

Санкт-Петербургский государственный университет

Программная инженерия
Кафедра системного программирования

Поляков Александр Романович

Автоматическая калибровка и синхронизация камеры и МЭМС-датчиков

Курсовая работа

Научный руководитель:
ст. преп. кафедры системного программирования Я. А. Кириленко

Санкт-Петербург
2018

Оглавление

Введение	3
1. Постановка задачи	5
2. Общие сведения	6
2.1. Камера-обскура	6
2.2. Поворот камеры	6
2.3. Эффект «rolling shutter»	7
2.4. Гироскоп	8
2.5. Метрики качества стабилизации	9
2.6. «Особые точки»	9
3. Обзор существующих решений	11
3.1. Калибровка по метрике	11
3.2. Калибровка методами OpenCV	11
3.3. Grid Search Calibration	12
4. Улучшенный алгоритм калибровки: Grid Search + Metric	14
5. Постановка эксперимента	15
Результаты	17
Список литературы	18

Введение

Качество кадров, получаемых с камер современных смартфонов, значительно расширяет возможности решения задач компьютерного зрения. Постепенно появляется всё большее число попыток применить эти практики в таких сферах компьютерного зрения как цифровая стабилизация видео [3], [6], [5], [13], дополненная реальность [7], 3D-реконструкция [12], [10], фотограмметрия на мобильных платформах и встраиваемых системах. Однако, данные алгоритмы требуют больших вычислительных ресурсов, что не позволяет использовать их на упомянутых выше платформах, особенно в режиме реального времени.

Наличие различных датчиков на этих платформах обусловлено их низкой стоимостью и одновременно высокой точностью, что позволяет эффективно использовать их показания. Так как большинство задач, упомянутых выше, так или иначе связаны с определением движения камеры (что является «узким местом» для многих алгоритмов), предпочтение дается МЭМС-датчикам (микроэлектромеханические датчики) движения — гироскопу и акселерометру [8], [2].

Расширение математической модели в алгоритмах компьютерного зрения информацией с МЭМС-датчиков не только повышает качество и уменьшает объем вычислений, но и приводит к появлению новых сложностей. В частности, помимо общих внутренних параметров камеры (фокусное расстояние, оптический центр, rolling shutter), необходимо учитывать параметры датчиков (например, погрешность гироскопа) и параметры модели камера-датчики (ориентация камеры и датчиков, их синхронизация). В случае рассматриваемых платформ (смартфоны и встраиваемые системы) существует большое количество различных моделей камер, датчиков и, как следствие, их комбинаций, а, значит, необходимые параметры разнятся от устройства к устройству. При желании масштабировать существующие алгоритмы компьютерного зрения, использующие МЭМС-датчики, на такие устройства, необходимо иметь алгоритм автоматической калибровки, позволяющий получить нужные значения параметров.

В данной работе представлены результаты исследования в области автоматической калибровки и синхронизации параметров камеры и МЭМС-датчиков.

1. Постановка задачи

Данная работа является продолжением исследования [1], посвященного цифровой стабилизации видео в режиме реального времени с использованием МЭМС-датчиков, и нацелена на разработку и реализацию алгоритма автоматической калибровки ключевых параметров для [1]: фокусного расстояния и параметра синхронизации камеры и гироскопа. Для достижения цели были поставлены следующие задачи.

1. Сделать обзор существующих алгоритмов автоматической калибровки и синхронизации камеры и МЭМС-датчиков.
2. Реализовать существующие алгоритмы автоматической калибровки и провести анализ их точности и производительности.
3. Предложить и реализовать улучшенный алгоритм автоматической калибровки, адаптированный к существующей математической модели в исследовании [1].
4. Разработать кроссплатформенную библиотеку, реализующую данный алгоритм, для дальнейшей интеграции в приложение-прототип стабилизации видео [1].

2. Общие сведения

Данный раздел посвящен определению общих понятий и математических моделей.

