Video stabilization and motion detection using Matlab video processing toolbox.

Paterne Sissinto
Morgan State Univeristy
Email: sissinto@yahoo.com

Abstract

This paper presents video stabilization and motion detection using Matlab Simulink. Simulink is a classroom learning tool that offers an environment for multi-domain simulation and model-based design for dynamic and embedded systems. The project utilizes that tool to solve not only a problem common to amateur videos but also a situational awareness issue. The stabilization process starts with computing the optical flow between successive frames. The motion vectors are computed per block of 4×4 pixels for a good representation of the jerkiness present in the video and faster processing. From the optical flow, there is an affine representation of the unwanted motions using four parameters. The four parameters represent the camera motion by rotation matrix and translation vector. The smoothing process of the jerky video uses the inverse rotation matrix and oppose translation vector. The motion detection is done by frame difference block provided by Simulink. A tracking function allows the localization of the moving object. Simulink processing blocks represent the different steps taken in implementing the project. The video for complete testing allows detection only after stabilization.

I Introduction

The current work presents the implementation of a project assigned as part of a graduate course in video processing. The project was to develop a video stabilization and motion detection system that can later be loaded on a reconfigurable board. That requirement suggested simple and efficient algorithms and the use of software that support design, simulation, implementation, and testing using embedded systems. When a camera undergoes vibrations or unwanted motions as in the case of a camera mounted on a moving vehicle or a hand-held camera, it records a jerky video created by small, random movements. The poor quality of such recording may cause the video not to be useful (surveillance task or pattern recognition for example). Video applications by computers and other devices are based on reconfigurable boards with limited resources. In order to develop applications on mobile platforms simple and efficient algorithms must be developed. To this end, this work explores and utilizes Matlab Simulink and its processing toolboxes to get rid of the disturbing effects of camera motion and identify motion in footage.

II Objective

The first part of this work will develop and implement video stabilization based on motion estimation with four-parameter model. The implementation consists in correcting a jerky video stream recorded with an unstable camera and uses tools offered by Matlab Simulink. The second part will explore different techniques for motion detection with a stationary background, and identify the techniques with interesting results. Finally, Matlab video processing toolbox will

essentially be used to implement stabilization and motion detection algorithms applied to pre-captured footage.

III Approach:

A Background

Video stabilization is an important step for many video processing works. It is used in a range of fields including, surveillance, tracking, situational awareness and diverse hand-held video recording devices. Multiple techniques have been developed and used for stabilization. Yao and his colleagues used principal component analysis to estimate unwanted inter-frame motion[1]. The accuracy of their method comes with extensive computation. Another way of getting rid of video jerkiness is the Dynamic Time Warping to overcome shape changes caused by undesirable camera motion. Unfortunately this technique is computationally unacceptable for embedded applications.[2] Hany Farid and Jeffrey B. Woodward[3] worked on stabilization based on differential optical flow.[4] That method requires low computation, reduces significantly jerkiness, and therefore is implemented in this work.

Camera held in hands or mounted on moving platforms records scenes while they undergo some motions themselves. That physical instability introduces unwanted motions that may alter significantly processing results.[5] Matlab Simulink is equipped with processing blocks that performs basics functions. By combining the available blocks with customized functions one can develop algorithms and remove disturbing jerkiness and unwanted camera movements.

Different works on object detection with stable background in real time video led to many algorithms. The most commonly used are the Running Gaussian Average, the Mixture of Gaussians, the Kernel Density Estimation and the Eigen Background techniques. An exhaustive comparison of all the techniques is not provided here. Following is an attempt to summarize challenges on accuracy and speed encountered by the authors of the experiments on these methods.[6, 7, 8, 9]

Table 1: Accuracy and speed of some background subtraction techniques

Methods	Accuracy	Speed
Running Gaussian Average	Detection of a lot of noise (false alert problem), low performance	Fastest due to simplicity in operations
Mixture of Gaussians	Reduces the noise, good adaptation to illumination changes	Acceptable speed
Kernel Density Estimation	Powerful method	Needs extra algorithm for real time application
Eigen backgrounds	Application need to be in line with training set for good results	Depends on the number of best eigenvectors

The mixture of Gaussians technique present great advantages over many algorithms but the computational effort associated with its implementation disqualifies it for real time embedded applications. The Gaussian average is very effective in detecting changes that occur in pixel intensities.[10] Javier Tolelo and his co-authors developed an embedded system that performs video processing on a Field-Programmable Gate Array (FPGA) board.[11] They took advantage of parallel implementation, reconfiguration capabilities and the ability to expand reconfigurable platforms to create a dynamic applications that running fast algorithms. In this work, frame difference will be used to identify and track an object in motion. Matlab simulink presents the advantage of offering necessary blocks to build the model and experiment the system.

