A Robust Global Motion Estimation Scheme for Sprite Coding

Hoi-Kok Cheung and Wan-Chi Siu

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

ABSTRACT

A new global motion estimation technique for sprite coding is presented in this paper. The proposed system manages to accurately register frames to a sprite without referencing the sprite. This allows motion estimation process to be performed in an environment free from the adverse influence of the sprite which is usually blurred by the erroneous motion estimation. Furthermore, our proposed system can figure out the frame having the highest resolution, base frame, in the video sequence and project all other frames to the space of this base frame. This can avoid the loss of information due to decimation. Moreover, the determination of the size of the sprite can be conducted before construction of the sprite which avoid clumsy process of manual sprite memory allocation. Experimental results show that our proposed technique manages to estimate global motion in high accuracy and indicate a good robustness against estimation error which is suitable for the sprite coding application in MPEG-4.

1. Introduction

Global motion estimation is one of the essential tasks in image processing and video compression fields. Recently, an object-based coding standard named MPEG4 was proposed and sprite coding is one of the core techniques for compression using global motion estimation. Sprite coding techniques involve a generation of a high-resolution image called sprite. The image is composed of information belonging to objects visible throughout the video sequence. Usually the background objects are coded and the corresponding sprite is called "background mosaic".

One of the classes of the sprite defined in MPEG4 is built off-line prior to the coding of individual frames[1]. During encoding process, global motion between the sprite and the current frame is estimated (long-term global motion parameters) (for relative motion between consecutive frames, it is called short-term motion) Dufaux and Konrad[2] and Szeliski[3] proposed to estimate the global motion by an iterative minimization of the Frame difference error and Smolic, Sikora and Ohm[4] proposed to employ a recursive closed-loop prediction scheme to reduce the error accumulation problem. After the estimation of global motion, frames are then blended to form the sprite. Nicolas[5] proposed to analytically determine the blending coefficient to increase the coding efficiency.

In this paper, we propose a new approach for the estimation of global motion parameters by directly estimating the relative motion between current frame and a chosen reference frame. This approach manages to give an accurate, stable and robust estimate of the motion. It can also greatly alleviate the problem of error accumulation by isolating the infrequent image registration error and stopping the error from propagating to the rest of the frames. The estimation is performed in three stages by roughly estimating the initial registration matrix first using the result of the previous frame and followed by the estimation of affine motion parameters. Finally, motion parameters using higher order motion model can be obtained using gradient descent.

2 The proposed system

We have noticed that the registration error is highly related to the source of the reference for motion estimation. In our proposed system, instead of using the sprite to be the reference for motion estimation, we choose a particular frame in the history of time. An overview of the motion estimation algorithm is shown in Fig. 1(a). Let the previous registered frame and the chosen reference frame be frame m and frame k respectively. A_{mk} is the estimated registration matrix projecting frame m onto the space of frame k. To register the current frame m+1, we first project frame m+1 into the space of frame k using registration matrix A_{mk} to generate frame z. Then, motion parameters are estimated between frame k and frame z using short-term motion estimation techniques. Note that if there is error in the estimation of matrix A_{mk}, the error can be compensated during the motion estimation between frame k and frame z. Therefore, our proposed system is robust against error which manages to stop the error from propagating. Finally, the estimated parameters are concatenated to A_{mk} to give $A_{m+1,k}$ and $A_{m+1,k}$ is again concatenated to the long-term motion parameters of the reference frame k, A_{k1}, to result the long-term motion parameters $A_{m+1,1}$.

