

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет: информационных технологий и компьютерной безопасности

Кафедра: компьютерных интеллектуальных технологий проектирования

РЕФЕРАТ

По дисциплине: «Проектная деятельность»

Тема: «Определение нетиповых объектов по видеоряду»

Выполнил работу студент группы МИИВТ-231 _____ Котельников В.В.
подпись, дата

Принял: _____ Чижов М.И.
подпись, дата

Воронеж 2024

Введение

Поиск нетиповых объектов является частью проблемы поиска шаблонов в данных. Шаблон – это своего рода образец, по которому можно изготовить одинаковые изделия. Так вот шаблон данных является таким же образцом, по которому можно определять схожие или идентичные фрагменты данных. Данная проблема (проблема поиска шаблонов данных) имеет долгую историю и в большинстве своем успешную. Нахождение закономерностей в данных играют большую роль в научных открытиях. Например, обширные астрономические наблюдения Тихо Браге в 16-ом веке позволили Иоганну Кеплеру открыть эмпирические законы планетарного движения, которые в свою очередь, послужили плацдармом для развития классической механики. Решение проблемы определения нетиповых данных в различных областях, может существенно упростить, обезопасить и улучшить жизнь людей

Цель исследовательской работы

Моя цель состоит в изучение вопроса об определении нетиповых объектов, а также их поведение. Для изучения мной выбрана проблема воровства в супер маркетах, магазинах. Основной целью является определение нетиповых моделей поведения, действий, нахождение закономерностей в тех или иных действиях, за которыми следуют кражи. Данное исследование поможет улучшить борьбу с нарушителями закона, облегчат человеческий труд в поиске нарушителей, ускорят поиск и главное помогут вовремя предотвращать преступления.

Задачи для достижения целей исследовательской работы:

- 1) Изучение методов определения объектов на изображении и по видеоряду
- 2) Сравнение методов и выбор оптимального для исследования
- 3) Проектирование архитектуры ПО
- 4) Реализация (программирование ПО)
- 5) Тестирование ПО

1. Изучение методов определения объектов на изображение и по видеоряду

1.1. Определение объекта с помощью поиска контуров

Поиск контуров – включает в себя набор различных математических методов, которые позволяют выявлять точки в цифровом изображении, в которых происходят резкие изменения яркости. Данные точки обычно организованы в виде набора кривых линий и называются краями, границами или контурами.

Чаще всего резкое изменение яркости изображения может соответствовать: различным материалам, различию в освещении различных частей сцены, перепадам глубины или изменению ориентации поверхности. Определение краев помогает установить границы и форму объекта.

Выделенные края могут быть двух типов: независящие и зависящие от точки обзора. Таким образом, независящие границы отображают такие свойства, как цвет и форма поверхности, а зависящие могут меняться в разных точках обзора и отображать геометрию сцены.

Существуют различные методы выделения краев:

1) Детектор границ Канни – данный алгоритм состоит из пяти отдельных шагов:

- сглаживание: размытие изображения для удаления шума;
- поиск градиентов: границы отмечаются там, где градиент изображения приобретает максимальное значение;
- подавление не максимумов: только локальные максимумы отмечаются как границы;
- двойная пороговая фильтрация: потенциальные границы определяются порогами;
- трассировка области неоднозначности: итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами.

2) Оператор Собеля – Данный оператор базируется на приближении значений градиента яркости изображения. Он вычисляет градиент яркости в каждой точке на изображении, тем самым находит величину изменения яркости и ее направление. Результат показывает изменения яркости изображения в каждой точке, т.е. вероятность нахождения точки на границы, а также ориентацию.

Можно сказать, что результат оператора в точке, которая находится в постоянной яркости – это нулевой вектор. А точки, находящиеся в границе с различной яркостью – это вектор, который пересекает границу в направлении возрастания яркости. Оператор Собеля использует фильтрацию изображения на основе свертки по горизонтали и вертикали, поэтому он легко вычисляется. Оператор использует матрицу 3×3 , благодаря которым свертывают исходное изображение для дальнейшего вычисления приближенных производных по горизонтальным и вертикальным направлениям.

