Robust Motion Estimation for Camcorders Mounted in Mobile Platforms

Jinhai Cai
ARCAA, School of Engineering Systems
Cooperative Research Centre for Spatial Information
Queensland University of Technology
Brisbane, QLD 4001, Australia
j.cai@ieee.org or j.cai@qut.edu.au

Rodney Walker
ARCAA, School of Engineering Systems
Queensland University of Technology
Brisbane, QLD 4001, Australia
ra.walker@qut.edu.au

Abstract

In this paper, we propose a novel camera motion estimation algorithm for mobile platforms. Videos captured by cameras mounted in mobile platforms suffer from jitter motions caused by various factors. It is vital to obtain accurate estimates of camera motions in order to remove these undesirable jitters. But existing motion estimation methods for mobile platforms have difficulties to deal with the interference of moving objects and estimation errors. We propose to estimate camera motions from histograms of local motions as the position of the highest peak in each motion histogram is insensitive to estimation errors and robust to the interference of moving objects. We also propose to use sorted arrays to implement histograms due to its advantages. Experiments show our proposed methods can achieve promising results.

1 Introduction

In recent years, video processing and computer vision have become increasingly important in many applications such as surveillance systems, unmanned aerial vehicle (UAV) systems, and personal video systems. This is partially due to the dramatic cost reduction of digit cameras and computers [1], and partially due to the recent advancement of computer vision and image processing technology. However, it is inevitable that there are some unwanted motion effects in videos taken by hand or from mobile platforms. The unwanted motions in videos are usually caused by shakes of cameras held by hands or mounted on cars, and caused by air turbulence, which deviate cameras mounted on UAVs away from their normal paths. Therefore it is essential to have a video stabilization algorithm to remove these undesirable motion effects thus to obtain good quality videos.

Usually, video stabilization systems can be divided into three categories. The first is the class of using hardware motion sensors [2] or mechanical devices such as accelerometers, gyros, and mechanical dampers [3] to reduce platform vibrations. The second is the class of object tracking based approaches [4, 5]. The third is the class of ego-motion estimation based approaches [6, 7]. As we focus on software solution, we will not discuss the first class approach in this paper. In the second class, a video stabilization system defines a target to track and the target could be a vehicle, a license plate, a person, lane markings, or a road sign. It can be used in surveillance and live television programs. In the third class, a video stabilization system usually comprises three components: motion estimation, motion filtering, and motion compensation [7].

In this paper, we will focus on the video stabilization based on ego-motion estimation. We will propose a novel algorithm for camera motion estimation based on motion histograms, as the positions of histogram peaks of are robust to moving objects.

2 Background

It is well known that human visual sensing is comfortable to smoothing motions but suffers significantly from high-frequency motions, known as jitters. Therefore, it is important to remove the high-frequency motions from videos to satisfy the human eye. In the past decade, many video stabilization algorithms have been developed for various applications.

If a camera is installed in a fixed location such as on a wall or a pole, the camera motion is usually caused by strong winds or under heavy traffic. In this case, the camera motion can be estimated by object tracking or feature tracking [4, 8], by finding a region with small accumulated motion [6] as the background is almost stationary in long terms. In order to reduce the computational cost, Marcenaro et al. [4] use a fixed set of points on a grid that is superimposed on an image. The method evaluates the motion transformation to be applied to the image by minimizing the

978-0-7695-3456-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DICTA.2008.48

mean square error between the corresponding pixels in the images of a sequence. The global motion is estimated by averaging the motions of points excluding outliers.

If the camera is mounted on a mobile platform, the motion of the camera contains two components: the intended motion and the undesired motion. In this case, some researchers take the advantages of the a-priori knowledge of application domains. In [5], a video stabilization algorithm was developed for a camcorder mounted on a moving vehicle by following the lane on the road. Therefore the algorithm is robust to moving objects. But it is difficult to apply this algorithm to other applications. In general case, the global motion estimation is derived from local motion vectors (LMVs) that are obtained from phase correction [9] or optical flows using block matching and differential methods [1, 6, 7, 10, 11, 12]. However, it is not easy to derive global motion from LMVs as there are four components in LMVs: the intended motion, motion jitter, motion of moving objects, and local motion estimation errors. Usually, the motion jitter and the motion estimation errors can be modelled as a random processing. However the interference from moving objects is difficult to remove as the impacts depends on many parameters such as moving objects' depths, speeds, and moving directions. In order to accurately estimate the global motions, many researchers use median filters or averaging LMVs after removing outliers [1] if a simple motion model such as translation is used.