Initially, the reference frame is chosen to be the first frame of the sequence. The current frame would replace the reference frame if one of the following two cases occurs. The first case is that the displaced frame difference between the registered current frame and the reference frame is large. This reflects that there is a large variation of illumination and the reference frame is no longer appropriate for motion estimation. The

0-7803-7761-3/03/$17.00 ©2003 IEEE

second case is that the relative displacement between the current frame and the reference frame is large. This can be detected if either the overlaping area between the registered current frame and the reference frame is smaller than a threshold T1 or the non-overlap area is greater than a threshold T2.

$$T_1 = w \times h \times Nr \text{ and } T_2 = w \times h - T_1 \qquad (1)$$

where w and h are the width and height of the frame respectively. Nr is a parameter defined between 0 and 1 controlling the value of the two thresholds. For each of the chosen reference frame, a number of pixel blocks are selected for block matching in the rear stages. The block selection technique employed is our earlier work on long-term global motion estimation[6]. The aim is to exclude blocks of pixels, which mainly consists of textured area or homogeneous area, from calculation to avoid aperture problem and to reduce the chance of trapping into a local minimum.

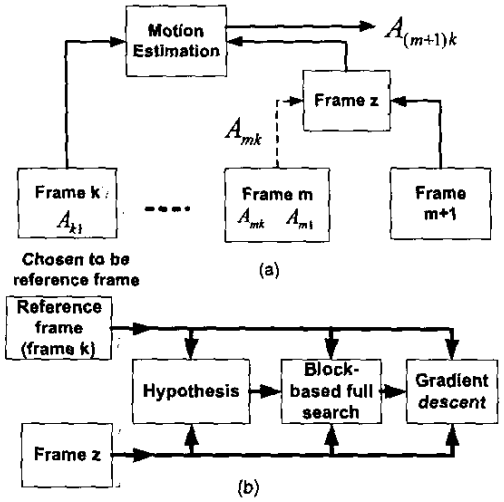

Fig. 1 Block diagram of the global motion estimation algorithm. (a) Frame m+1 is projected to frame z using registration matrix' A(mk). (b) Equivalent to the motion estimation block in (a)

The proposed motion parameters estimation algorithm comprises three stages as shown in Fig. 1(b). At the first stage, a rough estimation of the global motion between the reference frame and frame z is determined by some hypothesis tests. Then, the rough estimate is used as an initial guess of the solution in stage 2 and the motion parameters of affine motion model are computed using block matching technique. In the third stage, the result of the previous stage is used as an initial guess of the motion parameters of a higher order motion model, like the perspective motion model, and the parameters are refined using gradient descent algorithm.

2.1. Motion model

To efficiently describe the majority of the pixel motions between frames, one of the methods is to model the motion and parameterized it. Affine model is one of the models which can describe rigid object motion involving translation, rotation, magnification and shear.

$$\begin{pmatrix} x' \\ y' \\ s \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ s \end{pmatrix} \qquad (2)$$

Here, the coordinates of the point is expressed in terms of homogeneous coordinates. Coordinates $(x \quad y \quad s)^T$ represent coordinates $(x/s \quad y/s)^T$ on the image. To describe motion with perspective distortion, the perspective motion model has to be used.

$$\begin{pmatrix} x' \\ y' \\ s' \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ s \end{pmatrix} \qquad (3)$$

Describing the motion between consecutive frames over a short period, the affine motion model is generally sufficient. Higher order motion model can describe motion more loyally at the expense of higher computational load and more sensitive to noise [7]. Therefore, in this paper, we prefer to use the affine motion model in our experiments.

2.2. Initialization stage

In this stage, a rough estimation of the motion between the reference frame and frame z is performed. Note that the variation of the camera motion over a sequence is usually smooth. The inter-frame motion between the current frame and the previous frame is similar to the inter-frame motion between the last two frames. Three simple and fast tests can be performed to roughly estimate the camera motion and the corresponding sum of absolute difference(SAD) between images is calculated.

Hypothesis 1. This is used to classify that there is no relative motion between the current frame and the previous frame.

$$SAD_1 = \sum_{p \in S} |I_k(p) - I_z(p)| \qquad (4)$$

Hypothesis 2. This is used to classify that inter-frame motion between the current frame and the previous frame is the same as that of the last two frames.

$$SAD_2 = \sum_{p \in S} |I_k(p) - I_z(p, A_{m,m-1})| \qquad (5)$$

where $I_z(p, A_{m,m-1})$ represents the projected frame of I_z using $A_{m,m-1}$.

