Visual Odometry

Part II: Matching, Robustness, Optimization, and Applications

Колесо Одометрии ?????????????????

**Visual Odometry** is the process of determining the position and orientation of a robot by analyzing the associated camera images.

**Визуальная Одометрия \***

**Odometry** is the use of data from motion sensors to estimate change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location.

**Одометрия \***

**Egomotion** is defined as the 3D motion of a camera within an environment. In the field of computer vision, egomotion refers to estimating a camera's motion relative to a rigid scene. An example of egomotion estimation would be estimating a car's moving position relative to lines on the road or street signs being observed from the car itself. The estimation of egomotion is important in autonomous robot navigation applications.

определяется как 3D-движение камеры внутри среды. В области компьютерного зрения egomotion относится к оценке движения камеры относительно жесткой сцены. Примером оценки egomotion будет оценка движущегося положения автомобиля относительно линий на дорогах или уличных знаков, наблюдаемых от самого автомобиля. Оценка egomotion важна в автономных навигационных приложениях для роботов. [18]

Визуальная Одометрия – Оценка положения – это процесс оценки движения камеры в среде с помощью средств передвижения (транспортное ср-во, человек или робот) использующий в качестве входных данных одну или несколько камер, прикрепленных к одному их этих средств. Прикладные области включают робототехнику, носимые вычислю устройства, дополненную реальность, автомобили. Термин Визуальная Одометрия был поляризован в 2004 году Нистером в его революционной статье [1], но уже были изданы более ранние работы [88], [89]. Этот термин был выбран из-за его схожести с методом Колеса Одометрии, который постепенно оценивает движение транспортного ср-ва с помощью встроенного счетчика вращений этого колеса в течении времени. Аналогичным образом, метод Визуальной Одометрии управляется постепенной оценкой положения транспортного средства через проверку изменений, которые производятся движением на картинках от бортовых камер. Для эффективной работы метода Визуальной Одометрии должна быть достаточная освещенность окружающей среды и статичная сцена с достаточными текстурами, чтобы очевидное действие могло быть определено. Более того, последовательность кадров должна быть зафиксирована с обеспечением того, что они имеют достаточное перекрытие сцены. Преимущество метода Визуальной Одометрии, с уважением к методу Колесо Одометрии, в том, что метод Визуальной Одометрии не подвержен влиянию неровной местности или других неблагоприятных условий.

Это демонстрирует, что по сравнению с методом Колеса Одометрии, метод Визуальной Одометрии обеспечивает более точную оценку траектории, с вероятностью ошибки в местоположении от 0.1 до 2%. Это возможность сделать метод Визуальной Одометрии интересным дополнением к методу Колеса Одометрии и, дополнительно, другие навигационные системы, такие как глобальная система местоположений(GPS), внутренняя система измерений (IMUs), и метод Лазерной Одометрии (похож на метод Визуальной Одометрии, Лазерная Одометрия оценивает движение камеры транспортного средства внутри среды путем сопоставления последовательных лазерных сканов). При отказе GPS в окружающих средах, таких как подводная или воздушная, Визуальная Одометрия имеет крайнюю важность.

Две части этого пособия и их обзор обеспечат широкое введение в метод Визуальной Одометрии и исследование, которое проводилось с 1980 по 2011 год. Хотя первые два десятилетия доказывают множество независимых реализаций, только во время третьего десятилетия сделали преуспевающую систему, работающую в реальном времени, что привело к тому, что метод Визуальной Одометрии будет использоваться на других планетах двумя Марсоходами-вездеходами впервые. Первая часть повествовала о историческом обзоре первых 30 лет исследований в этой области, дискуссии о моделировании камер и их настройке, и описание оценки-движения главного трубопровода для двух схем – монокулярная и бинокулярная, описание плюсов и минусов каждой реализации [87].

In monocular VO, you can only say that you moved one unit in x, two units in y, and so on, while in stereo, you can say that you moved one meter in x, two meters in y, and so on.

stereo VO is usually much more robust (due to more data being available). But, in cases where the distance of the objects from the camera are too high ( as compared to the distance between to the two cameras of the stereo system), the stereo case degenerates to the monocular case.

[ https://avisingh599.github.io/vision/visual-odometry-full/ ]

Часть II (этот учебник) касается соответствия функций, надежности и приложений. В нем рассматриваются основные детекторы функции точки, используемые в VO, и различные схемы исключения выбросов. Особое внимание уделяется случайному выборочному консенсусу (RANSAC) и обсуждаются стратегии, направленные на его ускорение. Другие затронутые темы: моделирование ошибок, обнаружение петли (или распознавание местоположения) и настройка связки. Также даются ссылки на онлайн-код, готовый к использованию. Математические обозначения и понятия, используемые в этой статье, определены в части I этого урока и поэтому не повторяются здесь.

Выбор и согласование функций.

