A NOVEL BLOCK MOTION ESTIMATION MODEL FOR VIDEO STABILIZATION APPLICATIONS

Harish Bhaskar and Helmut Bez
Research School of Informatics, Loughborough University

Keywords: Video stabilization, motion compensation, motion estimation, genetic algorithms, kalman filtering.

Abstract: Video stabilization algorithms primarily aim at generating stabilized image sequences by removing unwanted shake due to small camera movements. It is important to perform video stabilization in order to assure more effective high level video analysis. In this paper, we propose novel motion correction schemes based on probabilistic filters in the context of block matching motion estimation for efficient video stabilization. We present a detailed overview of the model and compare our model against other block matching schemes on several real-time and synthetic data sets.

1 INTRODUCTION

Video data obtained from compact motion capture devices such as hand-held, head mounted cameras, etc. has gained significant attention in recent years. Video stabilization, as the name suggests, deals with generating stabilized video sequences by removing unwanted shakes and camera motion. Several methods have been proposed in the literature for accomplishing video stabilization. However, the accuracy of motion estimation is a key to the performance of video stabilization. (Y.Matsushita and H.Y.Shum, 2005) propose a combination of motion inpainting and deblurring techniques to accomplish robust video stabilization. Several other research contributions have been made to video stabilization including, probabilistic methods (A.Litvin and W.C.Karl, 2003), model based methods, etc. Methods such as (M.Hansen and K.Dana, 1994)(Y.Yao and R.Chellappa, 1995)(P.Pochec, 1995)(J.Tucker and Lazaro, 1993)(K.Uomori and Y.Kitamura, 1990), propose to combine global motion estimation with filtering to remove motion artifacts from video sequences. These schemes perform efficiently only under restricted conditions and are again limited by the efficiency of the global motion estimation methodology. (K.Ratakonda, 1998) have used an integral matching mechanism for compensating movement between frames. (T.Chen, 2000) propose a 3 stage video stabilization algorithm based on motion estimation. The process includes motion estimation for computing local and global motion parameters, motion smoothing for removing abrupt motion changes between subsequent frame pairs and finally a motion correction methodology for stabilization. In this paper we extend the work presented in (T.Chen, 2000) to accommodate a novel motion correction mechanism based on moving average filters and Kalman filtering alongside a motion estimation strategy that combines vector quantization based block partitioning with a genetic algorithm based block search for motion estimation.

2 PROPOSED MODEL

The video stabilization model proposed in this paper extends a parametric motion model proposed in (T.Chen, 2000). A detailed overview of the proposed model in the form of a pseudo code is as follows.

- Input at a time instant t two successive frame pairs of a video sequence, f_t & f_{t+1} where $1 \leq t \leq N$, where N is total number of frames in the video
- Image frame f_t is initially partitioned into 4 blocks using the vector quantization algorithm

Bhaskar H. and Bez H. (2007).
A NOVEL BLOCK MOTION ESTIMATION MODEL FOR VIDEO STABILIZATION APPLICATIONS.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 303-306
DOI: 10.5220/0001650803030306

described in the subsection below, **Note:** Every block represents an image region

- For every block b
  - The centroid (x_c, y_c) of the block is computed
  - A genetic algorithm as described below is used to accurately match the block in the successive frame f_{t+1}
  - If the genetic algorithm accurately matched the block in frame f_t to frame f_{t+1} (with error = 0), then the motion vector is evaluated as $(x^* - x, y^* - y)$ where (x^*, y^*) is the estimated transformed centroid of the block in frame f_{t+1}
  - If the genetic algorithm returned non-zero matching error then the process is repeated by further sub dividing block.
- The process is terminated either when no further splitting is needed or a predefined block size is reached.
- If the processed frame pair is (f_t, f_{t+1}) where $t = 1$, then proceed to next frame pair, otherwise if $t > 1$, then run motion correction using any of the proposed filter mechanisms specified to generate smoothed motion vectors $MV_{\aleph}$
- Compute the difference between the original motion vectors MV and the smoothed motion vectors $MV_{\aleph}$ adjust the original motion vectors by the factor of difference $MV_{comp} = MV \pm (MV - MV_{\aleph})$
- Generate Stabilized frames using the original motion vector MV and compensated motion vectors MV_{comp} and represent them as f^*_{t+1} and f^{*comp}_{t+1}
- Deduce the PSNR of the two versions of stabilized frames using, PSNR for a gray scale image is defined as:

  $$10\ log_{10}\left[\frac{255^2}{\frac{1}{HW}\sum_H \sum_W \|f_{t+1} - f_{comp}\|^2}\right] \tag{1}$$

  where, (H, W) is the dimensionality of the frames and f_{t+1} and $f_c omp$ are the intensity components of the original target and the motion compensated images which will equal f^*_{t+1} and f^{*comp}_{t+1} respectively. PSNR values generally range between 20dB and 40dB; higher values of PSNR indicate better quality of motion estimation.
- If $PSNR_{comp} \geq PSNR$ then use f^{*comp}_{t+1} as stabilized frame for subsequent analysis otherwise use f^*_{t+1}.