Hypothesis 3. This is used to classify that the object is under constant acceleration

$$SAD_3 = \sum_{p \in S} |I_k(p) - I_z(p, A^2_{m,m+1})| \qquad (6)$$

where $A^2_{m,m+1} = A_{m,m+1} A_{m,m+1}$ and S is the support defined as

$$S = I_k \cap I_z \cap I_z(A_{m,m+1}) \cap I_z(A^2_{m,m+1})$$

The set of motion parameters resulting the smallest SAD are taken to be the initial estimate of the relative motion.

2.3. Estimation of affine motion parameters

After the first stage, the estimate is concatenated to the registration matrix $A_{m,k}$ to form a rough estimate of registration matrix $A_{m+1,k}$. The current frame is

projected again into the space of frame k using the matrix $A_{m+1,k}$ and replaces the frame in frame z. Motion is then estimated between the reference frame k and frame z.

In this stage, the full search algorithm is employed to estimate motion vector of pixel blocks. Only the selected pixel blocks in the reference frame, as stated in session 2, is used for motion estimation. This can substantially cut the computational load while the information obtained is still sufficient to describe the global motion[6]. The registration matrix A_{zk} is obtained in a least square sense and is concatenated to the rough estimate of registration matrix $A_{m+1,k}$ to form the refined registration matrix $A_{m+1,k}$. Subsequently, the long-term registration matrix $A_{m+1,1}$ of the current frame can be computed by concatenating $A_{m+1,k}$ to $A_{k,1}$.

Generally speaking, the estimated registration matrix at this stage is sufficiently good to describe the motion. If the affine motion model is chosen for the system, the motion estimation phase has come to the end at this stage.

2.4. Refinement of parameters

If the system uses a higher order motion model like the perspective motion model, refinement of the previous stage result can be performed using a gradient descent algorithm. The gradient-based method attempts to minimize the displaced frame difference (DFD) and can be expressed as follows.

$$DFD(\mathbf{a}) = \sum_{i=1}^{N} \left[I_k(x_i) - I_z(x_i, A_{m+1,k}) \right]^2 \qquad (7)$$

where $\mathbf{a} = (a \;\; b \;\; c \;\; d \;\; e \;\; f \;\; g \;\; h)^T$ is the vector of motion parameters to be refined. Since the perspective motion model is non-linear, the motion parameters $\mathbf{a}$ have to be updated iteratively. The Levenberg-Marquardt algorithm is applied for the minimization [2,3,6]. Since the algorithm requires to provide a trial solution which is sufficiently close to the true solution, the estimated motion parameters of the previous stage is used. The algorithm refines the motion parameters iteratively until a convergence is reached or a maximum number of iterations is reached [1,8]. Finally, the registration matrix $A_{m+1,k}$ is formed and is concatenated to $A_{k,1}$ to result $A_{m+1,1}$.

3. Construction of sprite

Since the estimation of the registration parameters does not involve the sprite, sprite-building process is not necessary to be performed along with the motion parameters estimation process. In our proposed system, the sprite is built after all the frames in the sequence are registered. During the process of estimating the registration parameters of frames, the first frame of the sequence is regarded to be the base frame of the sprite. All the other frames are registered to the space of the base frame. If the sequence involves a variation of focal length, the resolution of the same part of the background can be different in the field of view of different frames. Therefore, after the registration of each frame, the area of the projected frame will be computed. Before the sprite-building process, the frame having the smallest projected area is chosen to be the new base frame as the frame has the highest resolution in the sequence over a particular area of the background. The area of the projected frame in the space of the base frame is computed by counting the number of pixels on an integer-pel grid lying within the projected area.