 Существует два основных подхода к поиску элементов и их соответствий. Первый заключается в поиске особенностей на одном изображении и отслеживании их на следующих изображениях с использованием методов локального поиска, таких как корреляция. Второй заключается в том, чтобы независимо определять характерные черты во всех изображениях и сопоставлять их на основе некоторой метрики подобия между их дескрипторами. Первый подход более подходит, когда изображения берутся с близлежащих точек зрения, тогда как последний более подходит, когда ожидается большое изменение движения или обзора. Ранние исследования в области VO были выбраны для прежнего подхода [2] - [5], а работы последнего десятилетия были сосредоточены на последнем подходе [1], [6] - [9]. Причина в том, что ранние работы были задуманы для небольших сред, где изображения были взяты с близлежащих точек зрения, в то время как в последние несколько десятилетий фокус переместился в крупномасштабные среды, и поэтому изображения воспринимаются как можно более далеко друг от друга от каждого, чтобы ограничить связанные с движением проблемы.

**Обнаружение особенностей**

Во время этапа обнаружения особенности на изображении просматриваются важные ключевые точки, которые могут хорошо совпадать с другими изображениями. Локальная черта, - это шаблон изображения, который отличается от его непосредственной окрестности по интенсивности, цвету и текстуре. Для VO точечные детекторы, такие как углы или капли, важны, потому что их положение в изображении можно точно измерить.

Угол определяется как точка на пересечении двух или более ребер. Капля - это шаблон изображения, который отличается от его непосредственной окрестности по интенсивности, цвету и текстуре. Это не край, ни угол. Привлекательные свойства, которые должны иметь детектор хорошего качества, это: точность локализации (как по положению, так и по масштабу), повторяемость (т. Е. Большое количество функций должно быть пересмотрено на следующих изображениях), вычислительная эффективность, надежность (к шуму, артефактам сжатия , размытие), отличительная особенность (так что функции могут быть точно сопоставлены на разных изображениях) и инвариантность (как для фотометрических (например, для освещения), так и для геометрических изменений [поворот, масштаб (масштабирование), искажение перспективы]}.

В литературе VO есть много точечных детекторов, таких как угловые детекторы (например, Moravec [2], Forstner [10], Harris [11], Shi-Tomasi [12] и FAST [13]) и блочные детекторы (SIFT [14], SURF [15] и CENSURE [16]). Обзор этих детекторов можно найти в [17]. Каждый детектор имеет свои плюсы и минусы. Угловые детекторы быстро вычисляются, но менее отличительны, в то время как блочные детекторы более отличительны, но медленнее для обнаружения. Кроме того, углы лучше локализованы в положении изображения, чем капли, но менее локализованы в масштабе. Это означает, что углы не могут быть повторены так часто, как капли после больших изменений масштаба и точки обзора. Тем не менее, капли не всегда являются правильным выбором в некоторых средах - например, SIFT автоматически игнорирует углы, в которых городские среды чрезвычайно богаты. По этим причинам выбор соответствующего детектора функции должен быть тщательно рассмотрен в зависимости от вычислительных ограничений, требований в реальном времени, типа среды и базовой линии движения (т. Е. Того, как снимаются соседние изображения). Примерное сравнение свойств и характеристик различных угловых и блочных детекторов приведено на рисунке 1. Обратите внимание, что SIFT, SURF и CENSURE не являются истинными аффинно-инвариантными детекторами, но эмпирически найдены как инвариантные до определенных изменений точки зрения. Оценка эффективности детекторов функций и дескрипторов для внутреннего VO приведена в [18] и для наружных сред в [9] и [19].

Каждый детектор признаков состоит из двух этапов. Во-первых, применить функцию функции-ответа на всем изображении [такую ​​как функция углового ответа в детекторе Харриса или оператор разностного гауссова (DoG) SIFT]. Второй шаг - применить не максимальное подавление на выходе первого шага. Цель состоит в том, чтобы идентифицировать всю локальную минимальную (или максимальную) функцию функции-ответа. Вывод немаксимального подавления представляет собой обнаруженные признаки. Трюк, чтобы сделать детектор инвариантным для масштабирования.

