Auto-calibration and synchronization of camera and MEMS-sensors

A.R. Polyakov <polyakov.alx@gmail.com>
A.V. Kornilova <kornilova.anastasiia@gmail.com>
I.A. Kirilenko <y.kirilenko@spbu.ru>
Saint Petersburg State University, Software Engineering
28 Universitetskiy prospect, Petergof, Sankt-Peterburg, Russia, 198504

Abstract. This article describes our ongoing research on auto-calibration and synchronization of camera and MEMS-sensors. The research is applicable on any system that consists of camera and MEMS-sensors, such as gyroscope. The main task of our research is to find such parameters as the focal length of camera and the time offset between sensor timestamps and frame timestamps, which is caused by frame processing and encoding. This auto-calibration makes possible to scale computer vision algorithms (video stabilization, 3D reconstruction, video compression, augmented reality), which use frames and sensor's data, to a wider range of devices equipped with a camera and MEMS-sensors. In addition, auto-calibration allows completely abstracting from the characteristics of a particular device and developing algorithms that work on different platforms (mobile platforms, embedded systems, action cameras) independently of concrete device's characteristics as well. The article describes the general mathematical model needed to implement such a functionality using computer vision techniques and MEMS-sensors readings. The authors present a review and comparison of existing approaches to auto-calibration and propose own improvements for these methods, which increase the quality of previous works and applicable for a general model of video stabilization algorithm with MEMS-sensors.

Keywords: camera calibration; auto-calibration; digital signal processing; rolling shutter; computer vision; grid search

1. Introduction

The high quality of frames, received from modern smartphone cameras, expands the frontiers of solutions in computer vision tasks. Lately, there are more and more attempts to scale current practices in such areas of computer vision as video

stabilization [1], [2], [3], [4], augmented reality[5], 3D reconstruction [6], [7], photogrammetry on mobile platforms and embedded systems. However, these algorithms demand big computational resources that not allows applying them to above-mentioned platforms and in real time.

The presence of numerous different sensors on these platforms, caused by the low cost of their production and high precision at the same time, allows using their data effectively. As the majority of above-stated tasks is any way connected with detection of camera movement (which is the "bottleneck" in most algorithms), the main preference is given to motion sensors – gyroscope and accelerometer [8], [9].

Expansion of mathematical model of computer vision algorithm not only increases quality and reduces calculations but gives rise to new difficulties. In particular, besides general intrinsic parameters of the camera (focal length, optical center, rolling shutter) there are parameters of sensors (i.e, bias for gyroscope) and parameters of model "camera-sensors" (camera and sensors orientation, camera and sensors synchronization parameters). Therefore, if desired to scale an algorithm to a large amount of platforms (for example, in case of mobile phones) automatic calibration of these parameters is needed. It is caused by a big variety of cameras, sensors and their combinations.

This work is a continuation of the research [10] conducted on a subject of real-time digital video stabilization using MEMS-sensors and aims to prototype and implement an algorithm of auto-calibration of key parameters for this task: focal length and parameters of synchronization of frames and gyroscope data.

2. Preliminaries

This section is devoted to basic definitions, general mathematical models, and agreements, which will come out throughout this work.

2.1 Pinhole camera model

Pinhole camera model (fig. 1) is a basic mathematical camera model, which describes a mapping from 3-dimentional real world to its projection onto the image. This mapping satisfies the formula, in which X is coordinates of a point in real world and x is coordinates of its projection. In addition, it depends on camera parameters: f – focal length, (ox, oy) – optical center [11].

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

2.2 Rotation camera model

In case of camera rotation in space using rotation operator R, we get the next relationship between two projections x_1 and x_2 of one point in space X caught at a different time t_1 (rotation R_1) and t_2 (rotation R_2) correspondingly (fig. 2).
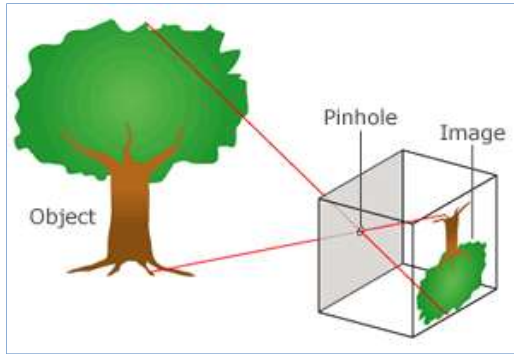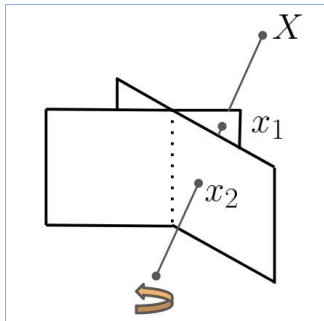
Fig. 1. Pinhole camera model