After the determination of base frame, the inverse of the base frame's registration matrix is concatenated to all registration matrixes. This operation is equivalent to projecting all the frames onto the space of the base frame and the projection of the base frame is equivalent to direct copying from the frame to the sprite without distortion as the registration matrix is an identity matrix. The next step is to determine the size of the sprite by computing the coordinates of the four corners of each projected frames. Subsequently, a transformation matrix of pure translation is computed that can map all the projected frames on the space of the base frame to a location such that all the coordinates of the pixels are positive. Then, the calculated matrix is concatenated to the registration matrix of all the frames. Finally, a sprite with the determined dimension is built by warping and blending all the frames onto it. A simple temporal average is employed in the blending process and a bilinear interpolation is carried out if the coordinates of projected pixels do not correspond to integer pixel.

4. Experimental Results

In this section, we will show some experimental results using our proposed system. Comparisons are made between systems using short-term motion estimation algorithm and long-term motion estimation algorithms. Three test sequences with CIF format are tested, namely "Stefan" (352x240), "Foreman" (352x288) and "Coast Guard" (352x240). All the sequences are provided with segmentation mask and the background objects of the sequences are encoded with various coders.

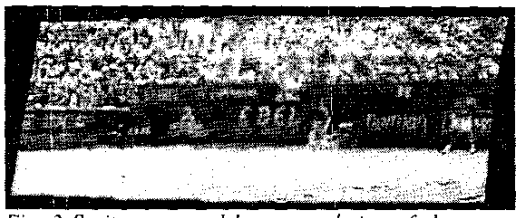

Fig. 2 Sprite generated by accumulation of short-term motion parameters using affine motion model

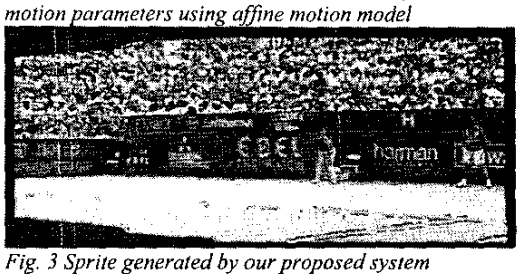

Fig. 3 Sprite generated by our proposed system

In our experiments, we chose to use affine motion model to describe the motion. Fig. 2 shows a sprite

generated with 150 frames of sequence Stefan by concatenating the estimated short-term registration matrix to form long-term parameters. It is noticed that sprite is blurred significantly because of the accumulated error, especially the area that the field of view of camera was moving to and fro. Fig. 3 shows the sprite generated by our proposed system with Nr = 0.1. The quality of the sprite is much better.

Sequence	Long-term (VM)[1] PSNR-Y[dB]/ Processing time (s)	Long-term [4] PSNR-Y[dB]/ Processing time (s)	Short-term [1] PSNR-Y[dB]/ Processing time (s)	**Proposed system** PSNR-Y[dB]/ Processing time (s)
Stefan (150)	20.889/ 5273	20.347/ 5354	18.955/ 750	**22.046/ 598**
Foreman (150)	28.057/ 3325	26.973/ 1905	27.941/ 511	**28.758/ 478**
Coast Guard (150)	23.586/ 5708	20.213/ 4654	22.294/ 1039	**23.538/ 1211**
Stefan (300)	18.871/ 9230	Failure	19.152/ 1966	**21.364/ 2140**

Table 1 Comparison of the quality of the reconstructed frames generated with different systems

Table 1 depicts the quality of the reconstructed frames generated by sprite coding systems with different motion estimation approaches. After the generation of sprite, frames are reconstructed by warping appropriate area from the sprites controlled by the corresponding registration matrixes. Comparison between the reconstructed frame and the original frame was done. The system proposed by the verification model[1] estimates the global motion in a hierarchical structure using gradient descent algorithm while the coder using long-term motion estimation scheme with closed-loop prediction feature employs the techniques proposed by Smolic, Sikora and Ohm[4].