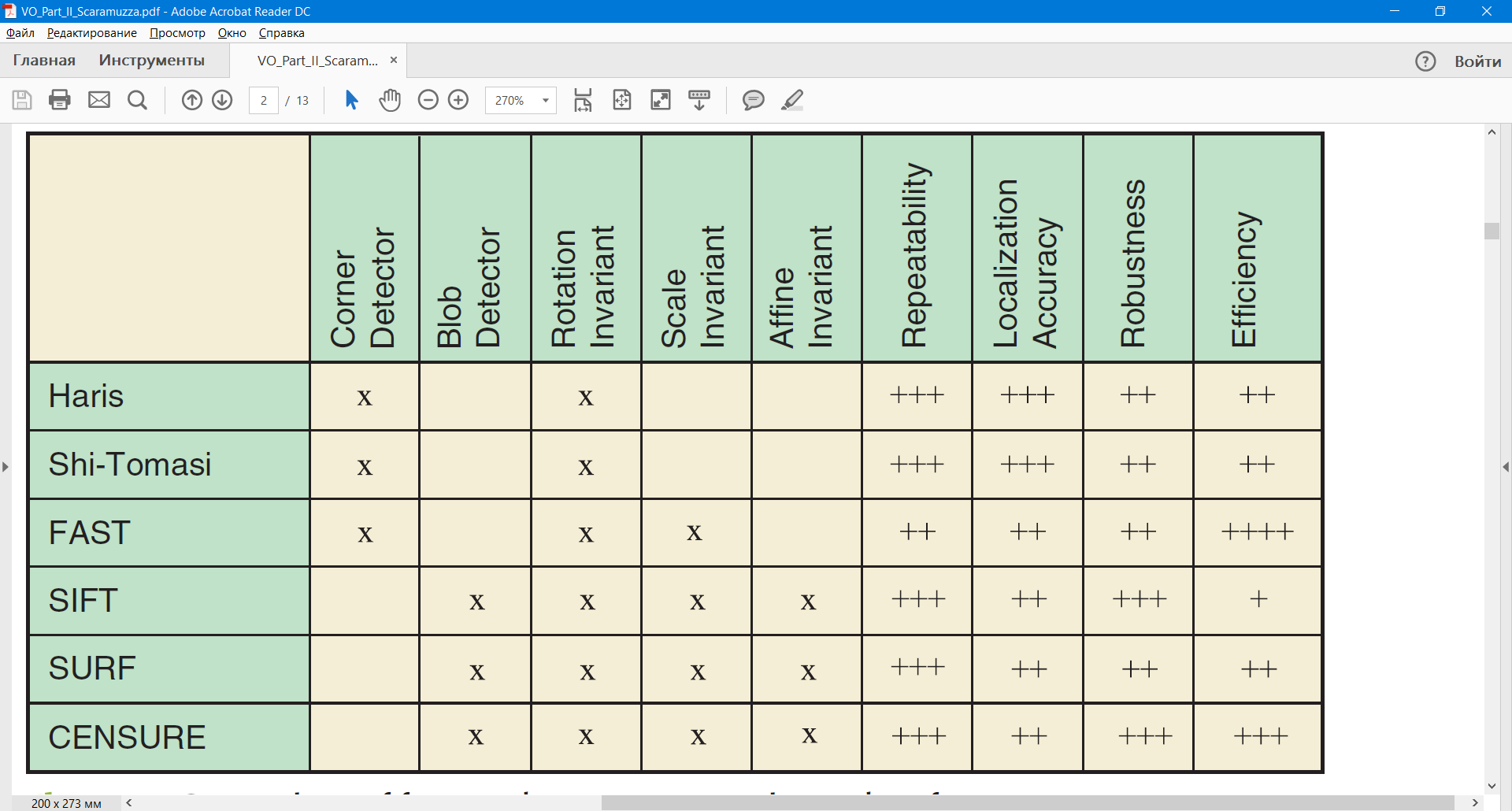


Рисунок 1. Сравнение характеристик детекторов: свойства и производительность.

Трюк в том, чтобы сделать детектор инвариантным к масштабированию, заключается в применении детектора в более ранних и поздних версиях одного и того же изображения [Рисунок 2 (a)]. Вместо этого инвариантность к перспективным изменениям достигается путем аппроксимации перспективного искажения как аффинного[[1]](#footnote-1).

SIFT[[2]](#footnote-2) – это функция, предназначенная для распознавания объектов и мест и найденная для выдающихся результатов. Детектор SIFT запускается путем свертывания верхней и нижней шкалы изображения с помощью DoG-оператора, а затем берет локальные минимумы или максимумы выхода в масштабах и пространстве (рис. 2). Сила SIFT - это надежный дескриптор, который будет объяснен в следующем разделе. Детектор SURF основывается на SIFT, но использует коробчатые фильтры для аппроксимации гаусса, что приводит к более быстрому вычислению по сравнению с SIFT, что достигается с помощью интегральных изображений [90].

**Дескриптор особенности(черты)**

На этапе описания черты область вокруг каждой обнаруженной черты преобразуется в компактный дескриптор, который можно сопоставить с другими дескрипторами. Простейшим дескриптором черты является ее внешний вид, то есть интенсивность пикселей в области вокруг точки объекта. В этом случае для сравнения интенсивностей можно использовать метрики ошибок, такие как сумма квадратов разностей (SSD) или нормализованная взаимная корреляция (NCC) [20]. В отличие от SSD, NCC хорошо компенсирует небольшие изменения яркости. Альтернативной и более надежной мерой подобия изображения является преобразование переписи [21], которое преобразует каждый патч изображения в двоичный вектор, представляющий, какие соседи имеют свою интенсивность выше или ниже интенсивности центрального пикселя. Сходство патчей измеряется через расстояние Хэмминга.

