Содержание

О редакторах

Рой Дэвис – почетный профессор факультета машинного зрения в Роял Холлоуэй, Лондонский университет. Он работал над многими аспектами зрения, от обнаружения признаков и подавления шума до робастного сопоставления образов и реализации практических задач зрения в реальном времени. Область его интересов включает автоматизированный осмотр объектов, наблюдение, управление транспортными средствами и раскрытие преступлений. Он опубликовал более 200 статей и три книги: *Machine Vision: Theory, Algorithms, Practicalities* (1990 г.), *Electronics, Noise and Signal Recovery* (1993 г.) и *Image Processing for the Food Industry* (2000 г.); первая из них не теряет популярности на протяжении 25 лет, а в 2017 г. вышло ее значительно расширенное пятое издание под названием *Computer Vision: Principles, Algorithms, Applications, Learning*. Рой является членом IoP и IET, а также старейшим членом IEEE. Он входит в редакционные коллегии журналов *Pattern Recognition Letters*, *Real-Time Image Processing*, *Imaging Science and IET Image Processing*. Он получил степень доктора наук в Лондонском университете; в 2005 г. был удостоен титула почетного члена BMVA, а в 2008 г. стал лауреатом премии Международной ассоциации распознавания образов.

Мэтью Тёрк – президент Технологического института Toyota в Чикаго (TTIC) и почетный профессор Калифорнийского университета в Санта-Барбаре. Его исследовательские интересы охватывают компьютерное зрение и взаимодействие человека с компьютером, включая такие темы, как автономные транспортные средства, распознавание лиц и жестов, мультимодальное взаимодействие, компьютерная фотография, дополненная и виртуальная реальность и этика ИИ. Он был главным организатором или ведущим нескольких крупных конференций, включая конференцию IEEE по компьютерному зрению и распознаванию образов, мультимедийную конференцию ACM, конференцию IEEE по автоматическому распознаванию лиц и жестов, международную конференцию ACM по мультимодальному взаимодействию и Зимнюю конференцию IEEE по приложениям компьютерного зрения. Он получил несколько наград за лучшую исследовательскую работу, а также различные премии и награды ACM, IEEE, IAPR и почетную премию Фулбрайта-Nokia за 2011–2012 гг. в области информационных и коммуникационных технологий.

Предисловие

Миновало почти десятилетие с тех пор, как произошел прорыв в разработке и применении *глубоких нейронных сетей* (deep neural network, DNN), и их последующий прогресс можно почти без преувеличения назвать выдающимся. Правда, этому прогрессу значительно способствовало появление специального оборудования в виде мощных графических процессоров; кроме того, возникло понимание, что *сверточные нейронные сети* (convolutional neural network, CNN) составляют важнейшую архитектурную основу, в которую можно встроить такие функции, как ReLU, упаковку, полностью связанные слои, распаковку и обратную свертку. По сути, все эти подходы помогли вдохнуть реальную жизнь в глубокие нейросети и резко расширить возможности их использования, поэтому первоначальный почти экспоненциальный рост их использования сохранился на весь последующий период. Мало того, что мощь нейросетевых технологий была впечатляющей, их применение значительно расширилось: от первоначального акцента на быстрое определение местоположения объекта и сегментацию изображения – и даже семантическую сегментацию – до применений, относящихся к видео, а не просто к анализу статичного изображения.

Было бы неправильно утверждать, что все развитие компьютерного зрения с 2012 г. было связано исключительно с появлением DNN. Свою роль сыграли и другие важные методы, такие как обучение с подкреплением, обучение с переносом, самообучение, лингвистическое описание изображений, распространение меток и такие приложения, как обнаружение новизны и аномалий, раскрашивание и отслеживание изображений. Тем не менее многие из упомянутых методов и области их применения получили новые стимулы и были пересмотрены и улучшены благодаря чрезвычайно быстрому внедрению DNN.

В этой книге мы попытались оценить, какие изменения произошли в области компьютерного зрения за минувшее десятилетие, насыщенное драматическими переменами. Сейчас самое время задаться вопросом, где мы находимся сейчас и насколько прочна база глубокого нейронного и машинного обучения, на которую опирается современное компьютерное зрение. Было ли это продуманное последовательное движение или слепой отчаянный рывок вперед? Не упускаем ли мы важные возможности и можем ли мы заглядывать в будущее с уверенностью, что движемся в правильном направлении? Или это тот случай, когда каждый исследователь может придерживаться своей собственной точки зрения и обращать внимание только на то, что представляется необходимым для его прикладной области, и если это так, то не ускользает ли от нас что-то важное при столь ограниченном подходе?

На самом деле есть и другие фундаментальные вопросы, на которые нужно найти ответ. Например, это сложный вопрос о том, до какой степени возможности глубокой нейросети можно повышать за счет качества обучающих данных; этот вопрос, по-видимому, применим к любому альтернативному

подходу, основанному на машинном обучении, независимо от того, относится ли он к DNN. Вряд ли фундаментальные ограничения нейросети зависят от того, каким способом ее обучали – обучение с подкреплением, самообучение или что-то другое. И обратите внимание, что люди вряд ли являются примером того, что можно каким-либо образом избежать интенсивного обучения; их способность к обучению с переносом лишь подтверждает, насколько эффективным может быть процесс обучения.

В этой книге мы стремимся не только представить передовые методики и подходы в области компьютерного зрения, но и разъяснить основополагающие принципы; мы взяли на себя роль преподавателей и, прежде чем представить читателю самые последние достижения, хотим сформировать у него понимание общей картины. Поэтому *глава 1* посвящена основам компьютерного зрения. Она начинается с детального анализа ранних подходов к компьютерному зрению, включая обнаружение признаков, обнаружение объектов, трехмерное зрение и появление DNN; далее мы переходим к визуальному слежению за объектами, которое рассматривается как пример прикладной области, где решающую роль могут играть DNN. Эта глава самая длинная в книге, потому что мы должны пройти путь с нуля до современных достижений; кроме того, она готовит почву для понимания ключевых идей и методов, описанных выдающимися экспертами в остальных главах.

Как будет показано в *главе 1*, обнаружение объектов – одна из самых сложных задач компьютерного зрения. В частности, эффективная система должна преодолевать такие проблемы, как искажение масштаба, окклюзия, переменное освещение, сложный фон и все факторы изменчивости, связанные с миром природы. *Глава 2* описывает различные методы и подходы, на которых основаны последние достижения. К ним относятся *слияние видимых областей* (region-of-interest pooling), *многозадачные потери* (multitask losses), *сети для предложения регионов* (region proposal networks), *привязки* (anchors), *каскадное обнаружение и регрессия* (cascaded detection and regression), *многомасштабные представления признаков* (multiscale feature representations), *методы дополнения данных* (data augmentation techniques), *функции потерь* (loss functions) и многое другое.

В *главе 3* подчеркивается, что недавние успехи в области компьютерного зрения в значительной степени связаны с появлением огромных массивов тщательно размеченных данных, необходимых для обучения моделей. В ней рассматриваются методы, которые можно использовать для обучения моделей распознавания на основе таких данных, требующие ограниченной ручной обработки. Помимо уменьшения количества размеченных вручную данных, необходимых для обучения моделей распознавания, необходимо снизить уровень подкрепления с сильного на слабый, в то же время разрешая релевантные запросы от оракула. Дан обзор теоретических основ и экспериментальных результатов, которые помогают достичь этого.

В *главе 4* рассматриваются вычислительные проблемы глубоких нейронных сетей, которые затрудняют их развертывание на оборудовании с ограниченными ресурсами. В ней обсуждаются методы сжатия моделей и поиска нейронной архитектуры, ориентированной на оборудование, с целью повышения эффективности глубокого обучения, уменьшения размера и ускоре-

ния нейронных сетей. В главе показано, как использовать *отсечение коэффициентов* (parameter pruning) для удаления избыточных весов, *факторизацию низкого ранга* (low-rank factorization) для уменьшения сложности, *квантование весов* (weight quantization) для уменьшения точности весов и размера модели, а также *дистилляцию знаний* (knowledge distillation) для переноса знаний из «черного ящика» бóльших моделей в меньшие.

В главе 5 обсуждается, как *глубокие генеративные модели* (deep generative models) пытаются восстановить низкоразмерную структуру целевых визуальных моделей. В ней показано, как использовать глубокие генеративные модели для достижения более управляемого синтеза визуальных паттернов посредством условной генерации изображения. Ключом к достижению этой цели является «распутывание» визуального представления, когда предпринимаются попытки разделить различные управляющие факторы в скрытом пространстве встраивания. Представлены три тематических исследования по *переносу стиля* (style transfer), *визуально-языковой генерации* (vision-language generation) и *синтезу лица* (face synthesis), чтобы проиллюстрировать, как этого добиться в условиях обучения без подкрепления или при слабом подкреплении.

Глава 6 посвящена актуальной проблеме реального мира – *распознаванию лиц* (face recognition). В ней обсуждаются современные методы, основанные на глубоком обучении, которые можно применять даже к неполным изображениям лица. В главе показано: (a) как создаются необходимые архитектуры глубокого обучения; (b) как такие модели можно обучать и тестировать; (c) как можно использовать *точную настройку* (fine tuning) предварительно обученных сетей для определения эффективных сигналов распознавания с полными и частичными данными о лице; (d) какие успехи достигнуты за счет последних разработок в области глубокого обучения; (e) каковы текущие ограничения методов глубокого обучения, используемых для распознавания лиц. В главе также упомянуты некоторые из нерешенных проблем в этой области.

В *главе 7* обсуждается важнейший вопрос о том, как перенести обучение из одной области данных в другую. Сюда относятся методы, основанные на дифференциальной геометрии, *разреженном представлении* (sparse representation) и глубоких нейронных сетях. Они делятся на два широких класса – дискриминационные и генеративные подходы. Первые включают обучение модели классификатора с использованием дополнительных потерь, чтобы сделать исходное и целевое распределения признаков похожими. Вторые используют генеративную модель для выполнения адаптации предметной области (домена): обычно междоменная генеративная состязательная сеть обучается для сопоставления образцов из исходного домена с целевым, а модель классификатора обучается на преобразованных целевых изображениях. Такие подходы проверяются на задачах междоменного распознавания и семантической сегментации.

В *главе 8* мы возвращаемся к задаче адаптации предметной области в контексте семантической сегментации, когда глубокие сети испытывают потребность в огромном количестве размеченных данных для обучения. Глава начинается с обсуждения различных уровней, на которых может осущест-

вляться адаптация, и стратегий их достижения. Затем рассматривается задача непрерывного обучения семантической сегментации. Хотя эта задача является относительно новой областью исследований, интерес к ней быстро растет, и уже представлено множество различных сценариев. Они подробно описаны вместе с подходами, необходимыми для их решения.