3) Векторный оператор Лапласа - Алгоритм Лапласа основан на поиске нулей. Он использует производные второго порядка. В отличие от алгоритмов, использующих градиентный подход для выделения границ. Лапласиан является скалярной функцией. Он находит применение для выделения границ следующим образом, лапласиан принимает максимальное значение на участках «перегибов» функции яркости. Из минусов данного алгоритма следует указать раздваивание границ, если она недостаточно резкая, т.е. имеются участки постоянного наклона. В этом случае поможет специальная обработка для устранения раздваивания линий.

4) Оператор Прюитт - Данный алгоритм работает по принципу определения максимального отклика на множестве используемой матрицы для обнаружения локальной ориентации границ в каждом пикселе. Для этого используются различные матрицы. Из одной матрицы можно получить восемь, переставляя ее коэффициенты. Максимальное значение каждого пикселя – это пиксель в полученном изображении. Его значения могут быть в

границах от 1 до 8, в зависимости от того, какая матрица дала наибольший результат. Этот метод еще называют подстановкой шаблонов границ. Изображению сопоставляется набор шаблонов, каждый из которых показывает расположение границ. Тогда расположение границы в пикселе формируется шаблоном, который больше всех соответствует близлежащему в окрестности пикселю.

5) Перекрестный оператор Робертса - Является одним из первых алгоритмов выделения границ. В данном методе используется суммарный вектор из двух диагональных векторов. Данный вектор показывает наибольшую разницу градиентов между охваченными точками. А направление данного вектора будет указывать наибольшую величину перепада между точками. Оператор Робертса чаще всего используют благодаря его быстрым вычислениям, но по качеству он проигрывает альтернативным алгоритмам из-за чувствительности к шуму.

1.2. Определение объекта при помощи каскадных классификаторов

Каскадирование является частным случаем ансамблевого обучения, основанного на объединении нескольких классификаторов, используя всю информацию, собранную с выхода из данного классификатора, в качестве дополнительной информации для следующего классификатора в каскаде. В отличие от ансамблей для голосования или штабелирования, которые являются многоэкспертными системами, каскадирование является многоступенчатым. Каскадные классификаторы обучаются с несколькими сотнями «положительных» образцов отдельных объектов и произвольными «негативными» изображениями того же размера. После обучения классификатора его можно применить к области изображения и определить объект, о котором идет речь. Для поиска объекта во всем кадре окно поиска перемещается по всему изображению с некоторым нахлестом. Этот процесс чаще всего используется при обработке изображений для обнаружения и

отслеживания объектов, в первую очередь для обнаружения и распознавания лиц. Ансамблевые методы — это набор слабых классификаторов (под слабостью классификатора подразумевается, что его ошибка на обучение выборки менее 50%, но более 0%). Объединяя их предсказания можно достичь более высокой точности классификации объектов из тестовой выборки.

1.3. Метод Виолы-Джонса

Метод Виолы-Джонса — это алгоритм, позволяющий обнаруживать объекты на изображении. Алгоритм может распознавать различные классы изображений, но основная его задача — обнаружение лиц. В библиотеке OpenCV он реализуется функцией (`cvHaarDetectObjects()`). С задачей обнаружения лиц на изображении человек справится легко, но компьютер нуждается в четких инструкциях и ограничениях. Чтобы сделать задачу более решаемой, метод Виолы-Джонса требует полного просмотра фронтальных вертикальных лиц. Таким образом, чтобы быть обнаруженным, все лицо должно смотреть на камеру и не должно быть наклонено ни в одну сторону. Хотя кажется, что эти ограничения могут немного уменьшить полезность алгоритма, поскольку шаг обнаружения чаще всего сопровождается шагом распознавания, на практике эти ограничения на позу вполне приемлемы. Принцип работы заключается в использовании изображения в интегральном представлении. (интегральное представление изображения - это матрица, размерность которой совпадает с размерами исходного изображения), что позволяет быстро вычислить необходимые объекты.

