Stabilization on a narrow-angle camera

1. Введение

1.1. Robust Video Stabilization Using Particle KeypointUpdate and l_1-Optimized Camera Path (2017)

1.1.1 Аннотация

Предлагаемый алгоритм состоит из трех этапов: (i) робастное обнаружение с использованием ключевых точек частиц между соседними кадрами; (ii) оценка пути камеры и сглаживание; и (iii) рендеринг для восстановления стабилизированного видео. В результате предлагаемый алгоритм позволяет оценить оптимальную гомографию путем переопределения важных характерных точек на плоской области с использованием ключевых точек с частицами.

Ранее 2D-видео стабилизация была осуществлена благодаря оценке оптимальной аффинной модели между кадрами, но такой подход не был устойчив к шумам, что приводило к значительным ошибкам при сопоставлении полученных кадров и динамическим окружением. Усовершенствование такого метода заключалось в использовании оптического потока для оценки траектории движения камеры, а также сглаживание пути движении камеры через гауссово ядро. Однако и такой подход оказался неудачным, так как не удалось стабилизировать несколько объектов с разными расстояниями в одно и то же время. Для повышения качества камеры стабилизированного видео используется стабилизированный подход, который указывает на поворот и масштабирование инвариантного пути камеры. Для этого многие исследователи прибегали к SIFT-признакам (FAST, SURF), MSE, RANSAC и энергетической функции для сглаживания пути камеры с уменьшением геометрических искажений. Также хорошо себя проявил подход с трекером фоновых признаков - KLT, но так или иначе несмотря на то, что методы 2D-видео стабилизации являются более быстрыми и надежными благодаря использованию линейного преобразования, они не позволяют оценить оптимальный путь камеры в регионах текстуры.

В настоящее время движение 3D-камеры оценивается на основе результатов сегментации изображения для улучшения качества видео. Был предложен метод стабилизации 3D-видео, использующий структуру от движения и пространственного деформирования для сохранения 3D-структур. Методы 3D-стабилизации могут давать более качественные результаты и пригодны для точного видеоанализа, однако реализация в сервисе реального или близкого к

реальному времени времени затруднена из-за высокой вычислительной сложности, и эти методы имеют общую проблему параллаксации, вызванной сбоем функции слежения за плоской областью.

1.1.2 Теоретические аспекты видео-стабилизации

Можно наблюдать геометрическое искажение видео из-за неправильного расположения пикселей в видеопоследовательности (рис. 1). В таком случае путь камеры не согласуется с системой координат камеры с точки зрения мировой системы координат. Так как искажение перспективы генерируется нежелательным сдвигом и вращением камеры, то геометрическое преобразование на выходе сенсора генерирует нестабильные кадры видеоизображения.

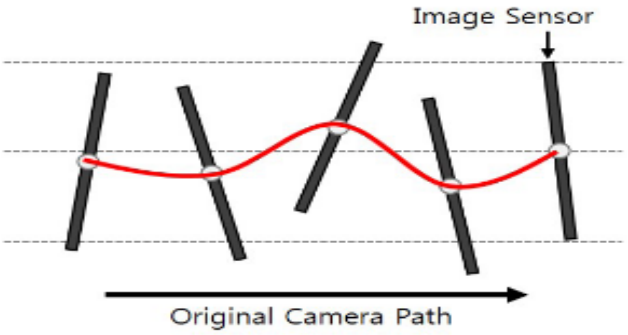


рис. 1 - причины геометрических искажений при стабилизации

Видео с тряской можно рассматривать как геометрически трансформированную версию идеально стабильного видео. Взаимосвязь между характерными точками исходного и трясущегося кадров определяется в однородной координате как:

$$q = Hp,$$

где H определяет гомографию, $p = [x, y, 1]^T$ - ключевую точку на оригинальном кадре, а $q = [x', y', 1]^T$ - соответствующую точку на колеблющемся кадре. Таким образом, неправильно оцененная гомография существенно снижает эффективность стабилизации видеоизображения при ошибочной траектории движения камеры. Для этого требуется извлекать наиболее функциональные точки, чтобы оценить оптимальную гомографию текстуры за меньшую область.

1.2.1 Извлечение и сопоставление ключевых точек для надежной стабилизации видеоизображения

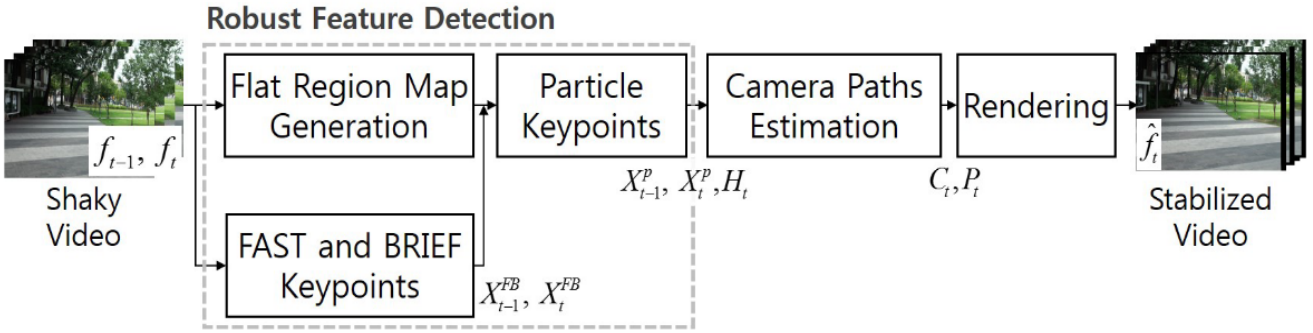


рис. 2 - блок-схема по алгоритму стабилизации видео

Пусть имеется два кадра из видео с тряской f_{t-1} и f_t. По ним генерируется карта плоской области с использованием FAST и BRIEF ключевых точек X_{t-1}^{FB} X_t^{FB} соответственно (рис. 2). После этого глобальный путь камеры C_t оценивается по оптимальной гомографии H_t, а сглаженный путь камеры P_t затем оценивается вариационным методом. В результате по оцененному пути камеры получается стабилизированный кадр f'_t.

1.2.2 Генерация карты плоской области для извлечения ключевых точек

Как уже говорилось, неточно оцененная гомография в бесструктурной области еще больше снижает эффективность стабилизации. Для решения этой проблемы предложенный метод генерирует карту плоской области и оптимальный путь камеры путем переопределения важных ключевых точек на плоской области.

Текстурированная область извлекается с помощью плоской карты области. Пространственно сглаженные картинки получаются путем свёртки изображений с тряской f_{t-1} и f_t с низкочастотным фильтром 3×3 Гаусса.

Кадры разделены на плоские и активные области с использованием абсолютной разницы исходного кадра и сглаженной версии. В результате, оценочная карта плоской области используется для переопределения робастных точек объекта. На рисунке 3 показано - исходный кадр с тряской и соответствующая плоская карта области.

(a) (b)

рис. 3 - Пример плоской карты региона: (a) входное изображение и (b) его карта плоской области с использованием предлагаемого метода

1.2.3 Робастные сопоставление признаков между смежными кадрами

Предложенный метод объединяет FAST и BRIEF для быстрого и точного извлечения ключевых точек объекта. FAST извлекает точки объекта путем сравнения интенсивности с 16 пикселями окрестности в окружности. Определяем угол, если интенсивность пикселей окрестности $I_{p\to x}$ ярче, чем у пиксела-кандидата I_p, или если они все темнее, чем у пиксела-кандидата I_p. Чтобы расположить пиксели окрестности в порядке количества информации о том, является ли пиксель кандидата углом, классификатор дерева решений тренируется с помощью итеративного алгоритма Dichotomiser 3 (ID3), ключевые точки p определяются как:

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t & (darker) \\ s, & I_p - t < I_{p\to x} < I_p + t & (similar) \\ b, & I_p + t \le I_{p\to x} & (brighter) \end{cases},$$

где p представляет окрестности, выбранные деревом решений с помощью алгоритма ID3, и t - порог сравнения интенсивности. В данном алгоритме t=0.2 для экспериментально лучшего результата. BRIEF идентифицирует локальные особенные точки путем сравнения интенсивности выборки пар. Гомография может быть вычислена очень эффективно, так как бинарная строка может быть сопоставлена по расстоянию Хэмминга при помощи операции XOR.