Fig. 2. Rotation camera model

$$x_1 = KR(t_1)X$$
$$x_2 = KR(t_2)X$$

By transforming these expressions, the following needed relationship is established:

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

Thus, the matrix of image transformation between moments in time t1 and t2 is defined as:

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

2.3 Rolling shutter effect

«Rolling shutter» (fig. 3, 4) is an effect arising on the majority of CMOS cameras, at which each row of the frame is shot at different time due to vertical shutter.

When shutter scans the scene vertically, the moment in time at which each point of the frame is shot, directly depends on the row it is located in. Thus, if i is the number of the frame and y is the row of that frame, then the moment, at which it was shot can be calculated this way:

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

where t_i is the moment when frame number i was shot, t_s is the time it takes to shot a single frame, h is the height of the frame. This can be used to make the general model more precise, when calculating the image transformation matrix.
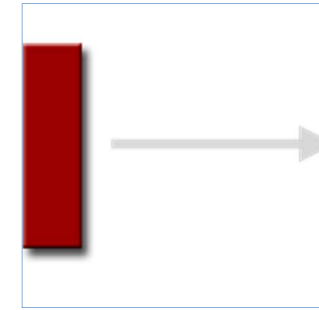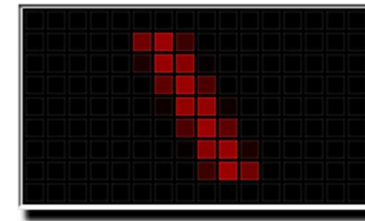

Fig. 3. Object movement

Fig. 4. Rolling-shutter effect during capturing the moving object

2.4 Gyroscope

The gyroscope is a sensor (MEMS-sensor in our case) which sends information about angular velocities of a body. Using this data and its timestamps, a rotation matrix (rotation operator) can be calculated through integration.

There are two approaches for integration data of gyroscope with different computational complexity and accuracy. The first approach is linear integration for receiving Euler angles and then their transformation to a rotation matrix, where θ – is rotation angle of one axis and ω – velocity over this axis between t and t + δ:

$$\theta(t + \delta) = \theta(t) + \int_t^{t+\delta} \omega(t)dt, (2)$$

This approach is applied only in case of insignificant and small rotations, because of the imperfection of Euler angles as an algebraic structure. The other and more complex approach is to use quaternions for data integration. This article [12] gives a full description about the integration of angular velocities using quaternions, and we tend to apply it.

2.5 Stabilization quality metrics

There are two main metrics which can estimate the quality of video stabilization of static scene – RMSE (root mean square error) and ITF (inter-frame transformation fidelity). The first is a comparison between two frames pixel-by-pixel using typical L2 metric. The ITF metric directly depends on PSNR (peak signal-to-noise ratio) parameter between two consecutive frames (k, k+ 1):

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

where Imax is maximum pixel intensity, and is counted as:

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

where N is count of frames in the video.

2.6 Features

In the computer vision, feature is a pattern that satisfies certain properties and can be detected on the image. One of directions of feature use is feature matching, which is mainly focused on searching of similar objects on two frames. In our work, we use feature matching to estimate how the camera moved through shooting.

In our experiments we have used two features types – ORB (Oriented FAST and rotated BRIEF) [13] and SIFT (Scale-Invariant Feature Transform) [14] which prove themselves as the most stable and robust in feature matching. SIFT is considered to exhibit the highest matching accuracies, but requires significant computational resources, while ORB is very fast but less precise [15].

2.7 Description of stabilization algorithm

At the moment stabilization algorithm, proposed in our previous paper [10], works as follows:

1) integrate gyroscope data (angular velocities and timestamps) using quaternions;

2) determine frame timestamp and corresponding rotation matrix;

3) count transformation camera matrix for every horizontal section of the frame (typically, there are several gyro reading per frame and, consequently, several rotation matrices);

4) transform every section using transformation matrix and combine them;

5) write transformed frame to the video.