Вслед за главой 1 в *главе 9* вновь подчеркивается важность визуального отслеживания как одной из основных классических проблем компьютерного зрения. Цель этой главы – дать обзор развития области, начиная с алгоритма Лукаса–Канаде и согласованных фильтров и заканчивая подходами, основанными на глубоком обучении, а также переходом к сегментации видео. Обзор ограничен целостными моделями для общего отслеживания в плоскости изображения, и особое внимание уделяется дискриминационным моделям, трекеру MOSSE (minimum output sum of squared errors, минимальная выходная сумма квадратов ошибок) и DCF (discriminative correlation filters, дискриминационные корреляционные фильтры).

Глава 10 развивает концепцию визуального отслеживания объектов еще на один шаг и концентрируется на долгосрочном отслеживании. Чтобы успешно справиться с этой задачей, отслеживание объектов должно решать серьезные проблемы, связанные с *распадом модели* (model decay), то есть с ухудшением качества модели из-за нарастающей погрешности, а также с исчезновением и появлением цели. Успех глубокого обучения оказал большое влияние на подходы к отслеживанию визуальных объектов, поскольку автономное обучение *сиамских трекеров* (Siamese tracker) помогает устранить распад модели. Однако, чтобы избежать возможности потери отслеживания в тех случаях, когда внешний вид цели значительно меняется, сиамские трекеры могут воспользоваться встроенными инвариантностями и эквивариантностями, допускающими вариации внешнего вида, не усугубляя распад модели.

В последние годы крепнет уверенность в том, что в динамичной среде видео и движущихся объектов – особенно когда идет речь о *распознавании действий и поведения* (action/behavior recognition) – жизненно важную роль играет понимание когнитивных функций человека. Обоснованность этого предположения полностью подтверждают следующие две главы. В *главе 11* описывается ориентированная на действия структура, которая охватывает несколько временных масштабов и уровней абстракции. Нижний уровень детализирует *характеристики объекта*, который совершает различные действия; средний уровень моделирует *отдельные действия*, а самый высокий уровень моделирует *деятельность*. Упор на использование характеристик понимания, геометрии, онтологий и ограничений, основанных на физике, позволяет избежать чрезмерного обучения характеристикам внешнего вида. Чтобы объединить восприятие на основе сигналов с *символьными знаниями* (symbolic knowledge), векторизованные знания согласовываются с визуальными признаками. Глава также включает обсуждение понятий *действия* (action) и *деятельности* (activity).

В *главе 12* рассматривается проблема *временно́й сегментации событий* (temporal event segmentation). Достижения когнитивной науки демонстрируют подходы к разработке высокоэффективных алгоритмов компьютерного зрения для пространственно-временной сегментации событий в видео

без необходимости использования каких-либо аннотированных данных. Во-первых, модель теории сегментации событий позволяет вычислять границы событий: затем следует временная сегментация с использованием фреймворка прогнозирования восприятия, временная сегментация вместе с рабочими моделями событий, основанными на *картах внимания* (attention map), и пространственно-временная локализация событий. Этот подход обеспечивает производительность на современном уровне при временно́й сегментации без подкрепления и пространственно-временной локализации действий, позволяя конкурировать с производительностью базовых моделей, обучаемых с полным подкреплением и требующих большого объема полностью аннотированных данных.

Методы обнаружения аномалий лежат в основе многих приложений, таких как анализ медицинских изображений, обнаружение мошенничества или видеонаблюдение. Эти методы также представляют собой важный шаг на пути развития искусственных *саморазвивающихся систем* (self-aware system), которые могут постоянно учиться в новых ситуациях. В *главе 13* представлен метод обнаружения аномалий с частичным подкреплением для этого типа саморазвивающихся агентов. Он использует возможности передачи сообщений в обобщенных динамических байесовских сетях для выявления аномалий на разных уровнях абстракции и различных типов данных временных рядов. Следовательно, обнаруженные аномалии могут быть использованы для обеспечения саморазвития системы за счет интеграции новых приобретенных знаний. В главе рассмотрено исследование по тематике обнаружения аномалий с использованием мультисенсорных данных от полуавтономного транспортного средства, которое выполняет различные задачи в закрытой среде.

Методы, основанные на моделировании и машинном обучении, отражали две доминирующие стратегии при решении различных проблем восстановления изображений, когда речь идет о низкоуровневом техническом зрении. Как правило, эти два метода имеют свои достоинства и недостатки; например, методы на основе моделей обладают гибкостью при решении различных проблем восстановления изображений, но обычно требуют долгой и трудоемкой настройки априорных значений для обеспечения хорошей производительности. С другой стороны, методы на основе машинного обучения демонстрируют более высокую эффективность и результативность по сравнению с традиционными методами на основе моделей, в основном из-за сквозного обучения, но, как правило, им не хватает гибкости для решения различных задач восстановления изображений. *Глава 14* знакомит читателей с методами plug-and-play и развертывания на базе глубоких нейросетей, которые продемонстрировали большие перспективы за счет использования как методов, основанных на обучении, так и методов, основанных на модели: основная идея методов глубокого plug-and-play заключается в том, что шумоподавитель на основе машинного обучения может неявно служить исходным изображением для методов восстановления изображений на основе модели, в то время как идея методов глубокого развертывания заключается в том, что путем развертывания моделей с помощью переменных алгоритмов разделения можно получить сквозную обучаемую итеративную

сеть, заменяя соответствующие подзадачи нейронными модулями. Следовательно, методы глубокого plug-and-play и глубокого развертывания могут унаследовать гибкость методов, основанных на моделях, сохраняя при этом преимущества методов, основанных на обучении.

Визуальные состязательные объекты (visual adversarial examples) – это изображения и видео, намеренно искаженные, чтобы ввести в заблуждение модели машинного обучения. В *главе 15* представлен обзор методов формирования помех для создания визуальных состязательных объектов, применяемых при оценке решения задач классификации изображений, обнаружения объектов, отслеживания движения и распознавания видео. Сначала определяются ключевые свойства состязательной атаки и типы возмущений, порождаемых атакой; затем анализируются основные варианты методов генерации состязательных атак на изображения и видео и исследуются применяемые при этом знания о целевой модели. Наконец, рассмотрены защитные механизмы, которые повышают устойчивость моделей машинного обучения к атакам со стороны противника и вероятность выявления манипуляций входными данными.

Вместе эти главы раскрывают заинтересованному читателю – будь то студент, исследователь или практический специалист – всю ширину и глубину современной методологии компьютерного зрения.

В заключение мы хотели бы выразить всем авторам нашу благодарность за огромный энтузиазм и самоотверженность, проявленные при работе над главами этой монографии. Благодаря их усилиям эта книга, как мы надеемся, станет надежным путеводителем в мире современного компьютерного зрения. Благодарим Тима Питтса из Elsevier Science за его постоянные советы и поддержку с самого начала и на протяжении всего времени, пока мы трудились над составлением этого сборника.

Рой Дэвис
Роял Холлоуэй, Лондонский университет,
Лондон, Соединенное Королевство

Мэтью Терк
Технологический институт Toyota в Чикаго,
Чикаго, Иллинойс, США
Май 2021 г.

Глава 1

Кардинальные перемены в области компьютерного зрения

Автор главы: **Рой Дэвис**,
Роял Холлоуэй, Лондонский университет, Эгам,
графство Суррей, Соединенное Королевство

Краткое содержание главы:

- обзор истории методов компьютерного зрения, включая операторы низкоуровневой обработки изображений, обнаружение 2D- и 3D-объектов, определение местоположения и распознавание, отслеживание и сегментацию;
- изучение развития методов глубокого обучения на основе искусственных нейронных сетей, включая взрывной рост популярности глубокого обучения;
- обзор методов глубокого обучения, применяемых для обнаружения признаков, обнаружения объектов, определения местоположения, распознавания и отслеживания объектов, классификации текстур и семантической сегментации изображений;
- влияние методов глубокого обучения на традиционную методологию компьютерного зрения.

1.1. Введение. Компьютерное зрение и его история

В течение последних трех-четырех десятилетий компьютерное зрение постепенно превратилось в полноценный научный предмет со своей методологией и областью применения. На самом деле у него так много областей применения, что трудно перечислить их все. Среди наиболее известных – распознавание объектов, наблюдение (включая подсчет людей и распозна-

вание номерных знаков), роботизированное управление (включая автоматическое управление транспортным средством), сегментация и интерпретация медицинских изображений, автоматический осмотр и сборка в заводских условиях, распознавание отпечатков пальцев и лиц, интерпретация жестов и многое другое. Для работы компьютерного зрения необходим поток данных из различных источников изображений, включая каналы видимого и инфракрасного спектра, трехмерные датчики и ряд жизненно важных медицинских устройств визуализации, таких как компьютерные и магнитно-резонансные томографы. К тому же данные должны включать положение, позу, расстояние между объектами, движение, форму, текстуру, цвет и многие другие аспекты. При таком разнообразии данных и изобилии действий и методов, используемых для их обработки, будет трудно обрисовать общую картину в рамках одной главы: следовательно, выбор материала неизбежно будет ограничен; тем не менее мы будем стремиться обеспечить прочную основу и дидактический подход к предмету.

Сегодня вряд ли можно представить компьютерное зрение без огромного прорыва, достигнутого в 2010-х годах, и, в частности, «взрыва глубокого обучения», который произошел примерно в 2012 г. Это событие значительно изменило саму суть предмета исследований и привело к достижениям и применениям, которые не только впечатляют, но и во многих случаях выходят далеко за рамки того, о чем люди мечтали даже в 2010 г. Наша книга в первую очередь посвящена самым передовым достижениям в области компьютерного зрения; роль этой вступительной главы состоит в том, чтобы обрисовать в общих чертах историю традиционной методологии, исследовать новые методы глубокого обучения и показать, как они изменили и улучшили более ранние (устаревшие) подходы.

На первом этапе будет полезно рассмотреть истоки компьютерного зрения, которое можно считать зародившимся в 1960-х и 1970-х гг., в основном как ответвление обработки изображений. В то время появилась техническая возможность захватывать целые изображения, а также удобно хранить и обрабатывать их на цифровых компьютерах. Первоначально изображения, как правило, записывались в бинарном виде или в оттенках серого, хотя позже стало возможным захватывать их в цвете. Исследователи уже тогда мечтали подражать человеческому глазу, распознавая объекты и интерпретируя сцены, но с доступными тогда маломощными компьютерами эти мечты были далеки от воплощения. На практике обработка изображений использовалась для *исправления* (tiding up) изображений и обнаружения признаков объектов, а распознавание изображений осуществлялось с использованием методов статистического распознавания образов, таких как *алгоритм ближайшего соседа* (nearest neighbor algorithm). Двумя основными локомотивами развития компьютерного зрения стали искусственный интеллект и биологическое зрение. Ограниченный объем книги не позволит нам здесь обсуждать эти аспекты; отметим лишь, что они заложили основу искусственных нейронных сетей и глубокого обучения (подробнее об этом в разделе 1.7).