Ключевые точки FAST извлекаются между двумя смежными видеокадрами f_{t-1} и f_t определяют распределение случайных ключевых

точек частиц. Дескрипторы генерируются с использованием BRIEF и сравниваются с помощью расстояния Хэмминга. Извлеченные ключевые точки FAST и BRIEF обозначаются как $X^{FB}_{t-1} = \{(x^1_{t-1}, y^1_{t-1}), \ldots, (x^M_{t-1}, y^M_{t-1})\}$и $X^{FB}_{t} = \{(x^1_{t}, y^1_{t}), \ldots, (x^M_{t}, y^M_{t})\}$. Затем случайным образом генерируются ключевые точки частиц в плоской области для обнаружения робастных характерных точек. Распределение N частичных точек $X^P_{t-1} = \{(x^1_{t-1}, y^1_{t-1}), \ldots, (x^M_{t-1}, y^M_{t-1})\}$и $X^P_{t} = \{(x^1_{t}, y^1_{t}), \ldots, (x^M_{t}, y^M_{t})\}$характеризуются как гауссовские функции $G(\bar{X}^{FB}_{t-1}, \sum_{t-1})$ и $G(\bar{X}^{FB}_{t}, \sum_{t})$ в плоских областях кадров f_{t-1}и f_tсоответственно. Дескриптор соответствует фреймам в смысле расстояния между ключевыми точками частиц и ключевыми точками FAST и BRIEF. Дескриптор D_t t-го кадра определяется как $D_t = X^P_t - X^{FB}_t$. Окончательные соответствия сопоставляются с помощью суммы квадратных разностей (SSD) дескрипторов двух кадров. Дескриптор используется для сопоставления робастных ключевых точек в плоской области с использованием ключевых точек частиц. Наконец, оптимальная гомография через RANSAC.

1.2.4 Оценка оптимального пути камеры

Традиционные методы стабилизации видео используют скользящее среднее Гауссова фильтра для сглаживания траектории камеры. Фильтр со скользящим средним может сгладить траекторию камеры, используя временной интервал соседних кадров. Гауссово ядро может удалить нежелательное движение, используя глобальное преобразование. Кроме того, при обрезке областей и увеличении размера искажений производительность стабилизации видео становится низкой. Для решения этой проблемы предложенный метод адаптивно сглаживает путь камеры с помощью алгоритма 1D TV (“Nonlinear total variation based noise removal algorithms”). Отверстия представляют собой пустую область в видеокадре, которая создается после перемещения кадра по сглаженному пути камеры. Для компенсации отверстий граница области стабилизированного видео обычно обрезается, а оставшаяся центральная область увеличивается, чтобы заполнить исходный размер видеокадра. Поэтому важно минимизировать область отверстий, чтобы сохранить исходное содержимое. В стабилизированном видео меньше дырок, так как 1D TV метод может сохранить исходную траекторию и удалить нежелательные отклонения.

Учитывая оптимальную гомографию H_t между f_{t-1}и f_t, генерируется глобальный путь камеры C_t. Угловые точки обозначаемые как

$V_t = \{(1,1),\ (1,h),\ (w,1),\ (w,\ h)\}$ с размерами кадра f_t с тряской как $w\ *\ h$ переходят V'_t, использую гомографию H_t. H_tможно рассматривать как матрицу преобразования движения камеры. Поэтому движение камеры между f_{t-1}и f_t можно понимать как разница между V'_t и V_t. Таким образом, глобальный путь камеры C_t есть $C_{t-1} + (V'_t - V_t)$. Найденный путь сглаживается через 1D TV, затем энергетическая функция для сглаженного пути P_i определяется как

$$E(P_t) = ||P_t - C_t||_2^2 + \lambda||AP_t||_1 \ (*), \qquad (1)$$

где A - матрица временного различия

$$A = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & & \\ & & & -1 & 1 \end{bmatrix}$$

и λ представляет весовые коэффициента для сглаживания. Первый член уравнения (1) приводит в действие сглаженный путь камеры, близкий к исходному, а второй удаляет шумные движения, искажающие путь камеры. Энергетическая функция уравнения (1) может быть минимизирована с помощью алгоритма итеративного свертывания. Результаты работы фильтр на рисунке 4.

Завершающим этапом стабилизации видео является реконструкция геометрически трансформированных кадров по траектории сглаженной камеры. Сглаженная гомография H'_tможет быть получена как разница между сглаженным путём камеры P_tи оригинальным C_t :

$$(C_t - P_t) + V_t = H'_t V_t,$$

где V_t представляет собой 4 угла картинки. Стабилизированный кадр f'_t генерируется как $f'_t = H'_t f_t$.

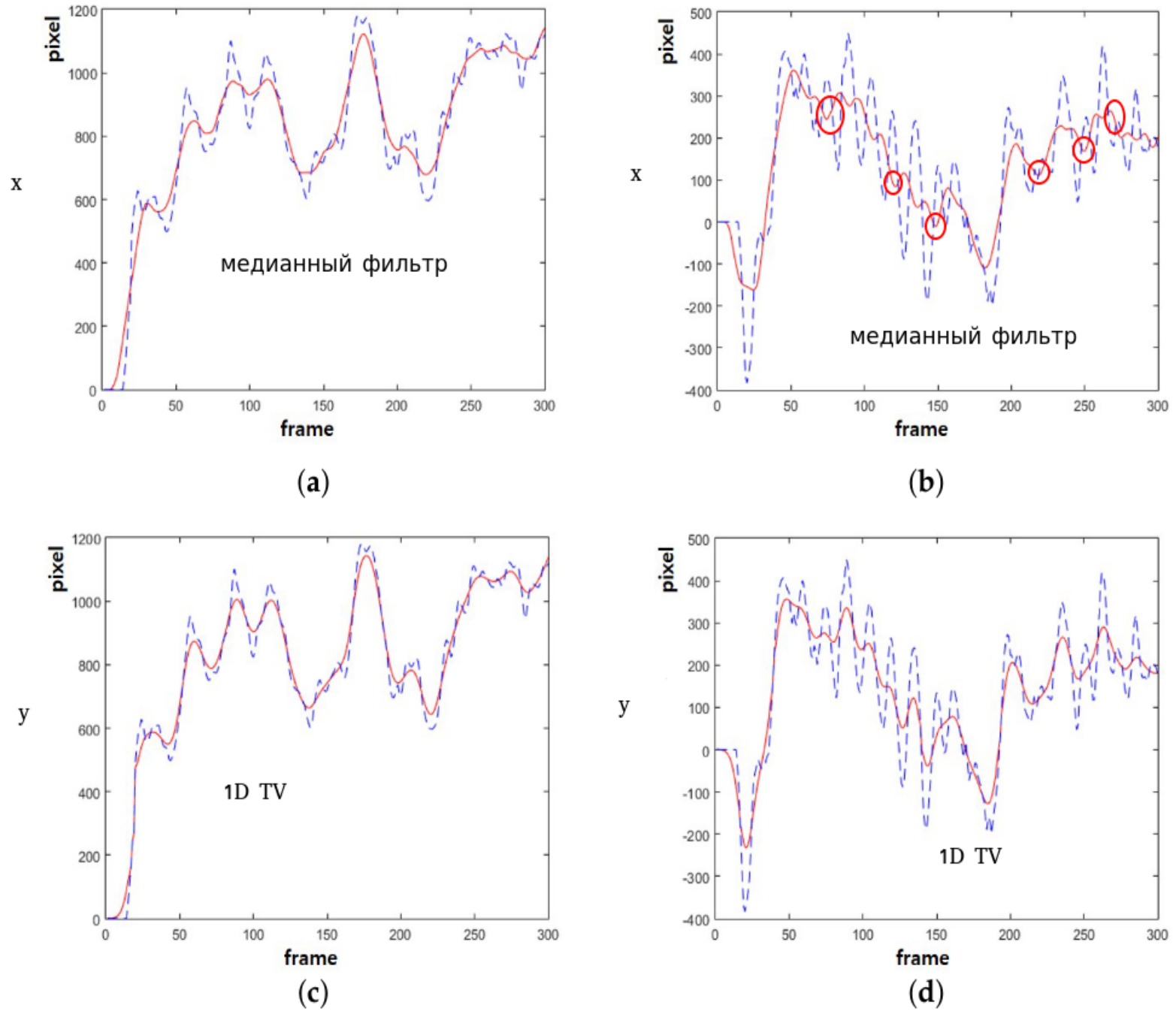


рис.4 - работа алгоритма 1D TV

Также важно оценивать искажение перспективы для объективной оценки методов стабилизации. По этой причине такие искажения оцениваются с помощью трансформации исходного и стабилизированного кадров. Гомографию стабилизированных последовательностей изображений можно определить как

$$P_t = B_t C_t,$$

где C_t и P_t соответственно представляют собой кумулятивную гомографию между соседними кадрами наблюдаемого дрожащего и стабилизированного видео и B_t матрицей трансформации. Перспективные искажения рассчитываются путем усреднения перспективных компонентов, поскольку гомография с искажениями определяет качество видео. Как показано в таблицах, предложенный метод стабилизации видео может успешно устранить нежелательное движение без перспективных искажений по сравнению с традиционными алгоритмами видеостабилизации.