2.1. Камера-обскура

Камера-обскура – математическая модель камеры, описывающая отображение объекта из трехмерного мира на двумерное изображение. Это отображение соответствует формуле, в которой X - это координаты точки в реальном мире, x - координаты проекции точки на изображении. Также формула зависит от параметров конкретной камеры. Для каждой камеры зададим соответствующую ей матрицу K с параметрами: f – фокусное расстояние, (o_x, o_y) – оптический центр [14].¹

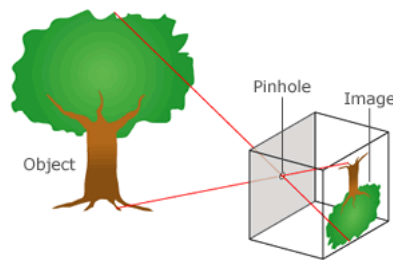


Рис. 1: Камера-обскура

$$\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{pmatrix} f_x & 0 & -o_x \\ 0 & f_y & -o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

2.2. Поворот камеры

В случае поворота камеры в пространстве, используя оператор поворота R , мы получаем следующее отношение между двумя проекциями x_1 и x_2 одной точки X в пространстве, запечатленной во время t_1 (поворот R_1) и t_2 (поворот R_2) соответственно.

¹Изображение взято с веб-сайта https://en.wikipedia.org/wiki/Pinhole_camera_model

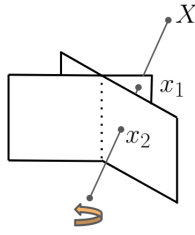


Рис. 2: Поворот камеры

$$x_1 = KR(t_1)X$$

$$x_2 = KR(t_2)X$$

Преобразовывая данные выражения, получаем следующее отношение:

$$x_2 = KR(t_2)R^T(t_1)K^{-1}x_1$$

Таким образом, матрица поворота между моментами времени t_1 и t_2 определяется как:

$$W(t_1, t_2) = KR(t_2)R^T(t_1)K^{-1}$$

$$x_2 = W(t_1, t_2)x_1$$

2.3. Эффект «rolling shutter»

«Rolling shutter» – эффект, возникающий на большинстве КМОП-камер, на которых считывание кадра с матрицы происходит построчно.²

Когда сцена сканируется вертикально, момент времени, в который считывается какая-либо точка, напрямую зависит от строки, в которой точка расположена. Тогда, если i – номер кадра, и y – номер строки в кадре, момент времени, в который была запечатлена точка, вычисляется следующим образом:

$$t(i, y) = t_i + t_s \frac{y}{h}$$

²Изображения взяты с веб-сайта <http://www.red.com/learn/red-101/global-rolling-shutter>



Рис. 3: Движение объекта

где t_i – момент времени, в который был снят кадр с номером i , t_s – время, необходимое для съемки одного кадра, h – высота кадра (количество строк). Эта информация может быть использована для увеличения точности вычисления матриц преобразования в общей математической модели.

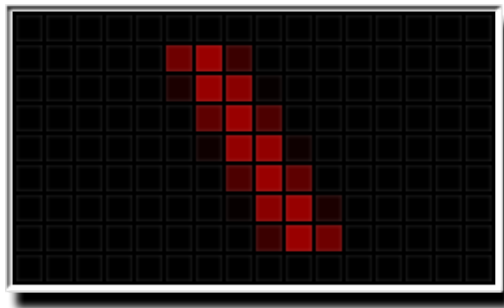


Рис. 4: Эффект rolling-shutter при считывании изображения с матрицы камеры

2.4. Гироскоп

Гироскоп – датчик (МЭМС-датчик в нашем случае), реагирующий на изменение ориентации тела в пространстве. Интегрируя его показания, может быть вычислена матрица поворота (оператор поворота).

Существуют два подхода интегрирования показаний гироскопа с различными вычислительными сложностями и точностями. Первый метод – интегрирование по конкретной оси без учета других. Здесь θ – угол поворота вокруг одной оси, ω – скорость поворота относительно этой оси в промежутке времени между t и $t + \delta$:

$$\theta(t + \delta) = \theta(t) + \int_t^{t+\delta} \omega(t)dt, \quad (1)$$

Такой подход используется в случае незначительных поворотов, причина этому – несовершенство углов Эйлера как алгебраической структуры. Другой, более сложный подход заключается в использовании кватернионов. Статья [4] дает полное описание интегрирования угловых скоростей с помощью кватернионов. Им мы воспользуемся в нашей работе.

2.5. Метрики качества стабилизации

Существуют две основные метрики, позволяющие определить качество стабилизации видео со статической сценой – RMSE (root mean square error) и ITF (inter-frame transformation fidelity).

Первая заключается в сравнении двух кадров попиксельно, используя стандартную L2-метрику.

Метрика ITF зависит от PSNR (peak signal-to-noise ratio) между двумя последовательными кадрами ($k, k + 1$):

$$PSNR(k) = 10 \log_{10} \frac{I_{max}}{MSE(k)},$$

где I_{max} – максимальная яркость пикселя, и подсчитывается как:

$$ITF = \frac{1}{N-1} \sum_{k=1}^{N-1} PSNR(k),$$

где N – количество кадров в видео.