Исправление изображений, вероятно, лучше описать как предварительную обработку: она может включать в себя ряд функций, где одной из самых важных является устранение шума. Вскоре было обнаружено, что использо-

вание алгоритмов сглаживания, в которых вычисляется среднее значение интенсивностей в окне вокруг каждого входного пикселя, применяемое для формирования отдельного сглаженного изображения, не только приводит к снижению уровня шума, но и к влиянию сигнала на самого себя (этот процесс также можно представить как уменьшение входной полосы пропускания для устранения большей части шума, с дополнительным эффектом устранения из входного сигнала компонентов высокой пространственной частоты). Однако эта проблема была в значительной степени решена за счет применения медианной, а не средней фильтрации, поскольку она работает за счет устранения выбросов на каждом конце локального распределения интенсивности – медиана является значением, наименее подверженным влиянию шума.

Типичные ядра фильтрации по среднему показаны ниже, причем второе из них более приближено к идеальной гауссовой форме:

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \quad \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}. \tag{1.1}$$

Оба они являются ядрами линейной свертки, которые по определению пространственно инвариантны в пространстве изображений. Общая маска свертки 3×3 задается выражением

$$\begin{bmatrix}c4 & c3 & c2\\ c5 & c0 & c1\\ c6 & c7 & c8\end{bmatrix}, \tag{1.2}$$

где локальным пикселям присвоены метки 0–8. Затем мы берем значения интенсивности в локальной окрестности изображения 3×3 как

$$\begin{bmatrix}P4 & P3 & P2\\ P5 & P0 & P1\\ P6 & P7 & P8\end{bmatrix}. \tag{1.3}$$

Воспользовавшись нотацией условного языка программирования наподобие C++, мы можем записать полную процедуру свертки в виде псевдокода:

```
для всех пикселей изображения выполнить {
  Q0 = P0*c0 + P1*c1+ P2*c2+ P3*c3 +P4*c4
    + P5*c5 + P6*c6 + P7*c7+ P8*c8;
}
```
(1.4)

До сих пор мы рассматривали маски свертки, которые представляют собой линейные комбинации входных интенсивностей: они отличаются от нелинейных процедур, таких как пороговая обработка, которые не могут быть выражены как свертки. На самом деле пороговая обработка очень широко применяется и может быть записана в виде следующего алгоритма:

```
для всех пикселей изображения выполнить {
  если (P0 < порог) A0 = 1; иначе A0 = 0;
}
```
(1.5)

Эта процедура преобразует изображение в оттенках серого в P-пространстве в бинарное изображение в A-пространстве. Здесь она используется для выделения темных объектов, представляя их как единицы на фоне нулей.

Мы завершаем этот раздел полной процедурой медианной фильтрации в пределах окрестности 3×3:

```
для (i = 0; i <= 255; i ++) hist[i] = 0;
для всех пикселей изображения выполнить {
    для (m = 0;m <= 8;m++) hist[P[m]]++;
    i = 0; sum = 0;
    пока (sum < 5){
        sum = sum+hist[i];
        i = i +l;
    }
        Q0 = i -1;
        для (m = 0;m <= 8;m++) hist[P[m]] = 0;
    }
```
(1.6)

Запись `P[0]` обозначает P_0, и так далее от `P[1]` до `P[8]`. Заметим, что операция нахождения медианы требует больших вычислений, поэтому время экономится только за счет повторной инициализации конкретных элементов гистограммы, которые фактически использовались.

Важная особенность процедур, описываемых уравнениями (1.4)–(1.6), заключается в том, что они берут входные данные из одного пространства изображений и выводят их в другое пространство изображений – процесс, часто описываемый как параллельная обработка, – тем самым устраняя проблемы, связанные с порядком, в котором выполняются вычисления отдельных пикселей.

Наконец, все алгоритмы сглаживания изображений, задаваемые уравнениями (1.1)–(1.4), используют ядра свертки 3×3, хотя, очевидно, можно использовать ядра гораздо большего размера: действительно, их можно реализовать иным путем, сначала преобразовывая в область пространственных частот, а затем систематически устраняя высокие пространственные частоты, хотя и с дополнительной вычислительной нагрузкой. С другой стороны, нелинейные операции, такие как медианная фильтрация, не могут быть реализованы подобным образом.

Для удобства остаток этой главы разделен на несколько частей следующим образом:

- часть A. Обзор операторов низкоуровневой обработки изображений;
- часть B. Выделение и распознавание 2D-объектов;
- часть C. Выделение трехмерных объектов и важность инвариантности;
- часть D. Отслеживание движущихся объектов;
- часть E. Анализ текстур;
- часть F. От искусственных нейронных сетей к методам глубокого обучения;
- часть G. Заключение.

В целом назначение этой главы состоит в том, чтобы обобщить ключевые понятия и достижения ранних – или «устаревших» – исследований в области компьютерного зрения и напомнить читателям об их значении, чтобы они могли более уверенно освоить новейшие разработки в этой области. Однако необходимость сделать такой выбор означает, что пришлось исключить многие другие важные темы.

1.2. Часть A. Обзор операторов низкоуровневой обработки изображений

1.2.1. Основы обнаружения краев

Обнаружение краев (edge detection) является наиболее важной и широко применяемой операцией обработки изображений. Для этого есть разные важные причины, но в конечном счете описание форм объектов по их краям и внутренним контурам уменьшает объем данных, необходимых для хранения изображения $N \times N$, с $O(N^2)$ до $O(N)$, тем самым значительно повышая эффективность последующего хранения и обработки. Кроме того, хорошо известно, что люди могут очень эффективно распознавать объекты по их контурам (иногда даже лучше, чем по полному изображению): легкость и достоверность распознавания двумерных эскизов и мультфильмов могут служить тому подтверждением.

В 1960-х и 1970-х годах было разработано значительное количество операторов обнаружения краев, многие из которых были в первую очередь интуитивно понятными, а это означает, что их оптимальность была под вопросом. Некоторые операторы применяли 8 или 12 масок-шаблонов для обнаружения краев с разной ориентацией. Как ни странно, прошло достаточно много времени, прежде чем возникло понимание, что, поскольку края являются векторами, для их обнаружения должно быть достаточно двух масок. Однако это не сразу устранило необходимость принятия решения о том, какие коэффициенты маски следует использовать в детекторах краев – даже в случае окрестностей 3×3, – и мы перейдем к дальнейшему изучению этого вопроса.

Далее мы исходно полагаем, что необходимо использовать 8 масок с углами, отличающимися на 45°. Однако 4 из этих масок отличаются от остальных только знаком, что делает ненужным их отдельное применение. На данный момент аргументы симметрии приводят к следующим маскам для 0° и 45° соответственно:

$$\begin{bmatrix} -A & 0 & A \\ -B & 0 & B \\ -A & 0 & A \end{bmatrix} \quad \begin{bmatrix} 0 & C & D \\ -C & 0 & C \\ -D & -C & 0 \end{bmatrix}. \tag{1.7}$$

Очевидно, что очень важно спроектировать маски так, чтобы они давали правильные ответы в разных направлениях. Чтобы выяснить, как это влияет

на коэффициенты маски, воспользуемся тем фактом, что градиенты интенсивности должны следовать правилам сложения векторов. Если значения интенсивности пикселей в окрестности 3×3 равны

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \tag{1.8}$$

представленные выше маски приведут к следующим оценкам градиента в направлениях 0°, 90° и 45°:

$$\begin{aligned} g_0 &= A(c + i - a - g) + B(f - d); \\ g_{90} &= A(a + c - g - i) + B(b - h); \\ g_{45} &= C(b + f - d - h) + D(c - g). \end{aligned} \tag{1.9}$$

Если сложение векторов должно быть допустимым, мы также имеем:

$$g_{45} = (g_0 + g_{90})/\sqrt{2}. \tag{1.10}$$

Приравнивание коэффициентов при $a, b, \ldots, i$ приводит к самосогласованной паре условий:

$$\begin{aligned} C &= B/\sqrt{2}; \\ D &= A\sqrt{2}. \end{aligned} \tag{1.11}$$

Далее обратите внимание на дополнительное требование – маски 0° и 45° должны давать одинаковые отклики при 22,5°. На самом деле за этим утверждением скрываются довольно утомительные алгебраические выкладки (Davies, 1986), которые показывают, что

$$B/A = (13\sqrt{2} - 4)/7 = 2{,}055. \tag{1.12}$$

Округляя значение этого выражения до 2, мы прямо приходим к маскам *оператора Собеля*:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{1.13}$$

применение которого дает карты компонентов g_x, g_y градиента интенсивности. Поскольку края являются векторами, мы можем вычислить локальную величину края g и направление θ, используя стандартные векторные формулы:

$$\begin{aligned} g &= [g_x^2 + g_y^2]^{1/2}; \\ \theta &= \arctan(g_y/g_x). \end{aligned} \tag{1.14}$$

Обратите внимание, что вычисления g и θ для всего изображения не будут свертками, поскольку они включают нелинейные операции.

Итак, в разделах 1.1 и 1.2.1 мы описали различные категории операторов обработки изображений, включая линейные и нелинейные операторы и операторы свертки. Примерами сверток (линейных операций) являются среднее и гауссово сглаживание и оценка компонента краевого градиента. Примерами нелинейных операций являются порог, вычисление краевого градиента и ориентации края. Следует отметить, что коэффициенты маски Собеля были получены в качестве побочного продукта, а не целенаправленно. Фактически они были разработаны для оптимизации точности ориентации краев. Заметим также, что, как мы увидим позже, точность ориентации имеет первостепенное значение, когда информация о краях передается в схемы расположения объектов, такие как преобразование Хафа.