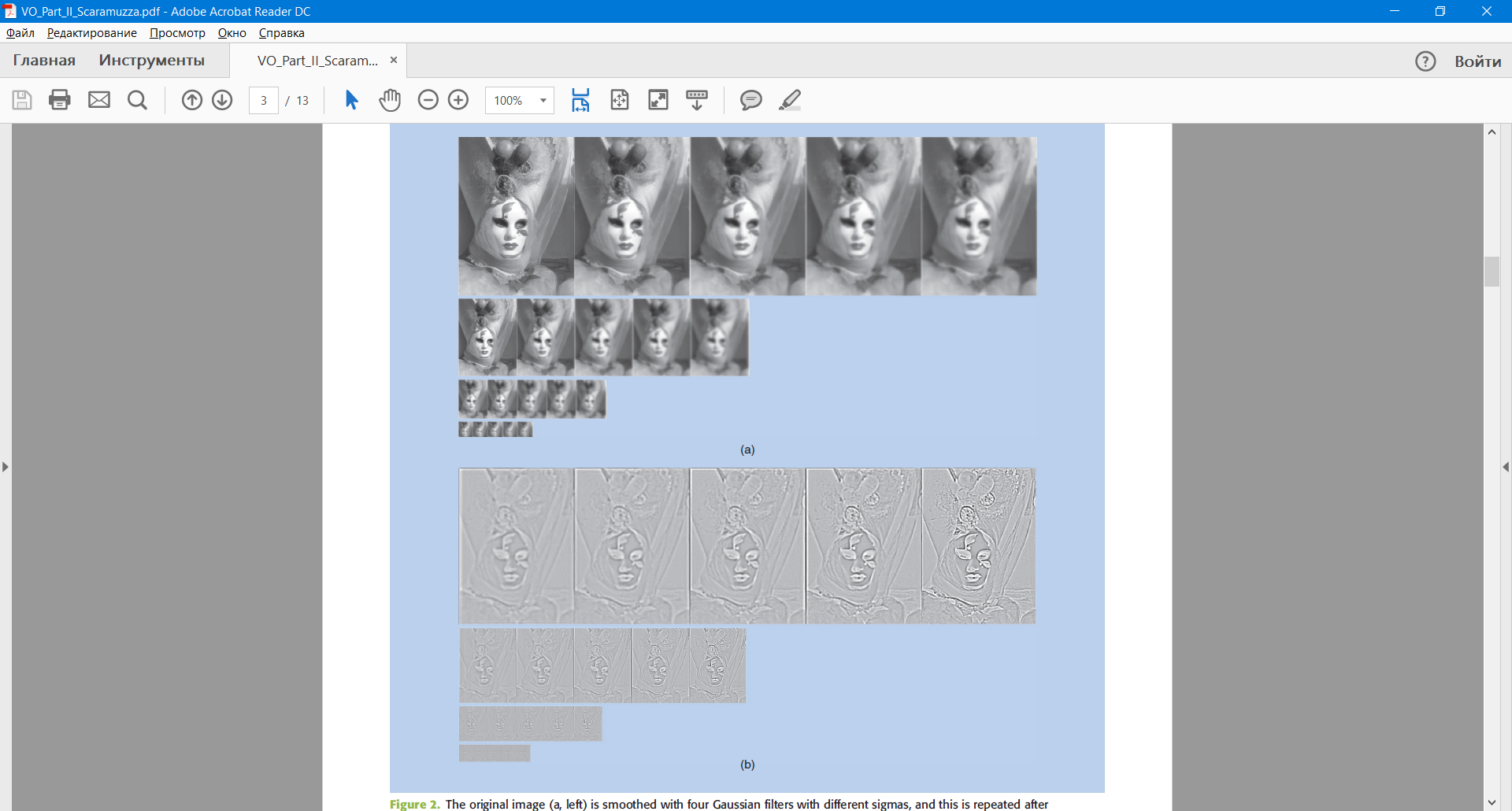


Рисунок 2. Исходное изображение (a, слева) сглаживается четырьмя гауссовскими фильтрами с разными сигмами, и это повторяется после понижающей дискретизации изображения второго фактора. Наконец, (б) изображения DoG вычисляются, принимая разницу между последовательными гауссовскими сглаженными изображениями. Функции SIFT можно найти в виде локальных минимумов или максимумов изображений DoG в масштабах и пространстве.

Во многих случаях локальный внешний вид этой функции не является хорошим дескриптором информации, передаваемой функцией, поскольку ее внешний вид будет изменяться с ориентацией, масштабом и изменениями точки зрения. На самом деле, SSD и NCC не являются инвариантными ни к одному из этих изменений, и поэтому их использование ограничено изображениями, снятыми на близлежащих позициях. Одним из самых популярных дескрипторов для точечных объектов является SIFT. SIFT-дескриптор - это в основном гистограмма локальных градиентных ориентаций. Патч вокруг функции разбивается на сетку 434. Для каждого квадранта строится гистограмма восьми градиентных ориентаций. Затем все эти гистограммы объединяются вместе, образуя 128-элементный дескрипторный вектор. Чтобы уменьшить влияние изменений освещения, дескриптор затем нормализуется до единицы длины. Дескриптор SIFT оказался устойчивым к изменениям освещения, вращения и масштаба, и даже до 60? изменения в точке зрения. Пример возможностей SIFT показан на рисунке 3. Отображается ориентация и масштаб каждой функции. Дескриптор SIFT может, вообще говоря, вычисляться для угловых или blob-функций; однако его производительность будет уменьшаться по углам, потому что по определению углы встречаются на пересечении ребер. Поэтому его дескриптор не будет таким отличительным, как для капли, которые, наоборот, лежат в сильно текстурированных областях изображения. В период с 2010 по 2011 год было разработано три новых дескриптора, которые намного быстрее вычисляются, чем SIFT и SURF. Простой двоичный дескриптор с именем BRIEF [22] стал популярным: он использует парные сравнения яркостей, взятые из патча вокруг ключевой точки. Несмотря на то, что он чрезвычайно быстро извлекается и сравнивается, он по-прежнему демонстрирует высокую дискриминационную силу в отсутствие поворота и изменения масштаба. Вдохновленный его успехом был разработан ORB [23], в котором рассматривается ориентационная инвариантность и оптимизация схемы выборки для пар значений яркости. В то же время BRISK [24] предоставляет детектор ключевых точек на основе FAST, который допускает инвариантность масштаба и вращения, и двоичный дескриптор, который использует настраиваемый шаблон выборки.