Кроме того, для оценки объективной производительности авторы использовали пиковые значения отношения сигнал/шум (PSNR) во временных соседних кадрах. PSNR определяется как

$$PSNR = 10\,log\,(MAX_f^2 / MSE),$$

где $MSE = \frac{1}{M}\frac{1}{N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\|f_{t-1}(x,y) - f_t(x,y)\|^2$ представляет собой среднюю квадратичную погрешность, а MAX_f максимальное значение интенсивности кадров.

Нестабильные видео с нежелательными движениями камеры имеют ограниченную производительность обнаружения объектов и отслеживания. Заключительный эксперимент проводится для того, чтобы продемонстрировать, может ли предложенный метод сыграть практическую роль предварительной обработки в различных системах видеоанализа. Трекер Lucas-Kanadefeature (LKT) для демонстрации производительности отслеживания объекта на трясущихся и стабилизированных видеороликах. Несмотря на то, что популярный LKT отслеживал робастные возможности с поворотом изображения и изменением точки обзора, у него есть фундаментальная проблема пропуска интересующих объектов на шатком видео. Предложенный метод может значительно повысить эффективность отслеживания объектов.

Предложенный метод стабилизации видео устраняет неустойчивые движения, оценивая оптимальный путь камеры с помощью робастного выделения ключевых точек в безтекстурной области, и сглаживает дрожание без задержки кадра с помощью метода вариационной оптимизации. Кроме того, предлагаемый метод особенно подходит для аппаратной реализации в карманных камерах, так как он оценивает оптимальный путь камеры в шатком видео, используя только четыре вершины в каждом кадре. В результате, предложенный алгоритм может успешно усилить шейк-видео, используя улучшенный 2D стабилизационный метод, основанный на ключевых точках частиц. Предлагаемый метод может быть использован для различных видеосистем, включая мобильные устройства формирования изображений, системы видеонаблюдения и информационные системы формирования изображений транспортных средств. Для преодоления вибрации видеоизображения, получаемого мобильными роботами на основе технического зрения, современная технология представляет систему стабилизации видеоизображения на базе полевого программируемого робота на базе массива затворов (FPGA) для применения в однокристальной встроенной системе для видеопотока в реальном времени. Предлагаемый метод может быть применен к этой системе для извлечения корректных характеристик в плоскости и улучшения качества стабилизированного видео. В последнее время в системе аэронаблюдения используется метод стабилизации видеоизображения для обнаружения объектов в широкой зоне. Авиационное видео, полученное с помощью подвижной камеры, не может избежать дрожания между кадрами, находящимися во временной близости. Поэтому алгоритм видеостабилизации является незаменимым этапом предварительной обработки для надежного обнаружения объектов в системе воздушного наблюдения. Предложенный метод позволяет определить значимые точки, которые трудно извлечь в плоской области или области с низким разрешением. Он может существенно повысить производительность условных методов стабилизации видеоизображения. Пользователи портативных камер общаются с видеоизображениями динамической активности, такими как ходьба, езда на велосипеде и пешие прогулки, и важно удалить нежелательные хрупкие движения. Предлагаемый алгоритм извлечения функций может быть гибко модифицирован для извлечения робастных ключевых моментов, а также может быть использован в облачном сервисе на базе вычислительной мощности сервера для повышения качества загружаемого видео. Дорожное видео первого лица может быть стабилизировано путем оптимальной оценки траектории движения камеры на основе обновления ключевых точек частиц в плоской области. Кроме того, в настоящее время личные видеоматериалы обобщаются в виде

видеозаписей с задержкой по времени из-за ограниченной зарядки аккумуляторов мобильных устройств и скорости беспроводной сети. В этом контексте предложенный метод может быть применен на этапе предварительной обработки алгоритма суммирования видео для устранения эффектов колебаний.

1.2 SIFT Features Tracking for Video Stabilization (2007)

1.2.1 Введение

Важным этапом любой цифровой системы стабилизации видео является фильтрация движения, при которой оценочное движение оценивается для распознавания намеренного движения: предложены и недавно модифицированы некоторые техники, такие как фильтрация Калмана и интегрирование векторного движения (MVI) для коррекции трансляционной и ротационной тряски в соответствии с реальными системными ограничениями. Недавно была представлена система стабилизации видео, основанная на особенностях SIFT. Она использует возможности SIFT для оценки межкадрового движения: затем фильтр Калмана распознает преднамеренное движение, а фильтр частиц снижает дисперсию ошибок. Как бы то ни было, эта система не модифицирует реализацию SIFT и не адаптирует ее к проблеме стабилизации видео. Следовательно, невозможно изменить параметры SIFT-алгоритма, чтобы улучшить характеристики. Более того, это требует больших вычислительных затрат: на каждом кадре должны выполняться как фильтрация Калмана, так и фильтрация частиц, а последняя подразумевает интенсивные вычисления для обновления весов покадрово.

Первая проблема, которую необходимо решить, связана с согласованием ключевых точек. В оригинальной работе она выполнена с использованием Евклидового расстояния между векторами дескрипторов и соотношения расстояний, а именно соотношения ближайшего соседнего расстояния и второго ближайшего, которое можно проверить по порогу отбрасывания ложных совпадений. Фактически, правильные совпадения должны иметь меньшие соотношения, а неправильные - более высокие, близкие.

В оригинальной реализации SIFT используется пороговое значение 0.8, чего недостаточно для общих целей: авторы исследовали корреляцию между соотношением расстояний и правильностью совпадений и обнаружили, что использование значения 0.6 в качестве порога лучше

подходит для отбрасывания неправильных совпадений: на самом деле, только несколько пар с ключевыми точками показывают такое низкое соотношение расстояний, но они с большей вероятностью будут правильными совпадениями, чем многие другие, которые имеют более высокие соотношения расстояний, как показано на рис. 5. Важно отметить, что на изображении среднего размера (352×288 пикселей) могут быть показаны даже тысячи ключевых точек, но межкадровое совпадение, выполненное в описанном виде, приводит лишь к нескольким сотням пар.

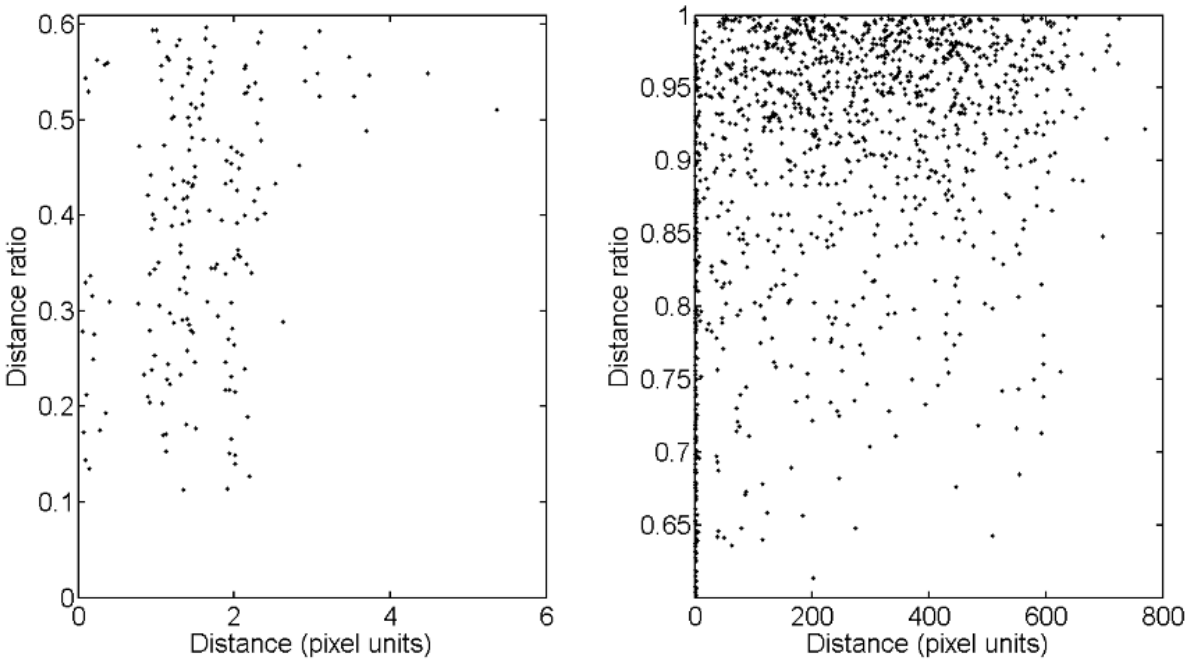


рис. 5 - Корреляция между расстоянием в пикселях (по оси X) и отношением расстояний (по оси Y) для среднеразмерного изображения: слева - для объектов с отношением расстояний менее 0,6, справа - для остальных объектов.

Результатом этого процесса сопоставления является список пар ключевых точек, которые могут быть легко использованы в качестве входа в алгоритм оценки движения, основанный на особенностях.