1.2.2. Оператор Кэнни

Детектор краев Кэнни изначально был создан как намного более точная замена основных детекторов краев, таких как детектор Собеля, и вызвал настоящий фурор после публикации в 1986 году (Canny, 1986). Для достижения столь высокой точности по очереди применяется ряд процессов:

1. Изображение сглаживается с помощью двумерного гауссиана, чтобы гарантировать, что поле интенсивности является математически корректной функцией.
2. Изображение дифференцируется с использованием двух одномерных производных функций, таких как функции Собеля, и вычисляется поле величины градиента.
3. Для утончения краев используется немаксимальное подавление вдоль направления нормали локального края. Это происходит в два этапа: (1) нахождение двух нецентральных красных точек, показанных на рис. 1.1, что включает интерполяцию величины градиента между двумя *парами* пикселей; (2) выполнение квадратичной интерполяции между градиентами интенсивности в трех красных точках для определения положения сигнала края пика с субпиксельной точностью.
4. Выполняется «гистерезисная» пороговая обработка: применение двух порогов t_1 и t_2 ($t_2 > t_1$) к полю градиента интенсивности; результатом является «не край», если $g < t_1$, «край», если $g > t_2$, а иначе это будет «край», только если он находится рядом с «краем». (Обратите внимание, что свойство «край» может распространяться от пикселя к пикселю в соответствии с приведенными выше правилами.)

Как отмечено в пункте 3, для определения местоположения пика амплитуды градиента может использоваться квадратичная интерполяция. Несложные алгебраические выкладки показывают, что для g-значений g_1, g_2, g_3 трех красных точек смещение пика от центральной красной точки равно $(g_3 - g_1)\sec\theta/[2(2g_2 - g_1 - g_3)]$: здесь $\sec\theta$ – это коэффициент, на который θ увеличивает расстояние между крайними красными точками.

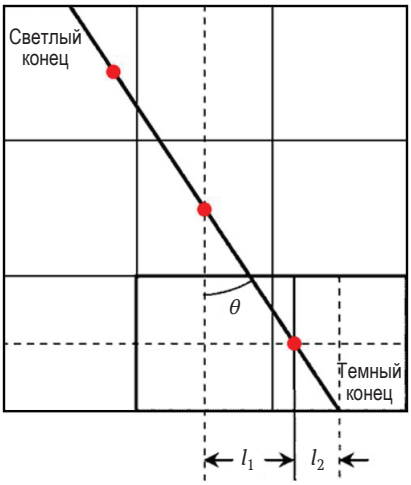


Рис. 1.1 ❖ Использование квадратичной интерполяции для определения точного положения пика амплитуды градиента

1.2.3. Обнаружение сегмента линии

В разделе 1.2.1 мы показали, как при помощи детектора краев всего с двумя масками вычисляется величина и ориентация признака края. Стоит подумать, можно ли использовать аналогичный векторный подход и в других случаях. Действительно, модифицированный векторный подход также можно использовать для обнаружения признаков *сегментов линии*. В этом можно убедиться, рассмотрев следующую пару масок:

$$\mathrm{L}_1 = A\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}; \quad \mathrm{L}_2 = B\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}. \tag{1.15}$$

Ясно, что можно построить еще две маски такого вида, но они отличаются от двух предыдущих только знаком и ими можно пренебречь. Таким образом, этот набор масок содержит ровно столько, сколько необходимо для векторного вычисления. В самом деле, если мы ищем темные полосы на светлом фоне, 1 может обозначать линию, а –1 может представлять светлый фон. (Нули можно рассматривать как «безразличные» коэффициенты, так как они будут игнорироваться в любой свертке.) Следовательно, L_1 представляет собой полосу 0°, а L_2 – полосу 45°. (Термин «полоса» используется здесь для обозначения сегмента линии значимой ширины.) Применяя тот же метод, что и в разделе 1.2.1, и определяя значения интенсивности пикселей, как в уравнении (1.8), находим:

$$\begin{aligned} l_0 &= A(d + f - b - h); \\ l_{45} &= B(c + g - a - i). \end{aligned} \tag{1.16}$$

Однако в данном случае недостаточно информации для определения отношения A к B, поэтому это должно зависеть от практических аспектов ситуации. Учитывая, что это вычисление выполняется в окрестности 3×3, неудивительно, что оптимальная ширина полосы для обнаружения с использованием вышеуказанных масок равна 1,0; эксперименты (Davies, 1997) показали, что согласование масок с шириной полосы w (или наоборот) дает оптимальную точность ориентации при $w \approx 1{,}4$, что имеет место при $B/A \approx 0{,}86$. Отсюда получается максимальная ошибка ориентации ~0,4°, что выгодно отличается от ~0,8° для оператора Собеля.

Воспользуемся формулами, аналогичными формулам в разделе 1.2.1, для псевдовекторного расчета коэффициента интенсивности линии l и ориентации сегмента линии θ:

$$l = [l_0^2 + l_{45}^2]^{1/2};$$
$$\theta = \frac{1}{2}\arctan(l_{45}/l_0). \tag{1.17}$$

Здесь мы были вынуждены включить коэффициент 1/2 перед арктангенсом: это потому, что отрезок прямой демонстрирует симметрию вращения на 180° по сравнению с 360° для обычных углов.

Обратите внимание, что это снова тот случай, когда оптимизация направлена на достижение высокой точности ориентации, а не, например, на чувствительность обнаружения.

Здесь стоит отметить два применения обнаружения линейных сегментов. Одним из них является осмотр сыпучих зерен пшеницы для обнаружения мелких темных насекомых, которые напоминают темные полосы: для этого использовались маски 7×7, разработанные на основе приведенной выше модели (Davies и др., 2003). Другим применением является определение расположения артефактов, таких как телеграфные провода на фоне неба или тросов, поддерживающих киноактеров, которые затем можно целенаправленно удалять.

1.2.4. Оптимизация чувствительности обнаружения

Оптимизация чувствительности обнаружения – задача, которая хорошо известна в радиолокации и очень эффективно применялась для этой цели со времен Второй мировой войны. По сути, эффективное обнаружение летательных аппаратов радиолокационными системами требует оптимизации *отношения сигнал–шум* (signal to noise ratio, SNR). Конечно, в случае радара обнаружение – это одномерная проблема, тогда как при построении изображений нам необходимо оптимально обнаруживать двумерные объекты на фоне шума. Однако шум изображения не обязательно является гауссовым белым шумом, как обычно предполагается применительно к радару, хотя удобно начать с этого предположения.

В радиолокации сигналы можно рассматривать как положительные пики на фоне шума, который обычно близок к нулю. В этих условиях применима хорошо известная теорема, которая гласит, что оптимальное обнаружение сигнала заданной формы достигается с помощью «согласованного фильтра», который имеет ту же форму характеристики, что и идеализированный входной сигнал. То же самое относится к изображениям, и в этом случае пространственный согласованный фильтр должен иметь ту же форму характеристики, что и идеальная форма искомого двумерного объекта.

Кратко рассмотрим математическую основу этого подхода. Во-первых, мы предполагаем набор пикселей, в которых производится выборка сигналов, что дает значения S_i. Затем мы выражаем желаемый фильтр в виде n-элементного весового шаблона с коэффициентами w_i. Наконец, предполагаем, что уровни шума в каждом пикселе независимы и подчиняются локальным распределениям со стандартными отклонениями N_i.

Очевидно, что суммарный сигнал, полученный от весового шаблона, можно записать в виде:

$$S = \sum_{(i=1)}^{n} w_i S_i, \tag{1.18}$$

тогда как общий шум, полученный от весового шаблона, будет характеризоваться его дисперсией:

$$N^2 = \sum_{(i=1)}^{n} w_i^2 N_i^2. \tag{1.19}$$

Следовательно, SNR равно

$$\rho^2 = S^2/N^2 = \left(\sum_{i=1}^{n} w_i S_i\right)^2 / \sum_{i=1}^{n} w_i^2 N_i^2. \tag{1.20}$$

Для нахождения оптимального SNR найдем производную

$$\partial\rho^2/\partial w_i = (1/N^4)[N^2(2SS_i) - S^2(2w_i N_i^2)] = (2S/N^4)[N^2 S_i - S(w_i N_i^2)], \tag{1.21}$$

а затем примем $\partial\rho^2/\partial w_i = 0$ и сразу получим

$$w_i = \frac{S_i}{N_i^2} \times \frac{N^2}{S}, \tag{1.22}$$

что можно записать проще как

$$w_i \propto \frac{S_i}{N_i^2}, \tag{1.23}$$

хотя знак пропорциональности можно заменить равенством без ограничения общности.

Обратите внимание, что если N_i не зависит от i (т. е. уровень шума не меняется на всей площади изображения), то $w_i = S_i$: это доказывает упомянутую выше теорему о том, что *пространственный* согласованный фильтр должен иметь тот же профиль интенсивности, что и двумерный объект, подлежащий обнаружению.

1.2.5. Работа с изменениями интенсивности фона

Помимо очевидной разницы в размерности, есть еще одно важное отличие зрения от радара: у последнего в отсутствие входного сигнала выходной сигнал системы колеблется и в среднем равен нулю. Однако в компьютерном зрении уровень фона обычно будет меняться в зависимости от окружающего освещения, а также в зависимости от входного изображения. По сути, решение этой проблемы заключается в использовании масок с нулевой суммой (или нулевым средним). Поэтому для такой маски, как в уравнении (1.2), мы просто вычитаем среднее значение $\bar{c}$ всех компонентов маски из каждого компонента, чтобы убедиться, что общая маска имеет нулевое среднее значение.

Чтобы убедиться, что использование стратегии нулевого среднего работает, представьте себе применение немодифицированной маски к окрестности изображения, показанной в уравнении (1.3), – допустим, мы получили значение K. Теперь добавим B к интенсивности каждого пикселя в окрестности; это добавит $\sum_n Bc_i = B\sum_n c_i = Bn\bar{c}$ к значению K. Но если мы сделаем $\bar{c} = 0$, то получим исходный вывод маски K.

В целом мы должны отметить, что стратегия нулевого среднего является лишь приближением, так как на изображении будут места, где фон варьируется между высоким и низким уровнями, поэтому невозможно точное устранение нулевого среднего (т. е. B нельзя рассматривать как постоянную над областью маски). Тем не менее если предположить, что изменение фона происходит в масштабе, значительно превышающем масштаб размера маски, эта стратегия должна работать адекватно.

Следует отметить, что аппроксимация с нулевым средним значением уже широко используется, как вы видели на примере масок ребер и сегментов линий в уравнениях (1.7) и (1.15). Этот подход также должен применяться к другим детекторам, таким как детекторы углов и отверстий.