The algorithm stabilizes video like a tripod, at now complex camera motion is not supported, but in progress.

3. Detailed problem description

As it was mentioned in the description of the stabilization algorithm, it directly depends on camera parameters: focal length, optical center and rolling shutter parameter. In most cases, all parameters besides focal length can be got from API of the device on which this algorithm runs (at the moment the major advantage is given to Android platforms). Thus, one of the main goals of this research is to find focal length, which is the most accurate for our stabilization algorithm.

The other significant direction is to synchronize frames received from the camera and data received from sensors (fig. 5). Mistiming is caused by the time needed for frame processing – scanning and encoding. Therefore, we need to find time offset of this processing to consider it in our model.
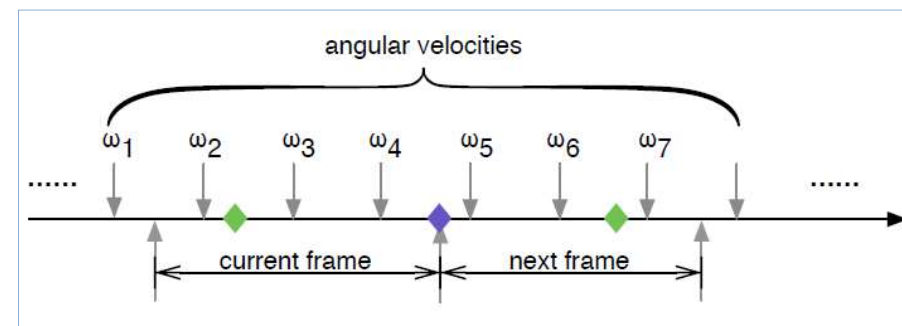

Fig. 5. Matching the time series of frames and gyroscope

Thus, the main goal of this research is to find the suitable focal length and time offset. Some of the described methods are wider and cover other parameters, and we also consider this information.

4. Calibration algorithms

In this section, we describe various approaches that we have tested during this research. The section contains a description of our basic method, review and implementation of the most known methods of calibration from other areas, and our improvements on these methods for our specific task.

4.1 Calibration based on stabilization metrics

focal length, time offset, rolling shutter

This simple approach is based on stabilization metrics described in section 2. Using ITF metric, we can estimate the quality of video stabilization after transformation of frames: the higher the value of metric – the better video is stabilized.

The approach determines three parameters: focal length, time offset and rolling shutter parameter and is as follows: detect a range and step of each parameter (for

example, range of focal length – [500, .., 1200] and step – 50) and find tuple of parameters on which metric is maximized using brute-force search.

It is worth noting, despite of the huge computational complexity this method gives the most accurate results due to the strong dependence on the current mathematical model.

4.2 OpenCV calibration method

focal length, optical center, distortion coefficients

This algorithm is applicable only in case of known geometry of subject which is on the scene. Also, the subject should contain easily distinguished feature points. This subject is usually called calibration pattern. We have used use the main calibration pattern which is supported by OpenCV – chessboard. It depends on such parameters as size of chessboard, the distance between cells and others.

The algorithm also determines distortion coefficients and is as follows:

1) count initial intrinsic parameters of the camera. Initial distortion coefficients are equal to zero;

2) estimate camera position using this initial parameters using PnP method;

3) using Levenberg-Marquardt algorithm minimize reprojection error – sum of square root distances between two matched point.

4.3 Grid search method

focal length, time offset

Using frames and gyroscope data, we can estimate the motion of camera in two ways:

1) use feature points on frames and estimate motion using the difference between matched points on consequence frames;

2) use data of gyroscope – measurements and their timestamps.

This approach is as follows. Firstly, we determine two functions which describe the average measure of camera motions in two ways – using feature points and using gyroscope measurements. These functions must depends on time and if necessary must have facilities for interpolation (data of gyroscope is discrete). Having these functions, that describes motion in different ways, we can estimate shift (time offset) of functions using cross-correlation.