2.6. «Особые точки»

В компьютерном зрении «особая точка» (features) – паттерн, обладающий определенными свойствами. Один из способов применения таких точек – выделение особых точек, нацелен на поиск похожих объектов на двух кадрах. В нашей работе мы используем особые точки для определения движения камеры между кадрами.

В экспериментах были задействованы два типа особых точек, выбранных по принципу простоты использования и точности : ORB (Oriented FAST and rotated BRIEF) [11] и SIFT (Scale-Invariant Feature Transform) [9]. SIFT дает более точные результаты, однако требователен к вычислительным ресурсам, в то время как ORB менее точен, но значительно выигрывает SIFT по скорости.

3. Обзор существующих решений

Данный раздел посвящен обзору существующих методов автоматической калибровки и синхронизации камеры и МЭМС-датчиков, также в нем рассмотрены основные преимущества и недостатки данных подходов.

3.1. Калибровка по метрике

Данный метод является действующим алгоритмом калибровки в исследовании [1]. Он основан на метрике качества стабилизации, описанной в пункте 2.5. Используя метрику ITF, мы можем определить качество стабилизации видео после трансформации кадров видео: чем больше значение метрики, тем лучше стабилизировано видео. Алгоритм находит следующие параметры: фокусное расстояние, временное смещение между камерой и гироскопом, rolling shutter. Так как нам известны возможные диапазоны значений всех параметров, простейшим решением будет полный перебор всех параметров. Именно это и делает данный алгоритм. Несмотря на большую вычислительную сложность, такой подход является самым точным, так как он сильно завязан на текущую математическую модель стабилизации видео.

3.2. Калибровка методами OpenCV

Алгоритм применим лишь в случае заранее известной геометрии объекта на сцене. Находит следующие параметры: фокусное расстояние, оптический центр, коэффициенты искажения³. Еще одним ограничением данного подхода является наличие у объекта на сцене легко выделяемых особых точек, называемых паттерном калибровки. Примером такого паттерна может служить шахматная доска.

Основные шаги алгоритма:

³https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

1. Подсчет начальных значений параметров камеры, коэффициенты искажения устанавливаются равными нулю.
2. Вычисление позиции камеры с помощью PnP-метода.⁴
3. Используя алгоритм Левенберга-Марквардта минимизируется ошибка репроецирования (re-projection error).

3.3. Grid Search Calibration

Используя кадры видео и показания гироскопа, можно двумя способами найти движение камеры:

1. Использовать особые точки на кадрах и находить искомое движение через разницу между соответствующими особыми точками на двух последовательных кадрах.
2. Использовать показания гироскопа.

Данный подход заключается в следующем: во-первых, вычисляются значения двух функций, описывающих движения камеры с помощью особых точек и показаний гироскопа. Эти функции должны зависеть от времени. Имея такие две функции, можно найти параметр синхронизации (временной сдвиг) камеры и гироскопа, используя кросс-корреляцию.

Определим функции:

$$r_f(t) = \frac{\sum_{m \in M(t)} (m_x - m'_x) + (m_y - m'_y)}{2|M(t)|(t_i - t_{i-1})}$$

$$r_g(t) = \frac{\omega_x(t) + \omega_y(t) + \omega_z(t)}{3}$$

На рис. 5 можно видеть, что описанные с помощью этих двух функций движения камеры, имеют схожий вид, так же по сдвигам на пиках можно судить о наличии временной задержки между этими двумя функциями, о которой упоминалось выше.