1.2.6. Теория, сочетающая согласованный фильтр и конструкции с нулевым средним

На первый взгляд идея нулевого среднего настолько проста, что может показаться, что она легко интегрируется с формулами согласованного фильтра из раздела 1.2.4. Однако применение нулевого среднего уменьшает количество степеней свободы согласованного фильтра на одну, поэтому необходимо изменить формальное представление согласованного фильтра, чтобы последний продолжал оставаться идеальным детектором. Дабы продолжить,

мы представляем случаи с нулевым средним и согласованным фильтром следующим образом:

$$(w_i)_{\text{z-m}} = S_i - \bar{S};$$
$$(w_i)_{\text{m-f}} = S_i/N_i^2. \tag{1.24}$$

Далее мы объединяем их в форму

$$w_i = (S_i - \tilde{S})/N_i^2, \tag{1.25}$$

где мы избежали тупика, попробовав гипотетический (т. е. пока неизвестный) тип среднего для S, который мы называем $\tilde{S}$. (Конечно, если эта гипотеза в конце концов приведет к противоречию, потребуется новый подход.) Применение условия нулевого среднего $\sum_i w_i = 0$ теперь дает следующее:

$$\sum_i w_i = \sum_i S_i/N_1^2 - \sum_i \tilde{S}/N_1^2 = 0; \tag{1.26}$$

$$\therefore \quad \tilde{S}\sum_i (1/N_1^2) = \sum_i S_i/N_1^2; \tag{1.27}$$

$$\therefore \quad \tilde{S} = \sum_i (S_i/N_1^2) \Big/ \sum_i (1/N_1^2). \tag{1.28}$$

Из этого мы делаем вывод, что $\tilde{S}$ должно быть взвешенным средним, в частности взвешенным средним по шуму $\tilde{S}$. С другой стороны, если шум равномерный, $\tilde{S}$ вернется к обычному невзвешенному среднему $\bar{S}$. Кроме того, если мы не применяем условие нулевого среднего (которого мы можем достичь, установив $\tilde{S} = 0$), уравнение (1.25) сразу возвращается к стандартному условию согласованного фильтра.

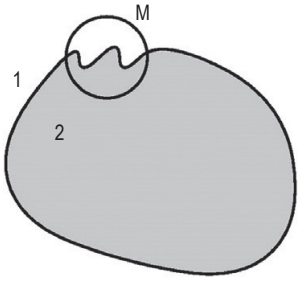
Формула для $\tilde{S}$ может показаться излишне обобщенной, поскольку N_i обычно почти не зависит от i. Однако если бы идеальный профиль был получен путем усреднения профилей реальных объектов, то вдали от его центра дисперсия шума могла бы быть более существенной. Действительно, для больших объектов это было бы явным ограничивающим фактором при таком подходе. Но для относительно небольших объектов и признаков дисперсия шума не должна чрезмерно варьироваться и должны быть достижимы полезные профили согласованного фильтра.

От себя хочу отметить, что основной результат, доказанный в этом разделе (ср. уравнения (1.25) и (1.28)), отнял у меня столько времени и усилий, что я начал было сомневаться в своей способности достичь его. Поэтому я стал называть его «последней теоремой Дэвиса».

1.2.7. Структура маски (дополнительные соображения)

Хотя формальное представление согласованного фильтра и полностью интегрированное к данному моменту условие нулевого среднего могут пока-

заться достаточно общими, чтобы обеспечить однозначную структуру маски, остается ряд аспектов, которые еще предстоит рассмотреть. Например, какого размера должны быть маски? И как их оптимально разместить вокруг каких-либо примечательных объектов или признаков? Чтобы ответить на этот вопрос, мы возьмем следующий пример довольно сложного признака объекта. Здесь область 2 – это обнаруживаемый объект, область 1 – фон, а M – область маски признака.



В этой модели мы должны рассчитать оптимальные значения весовых коэффициентов маски w_1 и w_2 и площадей областей A_1 и A_2. Мы можем записать общую мощность сигнала и шума из маски шаблона как:

$$\begin{aligned} S &= w_1 A_1 S_1 + w_2 A_2 S_2; \\ N^2 &= w_1^2 A_1 N_1^2 + w_2^2 A_2 N_2^2. \end{aligned} \tag{1.29}$$

Таким образом, мы получаем отношение мощности сигнал–шум (SNR):

$$f_{i,t+1} = f_{i,t} + \frac{\partial f}{\partial \phi}\frac{\partial \phi}{\partial t}. \tag{1.30}$$

Легко видеть, что если *обе* области маски увеличить по площади одинаково в η раз, то во столько же раз увеличится и ρ^2. Следовательно, мы можем оптимизировать маску, регулируя *относительные* значения A_1 и A_2 и оставляя общую площадь A неизменной. Давайте сначала исключим w_2, используя условие нулевого среднего (которое обычно применяется для предотвращения влияния изменений уровня интенсивности фона на результат):

$$w_1 A_1 + w_2 A_2 = 0. \tag{1.31}$$

Ясно, что мощность SNR больше не зависит от весов маски:

$$\rho^2 = \frac{S^2}{N^2} = \frac{(S_1 - S_2)^2}{N_1^2/A_1 + N_2^2/A_2}. \tag{1.32}$$

Далее, поскольку общая площадь маски A заранее определена, мы имеем:

$$A_2 = A - A_1. \tag{1.33}$$

Подстановка A_2 сразу дает нам простое условие оптимизации:

$$A_1/A_2 = N_1/N_2. \tag{1.34}$$

Принимая $N_1 = N_2$, мы получаем важный результат – *правило равных площадей* (Davies, 1999):

$$A_1 = A_2 = A/2. \tag{1.35}$$

Наконец, когда применяется правило равных площадей, правило нулевого среднего принимает форму:

$$w_1 = -w_2. \tag{1.36}$$

Обратите внимание, что многие случаи, например возникающие, когда передний план и фон имеют разные текстуры, можно смоделировать, полагая $N_1 \neq N_2$. В этом случае правило равной площади не применяется, но мы все еще можем использовать уравнение (1.34).

1.2.8. Обнаружение угла

В разделах 1.2.1 и 1.2.3 мы обнаружили, что только два типа признаков имеют векторную (или псевдовекторную) форму – края и линейные сегменты. Следовательно, в то время как эти признаки могут быть обнаружены с использованием всего лишь двух компонентных масок, ожидается, что все остальные признаки потребуют сопоставления со многими другими шаблонами, чтобы справиться с различными ориентациями. К этой категории относятся и *детекторы углов*, у которых типичные угловые шаблоны 3×3 имеют следующий вид:

$$\begin{bmatrix} -4 & 5 & 5 \\ -4 & 5 & 5 \\ -4 & -4 & -4 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & 5 \\ -4 & 5 & -4 \\ -4 & -4 & -4 \end{bmatrix}. \tag{1.37}$$

(Обратите внимание, что эти маски были настроены на форму с нулевым средним значением, дабы устранить эффекты различных условий освещения.)

Чтобы преодолеть очевидные проблемы сопоставления шаблонов, не последней из которых является необходимость использования ограниченного числа цифровых масок для аппроксимации аналоговых вариаций интенсивности, которые сами по себе заметно различаются от экземпляра к экземпляру, было предпринято много усилий по выработке более принципиального подхода. В частности, поскольку края определяются первыми производными поля интенсивности изображения, казалось логичным перейти к производным второго порядка. Одним из первых таких исследований был подход Боде (1978), в котором использовались операторы Лапласа и Гессе:

$$\begin{aligned} &\text{Лапласиан} = I_{xx} + I_{yy}; \\ &\text{Гессиан} = I_{xx} I_{yy} - I_{xy}^2. \end{aligned} \tag{1.38}$$

Они были особенно привлекательны, поскольку определены в терминах детерминанта и следа симметричной матрицы вторых производных и, таким образом, инвариантны относительно вращения.

На самом деле *оператор Лапласа* дает существенные отклики вдоль линий и краев и, следовательно, не особенно подходит для обнаружения углов. С другой стороны, *оператор Боде* (*оператор Гессе*), известный как «DET», не реагирует на линии и края, но дает значимые сигналы вблизи углов и, следовательно, полезен для построения детектора углов, хотя он реагирует одним знаком на одной стороне угла и обратным знаком на другой стороне угла: на самом углу дает нулевой ответ. Кроме того, другие исследователи подвергли критике специфические отклики оператора DET и обнаружили, что им необходим довольно сложный анализ, чтобы определить наличие и точное положение каждого угла (Dreschler, Nagel, 1981; Nagel, 1983).

Тем не менее Китчен и Розенфельд (Kitchen, Rosenfeld, 1982) показали, что они смогли преодолеть эти проблемы, оценив скорость изменения вектора направления градиента вдоль направления касательной горизонтального края и связав его с горизонтальной кривизной κ функции интенсивности I. Чтобы получить реалистичное представление о *силе угла*, они умножили κ на величину локального градиента интенсивности g:

$$\begin{aligned} C &= \kappa g = \kappa (I_x^2 + I_y^2)^{1/2} \\ &= \frac{I_{xx} I_y^2 - 2 I_{xx} I_x I_y + I_{yy} I_x^2}{I_x^2 + I_y^2}. \end{aligned} \tag{1.39}$$

Наконец, они использовали эвристику немаксимального подавления вдоль нормального направления края для дальнейшей локализации угловых положений.

Интересно, что Нагель (Nagel, 1983) и Шах и Джайн (Shah, Jain, 1984) пришли к выводу, что угловые детекторы Китчена и Розенфельда, Дрешлера и Нагеля, а также Зуниги и Харалика (Zuniga, Haralick 1983) по существу эквивалентны. Это не должно вызывать большого удивления, так как, в конце концов, можно было бы ожидать, что различные методы будут отражать одни и те же лежащие в основе физические явления (Davies, 1988) – определение производной второго порядка, которое можно интерпретировать как горизонтальную кривизну, умноженную на градиент интенсивности.

1.2.9. Оператор «особой точки» Харриса

На этом этапе Харрис и Стивенс (Harris, Stephens, 1988) разработали совершенно новый оператор, способный обнаруживать признаки угла, основанный не на производных второго порядка, а на производных первого порядка. Как мы увидим ниже, это упростило математическую составляющую, включая избавление от трудностей применения цифровых масок к аналоговым функциям. Фактически новый оператор смог выполнять функцию производной второго порядка, применяя операции первого порядка. Любопытно,

каким образом он извлекает соответствующую информацию о производных второго порядка. Чтобы понять это, нам нужно изучить его довольно простое математическое определение.

Оператор Харриса определяется локальными компонентами градиента интенсивности I_x, I_y в изображении. Определение оператора требует, чтобы область окна была определена и усреднялась $\langle . \rangle$, дабы занять все это окно. Начнем с вычисления следующей матрицы:

$$\Delta = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}. \tag{1.40}$$

Затем мы используем детерминант (det) и след (trace) для оценки углового сигнала:

$$C = \det \Delta / \operatorname{trace} \Delta. \tag{1.41}$$

(Опять же, что касается операторов Боде, значение использования только детерминанта и следа заключается в том, что результирующий сигнал будет инвариантным к угловой ориентации.)

Прежде чем приступить к анализу формы C, заметим, что если бы не проводилось усреднение, $\det \Delta$ был бы тождественно равен нулю: ясно, что только сглаживание, присущее операции усреднения, допускает разброс значений первой производной и тем самым позволяет результату частично зависеть от вторых производных.