Once we obtain camera motions, we can cancel the effect of unwanted motion from videos. In practice, a translational motion model is sufficient for most applications. In this case, low-pass filters and moving average filters are widely used to smooth camera motions. The smoothed camera motions are assumed to be the intended motions. The motion compensation is accomplished by shifting the position of the video frame with an amount equal with the difference between the accumulated intended camera motions and the accumulated gross motions. Usually the corrective displacement results in undefined regions or truncation of video size. Recently, [13] has proposed to use information from neighbouring frames to fill the undefined regions.

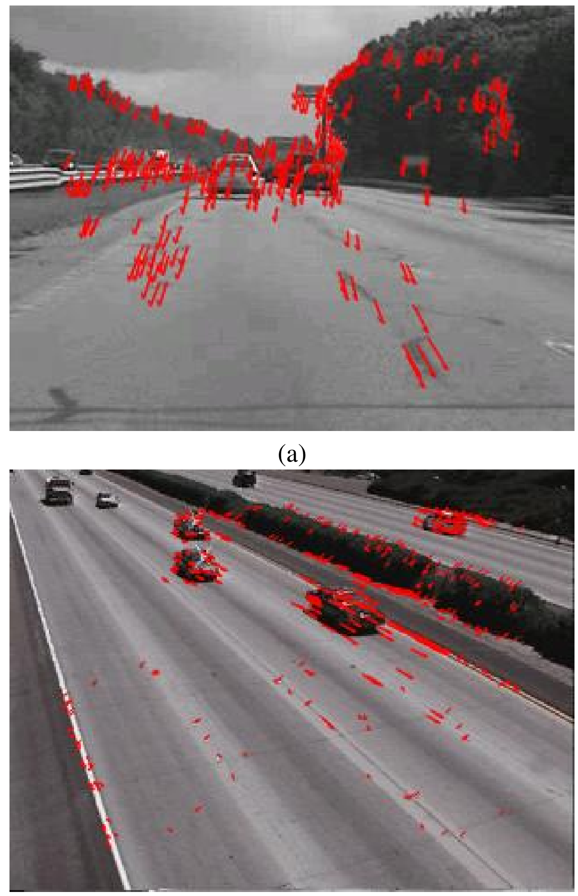
In this paper, we will focus on how to obtain accurate estimation of global motions.

3 Camera Motion Estimation using Local Motion Histograms for video stabilization

In this section, we will present a novel method to estimate camera motions using histograms.

3.1 Local motion estimation

The Lucas-Kanade method in pyramids [14, 15] is one of the most reliable and widely used techniques for estimating optical flow. We use the iterative Lucas-Kanade method in pyramids [16] for optical flow estimation. Figure 1 shows two examples of calculating optical flow using method in [16]. Figure 1 clearly demonstrates that the local motion estimation is affected by depth and moving objects. Figure 1(a) also shows that is not easy to remove outliers as their values are not far away from the true camera motions.



(a)

(b)

Figure 1. Optical flow estimation using Lucas-Kanade method in pyramids.

3.2 Camera motion estimation using histograms of local motions

The histogram of local motions at a given frame contains information about the camera motion, the jitter and moving objects. Usually, there are several peaks in one histogram of local motions. One peak is produced by the camera motion (global motion) and some peaks are caused by moving objects and estimation errors. It is also expected that the highest peak in a histogram is produced by the camera motion as moving objects usually occupy less than half of the scene.

In order to understand the impacts of moving objects to the estimation of camera motions, let's conduct the theory analysis first. Let $LMV_i(t)$ for $i = 1, 2, \cdots, N$ denote the local motion vector at the ith location in the tth frame. Each $LMV_i(t)$ can be expressed as follows:

$$LMV_i(t) = IC_i(t) + J_i(t) + M_i(t) + E_i(t), \quad 1 \leq i \leq N, \tag{1}$$

where N is the total number of local motion vectors, $IC_i(t)$ is the motion caused by the intended camera motion, $J_i(t)$ is the jitter, $M_i(t)$ is the motion of a moving object, and $E_i(t)$ is the estimation error. In some methods [6], scenes are assumed to be close to planar and thus $IC_i(t)$ and $J_i(t)$ are constants in a given frame. In this paper, we assume that the probability distribution function of $IC_i(t) + J_i(t)$ with their estimation error is Gaussian, $G(m, \sigma)$.

