Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp36-39)

Efficient Compression Technique for Panorama Camera Motion

H. FAROUK*, S. MASHALI*, M. RASHWAN**
* Computers and Systems Department
Electronic Research Institute
** Electronics and Communications Dept.
Faculty of Engineering, Cairo University
EGYPT

Abstract: - Video compression is vital for efficient storage and transmission of digital video signal. The hybrid video coding techniques based on predicative and transform coding are adopted by many video-coding standards such as ISO MPEG-1/2 and ITU-T H.261/262/263, owing to its high compression efficiency. Motion compensation is a predictive technique for exploiting the temporal redundancy between successive frames of video sequence. Block matching is a simple and effective motion estimation method to obtain the motion compensation prediction. In panorama camera motion, we can enhance the bit rate and minimize the encoding time by calculating the largest block This minimizes the number of motion vectors required for P picture coding which improves the bit rate by average ratio 1:975, which is the ratio between the largest block and the equivalent blocks in the standard MPEG

Key-word: - video compression - motion compensation - video coding - motion vector

1 Introduction

Motion compensation is a vital component of many video coding standards as ISO MPEG-1/2 and ITU-T H.261/262/263 due to its high efficiency in reducing temporal redundancy between successive frames. Block-based motion estimation is the most popular method to obtain motion compensated prediction. By dividing each frame into rectangular blocks of equal size, the motion estimator obtains a motion vector for each of the blocks within a search window in the reference frame, using the block matching algorithm (BMA). The full search algorithm (FS) is the most straightforward BMA, which provides an optimal solution by matching all the candidate blocks inside the search window. However, the computational complexity of FS is always too high for real-time implementation. . The motion information may be used in the motion estimation phase in the video compression process, as the direction of the global motion will be used as guide in the searching technique, the motion information will be used to minimize the search overhead. Camera operation information is very significant for the analysis and classification of video shots. We will classify the motion into 4 types, using a video indexing technique into: *pan* (turning the camera up or down and it will be the same as moving the camera up or down), *tilt* (turning the camera left or right and it will be the same as moving the camera left or right), zoom *in* and *out* (moving the camera away or closer and it will be the same as changing camera lens to magnify or de-magnify the view) and the fourth type will be fixed camera through the shot. Once the shots are identified, representative key frames may be extracted and the shots may be clustered to obtain hierarchical views of video. The motion analysis may be done on the uncompressed data or on the compressed data we will use the one applied in the uncompressed domain for the online applications

2 The shot detection algorithm

For any sequence of spatial frames we will define 5 keyblocks per frame as shown in Fig.1. The keyblocks A, B, C, D and E (the central keyblock) are blocks of size 16x16 pixels. The tests proved that block size of 16x16 pixels is the optimum size because it contains a reasonable amount of details. This block size is programmable in our algorithm to satisfy the requirement for more smooth frames (enlarge the block to enable the algorithm to detect the direction, may be taken 32x32 pixels or 64x64 pixels) or for more detailed frames (minimize block size because it is sufficient to detect the direction from block of size 8x8 pixels or even 4x4 pixels for the highly detailed one). In Fig.1, *TLx*, *TLy* are the distance of any keyblock from the neighbor edge, *TLx* for the distance in *X* direction, *TLy* for the distance in *Y* direction. *KBx*, *KBy* will be for the keyblock dimension in *X* direction and *Y* direction respectively. In our work usually *TLx*=*TLy*, *KBx*=*KBy*. Three main directions will be defined, direction *a* for *pan*, direction *b* for *tilt* and direction *c* for *zoom in/out.*

For any frame *N* and frame *N+1* the keyblocks will be transformed into luminance-chrominance domain YUV or (*YCrCb*) and only Y components (luminance)[6-7] will be used because it contain most of the entropy of the block. The conversion equation from the RGB domain to YUV domain will be through the next equation

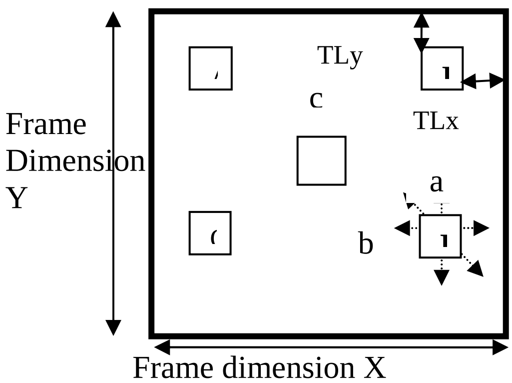

Fig.1: Keyblock with respect to the frame

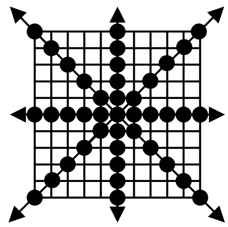

Fig.2: Test points for the keyblocks in the test window

Search strategy through search window will be done on the keyblock in frame *N* to determine its new position in frame *N+1*[8-9]. This search will be through 6 directions as in Fig.1, Fig.2.

The new position of the block in frame *N+1* will be for the minimum difference in the Y component in the whole directions that because, for example, if the test is in the horizontal direction, the motion may not be sharp horizontal so our result will get the global direction not the accurate position for the block.