Чтобы понять работу детектора в деталях, сначала рассмотрим его отклик для одиночного края (рис. 1.2a). Фактически здесь

$$\det \Delta = 0, \tag{1.42}$$

потому что I_x равен нулю во всей области окна.

Далее рассмотрим ситуацию в окрестностях угла (рис. 1.2b). Здесь:

$$\Delta = \begin{bmatrix} l_2 g^2 \sin^2\theta & l_2 g^2 \sin\theta\cos\theta \\ l_2 g^2 \sin\theta\cos\theta & l_2 g^2 \cos^2\theta + l_1 g^2 \end{bmatrix},$$

где l_1, l_2 – длины двух краев, ограничивающих угол, а g – контраст края, предполагаемый постоянным для всего окна. Теперь мы находим (Davies, 2005):

$$\det \Delta = l_1 l_2 g^4 \sin^2\theta, \tag{1.44}$$

а также

$$\operatorname{trace} \Delta = (l_1 + l_2) g^2; \tag{1.45}$$

$$\therefore \quad C = \frac{l_1 l_2}{l_1 + l_2} g^2 \sin^2\theta. \tag{1.46}$$

Это можно интерпретировать как произведение (1) коэффициента добротности λ, который зависит от длин кромок в пределах окна, (2) коэффициента контрастности g^2 и (3) коэффициента формы $\sin^2\theta$, который зависит от «резкости» края θ. Ясно, что C равно нулю при $\theta = 0$ и $\theta = \pi$ и максимально при $\theta = \pi/2$ – все эти результаты интуитивно верны и уместны.

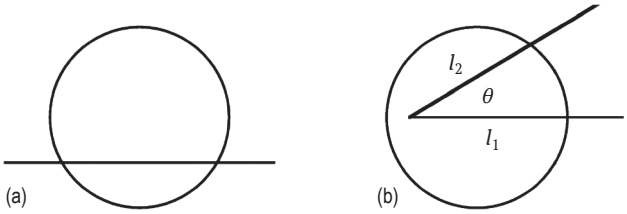


Рис. 1.2 ❖ Геометрическая иллюстрация расчета отклика линии и угла в круглом окне: (a) прямой край, (b) угол в общем виде. © IET 2005

Из этой формулы можно определить многие свойства оператора, в том числе тот факт, что пиковый сигнал возникает не в самом углу, а в центре окна, используемого для вычисления углового сигнала, хотя смещение уменьшается по мере того, как снижается острота угла.

1.3. Часть B. Локализация и распознавание двухмерных объектов

1.3.1. Подход к анализу формы на основе центроидного профиля

Двухмерные объекты обычно характеризуются формой их границ. В этом разделе мы рассмотрим, чего можно достичь, отслеживая границы объекта и анализируя полученные профили формы. Среди наиболее распространенных типов профилей, используемых для этой цели, выделяется *центроидный профиль*, в котором граница объекта наносится на карту с использованием полярных координат (r, θ), принимая центроид C границы за начало координат.

В случае круга радиуса R центроидный профиль представляет собой прямую линию на расстоянии R выше оси θ. На рис. 1.3 представлено пояснение, а также показаны два примера разбитых круглых объектов. В случае (a) окружность лишь слегка искривлена, и поэтому ее центроид C остается практически неизменным; следовательно, бóльшая часть центроидного графика остается на расстоянии R выше оси θ. Однако в случае (b) даже та

часть границы, которая не нарушена и не искажена, находится далеко не на постоянном расстоянии от оси θ: это означает, что объект невозможно узнать по его профилю, хотя в случае (a) нетрудно распознать его как слегка поврежденный круг. На самом деле мы уделяем столько внимания этим случаям в основном из-за того факта, что в случае (b) центроид смещается так сильно, что даже неизмененная часть формы не может быть немедленно распознана. Конечно, можно было бы попытаться исправить ситуацию, переместив центроид обратно в положение, соответствующее кругу, но это довольно сложная задача: во всяком случае, если исходная фигура не является кругом, много вычислений будет потрачено впустую до того, как станет понятна истинная природа проблемы.

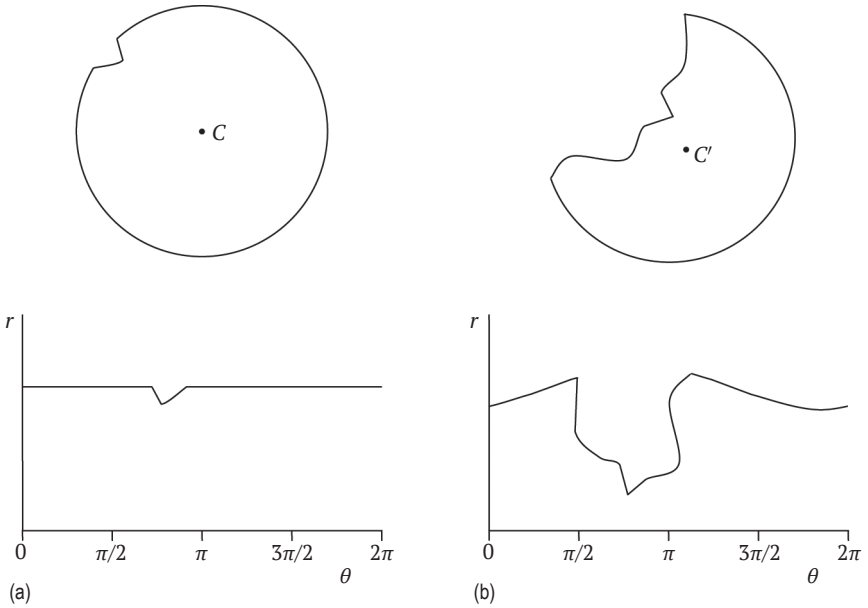


Рис. 1.3 ❖ Проблемы с дескриптором центроидального профиля: (a) представлен круглый объект с небольшим дефектом на его границе; под ним изображен соответствующий центроидный профиль; (b) представлен тот же объект, но на этот раз с грубым дефектом: поскольку центроид смещен в сторону C', *весь* профиль центроида сильно искажен

В целом мы можем заключить, что подход с центроидным профилем ненадежен и не рекомендуется. На самом деле это не означает, что его совсем не следует использовать на практике. Например, на конвейере для сыра или печенья любой предмет, который не распознается сразу по постоянному R-профилю, должен быть немедленно удален с конвейера; затем можно исследовать оставшиеся объекты более тщательно, чтобы убедиться, что их значения R приемлемы и демонстрируют надлежащую степень постоянства.

Робастность и ее значение

Не случайно здесь возникла идея *робастности*[1]. Она лежит в основе большей части дискуссий о ценности и эффективности алгоритмов, имеющих прямое отношение к компьютерному зрению. Основная проблема заключается в изменчивости объектов или любых иных сущностей, присущей компьютерным изображениям. Эта изменчивость может возникать по совершенно разным причинам: шум, различная форма объектов (даже одного и того же типа), различия в размере или расположении, трещины или дефекты, разное расположение камер и разные режимы просмотра. Кроме того, один объект может быть частично затенен другим или только частично находиться в определенном изображении (что дает эффекты, не отличающиеся от механического повреждения объекта).

Хотя хорошо известно, что шум влияет на точность измерения, можно подумать, что он с меньшей вероятностью повлияет на робастность. Однако нам необходимо отличать «обычный» тип шума, который мы можем описать как *гауссов шум*, от пикового или импульсного шума. Последние обычно описываются как выделяющиеся точки или «выбросы» в распределении шума. (Напомню, что мы уже видели, как медианный фильтр значительно лучше справляется с выбросами, чем средний фильтр.) Предметом *робастной статистики* является изучение темы нормальных значений и выбросов, а также то, как лучше всего справляться с различными типами шума. Исследования в этой области лежат в основе оптимизации точности измерения и достоверности интерпретации при наличии выбросов и грубых нарушений внешнего вида объекта.

Далее следует отметить, что существуют другие типы графических представлений границ, которые можно использовать вместо центроидного профиля. Один из них представляет собой график (s, ψ), а другой – производный профиль (s, κ). Здесь ψ – угол ориентации границы, а $\kappa(s)$, равный $d\psi/ds$, – локальная функция кривизны. Важно отметить, что эти представления не основаны на положении центроида, следовательно, его положение не нужно вычислять или даже оценивать. Несмотря на это преимущество, все такие представления граничных профилей имеют еще одну существенную проблему: если какая-либо часть границы закрыта, искажена или нарушена, сравнение формы объекта с шаблонами известной формы становится весьма затруднительным из-за разной длины границ.

Несмотря на эти проблемы, в подходящих ситуациях метод центроидного профиля имеет определенные преимущества, поскольку он способствует простоте измерения радиусов окружностей, легкости идентификации квадратов и других форм с выступающими углами и простому измерению ориентации, особенно для формы с выступающими углами.

Теперь осталось найти метод, который мог бы заменить метод центроидного профиля в тех случаях, когда могут возникать грубые искажения или *окклюзии* (загораживания одних объектов другими). В поисках такого метода мы переходим к следующему разделу, который знакомит с подходом преобразования Хафа.

---

[1] Под робастностью в статистике понимают нечувствительность к различным отклонениям и неоднородностям в выборке, связанным с теми или иными, в общем случае неизвестными, причинами. © academic.ru.

1.3.2. Схемы обнаружения объектов на основе преобразования Хафа

В разделе 1.3.1 мы рассмотрели, как круглые объекты могут быть идентифицированы по их границам с использованием подхода центроидного профиля к анализу формы. Этот подход оказался ненадежным из-за его неспособности справиться с грубыми искажениями формы и окклюзиями. В этом разделе мы покажем, что *преобразование Хафа* (Hough Transform) обеспечивает простой, но изящный способ решения данной проблемы. Используемый метод состоит в том, чтобы взять каждую краевую точку на изображении, переместить ее внутрь на расстояние R вдоль локальной нормали к краю и сохранить эту точку в отдельном изображении, называемом *пространством параметров*: R принимается за ожидаемый радиус кругов, которые должны быть локализованы. Результатом этого будет скопление точек (часто называемых «голосами») вокруг местоположений центров кругов. Фактически для получения точных оценок местоположений центров необходимо только найти значимые пики в пространстве параметров.

Этот процесс проиллюстрирован на рис. 1.4, из которого видно, что метод игнорирует некруглые части границы и идентифицирует только настоящие центры окружностей. Таким образом, подход фокусируется на данных, которые соответствуют выбранной модели, и не обращает внимания на нерелевантные данные, которые в противном случае приводят к значительному снижению робастности. Разумеется, данный метод зависит от точности оценки направлений нормалей к краям. К счастью, оператор Собеля способен оценивать ориентацию края с точностью до 1°, и его легко применять. Как показано на рис. 1.5, результаты могут быть весьма впечатляющими.