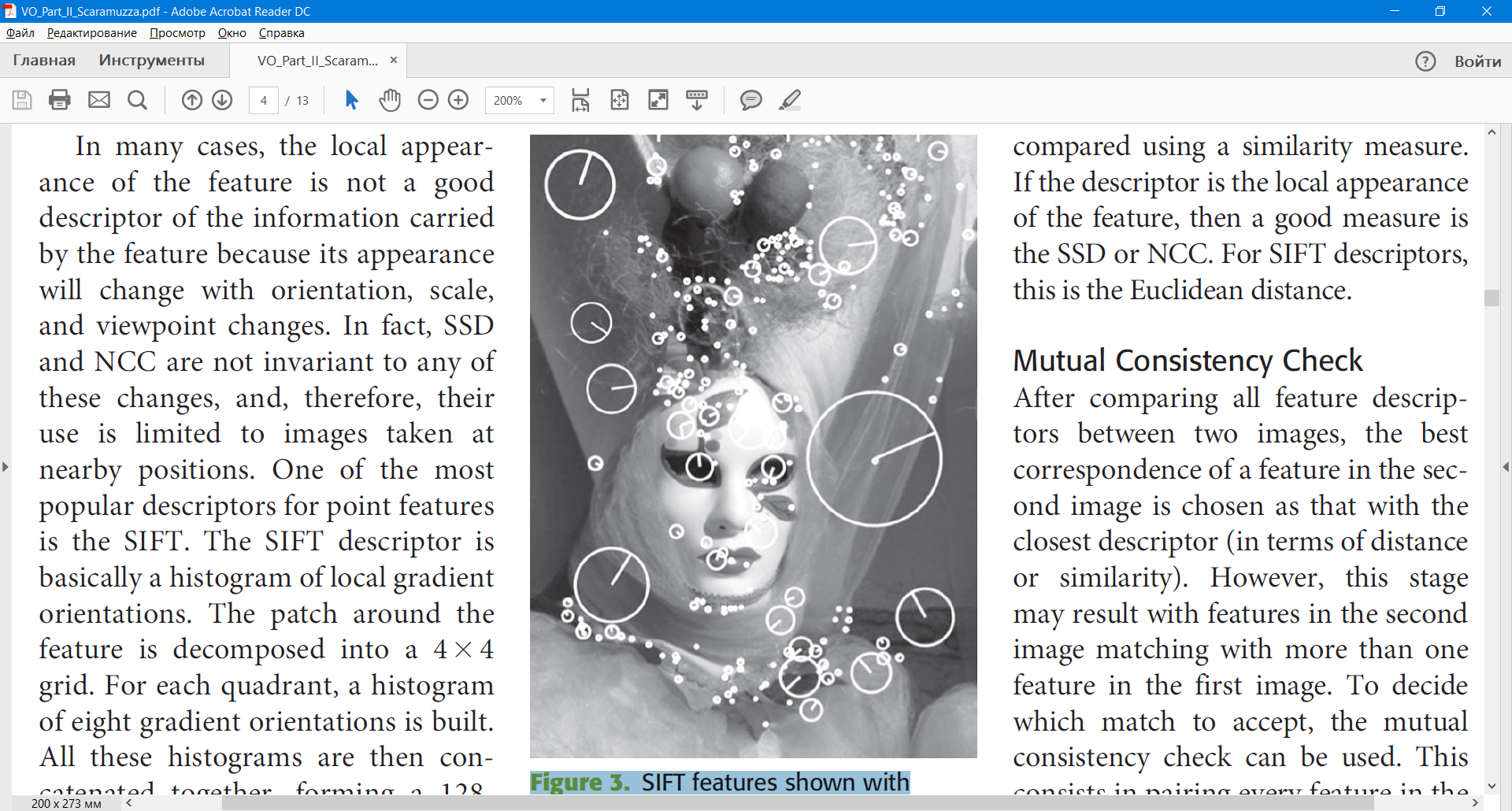


Рисунок 3. Функции SIFT, показанные с ориентацией и масштабом.