Now, we will use Gaussian distribution assumption to demonstrate the effectiveness of the proposed method. Let us assume that the distribution function of $IC_i(t) + J_i(t)$ with estimation errors at Y direction is $G(1, 2)$ (in solid line) and the distribution function of $M_i(t)$ with its estimation errors is $G(5, 4)$ (in dash line) as shown in Figure 2. The overall distribution function or the histogram (in dot line) is also shown in Figure 2. Clearly the effect of moving objects is moving the peak in the histogram from 1 to 1.29. However, if we used the averaging method, the estimated global camera motion would be 3 instead of 1.29 and the error of the global camera motion estimation would be 2 instead of 0.29. This example has clearly shown that the position of the highest peak is robust to imapcts of moving objects and estimation errors, thus the proposed method is relatively robust to imapcts of moving objects and estimation errors.

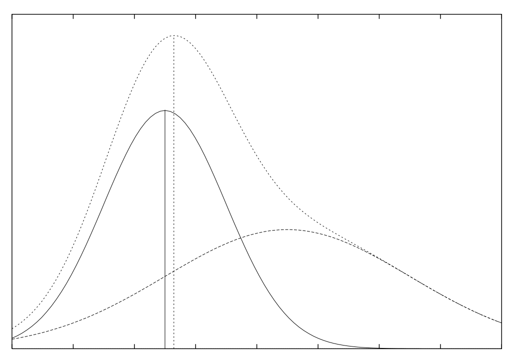


Figure 2. The effect of moving objects on the global camera motion estimation.

In reality, however, local motion histograms are not so smoothing and there are more than one peak in each histogram. Figure 3 shows a typical example of local motion histograms. In order to obtain an accurate estimate of the global camera motion at a given frame, it is necessary to smoothen the histogram and then to pick up the peak with the maximum value from the histogram. Another issue is how to determine the suitable number of bins of a histogram at a given frame. Outliers can greatly enlarge the range of a histogram and thus the range of each bin. If the selected number of bins is too large, there are too few samples in each bin and the statistics is not reliable, and therefore the camera motion estimation is not reliable. If the selected number of bins is too small, the range of each bin is too large and the accuracy of camera motion estimation will be reduced significantly.

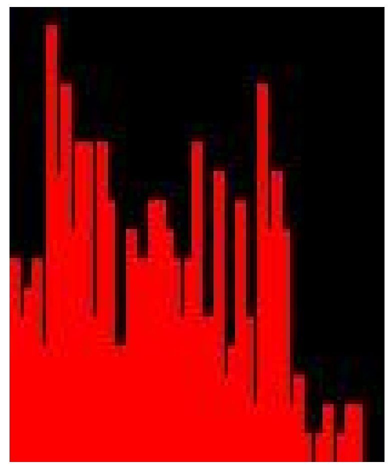


Figure 3. An example of local motion histograms.

3.3 Histograms implemented using sorted arrays

In order to avoid the selection of bin number for the histogram-based method, we use sorted arrays instead of directly using histograms to estimate the camera motion for each frame. Let us take the motion at the vertical direction as an example. For a given set of local motions at the vertical direction, we can get a sorted array for vertical local motions $\mathbf{SA^y}$:

$$\mathbf{SA^y}(t) = \{LMV_k^y(t)\}, \quad 1 \leq k \leq N, \tag{2}$$

where

$$LMV_k^y(t) \leq LMV_{k+\delta}^y(t). \quad \delta > 0 \;\&\; k + \delta \leq N. \tag{3}$$

If there are enough samples, the average difference between two local motions at the vertical direction is in proportionate to the inverse function of the histogram, that is:

$$LMV^y_{k+\delta}(t) - LMV^y_k(t) \propto \frac{1}{hist^y(t+\frac{\delta}{2})}, \quad (4)$$

where $hist^y(t)$ is the histogram of local motions at the vertical direction. According to eqn(4), we find that to obtain the camera motion at the vertical direction is equivalent to find the position index k to the sorted array which minimises the difference between two local motions, $LMV^y_{k+\delta}(t)$ and $LMV^y_k(t)$. So we have

$$p(t) = \arg\min_k \{LMV^y_{k+\delta}(t) - LMV^y_k(t)\}, \quad (5)$$

where $p(t)$ is the position in the sorted array that we have the minimum difference of two local motions between $p(t)$ and $p(t)+\delta$. We take the middle local motion as the camera motion:

$$CM^y(t) = LMV^y_{(p(t)+0.5\delta)}(t), \quad (6)$$

where $CM^y(t)$ is the estimated camera motion at the vertical direction at tth frame. Note, the estimated motions consist of intended camera motions, jitters, and partial estimation errors.