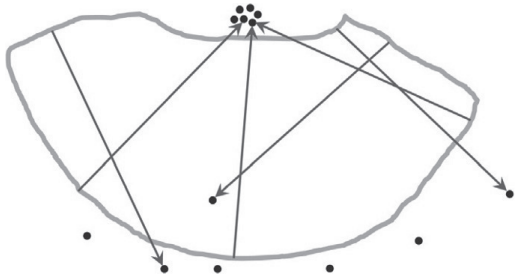


Рис. 1.4 ❖ Робастность преобразования Хафа при нахождении центра круглого объекта. Круглая часть границы дает центральные точки-кандидаты, которые фокусируются на истинном центре, тогда как неправильная ломаная граница дает центральные точки-кандидаты в случайных положениях. В данном случае граница примерно совпадает с границей сломанного печенья, показанного на рис. 1.5

Недостаток описанного выше подхода заключается в том, что ему требуется заранее известное значение R. Общее решение этой проблемы состоит в использовании трехмерного пространства параметров, в котором третье

измерение представляет возможные значения *R*, и последующем поиске наиболее значимых пиков в этом пространстве. Однако более простое решение включает в себя накопление результатов для диапазона вероятных значений *R* в одном и том же двумерном пространстве параметров – процедура, которая приводит к существенной экономии памяти и вычислений (Davies, 1988). На рис. 1.6 показан результат применения этой стратегии, которая работает как с положительными, так и с отрицательными значениями *R*. С другой стороны, в плоскости с одним параметром информация о радиальном расстоянии теряется из-за накопления всех голосов. Следовательно, потребуется дополнительная итерация процедуры для определения радиуса, соответствующего местоположению каждого пика.

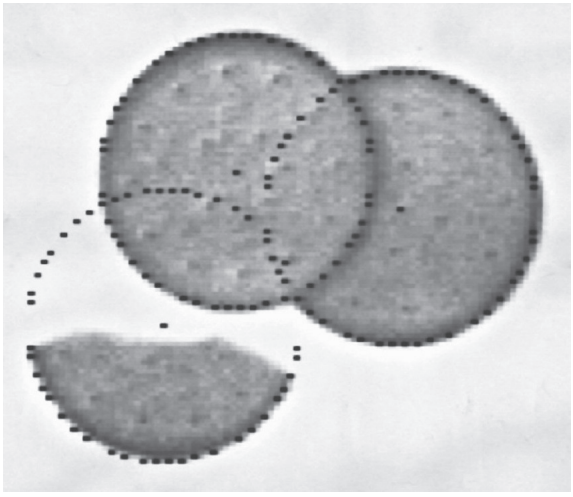


Рис. 1.5 ❖ Набор сломанных и перекрывающихся печений, демонстрирующий надежность метода определения центра. На точность метода указывают черные точки, каждая из которых находится в пределах 1/2 пикселя радиального расстояния от центра. © IFC 1984

Подход с преобразованием Хафа также можно использовать для обнаружения эллипса: два простых метода для этого случая представлены на рис. 1.7. Оба они воплощают непрямой подход, в котором используются *пары* краевых точек. В то время как *метод бисекции диаметра* требует значительно меньше вычислений, чем *метод хорд и касательных*, он более подвержен ложным обнаружениям, например когда два эллипса лежат рядом друг с другом на изображении.

Чтобы доказать правильность метода хорд и касательных, укажем на применимость этого метода для окружностей, а далее *свойство проективности* гарантирует, что он также сработает для эллипсов, потому что при ортогональной проекции прямые линии проецируются в прямые, средние точки в средние, касательные в касательные, а окружности в эллипсы; кроме того, всегда можно найти такую точку обзора, что окружность можно спроецировать на заданный эллипс.

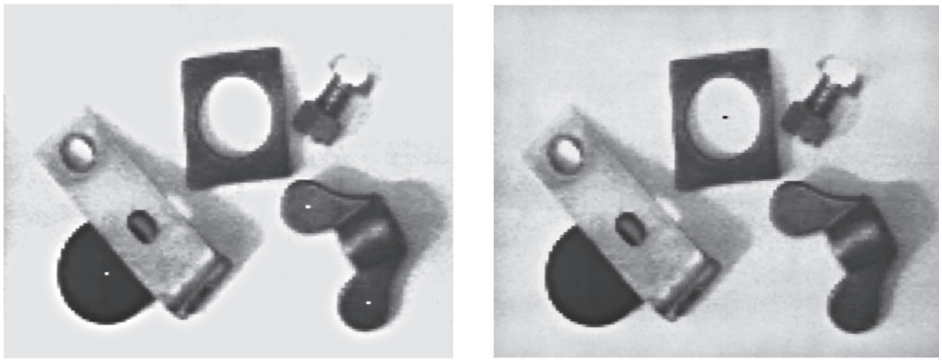


Рис. 1.6 ❖ Одновременное обнаружение объектов с разными радиусами: (a) обнаружение крышки объектива и барашковой гайки, когда предполагается, что радиусы находятся в диапазоне 4–17 пикселей; (b) обнаружение отверстий на том же изображении, когда предполагается, что радиусы попадают в диапазон от –26 до –9 пикселей (используются отрицательные радиусы, поскольку отверстия считаются объектами отрицательного контраста): ясно, что на *этом* изображении мог быть применен меньший диапазон отрицательных радиусов.

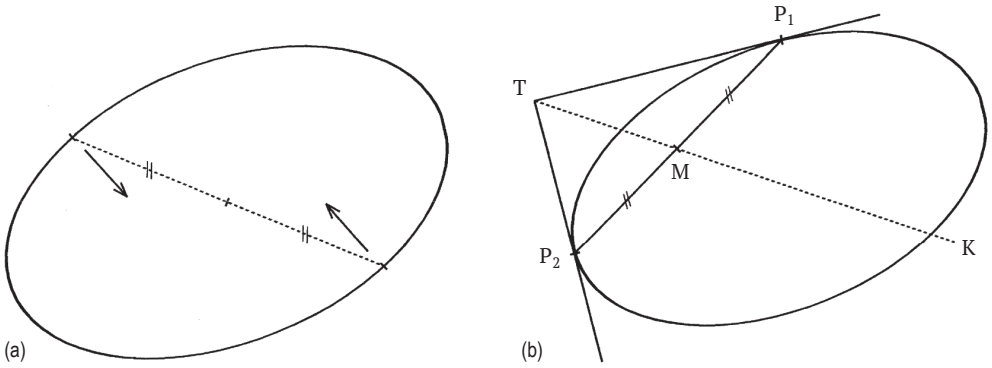


Рис. 1.7 ❖ Геометрическое представление двух методов обнаружения эллипсов: (a) в методе бисекции диаметра находят пару точек, для которых ориентации ребер антипараллельны. Середины таких пар накапливаются, и полученные пики принимаются за центры эллипсов; (b) в методе хорд и касательных касательные в точках P_1 и P_2 пересекаются в точке T, а середина отрезка P_1P_2 находится в точке M. Центр эллипса C лежит на полученной линии TM

Теперь мы переходим к так называемому *обобщенному преобразованию Хафа* (generalized Hough transform, GHT), которое использует более прямую процедуру обнаружения эллипса, чем два других метода, описанных выше.

Чтобы понять, как обобщается стандартный метод Хафа для локализации объектов произвольной формы, нам сначала нужно выбрать точку локализации L в шаблоне идеализированной формы. Затем нам нужно сделать так, чтобы вместо перемещения от краевой точки на фиксированное расстояние

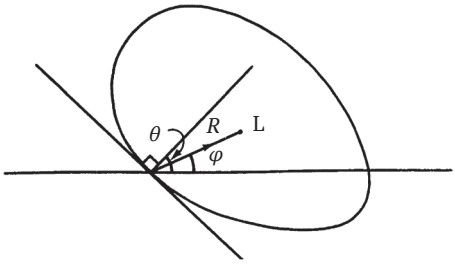


Рис. 1.8 ❖ Вычисление обобщенного преобразования Хафа

R непосредственно вдоль локальной нормали от края до центра, как в случае с окружностями, мы перемещались на соответствующее *переменное* расстояние R в *переменном* направлении φ так, чтобы прийти к L; R и φ теперь являются функциями направления нормали к локальному краю θ (рис. 1.8). В этих условиях голоса будут иметь пик в заранее выбранной точке локализации объекта L. Функции $R(\theta)$ и $\varphi(\theta)$ могут быть представлены аналитически в компьютерном алгоритме, а для совершенно произвольных форм они могут быть сохранены в виде интерполяционных таблиц. В любом случае схема основана на очень простом принципе, но при обобщении метода Хафа возникает важное усложнение, потому что мы переходим от изотропной формы (круг) к анизотропной форме, которая может иметь совершенно произвольную ориентацию.

Это означает добавление дополнительного измерения в пространство параметров (Ballard, 1981). Затем каждая точка края вносит свой вклад в набор голосов в каждой плоскости ориентации в пространстве параметров. Наконец, все пространство параметров просматривается в поисках пиков – наивысших точек, указывающих как на расположение объектов, так и на их ориентацию. Интересно, что GHT может обнаруживать эллипсы, используя одну плоскость в пространстве параметров, за счет применения *функции точечной экстраполяции* (point spread function, PSF) к каждой краевой точке, которая учитывает все возможные ориентации эллипса: обратите внимание, что PSF применяется на некотором расстоянии от краевой точки, чтобы центр PSF мог пройти через центр эллипса (рис. 1.9). Ограниченный объем главы не позволяет представить здесь детали вычислений (например, см. Davies, 2017, глава 11).