1.4. Метод Хафа

Преобразование Хафа — этот метод используется для поиска геометрических фигур на изображениях или видеоряде. На основе нужной

фигуры выбирается определенное количество параметров, необходимых для определения формы. Например, для прямой линии $y = mx + c$ существует два параметра. После чего находится значение параметров для каждого пикселя. Для конкретной прямой должна быть пара уникальных параметров. Следовательно, все точки на этой линии будут иметь одну и ту же пару параметров. Далее происходит голосование за каждую пару параметров, полученных из пикселей изображения. Пара с наибольшим количеством голосов соответствует прямой линии. Бывает так, что на изображение несколько линий, например, 10, следовательно, 10 точек получивших наибольшее количество голосов, представляют эти линии. Затем эти линии строятся с использованием параметров. Этот способ подходит для определения простых фигур.

1.5. Методы сегментации

Сегментация изображения представляет собой разбиение изображения на множество покрывающих его областей. Сегментация применяется во многих областях, таких как производство для индикации дефектов при сборке деталей, медицине для первичной обработки снимков.

1) Метод сегментации по водоразделам (WaterShed)

Алгоритм работает с изображением как с функцией от двух переменных $f = I(x, y)$, где x, y - координаты пикселя. Значением функции может являться интенсивность или модуль градиента. Для лучшего контраста можно взять градиент от изображения. Если по оси OZ откладывать абсолютное значение градиента, то в местах перепада интенсивности будут образовываться хребты, а в однородных регионах – равнины. После нахождения минимумов функции, начинается процесс заполнения «водой», который начинается с глобального минимума. Как только уровень воды достигает значения очередного локального минимума и начинается заполнения его водой. Когда два региона начинают сливаться, строится перегородка, чтобы предотвратить объединение областей. Вода продолжит

заполнять до тех пор, пока регионы не будут отделяться только искусственно построенными перегородками.

Данный алгоритм может быть полезным, если на изображение небольшое число локальных минимумов, в случае же их большого количества будет возникать избыточное разбиение на сегменты.

2) Метод сегментации MeanShift

Данный метод группирует объекты с близкими признаками. Пиксели со схожими признаками объединяются в один сегмент, на выходе получается изображение с однородными областями. В качестве координат в пространстве признаков можно выбрать координаты пикселя (x, y) компоненты *RGB* пикселя. Если изобразить пиксели в пространстве признаков, будет видно сгущение признаков в определенных местах.

При выборе в качестве признаков координат пикселей и интенсивностей по цветам в один сегмент будут объединяться пиксели с близкими цветами и недалеким расположением друг от друга. Следовательно, если выбрать другой вектор признаков, то объединение пикселей в сегменты будет идти по данному вектору. Например, если убрать из признаков координаты, то небо и озеро будут считаться одним сегментом, так как с наибольшей вероятностью пиксели этих объектов в пространстве признаков попали бы в один локальный максимум

Если объект, который нужно выделить, состоит из областей, сильно различающихся по цвету, то данный метод не сможет объединить эти регионы в один, и данный объект будет состоять из нескольких сегментов. Но зато данный метод хорошо справиться с однородным по цвету предметом на пестром фоне. Ещё метод MeanShift часто используют при реализации функции слежения за движущимися объектами.

3) Метод сегментации FloodFill

С помощью FloodFill (заливка или метод «наводнения») можно выделить однородные по цвету регионы. Для этого нужно выбрать

начальный пиксель и задать интервал изменения цвета соседних пикселей относительно исходного. Интервал может быть и несимметричным. Алгоритм будет объединять пиксели в один сегмент (заливая их одним цветом), если они попадают в указанный диапазон. На выходе будет сегмент, залитый определенным цветом, и его площадь в пикселях.

Такой алгоритм может быть полезен для заливки области со слабыми перепадами цвета однородным фоном. Одним из вариантов использования FloodFill может быть выявление поврежденных краев объекта. Например, если, заливая однородные области определенным цветом, алгоритм заполнит и соседние регионы, то значит нарушена целостность границы между этими областями.

4) Метод сегментации GrabCut

Это интерактивный алгоритм выделения объекта, разрабатывался как более удобная альтернатива магнитному лассо (чтобы выделить объект, пользователю требовалось обвести его контур с помощью мыши). Для работы алгоритма достаточно заключить объект вместе с частью фона в прямоугольник (grab). Сегментирование объекта произойдет автоматически (cut).

Могут возникнуть сложности при сегментации, если внутри ограничивающего прямоугольника присутствуют цвета, которые встречаются в большом количестве не только в объекте, но и на фоне. В этом случае можно поставить дополнительные метки объекта (красная линия) и фона (синяя линия).