**Соответствие функций**

Шаг согласования функций ищет соответствующие функции в других изображениях. На рисунке 4 показаны функции SIFT, согласованные между несколькими кадрами, наложенными на первое изображение. Набор совпадений, соответствующих одной и той же функции, называется дорожкой функций. Самый простой способ сопоставления функций между двумя изображениями - сравнить все дескрипторы функций в первом изображении со всеми другими дескрипторами функций во втором изображении. Дескрипторы сравниваются с использованием меры подобия. Если дескриптор является локальным видом функции, то хорошей мерой является SSD или NCC. Для дескрипторов SIFT это евклидово расстояние.

Проверка взаимной согласованности

После сравнения всех дескрипторов функций между двумя изображениями наилучшее соответствие функции во втором изображении выбирается так, как с ближайшим дескриптором (с точки зрения расстояния или сходства). Однако этот этап может привести к тому, что функции второго изображения совпадают с более чем одним признаком на первом изображении. Чтобы решить, какое совпадение принять, можно использовать проверку взаимной согласованности. Эта

состоит в объединении каждой функции во втором изображении с функциями первого изображения. В качестве правильных принимаются только пары соответствующих функций, которые взаимно имеют друг друга в качестве предпочтительного соответствия.

Ограниченное соответствие

Недостатком этого исчерпывающего соответствия является то, что он является квадратичным по числу признаков, что может стать непрактичным, когда количество функций велико (например, несколько тысяч). Лучшим подходом является использование структуры индексирования, такой как многомерное дерево поиска или хеш-таблица, для быстрого поиска объектов рядом с данной функцией. Более быстрое сопоставление функций - поиск потенциальных соответствий в

регионов второго изображения, где они ожидаются. Эти регионы могут быть предсказаны с использованием модели движения и трехмерного (трехмерного) позиционного положения (если доступно). Например, это имеет место в оценке движения на основе 3-D-to-2-D, описанной в части I этого руководства. Движение может быть задано дополнительным датчиком, таким как ИДУ, одометром колеса [25], лазером и GPS, или может быть выведено из предыдущей позиции, предполагая модель с постоянной скоростью, как это предложено в [26]. Затем предсказанная область вычисляется как эллипс ошибки из неопределенности движения и точки 3-D.