In this way, we can avoid to select the number of bins for histograms. Now, we need to set the parameter δ. We find that the performance of camera motion estimation is not sensitive to δ if it is in the range of 5% to 20% of the array size (the total number of local motions). We choose 10% of the array size in our experiments.

4 Motion smoothing and video stabilization

After we obtain camera motions, we need to smooth the motions to remove jitters. The most widely used approach for smoothing is to apply a low-pass filter on locations (accumulated camera motions). However, there is a practical problem: accumulated errors. If there is a significant estimation error occurred in one frame, the error will after all succeeding frames. An application system must deal with this problem. The final step of video stabilization is motion compensation.

4.1 Smoothing

4.1.1 Smoothing using a first-order IIR filter

Smoothing is an effective way to remove jitters and estimation errors as they are high-frequency components in the camera motions $\mathbf{CM}(t)$ or in the accumulated motions. In this paper, a first-order IIR filter is used to smooth accumulated motions:

$$\mathbf{SAM}(t) = \alpha\mathbf{SAM}(t-1) + (1-\alpha)\mathbf{ACM}(t), \quad (7)$$

and

$$\mathbf{ACM}(t) = \sum_{i=1}^{t} \mathbf{CM}(t), \quad (8)$$

where $\mathbf{SAM}(t) = \{SAM^x(t), SAM^y(t)\}$ is the smoothed accumulated motion vector, $\mathbf{ACM}(t) = \{ACM^x(t), ACM^y(t)\}$ is the accumulated camera motion vector and $\mathbf{CM}(t) = \{CM^x(t), CM^y(t)\}$ is the camera motion vector at tth frame, and α is the smoothing parameter. The reasons of using this first-order IIR filter are: (1) it can be used in real-time systems; (2) it requires little memory; (3) it involves little computations; (4) the smoothed motions produced by the filter are satisfactory to human's eyes if a suitable value is selected for α. According to our experiments, the suitable range of α is $[0.95,\ 0.98]$.

4.1.2 Smoothing using a non-causal low-pass filter

In some applications, videos are processed offline. So we can use non-causal low-pass filters to smoothen camera motions or accumulated motions. In this paper, a non-causal low-pass filter is used to smooth accumulated motions:

$$\mathbf{SAM}(t) = \sum_{i=-L}^{L} f(i) * \mathbf{ACM}(t-i), \quad (9)$$

where $f(i)$ is the non-causal low-pass filter and $L = 20$ frames.

4.2 Motion Compensation

In order to stabilize the video, we need to compensate the tth frame using offset. The offset is the accumulated jitter or unwanted motion from the beginning to the tth frame:

$$\begin{aligned}\mathbf{AJ}(t) &= \sum_{i=1}^{t} \mathbf{CM}(i) - \mathbf{SM}(i) \\ &= \mathbf{AJ}(t-1) + \mathbf{ACM}(t) - \mathbf{SAM}(t), \quad (10)\end{aligned}$$

where $\mathbf{AJ}(t)$ is the accumulated jitter till the tth frame.

As indicated in eqn(1), there are errors in estimated camera motions. Although the estimation errors are small in most cases, the accumulated error can be significant if the image sequence of the video is long. If there is a significant error in the estimated camera motion at one frame, the error will be in the accumulated jitter of all succeeding frames. The simple solution to the problem is:

$$\mathbf{AJ}(t) = \beta\mathbf{AJ}(t-1) + \mathbf{CM}(t) - \mathbf{SM}(t), \quad (11)$$

where β is the parameter to control the influence of estimation errors. When β is small, the influence of estimation errors in the current frame to the succeeding frames is very limited, but the remaining jitter motions are not trivial. If β is close to 1.0, the estimation errors in the current frame will after many succeeding frames. Usually, we choose β between 0.90 and 0.98.