B Methodology

Simulink provides an interactive graphical environment and customizable set of block libraries in order to design not only signal, image, and video processing systems but also communication and control systems. Simulink let users design, simulate, implement, and test their projects.

The steps followed in this design consist of stabilizing a feed then motion detection and tracking on footage. Figure 1 presents the system model.

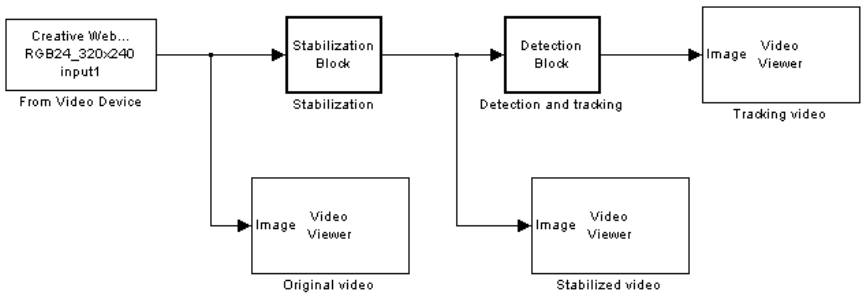

Figure 1: System Model

Simulink does not provide a built-in stabilization block; therefore a customized and advanced system is created in order to perform the operation of stabilization.

1) Video Stabilization algorithm

The goal of video stabilization is to remove from video sequences, disturbing jerkiness and unwanted camera movements. The steps followed in this work are the optical flow computation between adjacent frames then the fitting of the optical to a 4-parameter affine model and finally the smoothing process to reproduce the stabilized video. Figure 2 shows a representation of the stabilization process.

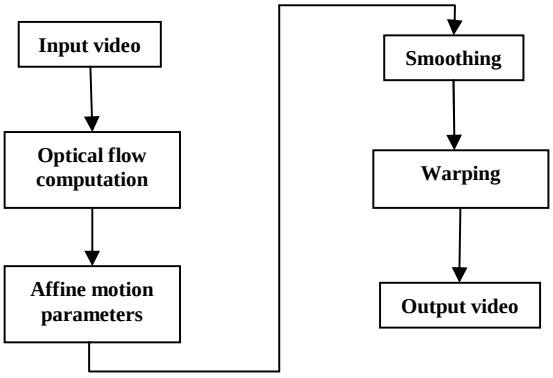

Figure 2: Video stabilization process

a) Optical flow

The optical flow of a video frame is a field of motion vector per pixel or sub-pixel. Multiple methods allow computing the optical flow among which partial differential equation based methods, gradient consistency based techniques and least squared methods. In this work, block-based representation is used. As done for class assignments, the motion vectors are computed per block of 4×4 pixels. Using small blocks has the advantage of representing the block by a unique motion vector. The assumption is that in small blocks the motion vectors have similar directions. Also the computation required is reduced to the fourth.

b) Camera Motion Estimation using Affine Model

The parametric model chose and used in this work is the affine representation. This model is chosen over bilinear and other polynomial models due to the fact that it can represent camera translation, rotation and scaling motion in a simple way using 4 parameters. The motion between two successive frames, *f(x,y,t-1)* and *f(x',y', t)* can be modeled as an affine mapping. The affine mapping function can be represented as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix}$$

where *a* and *b* control scaling and rotation; *c* and *d* control translation.