1.6. Методы глубокого обучения (Deep Learning)

Глубокое обучение - это область машинного обучения, которая использует искусственные нейронные сети с несколькими слоями для моделирования и решения сложных проблем. Алгоритмы глубокого обучения изучают представления данных через иерархическую структуру слоев, где каждый слой извлекает все более абстрактные и сложные характеристики из входных данных, позволяя сети делать точные прогнозы или классификации.

Глубокое обучение приобрело популярность в последние годы благодаря своей выдающейся производительности в различных приложениях, включая распознавание изображений, обработку естественного языка, распознавание речи и многие другие. Оно произвело революцию в таких областях, как компьютерное зрение, где модели глубокого обучения достигли сверхчеловеческой точности в таких задачах, как обнаружение объектов, сегментация и классификация. В обработке естественного языка модели глубокого обучения используются для классификации текстов, анализа настроений, машинного перевода и генерации языка.

1) Сети прямого распространения

Нейронные сети с прямолинейным движением состоят из входного, скрытого и выходного слоев нейронов. Входной слой представляет особенности входных данных, а скрытые слои выполняют вычисления над входными сигналами для получения конечного результата.

Каждый нейрон в нейронной сети с прямой передачей применяет функцию активации к взвешенной сумме своих входов для получения выходного сигнала. Обычные функции активации включают сигмоидную функцию, функцию выпрямленной линейной единицы (ReLU) и функцию гиперболического тангенса.

Нейронные сети с прямолинейным движением обучаются с помощью обратного распространения - алгоритма, который вычисляет градиент

функции потерь относительно весов сети и обновляет веса с помощью градиентного спуска.

2) Рекуррентные нейронные сети

РНС имеют контуры обратной связи, которые позволяют им поддерживать внутреннее состояние или память о предыдущих входах. Вход на каждом временном шаге обрабатывается рекуррентным слоем, а выход слоя подается обратно на следующий временной шаг в качестве входа.

РНС могут страдать от проблемы исчезающего градиента при обучении на длинных последовательностях, когда градиенты становятся слишком малы для эффективного обновления весов. Эта проблема может быть решена с помощью LSTM-сетей.

РНС используются для решения таких задач, как моделирование языка, машинный перевод и распознавание речи.

3) Сети с долговременной кратковременной памятью

LSTM используют механизм стробирования для избирательного забывания или запоминания информации из предыдущих входов. Ячейка LSTM состоит из ячейки памяти, входных ворот, выходных ворот и ворот забывания.

LSTM обучаются с помощью обратного распространения во времени, что является расширением обратного распространения, которое может работать с последовательностями различной длины.

LSTM используются для таких задач, как распознавание речи, моделирование языка и распознавание рукописного текста.

4) Генеративно-состязательные сети

GAN состоят из двух нейронных сетей: сети-генератора, которая производит поддельные образцы, и сети-дискриминатора, которая пытается отличить настоящие образцы от поддельных.

GAN обучаются в состязательной среде, где генератор пытается обмануть дискриминатор, а дискриминатор пытается обнаружить

поддельные образцы. Процесс обучения включает в себя обновление весов генератора и дискриминатора по очереди.

GAN используются для решения таких задач, как синтез изображений, генерация музыки и предсказание видео.

Существует также несколько методов обучения нейронных сетей:

1) Контролируемое обучение - это наиболее часто используемый метод обучения нейронных сетей, при котором модель обучается на наборе помеченных данных, чтобы делать прогнозы на новых данных. Наиболее популярным алгоритмом, используемым в контролируемом обучении, является алгоритм обратного распространения, который вычисляет ошибку на выходе и распространяет ее обратно через слои сети для корректировки весов.

2) Стохастический градиентный спуск (SGD) - это популярный алгоритм оптимизации, используемый в глубоком обучении. Он обновляет параметры модели в направлении отрицательного градиента функции потерь, что делает его эффективным при обучении крупномасштабных нейронных сетей.

3) Алгоритм Adam - это вариант стохастического градиентного спуска, который адаптирует скорость обучения для каждого веса в сети, что делает его более эффективным по сравнению с другими алгоритмами оптимизации. Он сочетает в себе преимущества алгоритмов стохастического градиентного спуска и оптимизации на основе импульса.