Let us determine these functions:

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

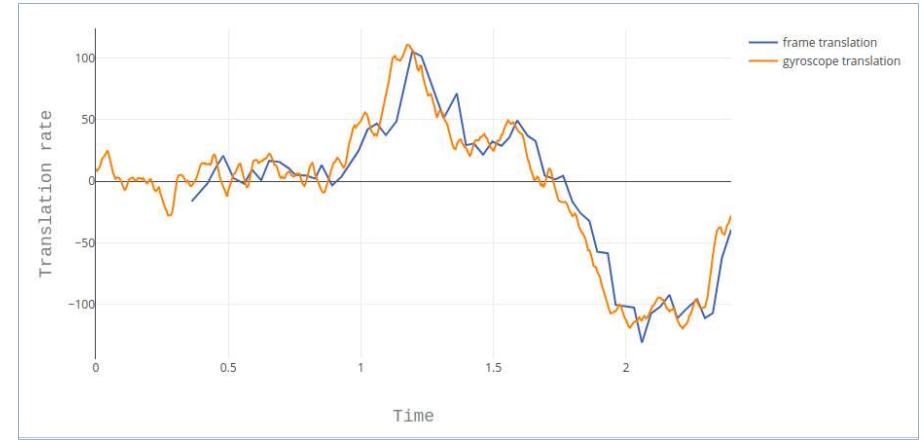$$r_g(t) = \frac{\omega_x(t) + \omega_y(t) + \omega_z(t)}{3}$$

Fig. 6. Time offset between frame and gyroscope

On the picture (fig. 6), you can see similar shape of these functions.

We have tried two typical cross-correlation functions to find offset:

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

If we have a set of possible offsets Td, we can find offset with a maximum value of correlation between frames and gyroscope functions:

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

Authors who support this approach tend to opinion that initial scale constant is a focal length value and try to find this constant like:

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

Using a method of the least squares:

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

4.4 Improvements for grid search method

This method presents a combination of two methods – method with stabilization metrics and method with grid search. The time offset is found by grid search method. If we have a set of possible focal lengths F and the calculated value of time offset, we can calculate a value of focal length. which maximizes stabilization metric:

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

This method is suitable very well in case of using these time offset and focal length in our video stabilization algorithm.

In addition, we have abandoned to take in account motion over zaxis, which is perpendicular to the camera matrix. This motion has non-linear correlation with linear angular velocity over this axis and leads to an error in the algorithm.

5. Results of prototyping

In this section, we will describe results of experiments and conditions in which they were conducted.

5.1 Dataset and environment

Our algorithm was tested on a dataset, which consists of video and gyroscope data from smartphones with the Android operating system. For these purposes, we have a special Android application, which records mp4 video file and csv format file with stamps for gyroscope and frame events. This application supports mobile platforms starting with 21 level Android API because of in this API event-driven scheme for camera frames was supported by camera2 interface. The csv file consists of two types of strings: «f» – for frames and «X, Y, Z, timestamp» – for gyroscope readings.

A framework for calibration algorithm comparison was implemented in Python using OpenCV 3.4 library. It consists of modules for video and gyroscope file parsing and a module for integration of gyroscope readings using quaternion. The framework also has opportunities for calculating metric statics for every method.

We have tested our algorithms on a dataset from the smartphone with the following parameters:

- Model number: Xiaomi Redmi 3S;
- Android version: 6.0.1 (build MMB29M).

5.2 Experiments

Inside our framework, we have implemented all described algorithms and compare them using stabilization quality metrics. We have tested algorithms on different scene types and with different camera movements. An algorithm with stabilization metric was considered as standard. All results are presented in tables. We compare grid search method using different cross-correlation functions and different feature detectors.

Experiments show that OpenCV algorithm has the worst result because of it is very sensitive for the scene (user needs to use chessboard or other pattern) and rotation and is not fit for our mathematical model. In the tables 1-3 you can see results of grid search algorithm without/with improvements (metric) in comparison with stabilization metric algorithm.

The algorithm is parametrized with feature types and shows the best results with the second cross-correlation function (similarity function).

Table 1. Result of calibration in case of 1-dimensional motion

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

Table 2. Result of calibration in case of 1-dimensional motion

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

Table 2. Result of calibration in case of 2-dimensional motion

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

The first two tables show the result of calibration in case of 1-dimentional motion. It is demonstrated that in case of ORB and SIFT features results are identical in accuracy. In addition, results show that in case of metric improvements focal length after calibration is equal to standard in comparison with simple grid search.