The error function used to estimate the affine parameter by minimization[3] is:

$$E(\vec{m}) \approx \sum_{x,y \in \Omega} \left[f(x,y,t) - f(ax - by + c, bx + ay + d) \right]^2$$

$$where \vec{m} = (a,b,c,d) \; and \; \Omega \; is \; the \; region \; concerned$$

Using a first order truncated Taylor series expansion, the minimization is simplified[3]:

$$E(\vec{m}) \approx \sum_{x,y \in \Omega} \left[f - \left(f + (ax - by + c - x) f_x - (bx + ay + d) f_y - f_t \right) \right]^2 = \sum_{x,y \in \Omega} \left[k - (\vec{c})^T \vec{m} \right]^2$$

$$k = f_t + x f_x + y f_y$$

$$\vec{c}^T = \left(x f_x \; y f_x \; x f_y \; y f_y \; f_x \; f_y \right)$$

Solving for the parameter values[3] yield:

$$\vec{m} = \left[\sum_{\Omega} \vec{c}\vec{c}^{T} \right]^{-1} \left[\sum_{\Omega} \vec{c}k \right]$$

The spatial/temporal derivatives are computed as[3]:

$$f_x(x,y,t) = \left(0.5 f(x,y,t) + 0.5 f(x,y,t-1) * d(x) * p(y)\right)$$

$$f_y(x,y,t) = \left(0.5 f(x,y,t) + 0.5 f(x,y,t-1) * d(x) * p(y)\right)$$

$$f_t(x,y,t) = \left(0.5 f(x,y,t) - 0.5 f(x,y,t-1) * d(x) * p(y)\right)$$

where * represents the convolution operator and d(.) and p(.) are 1-dimension separable filters: d(x) = (0.5 -0.5) and p(x) = (0.5 0.5); d(y) and p(y) are oriented vertically instead of horizontally.

c) Smoothing process

Smoothing relies on the motion representation parameters. In a process similar to the next-frame prediction technique used in motion estimation, the original frame can be warped with the rotation matrix and the translation vector in an inverse process. An L-level Gaussian can be used just like in the motion estimation process to obtain the output frame.

2) Motion Detection Algorithm

For motion detection, a simple frame difference technique is used since Simulink offers an absolute-difference block. The sub-processing block can be upgraded to any algorithm.

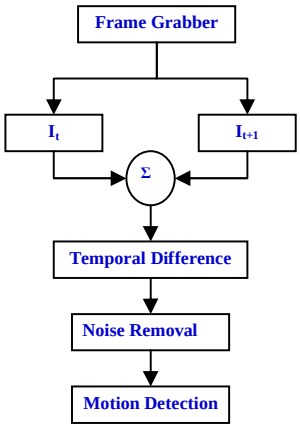

Figure 3: Motion detection using frame difference

For noise removal a Gaussian filter can be used but one needs to be cautious about blurring.

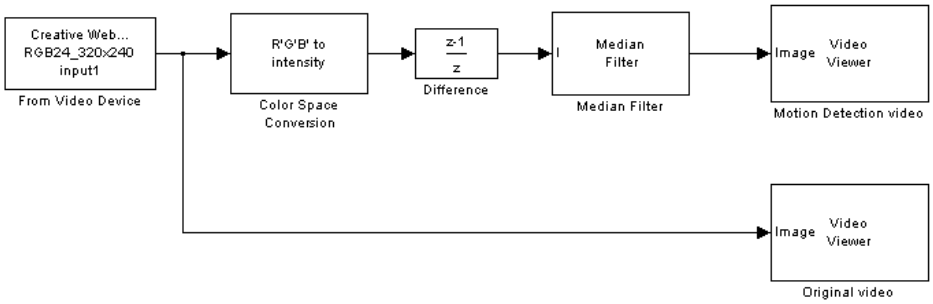

Figure 4: Motion detection using Simulink.

The web cam used for testing delivers a RGB output that is converted into single value intensity. The frame difference is done between current frame and one time-step-delayed frame. The rate of the video being 30 frames per second the time step is 1/30 of a second. Tracking is implemented following the model proposed by Matlab Simulink in Figure 5.