5 Experimental Results

We evaluated the proposed algorithm using various videos taken by cameras in UAVs, cars and held by hands, as shown in Figure 4. In each of these videos, there are jitters caused by car vibration, air turbulence and shaky hands. Figure 5 gives the comparison between original motions and smoothed motions, where the smoothing filter is the first order IIR filter. This figure clearly shows that high-frequency components in the original camera motions have been removed in the smoothed camera motions. Our experiments also demonstrated that human eyes have better visual perception to the stabilized videos than the original videos.

It is interesting to compare the performance of using a first-order IIR filter to using a non-causal low-pass filter for smoothing. We find that both filters can produce smoothed motions as shown in Figure 5(a) and Figure 6. However, the smoothed motions produced by the non-causal low-pass filter fit the original motions better and therefore there is smaller displacement in the stabilized videos. Thus, we recommend that an offline system use non-causal low-pass filter for motion smoothing.

It is also interesting to compare the performance of camera motion estimation using the proposed method to using the method of finding median values. We find that both methods produce very similar results in most cases. However, the method of finding median values may produce significant errors in camera motion estimation if there are moving objects in scenes, while the proposed method produces very accurate estimates of camera motions. Figure 7 shows such an example. This video clip was taken from a highway, therefore there are many moving motor vehicles on the road as shown in Figure 1(b). The video clip is almost stable except minor vibrations. As analysed in Section 3.2 and shown in Figure 2, moving objects can significantly degrade the performance of motion estimation using traditional methods such as averaging and finding median. This example demonstrates that the proposed method is robust to the influence of moving objects for camera motion estimation.

(a)

(b)

(c)

Figure 4. Images taken by a camera (a) in a moving vehicle, (b) in an UAV, (c) held by a hand.

6 Conclusion

In this paper, we presented a novel camera motion estimation algorithm using histograms of local motions for mobile platforms. As the highest peak in each histogram of local motions represents the camera motion and the peak position in the histogram of local motions is robust to the interference of moving objects and estimation errors. Therefore, it is very attractive to using local motion histograms for camera motion estimation. We also implemented the proposed algorithm using sorted arrays instead of directly using motion histograms, so that we can avoid selecting the number of bins for histograms, to which the performance is sensitive. Our theory analysis and experiments have proven that the proposed algorithm for camera motion estimation is robust to estimation errors and interference of moving objects. The proposed algorithm works well for both mobile and stabile platforms.

Acknowledgment

The authors would like to thank the Cooperative Research Centre for Spatial Information (CRC-SI), Australia, for supporting this research under the CRC-SI project 6.07.

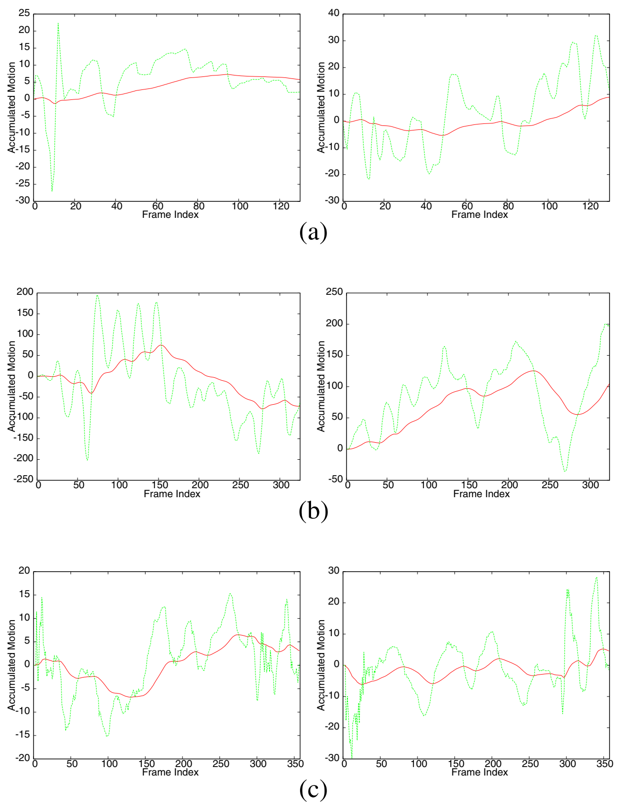


Figure 5. Image stabilization: Horizontal accumulated motions are in left and vertical accumulated motions are in right; original motions are in green-dash lines and smoothed motions are in red-solid lines. (a) in a moving vehicle, (b) in an UAV, (c) held by a hand.