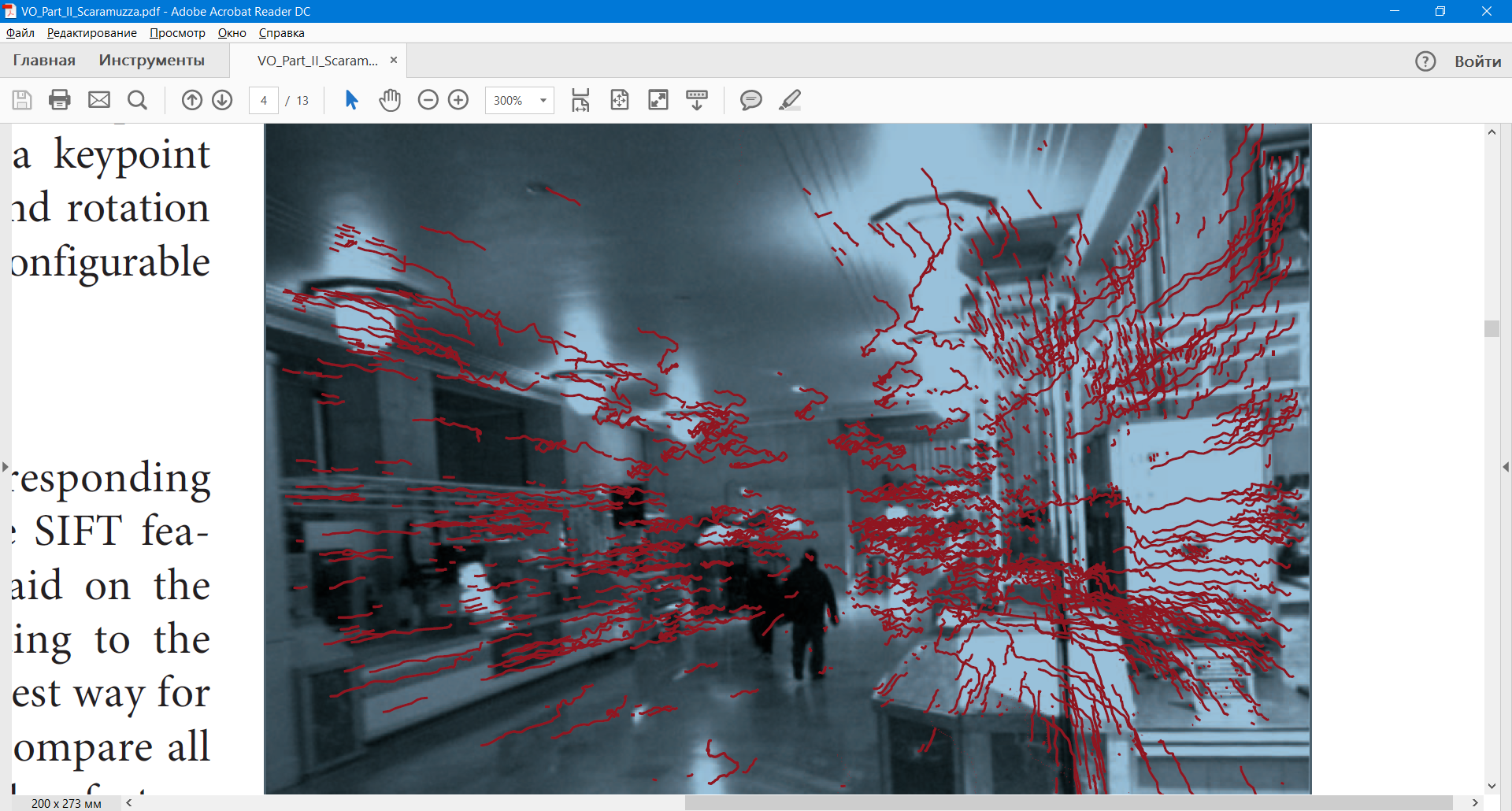


Рисунок 4. Дорожки SIFT.

В качестве альтернативы, если известна только модель движения, но не трехмерная позиция, соответствующее совпадение можно искать по эпиполярной линии во втором изображении. Этот процесс называется эпиполярным сопоставлением. Как можно видеть на рисунке 5, одна двумерная функция и два центра камеры определяют плоскость в трехмерном пространстве, которые пересекают оба изображения на две линии, называемые эпиполярными линиями. Эпиполярная линия может быть вычислена непосредственно из двухмерной функции и относительного движения камеры, как объяснено в части I этого урока. Каждая функция первого изображения имеет другую эпиполярную линию во втором изображении. В стереовизии вместо вычисления эпиполярной линии для каждой кандидатной функции изображения обычно исправляются. Выпрямление изображения - это переназначение пары изображений в новую пару изображений, где эпиполярные линии левого и правого изображений горизонтальны и выровнены друг с другом. Это имеет преимущество в облегчении поиска соответствия изображений, поскольку для каждой функции больше не нужно вычислять эпиполярные линии: корреспондент одной функции в левом (правом) изображении можно искать по этим функциям в правом (левом) изображении, что лежат в одном ряду. Выпрямление изображения может эффективно выполняться на графических процессорах (графических процессорах). В стереоизображении точно определено относительное положение между двумя камерами. Однако, если на движение влияет неопределенность, эпиполярный поиск обычно расширяется до прямоугольной области на некотором расстоянии от эпиполярной линии. В стереовизии преобразование SSD, NCC и переписи являются широко используемыми метриками подобия для эпиполярного сопоставления [91].

**Отслеживание функций**

Альтернативой независимому обнаружению признаков во всех изображениях-кандидатах, а затем их сопоставлению, является обнаружение признаков в первом изображении и затем поиск соответствующих совпадений на следующих изображениях. Этот подход, основанный на обнаружении, затем подходит для приложений VO, где изображения снимаются в близлежащих местах, где величина деформации движения и внешнего вида между соседними кадрами мала. Для этого конкретного приложения SSD и NCC могут работать хорошо. Однако, если функции отслеживаются с помощью длинных последовательностей изображений, их внешний вид может претерпеть большие изменения.