⁴<https://en.wikipedia.org/wiki/Perspective-n-Point>

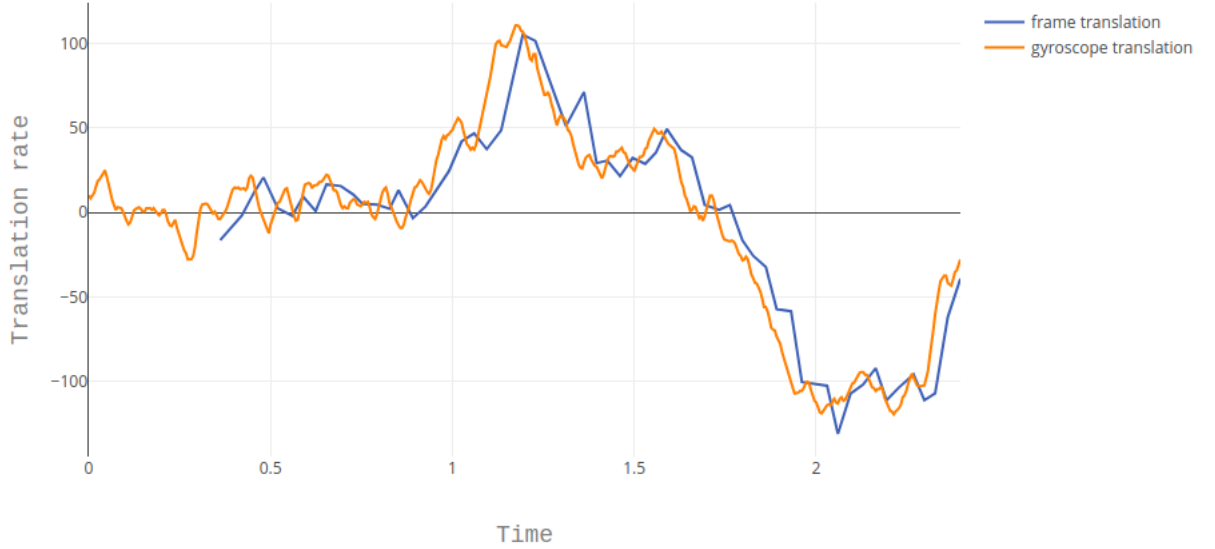


Рис. 5: Разница во времени между кадрами видео и показаниями гироскопа.

В данной работе мы опробовали две стандартные функции кросс-корреляции:

$$s(a, b) = a * b$$

$$s(a, b) = -|a - b|$$

Таким образом, если имеется диапазон возможных значений разницы во времени T_d , мы можем найти искомый параметр синхронизации камеры и гироскопа, найдя максимальную корреляцию между функциями $r_f(t)$ и $r_g(t)$:

$$offset = \arg \max_{t_d \in T_d} \sum_{t \in T} s(r_f(t - t_d), r_g(t))$$

Авторы, предложившие данный подход, заявляют, что фокусное расстояние – масштаб функции $r_f(t)$, и найти его можно следующим образом:

$$r_g(t) = f * r_f(t)$$

Используя метод наименьших квадратов:

$$f = \arg \min_f \sum_{i=1}^n (r_f(t_i + t_d) - f * r_g(t_i))^2$$

4. Улучшенный алгоритм калибровки: Grid Search + Metric

В рамках данной курсовой работы был предложен новый алгоритм, являющийся комбинацией калибровки по метрике и Grid Search калибровки. Параметр синхронизации камеры и гироскопа находится так же, как и в калибровке по метрике. Затем фокусное расстояние ищется как значение, при котором значение ITF метрики максимально:

$$f = \arg \max_{f \in F} ITF(f, t_d)$$

Такой подход хорошо себя показал в сочетании с прототипом алгоритма стабилизации [1]. Незначительным ограничением данного метода является требование к минимизации движения камеры вокруг оптической оси камеры, так как такие движения имеют нелинейную корреляцию с угловыми скоростями по этой оси, что приводит к ошибочным результатам работы алгоритма. Однако, стоит заметить, что данное ограничение не является существенным.

5. Постановка эксперимента

Внутри фреймворка, разработанного в исследовании [1], были реализованы и протестированы все вышеописанные алгоритмы автоматической калибровки. Видео для тестирования были получены со смартфона с параметрами:

- Модель: Xiaomi Redmi 3S;
- Версия Android: 6.0.1 (build MMB29M).

Тестирование проходило на видео с различными типами сцены (статическая и динамическая) и движения камеры. Алгоритм калибровки по метрике был выбран эталонным в силу его полного соответствия нашей математической модели камеры. В качестве функции кросс-корреляции была выбрана следующая функция, так как во время постановки экспериментов она показала лучшие результаты по сравнению с другими.

$$s(a, b) = -|a - b|$$

В качестве особых точек были взяты как ORB, так и SIFT. Также стоит заметить, что для тестирования алгоритмов из видео брались лишь первые 60 кадров (примерно 2 сек), что положительно сказывается на производительности, необходимой в смартфонах и встраиваемых системах.

Результаты сравнения алгоритмов представлены в таблицах.