1.2.2 Обучение модели

Для оценки движения, необходимого для наложения текущего кадра на предыдущий, используется набор локальных векторов движения, полученных при совпадении функций, для минимизации видимого движения. Для того, чтобы оценить этот глобальный вектор движения, локальные векторы движения должны соответствовать кадровой модели движения. Несмотря на то, что движение в видео является трехмерным, глобальное движение между кадрами может быть оценено с помощью

двухмерной линейной конформной модели, как правило, наилучший компромисс между эффективностью и сложностью. Данная модель описывает межкадровое движение, используя четыре различных параметра, а именно два сдвига, один угол поворота и коэффициент масштабирования:

$$\begin{cases} x_f = x_i \lambda \cos\theta - y_i \lambda \sin\theta + T_x \\ y_f = x_i \lambda \sin\theta + y_i \lambda \cos\theta + T_y \end{cases}$$

, где λ- параметр масштабирования, θ- угол поворота, T_x, T_y - сдвиги по осям.

Весь набор локальных векторов движения не содержит полезной информации для эффективной компенсации движения, так как, вероятно, включает в себя неправильные совпадения или коррекции, которые действительно принадлежат самоподвижным объектам в снимаемой сцене. Очевидно, что существуют корректные пары, которые представляют собой реальные дрожания камеры, но несколько точек просто не имеют отношения к такой информации.

Метод наименьших квадратов не дает хороших результатов, когда в общем количестве функций, как и в данном случае, присутствует значительная доля отклонений. Многие надежные методы оценки были разработаны и применены в компьютерном зрении, но для получения производительности в реальном времени реализовали упрощенную версию итерационного метода наименьших квадратов.

Уточнение итеративных наименьших квадратов может быть использовано для того, чтобы легко избежать промахов и уточнить решение. Оно выполняется путем определения решения наименьших квадратов с полным набором функций, вычисления статистики ошибок для набора данных по сравнению с вычисленным движением, а затем удаления любой ключевой точки, которая представляет ошибку больше заданного адаптивного порога. Эта методика хорошо работает, но для большей точности ее необходимо комбинировать с другими фильтрами, работающими с набором локальных векторов движения.

Изначально, когда имеет место очень большой локальный вектор движения, вряд ли окажутся правильными, поэтому такие сдвиги сразу же отбрасываются при помощи фиксированного порога по евклидовой норме локального вектора. Затем все локальные векторы движения используются для получения первой оценки методом наименьших квадратов. В то же время, функции отслеживаются при их движении через соседние кадры, так как некоторые функции, вероятно, появляются в нескольких

последовательных, и поэтому их локальные векторы движения гарантируют лучшую точность. После этого первого шага оценки движения каждая входная точка проверяется на соответствие найденным параметрам и оценивается ее погрешность. Так как каждая особенность связана с ключевой точкой на первом изображении, а другая - на втором, то первая точка трансформируется с использованием полученных параметров, выводя ожидаемую точку, которую можно сравнить с реальной второй точкой. Для выполнения этой задачи можно принять две различные меры по устранению погрешности:

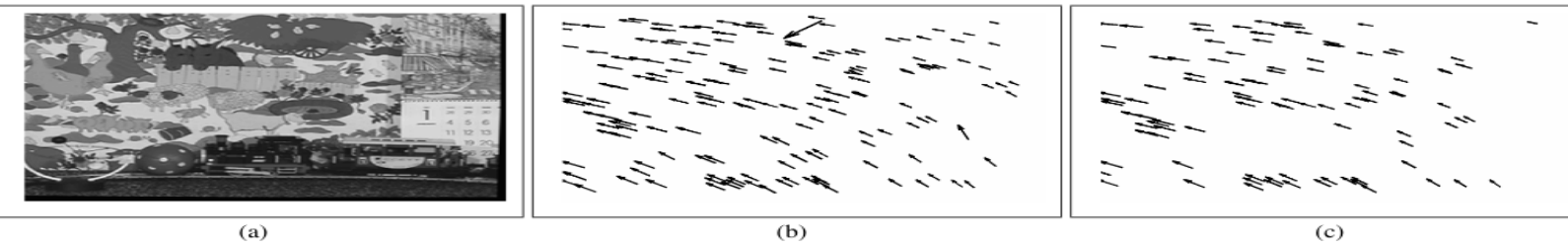
- *Евклидова дистанция* между ожидаемой и реальной точкой хорошо отбрасывает совпадения, которые не согласуются с найденными трансляционными компонентами, но могут ввести в заблуждение пограничную точку, когда происходит поворот.

- *угол поворота* между предполагаемой и реальной точкой, с учетом центра кадра: эта мера хорошо работает с вращающимися компонентами, но может привести к неправильному результату при применении на трансляционных компонентах.

Очевидно, что обе меры подходят для отбрасывания некорректных совпадений, поэтому лучшим выбором является применение двойной фильтрации: ошибка для каждой ключевой точки оценивается с использованием евклидового расстояния и угла поворота, затем каждой точке присваиваются обе ошибки. Также ошибки можно накапливать с определённым коэффициентом.

$$E_n^{cum}(k) = (1-\alpha)E_{n-1}^{cum}(k) + \alpha E_n(k)$$

Затем с помощью статистики совокупных ошибок вычисляется пороговый уровень: весь набор ранее отслеживаемых функций сортируется по двум кумулятивным ошибкам, а затем во второй ввод данных вставляются только ключевые точки первых 50% тезисов. Поскольку существует две меры предосторожности, каждая точка должна быть в первой половине множества, в противном случае она отбрасывается, как показано на рис. 6.



(a) (b) (c)

рис. 6 - Де-стабилизированная рамка (a) с локальными векторами движения после первой грубой оценки (b) и дофильтрация по кумулятивной погрешности (c): отбрасываются неправильные совпадения признаков.

Более того, использование кумулятивных ошибок, а не только реальных, улучшает результирующую стабилизацию: на самом деле, движущиеся объекты или движущиеся точки, которые могут легко ввести в заблуждение метод наименьших квадратов, аккумулируют свои ошибки в каждом кадре и, следовательно, имеют меньше шансов быть вставленными в конечный входной набор данных. Таким образом, становится возможным отбрасывать корректные совпадения, которые движутся не так, как это делает кадр, повышая тем самым точность компенсации движения.

1.2.3 Фильтрация движения

На самом деле, функция может перемещаться из кадра в соседний не только из-за дрожания камеры, но даже из-за намеренных перемещений или из-за того, что она принадлежит к движущемуся объекту в сцене. Этот локальный вектор движения может привести к неправильной оценке движения, так как стабилизация видео должна происходить только на дрожании камеры и не должна компенсировать нежелательные движения, влияющие на качество видео. Интеграция вектора движения может быть успешно использована для фильтрации кумулятивной кривой движения, т.е. движения текущего кадра относительно первого кадра. Вектор кумулятивного трансляционного движения обычно получается по сумме векторов движения всех предыдущих целых кадров: в этом случае он интегрируется с коэффициентом демпфирования δ, а интегрированный вектор движения кадра n генерируется из

$$IMV(n) = \delta IMV(n-1) + GMV(n)$$

где GMV(n) - вектор глобального движения между фраменами и кадром n-1, полученный итерационным методом наименьших квадратов. Таким образом, компенсация C(n), применяемая к кадру n, получается как

$$C(n) = IMV(n) - IMV(n-1)$$

и преднамеренное движение, таким образом, сглаживается и не компенсируется. Коэффициент демпфирования δ обычно выбирается в качестве фиксированного значения в диапазоне от 0 до 1, и в зависимости от требуемой степени стабилизации могут быть найдены различные значения: всегда существует компромисс между небольшим колебанием и следованием за намеренным движением с минимальной задержкой.

Однако использование фиксированного значения не обеспечивает достаточной гибкости, в то время как адаптивный коэффициент демпфирования δ является лучшим решением. Его величина зависит от суммы двух последних глобальных векторов движения, поскольку если эта сумма низка, то алгоритм предполагает наличие преднамеренной статической камеры и выбирает высокий коэффициент демпфирования для сильной стабилизации последовательности, в то время как высокое суммарное значение подразумевает большое движение и, следовательно, меньшее значение δ выбирается, следуя более близко за намеренным движением. Эта фильтрация применяется в зависимости от x и y трансляционных составляющих GMV.