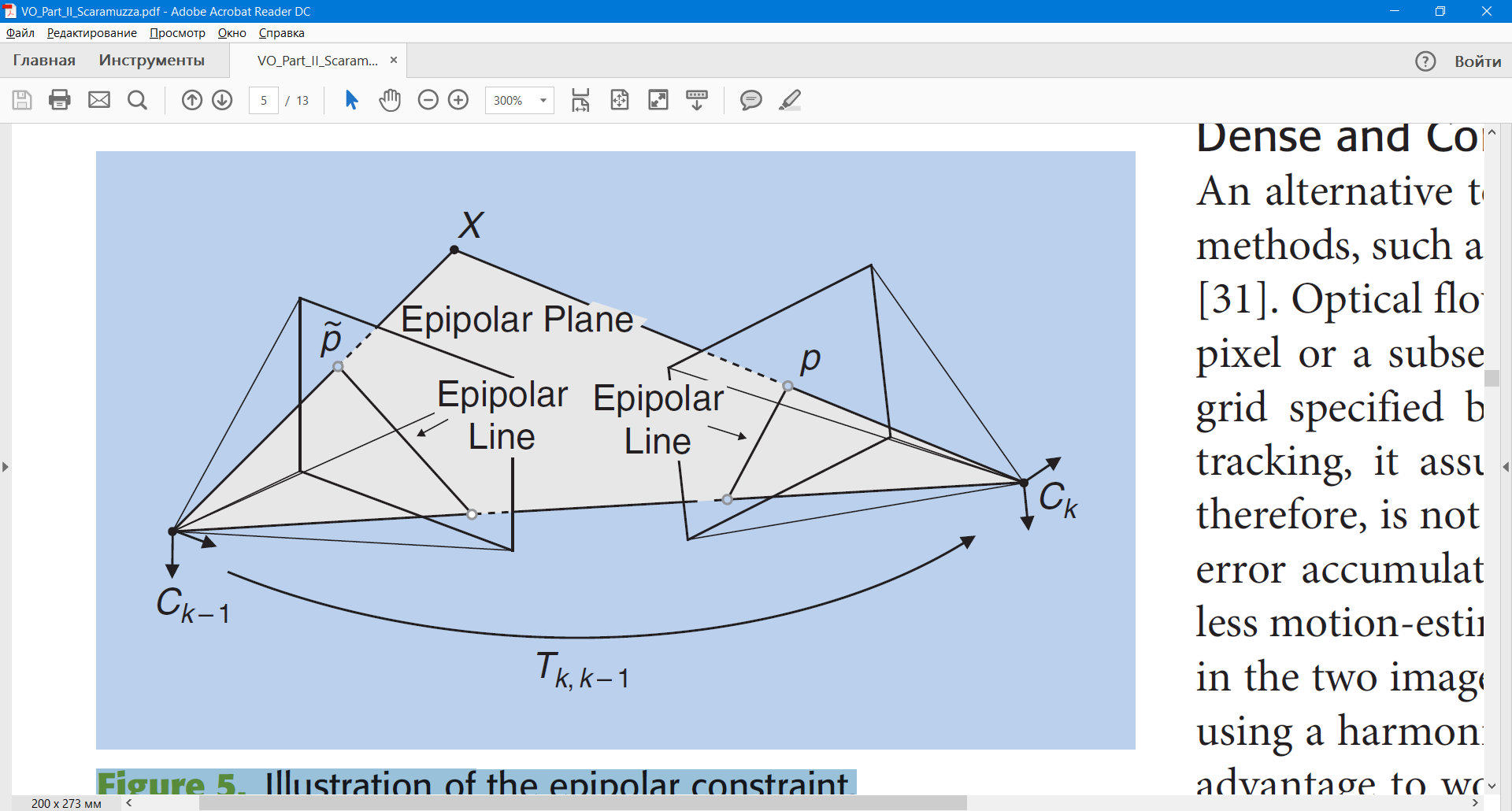


Рисунок 5. Иллюстрация эпиполярного ограничения.

В этом случае решением является применение модели аффинного искажения для каждой функции. Результирующий трекер часто называют трекером Kanade Lucas Tomasi (KLT) [12].

**Обсуждение**

Совместимость SIFT

Для сопоставления характеристик SIFT авторы сначала предложили тест отношения расстояния для использования на месте и обнаружения объекта [14]. Этот тест отношения расстояния принимает самое близкое соответствие (одно с минимальным евклидовым расстоянием) только в том случае, если отношение между ближайшим и вторым ближайшим совпадением меньше заданного пользователем порогового значения. Идея этого теста состоит в том, чтобы удалить совпадения, которые могут быть неоднозначными, например, из-за повторяющейся структуры. Порог для теста может быть установлен только эвристически, и неудачная догадка может также удалить правильные совпадения. Поэтому во многих случаях было бы полезно пропустить тест отношения и позволить RANSAC заботиться о выбросах, как описано в разделе «Удаление выброса».

Линии и ряды

Альтернативой точечным функциям для VO является использование линий или кромок, как предложено в [27] и [28]. Они могут использоваться в дополнение к точкам в структурированных средах и могут давать дополнительные сигналы, такие как направление (линии или кромки), а также ограничения планальности и ортогональности. В отличие от точек, линии сложнее сопоставить, потому что

линии, скорее всего, будут закрыты, чем точки. Кроме того, начало и конец сегмента линии кромки могут не существовать (например, окклюзии и линия горизонта).

Количество функций и распределение

Было обнаружено, что распределение признаков в изображении существенно влияет на результаты VO [1], [9], [29]. В частности, больше возможностей обеспечивает более стабильные результаты оценки движения, чем с меньшим количеством функций, но в то же время ключевые точки должны покрывать изображение как можно более равномерно. Для этого изображение можно разделить на сетку, и детектор функции применяется к каждой ячейке, настроив пороговые значения обнаружения, пока не будет найдено минимальное количество признаков в каждом субизображении [1]. Как правило, большой палец, 1000 функций является хорошим числом для изображения 6403480 пикселей.

Плотные и неправильные методы

Альтернативой экстракциям редкоземельных элементов является использование плотных методов, таких как оптический поток [30] или бескомпромиссные методы [31]. Оптический поток направлен на отслеживание, в идеале, каждого отдельного пикселя или подмножества всего изображения (например, всех пикселей в сетке, заданной пользователем). Однако, подобно отслеживанию признаков, он предполагает небольшое перемещение между кадрами и, следовательно, не подходит для приложений VO, так как ошибка движения быстро накапливается. Другой альтернативой являются безликие методы оценки движения, такие как [31]: все пиксели на двух изображениях используются для вычисления относительного движения с использованием гармонического преобразования Фурье. Преимущество этого метода заключается в том, что он работает особенно с изображениями с низкой текстурой, но чрезвычайно дорогостоящим (может занять до нескольких минут), а восстановленное движение менее точным, чем при использовании методов, основанных на характеристиках.

1. Affine Transformation –  отображение плоскости или пространства в себя. [[ref]](https://ru.wikipedia.org/wiki/%D0%90%D1%84%D1%84%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [↑](#footnote-ref-1)
2. Scale-invariant feature transform – [[ref]](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform). [↑](#footnote-ref-2)