Algorithm	Offset, μs	f	Metric
Metric (standard)	45	850	14.04
Grid Search + ORB	45	825	13.97
Grid Search + SIFT	45	950	13.33
Grid Search + Metric + ORB	45	850	14.04
Grid Search + Metric + SIFT	45	850	14.04

Algorithm	Offset, μs	f	Metric
Metric (standard)	45	850	16.25
Grid Search + ORB	50	850	16.10
Grid Search + SIFT	40	925	15.87
Grid Search + Metric + ORB	50	850	16.10
Grid Search + Metric + SIFT	40	850	15.53

Algorithm	Offset, μs	f	Metric
Metric (standard)	45	850	15.82
Grid Search + ORB	45	950	15.05
Grid Search + SIFT	50	825	15.31
Grid Search + Metric + ORB	45	850	15.82
Grid Search + Metric + SIFT	50	850	15.30

В первых двух таблицах представлены результаты калибровки в случае одномерного движения камеры. Видно, что особые точки ORB и SIFT в предложенном нами алгоритме «Grid Search + Metric» имеют одинаковую точность. Особо важным является то, что фокусное расстояние, найденное нашим алгоритмом совпадает с эталонным.

В третьей таблице представлены результаты калибровки в случае двумерного движения камеры. По результатам, данный случай эквивалентен предыдущему.

Эксперименты показали, что предложенный нами алгоритм «Grid Search + Metric» является наиболее подходящим для использования в математической модели [1]. В качестве конфигурации были выбраны показавшие наилучший результат функция кросс-корреляции

$$s(a, b) = -|a - b|$$

и особые точки ORB.

Результаты

В рамках данной курсовой работы были выполнены следующие задачи.

1. Сделан обзор существующих алгоритмов автоматической калибровки и синхронизации камеры и МЭМС-датчиков.
2. Реализованы и протестированы существующие аналоги алгоритмов автоматической калибровки камеры и МЭМС-датчиков.
3. Разработан новый алгоритм автоматической калибровки камеры и МЭМС-датчиков.
4. Разработанный алгоритм интегрирован в приложение-прототип стабилизации видео [1].

Данная работа была принята к докладу на конференции SYRCoSE, а также к публикации в сборнике трудов института системного программирования <http://www.ispras.ru/proceedings/>.

Список литературы

- [1] A.V. Kornilova I.A. Kirilenko N.I. Zabelina. Real-time digital video stabilization using MEMS-sensors // Proceedings of the Institute for System Programming. — 2017. — Vol. 29, no. 4. — P. 2220–6426.
- [2] Bell Steven, Troccoli Alejandro, Pulli Kari. A Non-Linear Filter for Gyroscope-Based Video Stabilization. — 2014.
- [3] CodingFlow: Enable Video Coding for Video Stabilization / Shuaicheng Liu, Mingyu Li, Shuyuan Zhu, Bing Zeng. — 2017. — 04. — Vol. 26. — P. 1–1.
- [4] Diebel James. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. — 2006.
- [5] Feature-based real-time video stabilization for vehicle video recorder system / Wu-Chih Hu, Chao-Ho Chen, Yi-Jen Su, Tzu-Hsing Chang // Multimedia Tools and Applications. — 2018. — Mar. — Vol. 77, no. 5. — P. 5107–5127. — URL: <https://doi.org/10.1007/s11042-017-4369-7>.
- [6] Grundmann Matthias, Kwatra Vivek, Essa Irfan. Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. — 2011.
- [7] Jia Chao, Evans Brian L. Online Calibration and Synchronization of Cellphone Camera and Gyroscope. — 2013.
- [8] Karpenko Alexandre, Jacobs David, Baek Jongmin. Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. — 2011.
- [9] Lowe David G. Distinctive Image Features from Scale-Invariant Keypoints // Int. J. Comput. Vision. — 2004. — . — Vol. 60, no. 2. — P. 91–110. — URL: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [10] Madgwick Sebastian O.H. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. — 2010.

- [11] ORB: An efficient alternative to SIFT or SURF / E. Rublee, V. Rabaud, K. Konolige, G. Bradski // 2011 International Conference on Computer Vision. — 2011. — Nov. — P. 2564–2571.
- [12] Ovren Hannes, Forsseñ Per-Erik. Gyroscope-based Video Stabilisation With Auto-Calibration. — 2013.
- [13] Subspace video stabilization / F Liu, M Gleicher, J Wang et al. — 2011.
- [14] Szeliski Richard. Computer Vision: Algorithms and Applications. — 2010.