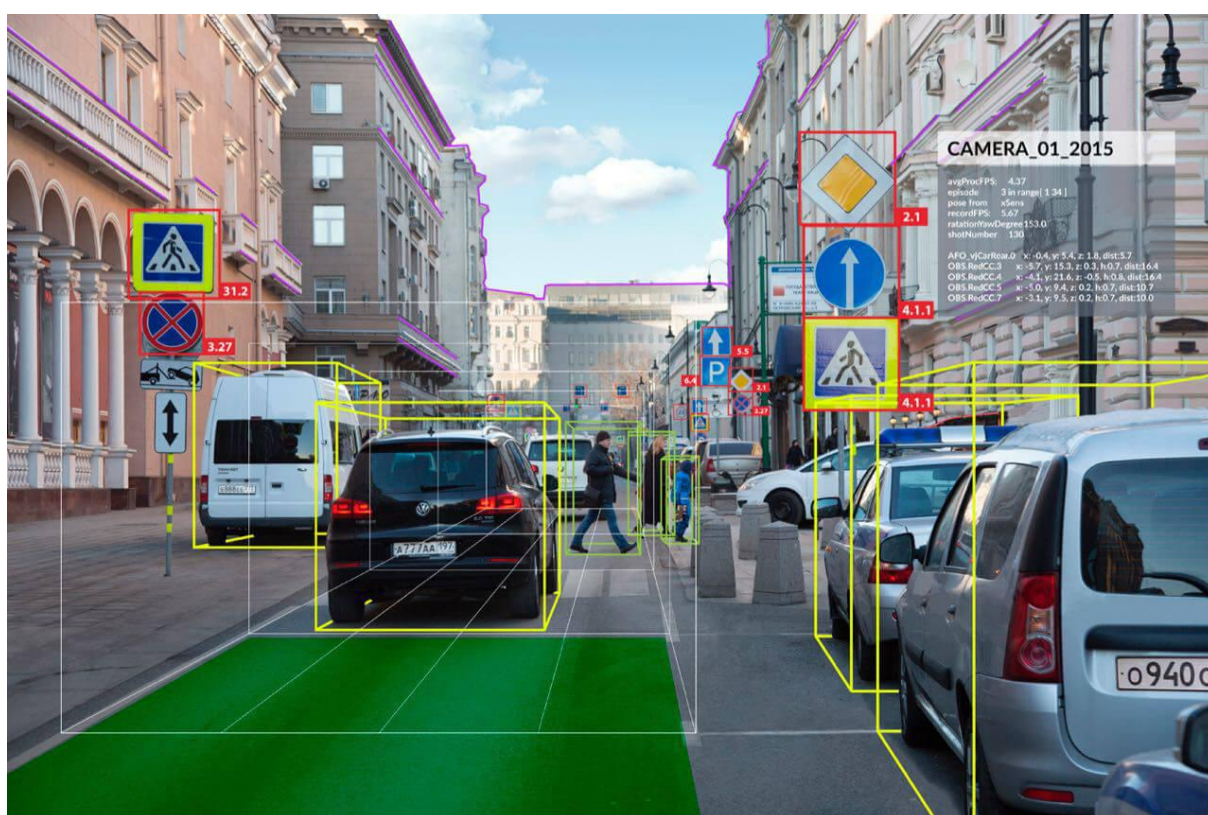


Рисунок 22. Пример использования интеллектуальной системы технического зрения в беспилотном автомобиле

В промышленной робототехнике также широко используют стереокамеры. Например, стерео сенсор Kuka_3D Perception распознает неструктурированное окружение в режиме реального времени и точно позиционирует объекты с разной кинематикой в пространстве. Робот

может фиксировать свое положение в пространстве с точностью до миллиметра и решать задачи еще быстрее и эффективнее. Благодаря интегрированной технологии распознавания KUKA_3D Perception обрабатывает данные изображения непосредственно в сенсоре и фиксирует текущее положение объекта с точностью до миллиметра. Камера оснащена встроенным графическим процессором, что позволяет запускать на ней нейросетевые модели, что также позволяет использовать ее в мобильной робототехнике.



Рисунок 23. Интеллектуальная камера KUKA_3D Perception

В ходе экспериментов при выполнении работы использовалась Intel RealSense, которая оснащена стереокамерой и инфракрасным датчиком, однако она не позволяет интегрировать интеллектуальные алгоритмы анализа и используется как сенсор.



Рисунок 24. Камера Intel RealSense

1.4. Заключение первой главы

Как показал обзор решений, интеллектуальные системы технического зрения широко применяются в современной робототехнике и активно развиваются. Использование нейросетевых моделей позволяет эффективно анализировать окружающую среду, однако такое решение не всегда является доступным для мобильных роботов и имеет высокую стоимость. Могут возникнуть сложности как со стороны аппаратного обеспечения, на котором должны работать алгоритмы, так и со стороны интеграции и масштабируемости решения. При решении некоторых задач также не всегда требуется настолько высокая производительность, поэтому использование аналитических и эвристических методов также не теряет своей актуальности.

2. Модели и алгоритмы нейро-эволюционной системы технического зрения

2.1 Архитектура нейро-эволюционной системы технического зрения

2.2. Выбор и настройка нейросетевой модели для оценки положения объектов в кадре

2.2.1 Принципы построения сверточных нейронных сетей

2.2.2 Разработанная архитектура модели

2.2.3 Процесс обучения модели

2.2.4 Оценка модели

2.3. Разработка эволюционного алгоритма для оценки формы объектов

2.3.1 Описание алгоритма RANSAC

2.3.1.1 Разработка математической модели примитива

2.3.1.2 Методика работы RANSAC

2.3.1.3 Перераспределение вероятностей выбора случайных точек

2.3.2 Описание генетического алгоритма

2.3.3 Оценка работы эволюционного алгоритма

2.4 Заключение второй главы

3. Разработка программного обеспечения для анализа изображения с бортовой камеры автономного робота

3.1 Структура программного обеспечения

3.2 Используемые инструменты

3.3 Порядок работы с программным обеспечением

4. Проведение экспериментальных исследований

4.1 Результаты работы эволюционного алгоритма

4.2 Результаты работы нейро-эволюционного алгоритма на данных с бортовой камеры

4.3 Заключение четвертой главы

5. Организационно-экономическая часть

Заключение

Используемая литература

Приложения