2.1 Motion Estimation

A brief description of the algorithms is specified.

2.1.1 Block Partitioning Based on Vector Quantization

For the block partitioning phase, we start by using vector quantization to provide the block matching scheme with the position of partitioning.

- Set the number of codewords, or size of the codebook to 4. This assumes that we need 4 regions to emerge out of the image frame during the quantization process.
- Initialize the positions of the codewords to $(\frac{w}{4}, \frac{h}{4})$, $(\frac{w}{4}, \frac{3h}{4})$, $(\frac{3w}{4}, \frac{3h}{4})$, $(\frac{3w}{4}, \frac{3h}{4})$ where (w, h) is the width and height of the block respectively. By this we assume that the worst case partition could be the quad-tree partition.
- Determine the distance of every pixel from the codewords using a specific criterion. The distance measure is the sum of differences in the gray intensity and the locations of the pixels.
- Group pixels that have the least distance to their respective codewords.
- Iterate the process again by recomputing the codeword as the average of each codeword group (class). If m is the number of vectors in each class then,

  $$CW = \frac{1}{m}\sum_{j=1}^{m} x_j \tag{2}$$

- Repeat until either the codewords don't change or the change in the codewords is small
- Associated with these 4 codewords, there are 4 configurations possible for partitioning the image frame into blocks. The configurations arise if we assume one square block per configuration. It is logical thereafter to find the best configuration as the center of mass of these 4 possible configurations. The center of mass will now be the partition that splits the image frame into blocks.

2.1.2 Genetic Algorithm Search

The inputs to the genetic algorithm are the block b_t and the centroid (x_c, y_c) of the block.

- Population Initialization: A population P of these n chromosomes representing (T_x, T_y, θ) is generated from uniformly distributed random numbers where,
  - $1 \leq n \leq limit$ and $limit$ (100) is the maximum size of the population that is user defined.
- To evaluate the fitness $E(n)$ for every chromosome n:

  - Extract the pixels locations corresponding to the block from frame f_t using the centroid (x_c, y_c) and block size information
  - Affine Transforming these pixels using the translation parameters (T_x, T_y) and rotation angle θ using,
    $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
  - If b_t represents the original block under consideration, b^*_{t+1} represents the block identified at the destination frame after transformation and (h, w) the dimensions of the block, then the fitness E can be measured as the mean absolute difference (MAD).

$$MAD = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \left| b_t(i,j) - b^*_{t+1}(i,j) \right| \tag{3}$$

- Optimization: Determine the chromosome with minimum error $n_{emin} = n$ where E is minimum. As this represents a pixel in the block, determine all the neighbors (NH_k) of the pixel, where $1 \leq k \leq 8$.
  - For all k, determine the error of matching as in Fitness evaluation.
  - If $E(NH_k) < E(n_{emin})$, then $n_{emin} = NH_k$
- Selection: Define selection probabilities to select chromosomes for mutation or cloning. Perform cross-over and mutation operations by swapping random genes and using uniform random values.
- Termination: Three termination criterion such as zero error, maximum generations and stall generations. Check if any condition is satisfied, otherwise iterate until termination.

2.2 Motion Smoothing

The work of (T.Chen, 2000) suggested the use of a moving average low pass filter for this process. In this paper, we extend the moving average filter to an exponentially weighted moving average filter.

2.2.1 Exponentially Weighted Moving Average Filter

A detailed pseudo code describing the process is as follows.

- Set the number of frame pairs across which the moving average filter to be any scalar J
- Compute the parameter alpha $\propto$ = $(1 \div J)$
- Compute the weighting factors for every frame pair between 1 and J as w $=\propto^{i-1}$ $\times(1-$ $\propto)$, where, $1 \leq i \leq J$ (Use these weighting factors as a kernel for the convolution process)
- Generate a vector of the motion vectors and rotation parameter theta across all frames; MV and θ
- Perform Convolution to generate the smoothed motion vectors, $MV_{\aleph} = MV \otimes w$ and $\theta_{\aleph} = \theta \otimes w$

2.2.2 Kalman Filter

A 2D Kalman filter can be used to predict motion vector of successive frames given the observation or motion vectors of the previous frames. An algorithm describing the smoothing process is listed below.