Пиковое соотношение сигнал-шум (PSNR) является полезной мерой ошибки для численной оценки того, насколько хорошо выполняется стабилизирование: PSNR между кадрами n и n + 1 есть

$$PNSR(n) = 10 \log_{10} \frac{I_{MAX}^2}{MSE(n)}$$

PNSR измеряет, насколько изображение похоже на другое: в этом случае полезно оценить, насколько последовательность стабилизирована алгоритмом, так как последовательные кадры в результирующей последовательности должны быть более непрерывными, чем входная и, следовательно, PSNR должен увеличиваться от начальной последовательности к конечной. Внутрикадровая точность преобразования затем используется, для объективной оценки стабилизации, вносимой алгоритмом:

$$ITF = \frac{1}{N_{frame} - 1} \sum_{k=1}^{N_{frame}-1} PSNR(k)$$

В итоге, был представлен новый подход к стабилизации видео, основанный на извлечении и отслеживании SIFT-функций через видеокадры. Мы используем алгоритм оценки движения, который отслеживает SIFT-функции, извлеченные из видеокадров, а затем оценивает их траекторию для оценки межкадрового движения. Модифицированная версия метода Iterative Least Squares (Наименьших Итеративных Квадратов), позволяющая избежать ошибки оценки песка преднамеренного движения камеры, фильтруется с помощью Adaptive Motion Vector Integration (Адаптивная Интеграция Векторов Движения). Эксперименты подтвердили эффективность метода. Дальнейшие работы включают в себя усовершенствования в извлечении SIFT и на этапе фильтрации движения для обработки видео различного разрешения.

Протестировать можно на http://trace.eas.asu.edu/ (foreman.cif)

1.3 Real-Time Video Stabilization for Unmanned Aerial Vehicles (2011)

Основное отличие от предыдущих статей - сглаживание сплайнами (“Spline smoothing”) + также можно использовать вместо обычного SIFT улучшенную версию PCA-SIFT (https://www.cse.unr.edu/~bebis/CS491Y/Papers/Ke04.pdf), но скорость совсем мала: 2fps. Также имеется возможность сглаживать не динамически последовательность кадров, а скользящим окном, но из-за этого появляется оффлайновое ограничение.

Возвращаясь к Сплайновому сглаживанию: параметры накапливаются, после чего и сглаживаются сплайном. Заметим также, что компенсация изображений не может быть рассчитана непосредственно из параметров, вычисленных в уравнения Аффинного преобразования, так как нежелательное движение датчиков и нормальное движение должно быть отделено заблаговременно. Нормальное движение, по-видимому, отличается от нежелательного движения датчиков, первый - медленный и следует определенным правилам, второй - быстрый и случайный, но непредсказуемый. На данном этапе традиционные методы обычно используют низкочастотный фильтр для сглаживания параметров движения с целью поддержания низкочастотного движения и отклонения высокочастотного движения. Однако, этот метод всегда приводит к нежелательному эффекту, т.е. стабилизированные изображения представляют собой несколько интервалов, отстающих от реального видео.

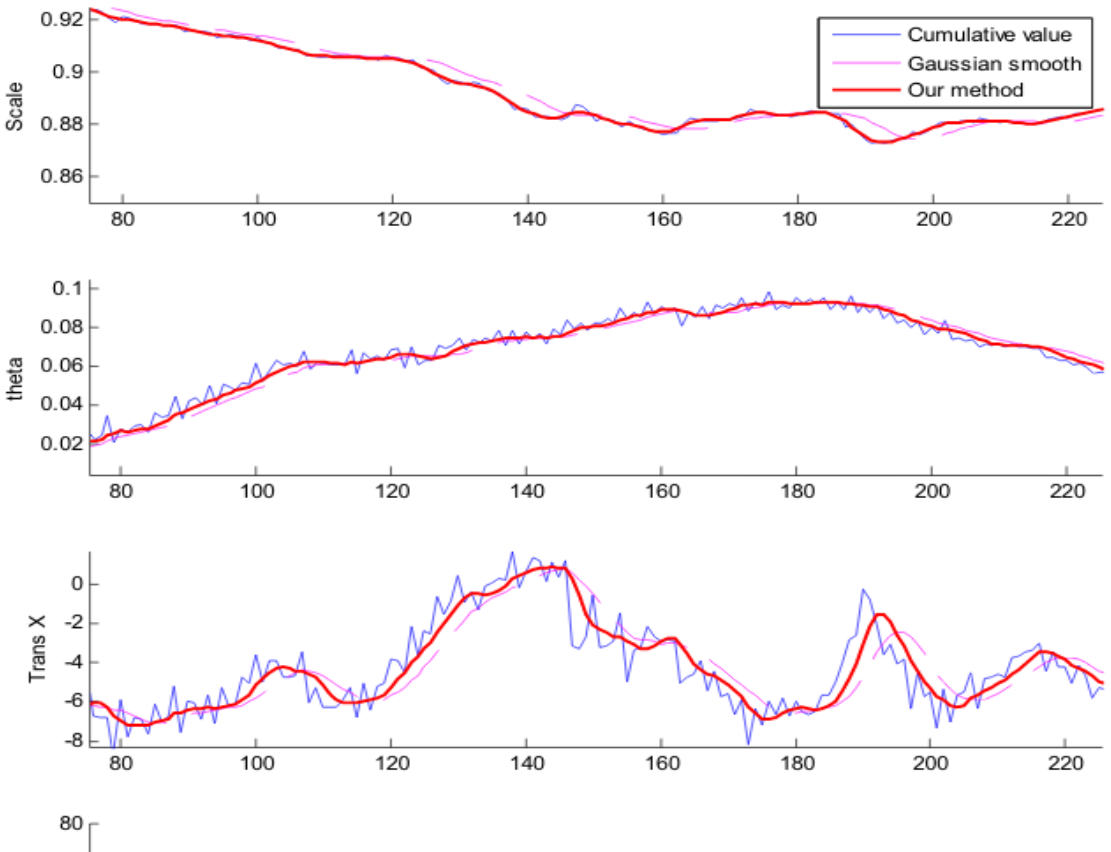
$$Frame_i^{smoothed} = A_i^{cumulative,smoothed} \cdot (A^{cumulative})^{-1} \cdot Frame_i$$

Способов сглаживания накопленных последовательностей трансформированных изображений много: числовая оптимизация, фильтр Калмана, либо же фильтр Гаусса, который своим ядром проходит по параметрам. Однако последний метод всегда приводят к нежелательному результату: стабилизированные изображения представляют собой несколько интервалов (в зависимости от ширины окна для сглаживания), отстающих от реального видео. Здесь для более простого подхода к свертыванию временной последовательности параметров кумулятивного преобразования используется кубический сплайн, основанный на сглаживании. Это позволяет убрать высокочастотный шум (или колебания камеры) при прохождении любого кумулятивного эффекта, например, из плавного панорамирования или от слежения за движением. Кубический сглаживающий сплайн g(t) генерируется с целью минимизации:

$$\mu \sum_{j=0}^{n-1} w(j)\left|P_j - g(j)\right|^2 + (1-\mu)\int \lambda(t)\left|D^2 g(t)\right|^2 dt$$

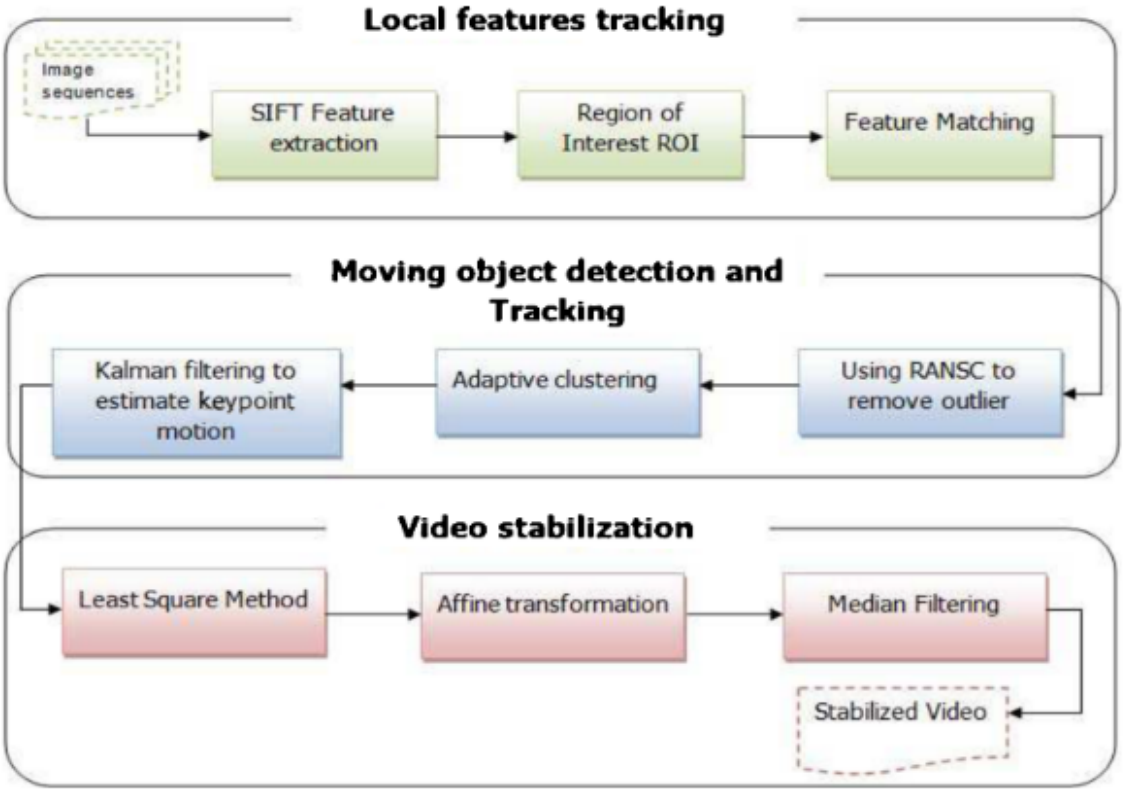
, где

μ – сглаживающий параметр, ω- вес штрафа, λ- для кусочно-постоянной весовой функции при грубом измерении, P - корреспондирование точки



Из возможностей улучшения можно рассмотреть 3х-канальных подход с детектированием ключевых точек, а также добавление таких признаков, как контур (а какими тогда будут дескрипторы?)