It is noticed that our proposed system (with Nr = 0.1) significantly outperforms the system with short-term motion estimation which suffers from serious error accumulation problem. For the systems with long-term motion estimation, the quality of the sprite plays an important role in the motion estimation process. However, due to various reasons including motion modeling error and motion estimation error, the quality of the sprite continuously deteriorates throughout the coding process and the reliability of the sprite for reference decreases. Our proposed system estimates motions by directly referencing a chosen reference frame which avoid the adverse influence from the sprite. Therefore, our proposed system is more robust against error and outperforms all the other coders. Meanwhile, the computational complexity of our proposed system is very low comparing to the other coders.

Fig. 4 show the estimated scale factors in x and y directions of each frame in the sprite for 300 frames of sequence 'Stefan'. Notice that the scale factors of the first frame is slightly greater than 1 because it is not the frame having the highest resolution in the view of field. Only frame 118, the base frame, has scale factor equal to 1 and is copied directly to the sprite without distortion. All the other frames are scaled up and mapped to the space of frame 118. This can avoid information loss due to decimation operation.

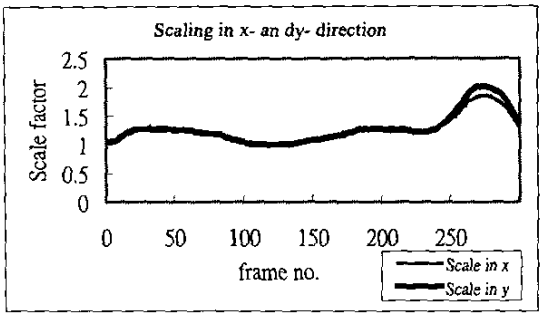

Fig. 4 Estimated Scale factors in x- and y- direction for sequence Stefan

5. Conclusion

We have presented a new technique in estimating the global motion between frames. The proposed motion estimation scheme directly registers the current frame with a chosen reference frame. This approach manages to avoid using the sprite to be the reference for motion estimation and prevent the registration error of the previous frames from propagating. Therefore, our system is very robust against error and can estimate motion accurately. Moreover, as our motion estimation scheme does not involve the sprite, sprite can be built after all the frames are registered. This allows the system to choose the frame having the highest resolution of the scene to be the base frame to avoid information loss due to decimation. Moreover, the size of the sprite can be accurately computed before the sprite is built which avoid blindly assigning a large buffer for sprite before encoding process starts.

6. References

[1] S.Fukunaga, Y.Nakaya, S.H.Son, and T.Nagumo, "MPEG-4 Video Verification Model version 14.2", ISO/IEC JTC1/SC29/WG11 5477, Maui, December 1999

[2] F. Dufaux and J.Konrad, "Efficient, Robust, and Fast Global Motion Estimation for Video Coding", Image Processing, IEEE Transactions on Image Processing, Vol. 9, Issue 3, March 2000

[3] R. Szeliski, "Video Mosaics for Virtual Environments", IEEE Computer Graphics and Applications, Vol. 16, Issue 2, March 1996

[4] A. Smolic, T. Sikora and J. R. Ohm, "Long-Term Global Motion Estimation and Its Application for Sprite Coding, Content Description, and Segmentation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9 Issue 8, Dec 1999

[5] N. Nicolas, "New Methods for Dynamic Mosaicking", IEEE Transactions on Image Processing, Vol 10, Issue 8, Aug. 2001

[6] H.K. Cheung and W.C.Siu, "Fast Global Motion Estimation for Sprite Generation", IEEE International Symposium on Circuits and Systems, Vol 3, 2002

[7] Z. Sun and A. M. Tekalp, "Trifocal Motion Modeling for Object-Based Video Compression and Manipulation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 5, Sep 1998

[8] W.H. Press et al., 'Numerical Recipes in C: The Art of Scientific Computing', second edition, Cambridge University Press, Cambridge, England, 1992