- Initialize the state of the system using (x, y, dx, dy), where (x, y) is the observation (i.e. the centroid of the block) and (dx, dy) is the displacement of the centroids. The values of state can be initialized using the motion estimates between the first successive frame pair.
- The state of the system S at time instant $t + 1$ and the observation M at time t can be modeled using

$$S(t + 1) = AS(t) + Noise(Q) \tag{4}$$

$$M(t) = S(t) + Noise(R) \tag{5}$$

- Initialize A and noises Q, R as Gaussian.
- Perform the predict and update steps of standard Kalman filter
  - Initialize state at time instant t_0 using $S_0 = B^{-1}M_0$ and error covariance $U_0 = \begin{bmatrix} \in & 0 \\ 0 & \in \end{bmatrix}$
  - Iterate between the predict and update steps
  - Predict: Estimate the state at time instant $t + 1$ using $S^-_k = AS_{k-1}$ and measure the predicted error covariance as $U^-_k = AU_{k-1}A^T + Q$
  - Update: Update the correct, state of the system $S_k = S^-_k + K(M_k - BS^-_k)$ and error covariance as $U_k = (I - KB)U^-$
  - Compute K, the Kalman gain using $K = U^-_k B^T (BU^-_k B^T + R)^{-1}$
- Smooth the estimates of the Kalman filter and present the smoothed outcomes as $MV_{\aleph}$

3 RESULTS AND DISCUSSION

Here, in this section, we present some sample results of the stabilization task on wildlife videos taken at a zoological park. Performance of the video stabilization scheme can only be visually evaluated. We provide some sample frames illustrating the quality of video stabilization. Figure 1 compare the video stabilization quality of the base-line model versus the proposed model. As we can clearly visualize there is quite a increased quality in the stabilized version of the proposed model in comparison to the stabilized version of the base model. The motion correction scheme using the Kalman filter was sufficient to smooth the motion vector correctly. The reason to this is because, the changes observed in the capture was linear. Similarly in figures 2, we compare the quality of video stabilization using another sample clip from the same wildlife video. The movement of the camera in this sequence was more abrupt and random in directions. We observed that the proposed model using Kalman filter could not handle these changes well and as well generate a good quality stabilized output. However, the motion correction mechanism using the exponentially weighted moving average filter could produce much better results.

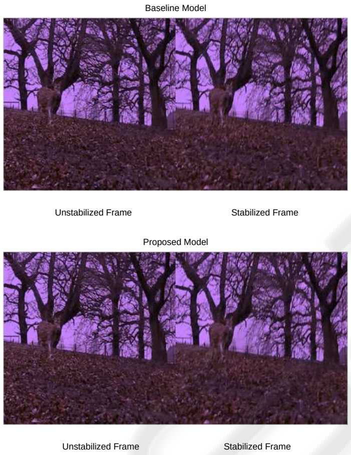

Figure 1: Model Performances on Video Sample Clip 3.

4 CONCLUSION

In this paper, we have presented a novel mechanism of motion correction and block based motion estimation strategy that combines vector quantization based block partitioning mechanism with the genetic algorithm based block search scheme applied to video stabilization. The model was tested on several real time datasets and the results have revealed a high degree of performance improvement when compared to existing video stabilization model based on motion estimation and filtering.

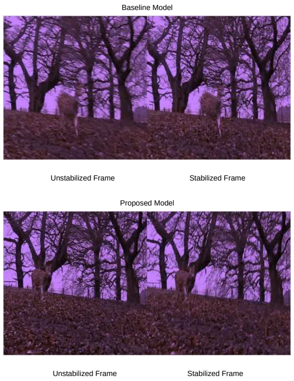

Figure 2: Model Performance on Video Sample Clip 6.

REFERENCES

A.Litvin, J. and W.C.Karl (2003). Probabilistic video stabilization using kalman filtering and mosaicking. *Image and Video Communication and Processing*.

J.Tucker and Lazaro, A. S. (1993). Image stabilization for a camera on a moving platform. In *IEEE Conference on Communications, Computers and Signal Processing*, pages 734 – 737.

K.Ratakonda (1998). Real-time digital video stabilization for multi-media applications. In *Proceeding of the 1998 IEEE International Symposium on Circuits and Systems*, pages 69–72.

K.Uomori, A.Morimura, H. T. and Y.Kitamura (1990). Automatic image stabilizing system by full-digital signal processing. In *IEEE Transactions on Consumer Electronics*, pages 510–519.

M.Hansen, P. and K.Dana (1994). Real-time scene stabilization and mosaic construction. *Image Understanding Workshop Proceedings*.

P.Pochec (1995). Moire based stereo matching technique. In *Proceedings of ICIP*, pages 370–373.

T.Chen (2000). Video stabilization algorithm using a block based parametric motion model. Master's thesis.

Y.Matsushita, E.Ofek, X. and H.Y.Shum (2005). Full-frame video stabilization. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Y.Yao, P. and R.Chellappa (1995). Electronic image stabilization using multiple image cues. In *Proceedings of ICIP*, pages 191–194.