4) Неподконтрольное обучение используется, когда нет доступных маркированных данных, и модель должна изучить основные закономерности и структуры в данных. Автоэнкодеры являются одним из наиболее часто используемых алгоритмов обучения без наблюдения в глубоком обучении.

5) Вариационные автоэнкодеры - это тип автоэнкодера, который изучает вероятностное распределение данных и может генерировать новые образцы данных путем выборки из изученного распределения.

6) Генеративные модели используются для генерации новых образцов данных, которые похожи на обучающие данные.

Вариационные автоэнкодеры и генеративные адверсарные сети (GAN) - наиболее часто используемые генеративные модели в глубоком обучении.

7) Обучение с подкреплением используется для обучения моделей принятию решений в динамической среде. Модель обучается, получая обратную связь в виде вознаграждений или штрафов за свои действия. Q-learning и Deep Q-Networks (DQNs) являются широко используемыми алгоритмами обучения с подкреплением в глубоком обучении.

8) Метод Actor-Critic - это комбинация обучения с подкреплением и глубокого обучения. Модель учится предсказывать ценность своих действий (Критик), а также учится предпринимать действия, которые максимизируют предсказанную ценность (Актер).

1.7. Обучение без учителя

Обучение без учителя (unsupervised learning, неконтролируемое обучение) – класс методов машинного обучения для поиска шаблонов в наборе данных. Данные, получаемые на вход таких алгоритмов обычно не размечены, то есть передаются только входные переменные X без соответствующих меток y . Если в контролируемом обучении (обучении с учителем, supervised learning) система пытается извлечь уроки из предыдущих примеров, то в обучении без учителя – система старается самостоятельно найти шаблоны

1) Метод k-средних

Кластеризация K-средних представляет собой одну из самых простых реализаций, суть которой заключается в итеративной инициализации центроидов для каждого кластера на основе среднего арифметического расположенных в нём наблюдений, а также их переопределении путём минимизации суммарного квадратичного отклонения от центроидов кластеров.

Существуют различные вариации алгоритма K-Means, которые модифицируют его шаги или функцию потерь для улучшения производительности, а также применимости к разным типам данных. К самым популярным вариациям относятся следующие:

- Lloyd's algorithm — это классический вариант K-Means, который хорошо работает для сферических кластеров с одинаковой плотностью, но может давать плохие результаты для других форм или размеров кластеров.

- Elkan algorithm — более быстрый вариант классического K-Means, который использует неравенство треугольника для уменьшения количества вычислений расстояний между объектами и центроидами, что может быть эффективнее на некоторых наборах данных с хорошо определёнными кластерами, однако требуется больше памяти из-за выделения дополнительного массива размера $(n_samples, n_clusters)$.

- Mini-batch K-Means — модификация классического K-Means, использующая случайные подвыборки данных на каждой итерации для обучения. Хорошо подходит для больших датасетов.

- K-Medoids — вариант K-Means, который в качестве центроидов выбирает реальные точки (медоиды) из данных, а не их средние значения, что повышает устойчивость к выбросам.

- K-Modes — вариант алгоритма K-Means для работы с категориальными данными, который выбирает один из объектов в кластере в качестве моды и минимизирует сумму расстояний Хэмминга между модой и

объектами в кластере. Расстояние Хэмминга представляет из себя количество позиций, в которых значения векторов не совпадают.

2) Affinity Propagation

Алгоритм Affinity Propagation (AP) — это алгоритм кластеризации, основанный на передаче сообщений между данными. В отличие от других методов кластеризации, таких как К-средних, AP не требует предварительного задания числа кластеров. Вместо этого он автоматически определяет количество кластеров на основе входных данных.

Вот основные шаги алгоритма Affinity Propagation:

- подготовка: Вычисляется матрица сходства для всех пар точек данных. Элементы этой матрицы представляют собой меру сходства между парами точек;
- отправка сообщений: Алгоритм итеративно обменивается двумя типами сообщений между парами точек — "responsibility" (ответственность) и "availability" (доступность);
- выбор примеров: После сходимости каждая точка данных выбирает другую точку в качестве примера, и точки, выбравшие себя, становятся центрами кластеров;
- формирование кластеров: Все точки данных присоединяются к кластеру своего примера, формируя окончательные кластеры.