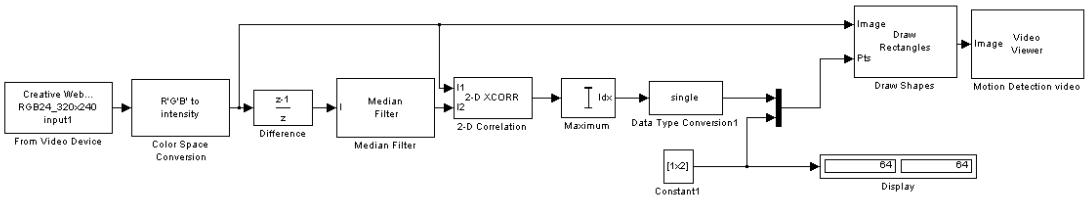

Figure 5: Tracking Model (From Simulink)

VI Results and Analysis

In order to realize video stabilization, the inter-frame motion was computing using 4x4-block motion vectors. The motion of the camera is then fitted into a 4-parameter affine representation. One can to use more than 4 parameters. More parameters provide better accuracy but require more computation. Testing is going on and not completed yet. Primary testing has been done for frame difference using Simulink.

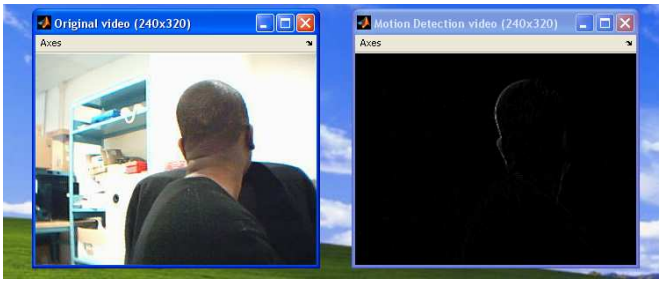

Figure 6: Original and motion detection frames

In figure 6 the subject makes a slight motion. Tests have showed that filtering is needed after motion detection to reduce blurring caused by motion. After testing the median filter was

show because it provided neat detection not only in presence of important motions but also in presence of slight motions. Detected motion triggers motion tracking in the output video.

V Conclusion

The use of a model-based design and customizable set of block libraries instilled in students the desire to model various projects in speech processing and wireless communications. The design environment has the advantage of clearly showing all the algorithms involved in a project as well as their interconnections. The development of the stabilization and detection algorithm for Simulink completed. The testing of the stabilization process needs to be completed in order to get results and analyze the selected algorithm. Interesting for this project is the video recorded for final testing has moving objects that can be detected only after stabilization.

References

1. Yao Shen, Parthasarathy Guturu, Thyagaraju Damarla, Bill P. Buckles, Senior ,and Kameswara Rao Namuduri, "Video Stabilization Using Principal Component Analysis and Scale Invariant Feature Transform in Particle Filter Framework", IEEE Transactions on Consumer Electronics, Vol. 55, No. 3, AUGUST 2009

2. Angelo Bosco, Arcangelo Bruna, Sebastiano Battiato, Giuseppe Bella, and Giovanni Puglisi "Digital Video Stabilization through Curve Warping Techniques" IEEE Transactions on Consumer Electronics, Vol. 54, No. 2, MAY 2008

3. Hany Farid and Jeffrey B. Woodward, "Video Stabilization and Enhancement" TR2007-605, Dartmouth College, Computer Science

4. J.L. Barron D.J. Fleet S.S. Beauchemin, T.A. Burkitt "Performance of Optical Flow Techniques" Multimedia and Expo, 2006 IEEE International Conference on 9-12 July 2006 Page(s):241 – 244

5. Jen. Hsiao, C. Hsu, T. Shih, P. Hsu, S. Yeh and B. Wang "Real-time video stabilization for the Rescue Robot" ICROS-SICE International Joint Conference 2009.

6. C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real Time Tracking" Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no 8, pp. 747-757, August 2000.

7. A. Elgammal, D. Harwood, and L.S. Davis, "Nonparametric model for background subtraction," Proc. ECCV 2000, pp. 751-767, June 2000.

8. Massimo Piccardi, "Background Subtraction Techniques: a review," IEEE International Conference on Systems, Man and Cybernetics, pp. 3099-3104 2004.

9. A. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no.11, pp.1499-1504, November 2003.

10. Eduardo Monari, Charlotte Pasqual, "Fusion of Background Estimation Approaches for Motion Detection in Non-Static Background", IEEE Advanced Video and Signal Based Surveillance, pp.347-352, 2007

11. F. Janvier Toledo, J. Javier Martinez and J. Manuel Ferrandez, “FPGA-based platform for image and video processing embedded systems” 2007 IEEE.