The third table describes results of calibration in case of 2-dimentional motion. Results are equal to the case of 1- dimentional motion. As we discussed earlier, the algorithm does not consider 3-dimentional motion because of constraints of grid search model.

5.3 Main results

To sum up, experiments have demonstrated that:

1) grid search method shows the better result for our mathematical model of camera and camera motion;

2) using grid search method, the best calibration result is achieved with the second cross-correlation function (similarity function);

3) ORB and SIFT features show equals results in search of the time offset, therefore we can use ORB as a faster method of feature matching;

4) our improvements of grid search with stabilization metric allow to find focal length which is equal to standard;

5) the algorithm supports only two-dimensional motion (except motion over, axis which is perpendicular to camera matrix), but this is not a strong restriction for users, therefore, our algorithm can be used on a large scale.

6. Conclusion

As lately cameras and motion sensors (gyroscope, accelerometer) very often tend to occur on one platform (smartphones or embedded systems), the quantity of the algorithms, using their joint information, has significantly increased. These algorithms directly depends on parameters of the system «camera-sensors» such as focal length, rolling shutter, synchronization parameters, which differ from platform to platform, and therefore these parameters must be calibrated for increasing of scalability.

Our work proposes the method for auto-calibration of focal length and time series offset (synchronization parameter), which is the most suitable for our video stabilization algorithm using MEMS-sensors. We have review different approaches and choose the nearest for our specific task. We have found parameters for this method, which increase the quality of the calibration algorithm.

It worth noting that proposed algorithm can be scaled not only for stabilization video task. It can be scaled for all algorithms, which support our mathematical model of camera and camera movement.

In the future, we plan to expand the count of calibration parameters with rolling shutter parameter and parameter of relative orientation of the camera and sensor axes.

Acknowledgment

References

[1]. S. Liu, M. Li, S. Zhu, and B. Zeng. Codingflow: Enable video coding for video stabilization. IEEE Transactions on Image Processing, vol. 26, Issue 7, Apr. 2017, pp. 3291-3302

[2]. M. Grundmann, V. Kwatra, and I. Essa. Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. In Proc. of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), 2011, pp. 225-232

[3]. W.-C. Hu, C.-H. Chen, Y.-J. Su, and T.-H. Chang. Feature-based real-time video stabilization for vehicle video recorder system. Multimedia Tools and Applications, vol. 77, no. 5, Mar. 1, 2018, pp. 5107–5127

[4]. F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. ACM Transactions on Graphics, vol. 30, issue 1, 2011, pp. 4:1-4:10

[5]. D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui. Mobile augmented reality survey: From where we are to where we go. IEEE Access, vol. 5, pp. 6917–6950, 2017

[6]. A. Bethencourt and L. Jaulin. 3d reconstruction using interval methods on the kinect device coupled with an IMU. International Journal of Advanced Robotic Systems, vol. 10, no. 2, 2013

[7]. J. Rambach, A. Pagani, S. Lampe, R. Reiser, M. Pancholi, and D. Stricker. Fusion of unsynchronized optical tracker and inertial sensor in EKF framework for in-car augmented reality delay reduction. In Proc. of the 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Oct. 2017, pp. 109–114

[8]. A. Karpenko, D. Jacobs, and J. Baek. Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. Stanford Tech Report CTSR 2011-03, 2011, 15 p.

[9]. S. Bell, A. Troccoli, and K. Pulli. A Non-Linear Filter for Gyroscope-Based Video Stabilization. Lecture Notes in Computer Science, vol 8692, 2014, pp. 294-308

[10]. Kornilova A.V., Kirilenko I.A., Zabelina N.I. Real-time digital video stabilization using MEMS-sensors. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 4, 2017, pp. 73-86. DOI: 10.15514/ISPRAS-2017-29(4)-5

[11]. R. Szeliski. Computer Vision: Algorithms and Applications. Springer-Verlag London, 2010, 812 p.