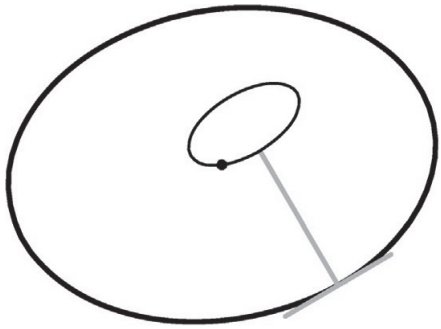


Рис. 1.9 ❖ Использование формы PSF, учитывающей все возможные ориентации эллипса. PSF позиционируется серыми вспомогательными линиями так, чтобы она проходила через центр эллипса (черная точка)

1.3.3. Применение преобразования Хафа для обнаружения линий

Преобразование Хафа (HT) также может применяться для обнаружения линий. Ранее было отмечено, что лучше избегать обычного уравнения с коэффициентом наклона и точкой пересечения вида $y = mx + c$, потому что для почти вертикальных линий требуются почти бесконечные значения m и c. Вместо этого использовалась «нормальная» (θ, ρ) форма прямой линии (рис. 1.10):

$$\rho = x\cos\theta + y\sin\theta. \tag{1.47}$$

Для применения метода в этой форме множество прямых, проходящих через каждую точку P_i, представляют в виде множества синусоид в пространстве (θ, ρ): например, для точки $P_1(x_1, y_1)$ синусоида имеет уравнение:

$$\rho = x_1\cos\theta + y_1\sin\theta. \tag{1.48}$$

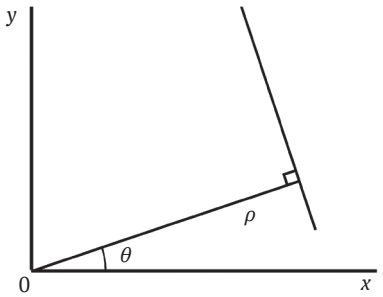


Рис. 1.10 ❖ Нормальная параметризация прямой линии в пространстве (θ, ρ)

После накопления голосов в пространстве (θ, ρ) пики указывают на наличие линий в исходном изображении.

Была проделана большая работа (см., например, Dudani, Luk, 1978) для ограничения погрешностей определения местоположения линии, возникающих по разным причинам: шум, дискретизация, эффекты фрагментации линии, эффекты небольшой кривизны линии, сложность оценки точных положений пиков в пространстве параметров. Кроме того, важна проблема локализации продольной линии. Для последнего из этих процессов Дудани и Лук (Dudani, Luk, 1978) разработали метод «*xy*-группировки», который предусматривал проведение анализа связности для каждой линии. Затем сегменты линии подлежали объединению, если они разделены промежутками менее ~5 пикселей. Наконец, сегменты короче определенной минимальной длины (также обычно ~5 пикселей) игнорировались как слишком незначительные, чтобы облегчить интерпретацию изображения.

В целом мы видим, что все описанные выше формы HT значительно выигрывают благодаря наличию механизма *накопления доказательств* (accumulating evidence) с использованием схемы голосования. Этот механизм является источ-

ником высокой робастности метода. Вычислительные процессы, используемые НТ, можно описать скорее как индуктивные, а не дедуктивные, поскольку наличие пиков приводит к *гипотезам* о присутствии объектов, которые в принципе должны быть подтверждены другими доказательствами, тогда как *дедукция* привела бы к немедленному доказательству присутствия объектов.

1.3.4. Использование RANSAC для обнаружения линий

RANSAC – это альтернативная схема поиска на основе моделей, которую часто можно использовать вместо НТ. Дело в том, что она очень эффективно работает при обнаружении линий, поэтому заслуживает отдельного внимания. Стратегию поиска можно рассматривать как схему голосования, но она используется иначе, чем в НТ. Она выдвигает последовательность гипотез о целевых объектах и определяет поддержку каждой из них, подсчитывая, сколько точек данных согласуется с ними в разумных (например, $\pm 3\sigma$) пределах (см. рис. 1.11). Как и следовало ожидать, для любого потенциального искомого объекта на каждом этапе сохраняются только гипотезы с максимальной поддержкой.

Давайте разберем, как RANSAC используется для обнаружения линий. Как и в случае с НТ, мы начинаем с применения детектора краев и определения местоположения всех краевых точек на изображении. Как мы увидим, RANSAC лучше всего работает с ограниченным количеством точек, поэтому полезно найти краевые точки, которые являются локальными максимумами градиента интенсивности изображения. Далее, все, что необходимо, чтобы сформулировать гипотезу прямой линии, – это взять любую пару граничных точек. Для каждой гипотезы мы проходим по списку N краевых точек, определяя, сколько точек M поддерживает гипотезу. Затем мы берем другие гипотезы (другие пары краевых точек) и на каждом этапе оставляем только ту, которая дает максимальную поддержку $M_{\max}$. Этот процесс показан в листинге 1.1.

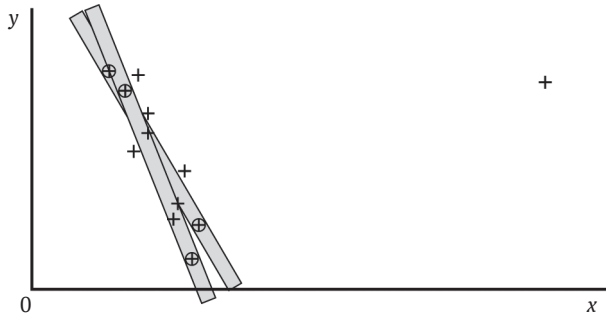


Рис. 1.11 ❖ Метод RANSAC. Здесь знаки + указывают точки данных, по которым нужно попытаться подогнать линии, а также показаны два экземпляра пар точек данных (обозначенных знаками ⊕), через которые проведены гипотетические линии. Каждая предполагаемая линия имеет область допуска $\pm t$, в пределах которой ищется поддержка максимального количества точек данных. Линия с наибольшей поддержкой считается наиболее подходящей

Листинг 1.1 ❖ Базовый алгоритм RANSAC для поиска линии с наибольшей поддержкой. Этот алгоритм возвращает только одну линию; точнее, он возвращает модель линии, которая имеет наибольшую поддержку. Линии с меньшей поддержкой в итоге игнорируются

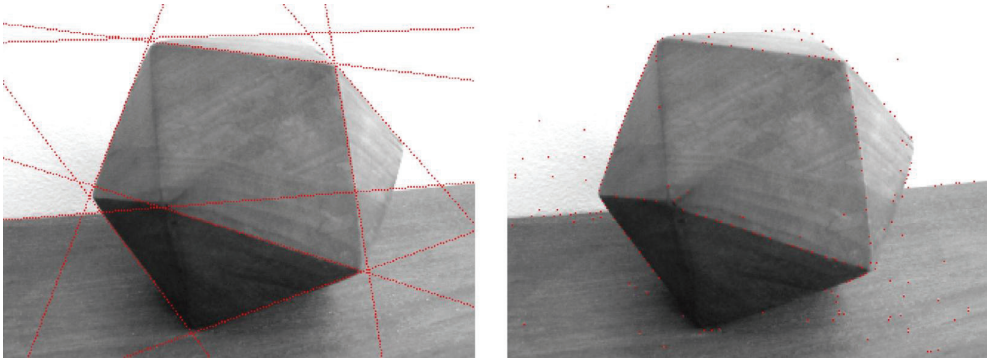
```
Mmax=0;
для всех пар краевых точек {
  найти уравнение линии, определяемое двумя точками i, j;
  M = 0;
  для всех N точек в списке
    если (точка k находится в пределах порогового расстояния d от линии) M++;
  если (M > Mmax) {
    Mmax = M;
    imax = i;
    jmax = j;
    // это гипотеза, имеющая максимальную поддержку на данный момент
  }
}
/* если Mmax > 0, (x[imax], y[imax]) и (x[jmax], y[jmax]) будут координатами точек,
определяющих линию с наибольшей поддержкой */
```

Алгоритм в листинге псевдокода 1.1 соответствует поиску центра самого высокого пика в пространстве параметров, как и в случае HT. Чтобы найти все линии на изображении, наиболее очевидной стратегией является следующая: найти первую линию, затем удалить все точки, поддерживающие ее; потом найти следующую линию и устранить все точки, поддерживающие ее; повторять, пока все точки не будут исключены из списка. Процесс может быть записан более компактно в таком виде:

```
повторить {
  найти линию;
  удалить поддерживающие точки;
}
пока не закончатся точки данных;
```

Как сказано выше, RANSAC предполагает довольно значительную вычислительную нагрузку, составляющую $O(N^3)$, по сравнению с $O(N)$ для алгоритма HT. Следовательно, при использовании RANSAC лучше каким-то образом уменьшить N. Это объясняет, почему полезно сосредоточиться на локальных максимумах, а не использовать полный список краевых точек. Однако в качестве альтернативы можно использовать повторную случайную выборку из полного списка до тех пор, пока не будет проверено достаточное количество гипотез, чтобы быть уверенным в том, что обнаружены все значимые линии. К слову, эти идеи отражают первоначальное значение аббревиатуры RANSAC, которая расшифровывается как RANdom SAmpling Consensus – консенсус с произвольными данными, в том смысле, что любая гипотеза должна формировать консенсус с доступными подтверждающими данными (Fischler, Bolles, 1981). Степень уверенности в том, что все значимые линии обнаружены, можно вычислить как обратную величину риска того, что значимая линия будет пропущена из-за пропуска репрезентативной пары точек, лежащих на линии.

Теперь мы можем рассмотреть результаты, полученные путем применения RANSAC к частному случаю поиска прямых линий. В описанном тесте в качестве гипотез использовались пары точек, а все краевые точки представляли собой локальные максимумы градиента интенсивности. Случай, показанный на рис. 1.12, соответствует обнаружению деревянного бруска в форме икосаэдра. Обратите внимание, что одна линия справа на рис. 1.12a была пропущена, потому что пришлось установить нижний предел уровня поддержки для каждой линии: это было необходимо, потому что ниже этого уровня поддержки количество случайных коллинеарностей резко возрастало даже для относительно небольшого числа краевых точек, показанных на рис. 1.12b, что приводит к резкому увеличению числа ложноположительных линий. В целом этот пример показывает, что RANSAC является очень важным претендентом на определение местоположения прямых линий в цифровых изображениях. Здесь не обсуждается тот факт, что RANSAC полезен для получения надежной подгонки ко многим другим типам форм как в 2D, так и в 3D.



(a) (b)

Рис. 1.12 ❖ Обнаружение прямых линий с использованием метода RANSAC: (a) исходное изображение в оттенках серого с прямыми краевыми линиями, обнаруженными с использованием метода RANSAC: (b) краевые точки, переданные в RANSAC для получения (a): это были локальные максимумы градиентов изображения. В (a) пропущены три ребра икосаэдра. Это потому, что они представляют собой края с низким контрастом и низким градиентом интенсивности. Фактически RANSAC также упустил четвертый край из-за наличия нижнего предела уровня поддержки (см. текст выше)

Наконец, следует упомянуть, что RANSAC менее, чем HT, подвержен влиянию алиасинга (ступенчатого искажения) вдоль прямых линий. Это связано с тем, что пики HT, как правило, фрагментируются из-за алиасинга, поэтому наилучшие гипотезы трудно получить без агрессивного сглаживания изображения. Причина, по которой RANSAC выигрывает в этом контексте, заключается в том, что он полагается не на точность отдельных гипотез, а скорее на их количество: стратегия исходит из того, что достаточное количество гипотез можно легко генерировать и столь же легко отбрасывать.