Преимущества AP включают способность работать с данными различной природы и возможность нахождения кластеров с различными размерами. Однако алгоритм может быть вычислительно затратным на больших наборах данных и требует тщательного выбора параметра предпочтения, который влияет на количество кластеров.

Алгоритм Affinity Propagation используется в различных приложениях, где необходимо эффективно разделить данные на группы, например, в биоинформатике для кластеризации генов, в компьютерном зрении для сегментации изображений и в социальных науках для выявления групп людей с похожими интересами.

3) Агломеративная кластеризация

Агломеративная кластеризация — это тип иерархической кластеризации, используемый для группировки объектов в кластеры на основе их сходства. Этот метод строит иерархию кластеров шаг за шагом, объединяя на каждом этапе наиболее близкие кластеры или точки. Вот подробное описание алгоритма:

- инициализация: Начинаем с того, что каждая точка данных представляет собой отдельный кластер. Таким образом, если у нас есть N точек данных, мы начинаем с N кластеров;
- вычисление матрицы расстояний: Создается матрица расстояний, где каждый элемент матрицы представляет расстояние между парами кластеров. Расстояние может быть измерено различными способами, например, методом ближайшего соседа, методом дальнего соседа, методом среднего расстояния или методом Варда;
- объединение кластеров: На каждом шаге алгоритма два кластера, которые находятся на наименьшем расстоянии друг от друга, объединяются в один кластер. Это уменьшает общее количество кластеров на один;
- обновление матрицы расстояний: После объединения кластеров матрица расстояний обновляется, чтобы отразить расстояния между новым кластером и оставшимися кластерами;
- повторение: Шаги 3 и 4 повторяются до тех пор, пока все точки данных не будут объединены в один кластер, или до достижения заранее определенного числа кластеров;
- построение дендрограммы: В процессе агломерации можно построить дендрограмму, которая визуальнo показывает, как точки данных объединяются в кластеры на различных уровнях иерархии.

Агломеративная кластеризация широко используется в различных областях, таких как биоинформатика для группировки генов или белков,

маркетинг для сегментации клиентов, социальные науки для классификации индивидуумов по группам и многие другие. Одним из преимуществ этого метода является его визуальная интерпретируемость и гибкость в выборе метрики расстояния. Однако он может быть вычислительно сложным для больших наборов данных и чувствителен к выбору метрики расстояния.

4) Спектральная кластеризация

Спектральная кластеризация — это алгоритм кластеризации, который использует спектр (собственные значения) матрицы сходства для уменьшения размерности пространства перед кластеризацией в меньшем количестве измерений. Этот метод особенно полезен, когда структура кластеров может быть выявлена только в нелинейном пространстве.

Принцип работы алгоритма Спектральной кластеризации:

- формирование матрицы сходства: Сначала создается матрица сходства S , которая отражает степень сходства между каждой парой точек в данных. Обычно это делается с использованием гауссовского (радиально-базисного) ядра;
- стандартизация матрицы: Для улучшения качества кластеризации матрица сходства может быть преобразована в лапласиан L путем вычитания из диагональной матрицы степеней D матрицу сходства;
- вычисление собственных значений и векторов: Рассчитываются собственные значения и собственные векторы матрицы лапласиана L . Собственные векторы, соответствующие наименьшим собственным значениям (исключая нулевое), используются для дальнейшей кластеризации;
- формирование нового пространства признаков: Собственные векторы, соответствующие k наименьшим собственным значениям, формируют новое пространство признаков, где k — это количество кластеров, на которое мы хотим разделить данные;

- кластеризация в новом пространстве: На последнем этапе применяется алгоритм кластеризации, например К-средних, к данным в новом пространстве признаков для формирования окончательных кластеров.

Преимущества спектральной кластеризации включают ее способность обнаруживать сложные структуры кластеров, которые могут быть нелинейно разделимыми, и ее относительную устойчивость к шуму и выбросам. Однако алгоритм может быть вычислительно сложным из-за необходимости вычисления собственных значений и векторов больших матриц, и его производительность сильно зависит от выбора параметров, таких как выбор ядра и его параметров.