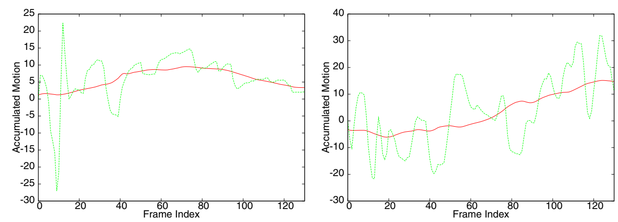


Figure 6. Motion smoothing using using a non-causal low-pass filter.

References

[1] Q. Luo and T.M. Khoshgoftaar, "An Empirical Study on Estimating Motions in Video Stabilization", Proc. of the IEEE International Conference on Information Reuse and Integration, pp.360-366, 2007.

[2] M. Oshima, T. Hayashi, S. Fujioka, T. Inaji, M. Mitani, J. Kajino, K. Ikeda, and K. Komoda, "VHS camcorder with electronic image stabilizer," IEEE Trans. on Consumer Electronics, Vol.35, No.4, pp.749-758, 1989.

[3] R. Kurazume and S. Hirose, "Development of image stabilization system for remote operation of walking robots," Proc. of International Conference on Robotics Automation, pp.1856-1861, 2000.

[4] L. Marcenaro, G. Vernazza, and C.S. Regazzoni, "Image Stabilization Algorithms for Video-Surveillance Applications", Proc. of International Conference on Image Processing, pp.I:349-352, 2001.

[5] Y. M. Liang, H. R. Tyan, S. L. Chang, H. Y. M. Liao, and S. W. Chen, "Video Stabilization for a Camcorder Mounted on a Moving Vehicle," IEEE Trans. on Vehicular Technology, Vol.53, No.6, pp.1636-1648, November 2004.

[6] A.U. Batur and B. Flinchbaugh, "Video Stabilization with Optimized Motion Estimation Resolution", Proc. of International Conference on Image Processing, pp.465-468, 2006

[7] M. Tico and M. Vehvilainen, "Constraint Motion Filtering for Video Stabilizationsing," Proc. of International Conference on Image Processing, pp.569-572, 2005.

[8] M. Ben-Ezra, S. Peleg, and M. Werman, "Real-Time Motion Analysis with Linear Programming," Proc. of International Conference on Computer Vision, pp.703-709, 1999.

[9] B. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," IEEE Transactions on Image Processing, Vol.5, No.8, pp.1266-1271, 1996.

[10] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by features tracking," Proc. of International Conference on Image Analysis and Processing, pp.665-667, 1999.

[11] J.S. Jin, Z. Zhu, and G. Xu, "A stable vision system for moving vehicles," IEEE Trans. Intelligent Transportation Systems, Vol.1, No.1, pp.32-39, March 2000.

[12] J. Cai, "Fast Stereo Matching: Coarser to Finer with Selective Updating," Proc. of International Conference on Image and Vision Computing New Zealand, pp.266-270, 2007.

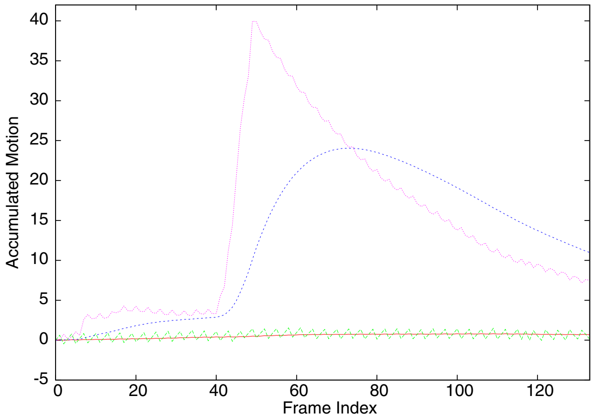


Figure 7. Comparison of motion estimation using the proposed method and using the method of finding median: Green-dash line is the estimated accumulated motions using the proposed method and red-solid line is its smoothed accumulated motions; Pink-dot line is the estimated accumulated motions using the method of finding median and the blue-dashdot line is its smoothed accumulated motions.

[13] Full-Frame Video Stabilization with Motion Inpainting Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-Frame Video Stabilization with Motion Inpainting," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.28, No.7, pp.1150-1163, 2006.

[14] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," Carnegie Mellon Univ., Tech. Report. CMU-CS-91-132, 1991.

[15] J. Shi and C. Tomasi, "Good Features to Track," Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pp.593-600, 1994.

[16] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker," with OpenCV distribution, 2000.