In the case that the 5 keyblocks give the same direction, the results will be analyzed as follows:

All keyblocks shifted up or all keyblocks shifted down:

So, it is *pan* motion (or translate to up/down) and this is an information given to the motion estimation phase in the encoding process to create the motion vector much faster by starting search from the global position got from the proposed algorithm. This direction detected from the global motion analysis could be information given to the searching algorithm proposed in [8] to determine the quarter of the windows to start search inside.

All keyblocks shifted left or all keyblocks shifted right :

So, it is *tilt* motion (or translate to left/right) and from this result we will do the same as *pan* motion in case 1.

All keyblocks shifted towards the center of the frame except keyblock E is fixed (*zoom*

Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp36-39)

***out*) or all keyblocks shifted away from the center of the frame except keyblock E is fixed (*zoom in*):**

So, the decision is *zoom* process. From the step value in the searching we can get a conclusion about the speed of zooming i.e. if the zoom is very fast so it is more efficient to code the frames as I frame because most commonly the motion estimation will fail to catch matched blocks. Or we can apply block based motion estimation technique for zooming as subsampling or over sampling.

No keyblocks motion:

So it is random motion inside the same scene then we can apply indexing technique based on shot cut. This technique depends on calculating the DCT for the blocks while encoding process and from the histogram of the DC coefficient of the DCT block calculate the shot cut. The shot cut has two types, sharp cut and gradual cut.

3 Panorama Camera motion

A special type of pan and tilt camera motion is the panorama which is camera scan for a view horizontally or vertically with the assumption that the scene content is ultimately fixed

The panorama camera shot algorithm depends on considering the largest block in the frame for example if the frame size is 352x240, 20 pixels margin will be left from all edges so the block size will be 312x200. This will be a special case from block based motion estimation, the new position of the block will be detected and represent it with one motion vector this will achieve a good enhancement in the bit rate as:

The classical block size is 8x8

Our block size is 312x200

Our block replace number of blocks =(312x200)/(8x8) =975 blocks

One motion vector will be created to represent this block and replace 975 motion vectors

This will be in case of P picture . P picture represents around 50% of the bit rate according to table 2.4

Where

I picture represents 150 kbps

and P picture represents 50 kbps

and B picture represents 20 kbps

and in the GOP of 15 frames we have 1xI +5xP +10xB =

150+5x50+10x20 = 500 kbps

P type represents 250 kbps

So, this technique will enhance the bit rate in this special type of motion with the ratio of 1:975 for the P type and with the ratio of 1:975/2 i.e. 1:487.5 for the total bit rate (since P represent 50% of the total bit rate)

4 Results

Fig.3, Fig.4 give examples for the panorama shot as the motion is to the left direction. Notice the shift in the block marked with the black border in Fig.3.a to the right direction in the Fig.3.b and in Fig.4.a to the up direction in the Fig.4.b

a) Original frame with black border around the estimated block

b) Processed frame with black border around the compensated block

Fig.3: Panorama shot sample with 2 frames a) the original frame N with the block 312x200 pixels b) the same block in the new position in frame N+5

a) Frame 15 from camera moving down shot

b) Frame 25 from camera moving down shot

Fig.4: Panorama shot sample moving down with 2 frames a) the original frame N with the block 312x200 pixels b) the same block in the new position in frame N+10

5 Conclusion

Special shots, corresponding to view scanning in horizontal or vertical directions, are called Panorama shots. For these shot very large block of the frame can be characterized with a single motion vector. The number of motion vectors required will reduce the bit rate by the ratio of 1:975 for P frames.

References

[1] Hampapur, A., Jain, R., and Weymouth, T., "Digital Video Segmentation", Proc. ACM Multimedia 94, San Francisco, CA, October, 1994, pp. 357-364.

[2] Shahraray, B., "Scene Change Detection and Content-Based Sampling of Video Sequences", in Digital Video Compression: Algorithms and Technologies, Arturo Rodriguez, Robert Safranek, Edward Delp, Editors, Proc. SPIE 2419,February, 1995, pp. 2-13.

[3] M Pilu, "On using raw MPEG motion vectors to determine global camera motion", HP Laboratories Bristol, HPL pp. 97-102, Aug. 1997.

[4] Chok-Kwan Cheung, "Fast Motion Estimation Techniques for Video Compression",1998, Online web site: "http://www.image.cityu.edu.hk/~ckcheung/thesis/".

[5] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion estimation", IEEE Transactions On Image Processing, Vol. 9, pp. 287-290, Feb. 2000.

[6] H. Farouk, S. Mashali, M. Rashwan, A. Nassar, "New fast adaptive matching criterion for block-based motion compensation", Application of Digital Image Processing XXIII, SPIE, 30 July to 4 Aug 2000.

[7] H. Farouk, S. Mashali, M. Rashwan, A. Nassar, "Fast Memorized Search Technique for Block-Based Motion Estimation", Applications of Digital Image Processing XXIV, SPIE, 29 July to 3 Aug 2001.

[8] H. Farouk, S. Mashali, M. Rashwan, A. Nassar, "Video Compression Improvement based on camera motion", *IEEE ISSPIT 2001,Cairo, Egypt.*