[12]. J. Diebel. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Matrix, 58, 2006, pp. 1-35

[13]. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In Proc. of the 2011 International Conference on Computer Vision, Nov. 2011, pp. 2564–2571

[14]. D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110

[15]. E. Karami, S. Prasad, and M. S. Shehata. Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images. arXiv:1710.02726 [cs.CV], 2017

Автоматическая калибровка и синхронизация камеры и МЭМС-датчиков

А.Р. Поляков<polyakov.alx@gmail.com>
А.В. Корнилова <kornilova.anastasiia@gmail.com>
Я.А. Кириленко <y.kirilenko@spbu.ru>
Санкт-Петербургский Государственный Университет,
Кафедра Системного программирования
198504, г. Санкт-Петербург, г. Петергоф, Университетский проспект, 28

Аннотация. Данная статья описывает текущие исследования по теме автоматической калибровки и синхронизации камеры и МЭМС-датчиков. Результаты исследования применимы к любой системе, имеющей камеру и МЭМС-датчики, примером которых является гироскоп. Основная задача нашего исследования – нахождение таких параметров системы камера-датчики, как фокусное расстояние камеры и разница во времени между считыванием показания датчика и считыванием кадра камеры, вызванная необходимостью предобработки "сырого" кадра и переводом его в определенный формат. Автоматическая калибровка позволяет применять алгоритмы компьютерного зрения (цифровая видео стабилизация, 3D-реконструкция, сжатие видео, дополненная реальность), использующие кадры видео и показания датчиков, на большем количестве устройств, оснащенными камерой и МЭМС-датчиками. Также автоматическая калибровка позволяет полностью абстрагироваться от характеристик конкретного устройства и разрабатывать алгоритмы, работающие на различных платформах (мобильные платформы, встраиваемые системы, экшн-камеры). Статья описывает общую математическую модель, необходимую для реализации данной функциональности, используя методы компьютерного зрения и показания МЭМС-датчиков. Авторы проводят обзор и сравнение существующих подходов к автоматической калибровке, а также предлагают свои улучшения, повышающие качество существующих алгоритмов.

Ключевые слова: калибровка камеры; автоматическая калибровка; обработка цифровых сигналов; компьютерное зрение

Список литературы

[1]. S. Liu, M. Li, S. Zhu, and B. Zeng. Codingflow: Enable video coding for video stabilization. IEEE Transactions on Image Processing, vol. 26, Issue 7, Apr. 2017, pp. 3291-3302

[2]. M. Grundmann, V. Kwatra, and I. Essa. Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. In Proc. of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), 2011, pp. 225-232

[3]. W.-C. Hu, C.-H. Chen, Y.-J. Su, and T.-H. Chang. Feature-based real-time video stabilization for vehicle video recorder system. Multimedia Tools and Applications, vol. 77, no. 5, Mar. 1, 2018, pp. 5107–5127

[4]. F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. ACM Transactions on Graphics, vol. 30, issue 1, 2011, pp. 4:1-4:10

[5]. D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui. Mobile augmented reality survey: From where we are to where we go. IEEE Access, vol. 5, pp. 6917–6950, 2017

[6]. A. Bethencourt and L. Jaulin. 3d reconstruction using interval methods on the kinect device coupled with an IMU. International Journal of Advanced Robotic Systems, vol. 10, no. 2, 2013

[7]. J. Rambach, A. Pagani, S. Lampe, R. Reiser, M. Pancholi, and D. Stricker. Fusion of unsynchronized optical tracker and inertial sensor in EKF framework for in-car augmented reality delay reduction. In Proc. of the 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Oct. 2017, pp. 109–114

[8]. A. Karpenko, D. Jacobs, and J. Baek. Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. Stanford Tech Report CTSR 2011-03, 2011, 15 p.

[9]. S. Bell, A. Troccoli, and K. Pulli. A Non-Linear Filter for Gyroscope-Based Video Stabilization. Lecture Notes in Computer Science, vol 8692, 2014, pp. 294-308

[10]. Kornilova A.V., Kirilenko I.A., Zabelina N.I. Real-time digital video stabilization using MEMS-sensors. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 4, 2017, pp. 73-86. DOI: 10.15514/ISPRAS-2017-29(4)-5

[11]. R. Szeliski. Computer Vision: Algorithms and Applications. Springer-Verlag London, 2010, 812 p.

[12]. J. Diebel. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Matrix, 58, 2006, pp. 1-35

[13]. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In Proc. of the 2011 International Conference on Computer Vision, Nov. 2011, pp. 2564–2571

[14]. D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110

[15]. E. Karami, S. Prasad, and M. S. Shehata. Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images. arXiv:1710.02726 [cs.CV], 2017