1.4 Video stabilization with moving object detecting and tracking for aerial video surveillance



Как вариант, можно отсеивать динамические объекты через Ransac (критерий Хи-квадрат) + диаграмма Вороного. Например, когда на нас едет поезд, то без такой фильтрации алгоритм будет неустойчивым, даже если пытаться аккуратно подбирать ROI, т.е. появляется подзадача - отделение движение камеры от движения динамических объектов, чтобы убрать общий шум. Для первого приближения можно считать, что ускорения всех объектов одинаковы.

1.5 Fast Feature-Based Video Stabilization without Accumulative Global Motion Estimation (https://sci-hub.do/https://ieeexplore.ieee.org/document/6311347)

Конечно, использование ORB-детектор сильно устойчивее SIFT. Улучшенный подход к сглаживанию в данной работе аналогичен методу сглаживания, предложенному в (https://sci-hub.do/https://ieeexplore.ieee.org/document/1634345), разница между ними заключается в том, что улучшенный подход непосредственно вычисляет параметры преобразования с использованием нестабильных входных кадров и стабилизированных выходных кадров (USF), в то время как предыдущий метод использует только исходные входные кадры (UOF). Таким образом, улучшенный метод сглаживания позволяет получить более сглаженную видеопоследовательность.

Пусть T^{j}_{i} есть переход из координаты i в j. Соседние кадры обозначим как $NF_c = \{n: c - k \leq n \leq c + k\}$, тогда сглаживающая трансформация есть

$$S_c = \sum_{j \in NF_c} T_c^j * G(k),$$

$$T_c^j = \begin{cases} T_{USF}(j), & j < c \\ T_{UOF}(j), & j > c \end{cases},$$

, где $T_{USF}(j)$ трансформация параметров из текущего кадра к выходному стабилизированному кадру j, в то время как $T_{UOF}(j)$ есть трансформация между текущим кадром и нестабилизированным кадром j, G(k) - гауссово ядро. В итоге, стабилизированное изображение есть

$$I'_c = S_c I_C$$

Разумеется, k нужно подбирать, учитывая, что при его увеличении такой подход улучшает качество. Эмпирически это значение было подобрано как равным 6, но и скорость работы алгоритма при таком радиусе заметно возрастает по сравнению с r = 3. (https://sci-hub.do/https://ieeexplore.ieee.org/document/1634345).

Также стоит сказать, что ORB более чувствителен к изменению заднего фона, что также нужно учитывать при решение определённых задач.

1.6 Digital video stabilization based on adaptive camera trajectory smoothing

1.6.1 Введение

Существует несколько глобальных подходов к стабилизации видео:

- Механическая стабилизация: сенсоры сами детектируют сдвиг и компенсируют его
- Оптическая стабилизация. широко используется в фотокамерах и состоит из механизма компенсации углового и поступательного движения камер, стабилизирующего изображение перед его записью на сенсор
- Цифровая стабилизация без использования каких-либо специальных устройств. Решения в такой стабилизации основаны на основе интенсивности пикселей, которые непосредственно используют текстуру изображений в качестве вектора движения, либо на основе ключевых точек, которые локализуют набор соответствующих точек в соседних кадрах, как и было описано в предыдущих статьях (подход с ключевыми точками быстрее). На этом этапе могут быть использованы такие методы, как извлечение ROI, чтобы избежать резки определенных объектов или регионов, которые должны быть важны для наблюдателя.

  Последние усовершенствования 2D-методов сделали их сопоставимыми с 3D-методами по качеству. Например, использование L1-нормы оптимизации может генерировать путь камеры, который следует кинематографическим правилам, чтобы рассматривать отдельно постоянные, линейные и параболические движения. Модель на основе сетки, в которой вычисляются несколько траекторий в разных местах видео, оказалась эффективной при работе с параллаксами без использования 3D-методов. С другой стороны, 3D методы обычно строят трехмерную модель сцены с помощью структурно-двигательных (SFM) методов сглаживания движения, обеспечивающих более качественную стабилизацию, но при более высоких вычислительных затратах. Поскольку они обычно имеют серьезные проблемы при работе с крупными объектами на переднем плане, 2D-методы в целом являются предпочтительными в практике.

  Несмотря на то, что 3D-методы не являются распространенными, они могут заполнять недостающие части кадра, используя информацию из нескольких других кадров. В последнее время 2D- и 3D-методы были расширены для работы со стереоскопическим видео. Появились гибридные подходы для получения эффективности и робастности

2D-методов в дополнение к высокому качеству 3D-методов. Некоторые из них основаны на таких концепциях, как подпространство траекторий и эпиполярное смещение (https://sci-hub.do/10.1145/2231816.2231824).

1.6.2 Адаптивное сглаживание гауссианой.

Предлагается брать размер окна $M = n/3 - 1$, где n - число картинок в видео, но на большой видеопоследовательности или в режиме онлайн это может сильно ухудшить работу алгоритма из-за постоянного накопления ошибок. Так как различные моменты видео будут иметь различное количество колебаний, эта работа применяет гауссовский фильтр адаптивно, чтобы удалить только нежелательное движение камеры.

Сглаживание интенсивного движения может привести к появлению видео с небольшим количеством пикселей. Кроме того, этот тип движения, как правило, является желательным движением камеры, которое не должно быть сглажено. Поэтому параметр σ вычисляется таким образом, что в этих регионах он имеет меньшие значения. Таким образом, траектория будет сглажена при рассмотрении отдельного значения для r_i в каждой точке i. Для определения значения σ_i применяется скользящее окно размером в два раза большим, чем измеряется частота кадров.

$$r_i = \left(1 - \frac{\mu_i}{max\_value}\right)^2$$

, где max_value соответствует либо ширине в траектории горизонтальной трансляции, либо высоте в вертикальной траектории трансляции. Значение μ_i вычисляется таким образом, чтобы придать больший вес точкам, расположенным ближе к i, где μ_i выражается как:

$$\mu_i = \frac{\sum_{j \in W_i, j \neq i} G(|j-i|, \sigma_\mu)\,\Delta_j}{\sum_{j \in W_i} G(|j-i|, \sigma_\mu)}$$

где j - это индекс каждой точки в окне i, где в качестве G() - гауссовская функция с σ, вычисленная как $\sigma_\mu = FPS(1 - CV)$,

$$CV = \frac{std(\forall t_i \mid i \in W_i)}{avg(\forall t_i \mid i \in W_i)}$$

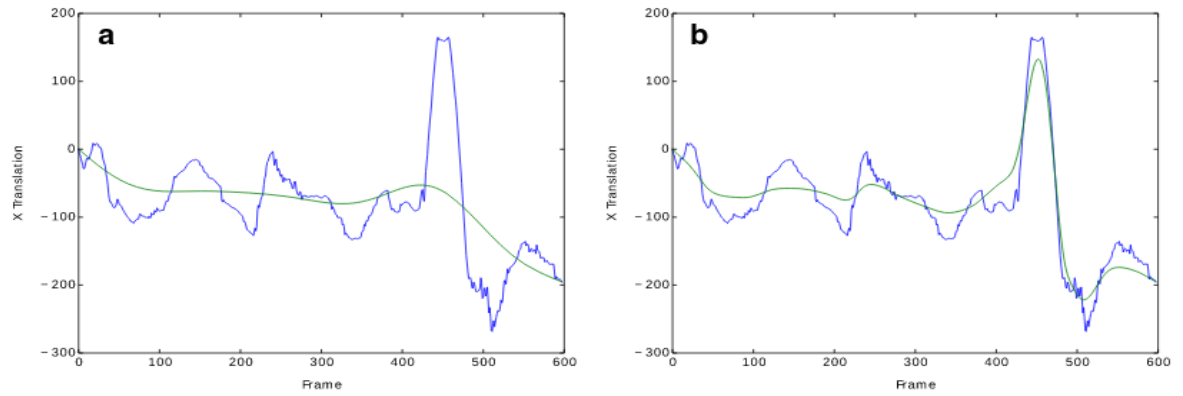


Fig. 4 Smoothing of camera motion trajectories. **a** Gaussian filter with $\sigma = 40$. **b** Adaptive Gaussian filter