Спектральная кластеризация широко применяется в областях, где данные имеют сложную структуру, включая обработку изображений, биоинформатику и распознавание речи.

1.8. Методы оптического потока

Оптический поток (ОП) – изображение видимого движения, представляющее собой сдвиг каждой точки между двумя изображениями. По сути, он представляет собой поле скоростей (т. к. сдвиг с точностью до масштаба эквивалентен мгновенной скорости). Суть ОП в том, что для каждой точки изображения $I_1(x, y)$ находится такой сдвиг (dx, dy) , чтобы исходной точке соответствовала точка на втором изображении $I_2(x + dx, y + dy)$. Как определить соответствие точек – отдельный вопрос. Для этого надо взять какую-то функцию точки, которая не изменяется в результате смещения. Обычно считается, что у точки сохраняется интенсивность (т. е. яркость или цвет для цветных изображений), но можно считать одинаковыми точки, у которых сохраняется величина градиента, гессиан, его величина или его определитель, лапласиан, другие характеристики. Очевидно, сохранение интенсивности дает сбои, если меняется освещенность или угол падения света. Тем не менее, если речь идет о видеопотоке, то, скорее всего, между двумя кадрами освещение сильно не изменится, хотя бы потому, что между

ними проходит малый промежуток времени. Поэтому часто используют интенсивность в качестве функции, сохраняющейся у точки.

1) Метод Лукаса-Канаду

Метод Лукаса-Канаде — это популярный алгоритм в компьютерном зрении для отслеживания движения объектов между двумя последовательными кадрами изображения. Он основан на идее, что поток (движение) пикселей между двумя кадрами является постоянным в небольшом окне. Вот подробное описание метода:

- выбор точек интереса: Сначала выбираются точки интереса на первом изображении, которые хорошо отслеживаются, например, углы или яркие пятна;

- вычисление градиентов: Для каждой точки интереса вычисляются градиенты интенсивности изображения по горизонтали и вертикали. Эти градиенты показывают, как интенсивность изображения изменяется вокруг точки;

- оценка оптического потока: Затем решается система уравнений, которая минимизирует разницу в интенсивности между соответствующими точками на двух последовательных кадрах. Это делается с использованием метода наименьших квадратов;

- вычисление векторов движения: Решение системы уравнений дает вектор движения для каждой точки интереса, который указывает, как и куда она переместилась между кадрами;

- итерационное уточнение: Если необходимо, процесс может быть повторен итерационно, чтобы уточнить оценку движения.

Метод Лукаса-Канаде широко используется для задач, связанных с отслеживанием движения, стабилизацией видео и 3D-реконструкцией. Он эффективен, когда движение между кадрами невелико, и предполагает, что движение является гладким на протяжении небольшого окна. Однако метод может не справляться с большими движениями или быстрыми изменениями освещения.

2) Метод Horn–Schunck

Метод Horn–Schunck — это классический алгоритм для определения оптического потока, который был предложен Бертраном Хорном и Брайаном Шунком в 1981 году. Оптический поток — это распределение видимых движений объектов, поверхностей и краев в визуальной сцене, вызванное относительным движением между наблюдателем и сценой. Метод Horn–Schunck основан на двух ключевых предположениях:

- яркость остается постоянной: Предполагается, что яркость пикселя остается постоянной во времени, то есть изменения в яркости изображения вызваны только движением;

- гладкость потока: Предполагается, что оптический поток медленно меняется по сцене, за исключением краев объектов;

Описание алгоритма:

- вычисление градиентов: Для каждого пикселя изображения вычисляются градиенты яркости по времени и пространству. Это включает в себя градиенты по горизонтали;

- инициализация: Векторы оптического потока инициализируются нулевыми значениями на всем изображении;

- итерационный процесс: На каждой итерации для каждого пикселя решается уравнение оптического потока, которое минимизирует разницу в яркости и обеспечивает гладкость потока. Это достигается путем минимизации функции стоимости, которая включает в себя данные и гладкость;

- решение уравнений: Система уравнений решается с использованием метода релаксации, который обновляет значения оптического потока на основе средних значений соседних пикселей;

- повторение до сходимости: Процесс повторяется до тех пор, пока изменения в оптическом потоке не станут незначительными, что указывает на сходимость алгоритма.