Table 2 Comparison between Gaussian filter and Kalman filter

No. of videos	Original	Gaussian filter $\sigma = 40$		Kalman filter	
	ITF	ITF	Hold pixels (%)	ITF	Hold pixels (%)
1	18.793	27.738	69.276	25.888	71.000
2	20.390	29.331	71.750	27.201	74.771
3	16.186	22.559	72.972	22.122	73.003
4	19.965	33.380	48.958	26.298	54.903
5	23.277	28.660	2.540	25.991	4.958
6	19.681	29.804	67.891	25.576	73.507
7	24.109	28.510	60.495	28.063	57.167
8	17.881	25.448	70.648	24.081	72.287
9	19.248	23.251	25.797	21.426	33.818
10	12.972	18.453	17.519	16.680	27.204
11	21.487	26.826	43.599	25.704	52.875
12	15.081	0	0	20.219	2.686
13	23.841	30.621	70.312	28.200	71.875
14	18.065	20.265	7.448	20.902	7.642
Average	19.355	24.631	44.953	24.167	48.406

Table 3 Comparison between semi-adaptive Gaussian filter and adaptive Gaussian filter

No. of videos	Original	Semi-adaptive Gaussian filter		Locally adaptive Gaussian filter	
	ITF	ITF	Hold pixels (%)	ITF	Hold pixels (%)
1	18.793	27.620	70.745	27.455	74.500
2	20.390	29.331	71.750	28.914	75.781
3	16.186	22.559	72.972	22.090	76.056
4	19.965	33.380	48.958	27.931	62.465
5	23.277	27.814	8.312	27.360	53.385
6	19.681	29.804	67.891	29.077	70.838
7	24.109	28.510	60.495	28.876	73.667
8	17.881	25.448	70.648	25.182	73.284
9	19.248	21.845	35.750	21.435	57.139
10	12.972	17.465	27.907	16.381	70.296
11	21.487	26.826	43.559	25.659	57.260
12	15.081	19.827	16.611	17.895	59.847
13	23.841	30.621	70.312	29.987	71.719
14	18.065	19.759	39.045	19.773	54.146
Average	19.355	25.772	50.353	24.858	66.455

Полуадаптивная версия поддерживает больше пикселей в видео, в которых оригинальный Гауссовский фильтр имел проблемы, так как к ним был применен σ=20. Однако, количество пикселей в кадрах меньше, чем в других видео. Это связано с тем, что во многих случаях σ=20 всё ещё является очень высоким значением. С другой стороны, меньшее значение σ могут игнорировать колебания, присутствующие в других примерах видео, создавая таким образом видео, не достаточно стабилизированное и, следовательно, с меньшим значением ITF. Поэтому, как видно из Таблицы 3, локально адаптивная версия, чья интенсивность сглаживания изменяется по траектории, получила значения ITF, сопоставимые с оригинальной и полуадаптивной версиями, сохраняя, возможно, большее количество пикселей.

Предлагаемый фильтр присваивает σ отдельные значения вдоль траектории камеры, учитывая, что интенсивность колебаний меняется на протяжении всего видео.

1.7 Real-time optical flow-based video stabilization for unmanned aerial vehicles

В целом, алгоритмы стабилизации видео следуют трем основным этапам: (1) определение движения, (2) компенсация движения и (3) композиция изображения. Большинство методов вращаются вокруг нахождения двухмерной модели движения (например, гомография) для расчета глобальной траектории движения. Затем к траектории применяется низкочастотный фильтр для отсеивания высокочастотных колебаний. Затем низкочастотный параметр применяется к кадрам с помощью деформации. Этот каркас очень эффективен для сцен с небольшим динамическим движением, что применимо к воздушным видео, однако оценка глобальной траектории движения с помощью окна кадров не может достичь мгновенной стабилизации, как это требуется для обработки в реальном времени.

Отмечается, что вышеуказанные методы работают в автономном режиме или с задержкой в несколько кадров. Для задач в реальном времени, кадры должны быть немедленно стабилизированы и представлены с другими задачами, такими как отслеживание и обнаружение объектов.

Также если говорить про область обнаружения, то причина ограничения области заключается в том, что пиксели, находящиеся вблизи краев кадра, имеют высокую вероятность появления в следующем кадре. Таким образом, мы можем предотвратить обнаружение ключевых точек на границах изображения, сэкономив время на вычисление, необходимое для совпадения тех ключевых точек, которые имеют высокую вероятность исчезновения в следующем кадре

Оценивая перемещение угловых точек, нет необходимости обнаруживать углы в следующей раме. А так как перемещение углов известно в следующей рамке, то согласование уже выполнено. Подгонка оптического потока повторяется для пяти кадров с детектированием углов только для первого кадра, а затем происходит повторное детектирование углов в шестом кадре. При этом тоновое совпадение не нарушает двух основных предположений оптического потока:

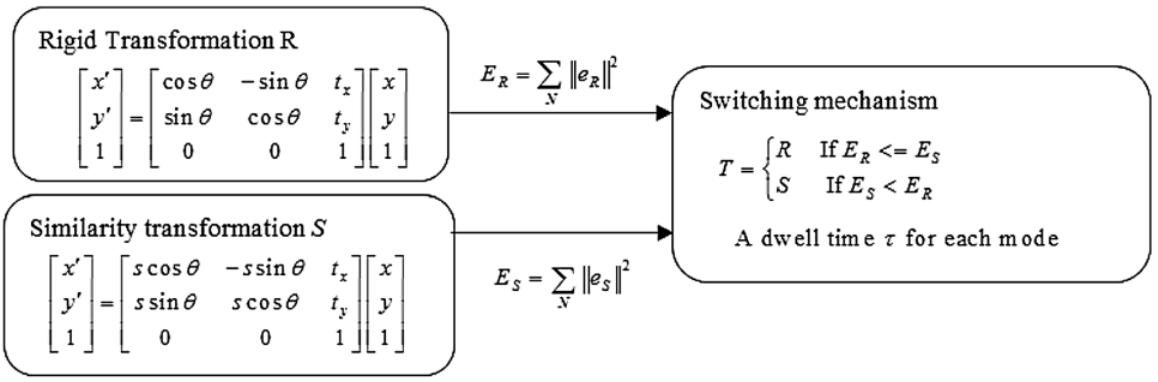
- интенсивность пикселей объекта не меняется между последовательными кадрами
- соседние пикселы имеют похожее движение

Предположения не срабатывают при большом движении. Таким образом, используются пирамиды для удаления малых движений, а большие движения становятся малыми, когда мы поднимаемся выше в пирамиде. Теперь, применяя Лукас-Канаду, оптический поток получается вместе с масштабом. Таким образом, векторы движения - это координаты ключевых точек, существовавших в предыдущем кадре и переместившихся на новое место в текущем кадре. После получения новых местоположений ключевых точек, мы знаем, что существует набор углов, которые находятся как в предыдущем кадре, так и в текущем кадре. Это позволяет нам создать матрицу преобразования внутри кадра, которая используется для моделирования движения и композиции изображения.
В предыдущих работах оценка движения выполнялась либо с использованием фильтров частиц, либо с помощью вариационных методов, и в основном с использованием методов согласования, основанных на инвариантном преобразовании признаков (SIFT) масштаба, все из которых не подходят для реализации в реальном времени, рассматриваемой в данной работе.
В общем случае оптимизация и подбор движения аффинного преобразование [A|t] формулируется следующим образом:

$$[A^*|t^*] = \arg\min \sum_i ||\mathrm{dst}[i] - A\mathrm{src}[i]^T - t||^2$$

, при этом требуется минимум 6 невырожденных корреспондированных точек, так как степень свободы у A = 6.

E_r и E_s представляют собой общее корневое среднеквадратическое расстояние между N углами предыдущего кадра и текущего после применения соответствующих преобразований. Если среднеквадратическое расстояние для жесткого преобразования меньше, чем для преобразования подобия, то механизм выберет вычисление жесткого преобразования для последовательности кадров (т.е. время выдержки). Для достижения баланса между скоростью и стабильностью, для переключения между жестким преобразованием и преобразованием схожести используется время выдержки в 20 кадров



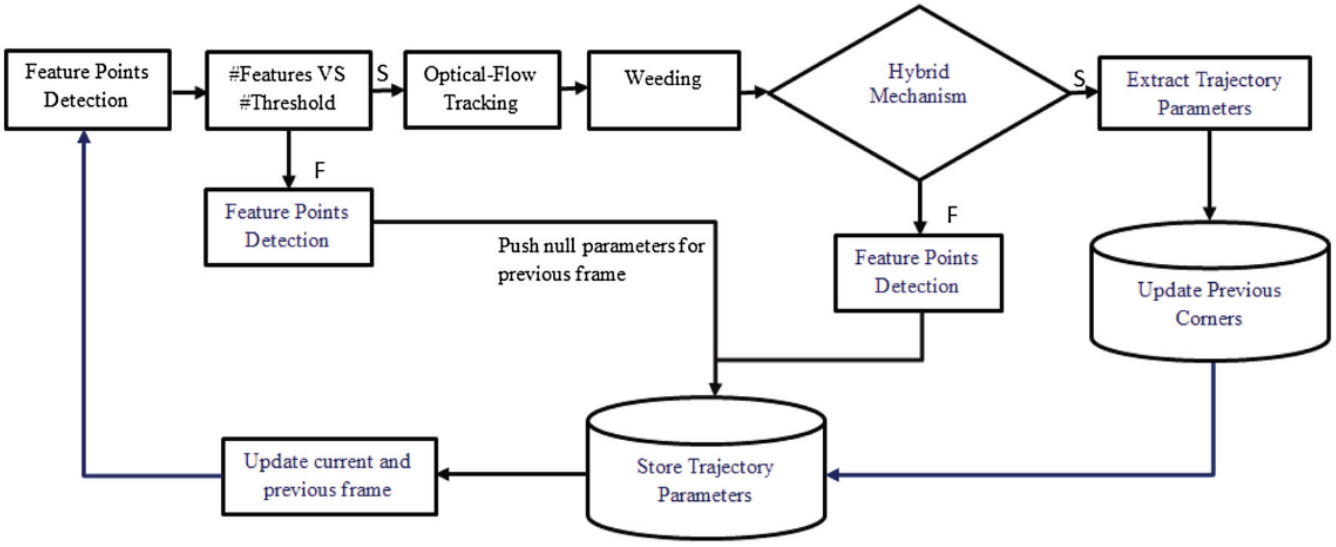
На исходном каркасе как жесткие, так и аффинные матрицы оцениваются из набора ключевых точек, которые детектируются и сопоставляются с помощью оптического потока. Затем с помощью двух матриц преобразования оцениваются два набора движений. Вычисляется среднее расстояние между новыми точками и местами расположения ключевых точек для обоих преобразований. Если среднее расстояние между точками, генерируемыми жестким преобразованием, меньше, чем среднее расстояние между точками, генерируемыми преобразованием подобия, то механизм останется жестким преобразованием или переключится на трансформирование подобия. Таким образом, алгоритм вернет трансформирование, которое дает меньшее изменение расстояния между точками.

Гибридный механизм переключается между жестким и частичным аффинным преобразованием. Это означает, что мы стабилизируем колебательные движения, используя только до четырех степеней свободы, а именно трансляцию в направлении X, трансляцию в направлении Y, масштабный коэффициент и коэффициент вращения. Это требуется, чтобы:

1) при выборе трансформации с меньшим общим корневым и квадратным расстоянием в последовательных кадрах, переключающий механизм динамически стабилизирует и уменьшает дрожание.
2) выбирая преобразование с меньшим количеством параметров, механизм переключения по возможности сокращает вычислительное время

На рисунке снизу представлен обзор рабочего процесса оценки движения в предлагаемом алгоритме. Он начинается с определения углов в первом кадре, а затем использует оптический поток для определения и отслеживания углов в последующих кадрах. В редких случаях, если количество обнаруженных углов слишком мало, то для этого кадра стабилизация пропускается. Кроме того, каждые

пять кадров переопределяются ключевые точки для поддержания хорошего качества оценки движения.



За отслеживанием оптического потока следует этап “прополки”, который проверяет последовательность потока в обратном направлении, тем самым устраняя плохое совпадение. Для каждых двадцати кадров используется гибридный механизм, позволяющий определить наилучшее преобразование, которое необходимо оценить. Затем из оценки извлекаются и сохраняются траектории параметров. Процесс повторяется для общего количества кадров в видео.

На этапе компенсации движения мы используем базовый метод накопления параметров траектории с последующим их сглаживанием с помощью окна усреднения. Исходная траектория параметров накапливается в окне кадров, что в два раза больше величины сглаживания по градиенту, а затем результат сложения каждого параметра кумулятивно усредняется. Далее к исходной траектории добавляется разность между усредненной траекторией и накопленной траекторией. Полученная величина называется сглаженной траекторией.

Обратите внимание, что окно большого среднего размера подвержено риску включения релевантных кадров по сравнению с целевой рамкой. И наоборот, небольшой радиус сглаживания приводит к неадекватному и неэффективному сглаживанию высокочастотных параметрических траекторий. Поэтому необходимо находить сбалансированные настройки.

Следует отметить, что траектория параметра - это достаточно абстрактная величина, которая не обязательно имеет прямое отношение к движению, вызываемому камерой. Для простой панорамной сцены со статическими объектами она имеет прямую связь с абсолютным положением изображения. Важным

моментом является то, что траектория может быть сглажена, даже если она не имеет никакой физической интерпретации.

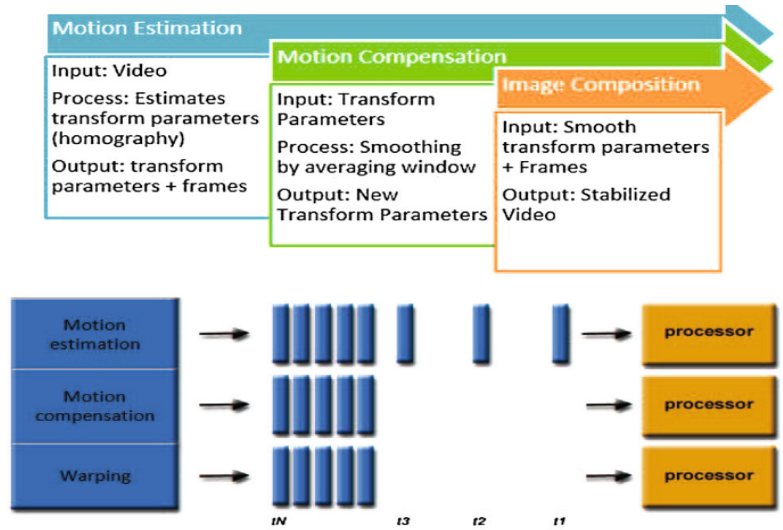


Fig. 11 Multi-threaded concurrent work flow

Реализация в реальном времени состоит из трех потоков, как показано на рисунке сверху, которые способны работать независимо друг от друга. Параллельные вычисления используются для разделения алгоритма стабилизации видео на три части: поток 1 (оценка движения), поток 2 (компенсация движения) и поток 3 (деформация). В первом потоке оценка движения выполняется для первых двадцати входящих кадров (радиус сглаживания для компенсации движения установлен на десять), а второй поток ждет завершения этого процесса. Эта задержка обусловлена тем, что для компенсации движения и деформации требуются сглаженные траектории параметров, которые могут быть сгенерированы только после обработки первых двадцати кадров. С 21 кадра все три потока работают одновременно. Другими словами, оценка движения между 20-м и 21-м кадрами получается с помощью потока 1, в то время как другие потоки вычисляют сглаженную траекторию, используя информацию, генерируемую со второго кадра на 21-й. Весь процесс продолжается бесконечно, пока не закончится видеопоток.

Так как сами потоки обрабатывают кадры намного быстрее, чем частота кадров, тайминги потоков в конечном итоге догоняют тайминги доступности кадров. Как только поток оценки движения сможет открыть видеопоток, он будет продолжать оценивать матрицу гомографии между каждым кадром до тех пор, пока не достигнет конца видеопотока. Поток компенсации движения будет запущен, как

только поток оценки движения завершит оценку для двадцати кадров (время выдержки).

Поток компенсации движения усредняет параметры преобразования в памяти для параметров преобразования. С этого момента всякий раз, когда поток компенсации движения оценивает новую гомографию, поток компенсации движения удаляет самые старые параметры преобразования и вставляет самые последние параметры, а затем выполняет сглаживание. Напомним, что этап компенсации движения не изменился по сравнению с предыдущей реализацией. Тем не менее, на этом этапе больше не нужно работать с целыми параметрами видео, вместо этого он собирает данные о параметрах преобразования, заданных потоком компенсации движения, и управляет ими динамически.

Как только компенсация движения выдаст сглаженные параметры последнего кадра, он вставит данные в хранилище, чтобы поток композиции изображения смог извлечь данные для отображения или хранения соответственно