International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 3, Number 8 (2013), pp. 779-784
© International Research Publications House
http://www. irphouse.com /ijict.htm

Video Compression

Seema Gupta

Ideal Institute of Management & Technology, Delhi.

Abstract

With the increasing popularity of technologies such as Internet streaming video and video conferencing, video compression has became an essential component of broadcast and entertainment media. Motion Estimation and compensation techniques, which can eliminate temporal redundancy between adjacent frames effectively, have been widely applied to popular video compression coding standards such as MPEG-1,MPEG-2, MPEG-4.The most commonly used motion estimation scheme is the full search block matching algorithm, which searches for the best candidate block among all the blocks in a search area of larger size in terms of either the mean-square error or the mean of the absolute frame difference. Though the computational complexity of this approach is very high, PSNR is also high. In this paper, we compare with Three step search & Diamond Search Algorithms in terms of computational complexity and PSNR. It searches for the best motion vectors in a coarse to fine search pattern. Full search Motion Estimation algorithm is not fit for real-time applications because of its unacceptable computational cost.

Keywords: Motion Estimation, Diamond Search, Three Step Search, Full Search.

1. Introduction

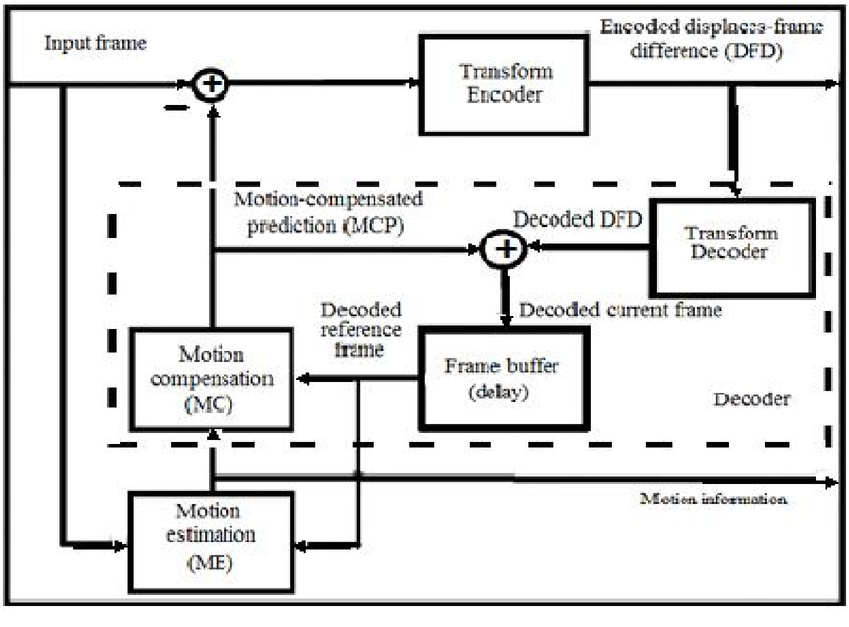
Video compression is needed for efficient storage of video, for efficient data transfer among various components of a video system, to reduce computational resources used in video processor hence reducing overall cost. There are two types of video compression techniques: lossless & lossy compression. In lossless compression, there is no information loss. The image reconstructed exactly the same as the original. Whereas in lossy compression, some information loss is tolerable.

1.1 Intra-frame & Interframe Compression

There are two types of redundancies in the video sequences, spatial redundancy & temporal redundancy. Redundancy among neighbouring pixels in an image is called as spatial redundancy & the coding technique which reduces the spatial redundancies in an image are called the intraframe coding. Whereas, redundancy between adjacent frames in a sequence of image is called as temporal redundancy & the coding technique which reduces the temporal redundancies are called interframe coding.

1.2 Motion Compensation

Motion estimation is defined as searching the best motion vector, which is the placement of the coordinate of the best similar block in previous frame for the block in current frame. Block-based matching algorithms find the optimal motion vectors which minimize the difference between reference block and candidate blocks. The idea behind block matching is to divide the current frame into a matrix of macro blocks that are then compared with corresponding block and its adjacent neighbours in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to p pixels on all fours sides of the corresponding macro block in previous frame. This p is called as the search parameter.



Wireless video conferencing application.

2. Matching Criteria for Motion Estimation

Three popular matching criteria used for block-based motion estimation are:

1. Mean of squarred error (MSE)
2. Sum of absolute difference (SAD)
3. Matching pel count (MPC)

MSE Criterion

Considering (k−l) as the past references frame l > 0 for backward motion estimation, the mean square error of a block of pixels computed at a displacement (i, j) in the reference frame is given by:

$$\text{MSE(i,j)=1/N}^2 \sum_{0}^{N-1}\sum_{0}^{N-1}[S(n1,n2,k)-s(n1+i,n2+j,k-l)]^2$$

SAD

The SAD measure at displacement (i, j) is defined as vector is determined in a manner similar to that the motion for MSE as

$$\text{SAD(i,j)=1/N}^2 \sum_{0}^{N-1}\sum_{0}^{N-1}[S(n1,n2,k)-s(n1+i,n2+j,k-l)]$$

$$\text{[d1,d2]=} \arg \underline{\min_{i,j}}[SAD(i,j)]$$

MPC Criterion

The count for matching and the displacement (i,j),for which the count is maximum correspond to the motion vector.

We define a binary valued function count(n1, n2) $\forall$ (n1, n2) $\in$ B as

count(n1, n2) = 1 if |s(n1, n2, k) −s(n1 + i, n2 + j, k − l)| $\leq \theta$

0 otherwise

Where θ is a pre-determined threshold. The matching pel count (MPC) at displacement (i, j) is defined as the accumulated value of matched pixels as given by

$$\text{MPC(i,j)=} \sum_{0}^{N-1}\sum_{0}^{N-1}[count(n1,n2,)]$$

$$\text{[d1,d2]=} \arg \underline{\max_{i,j}}[MPC(i,j)]$$

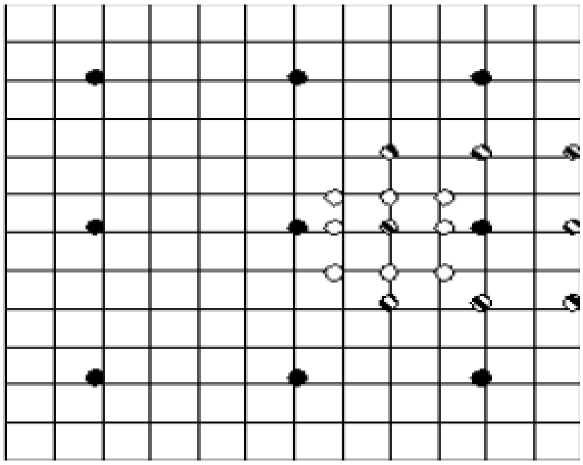
3. Block Mathching Algorithm

3.1 Full Search

Full search algorithm is the simple method for motion estimation. In searching for the best match, the correlation window is moved to each candidate position within the search window. There are a total (2p+1) *(2p+1) positions that need to be examined, where p is the search range for the block. The minimum dissimilarity gives the best match. The full search is brute force in nature and it delivers good accuracy in searching for the best match. But because of a large amount of computation is involved, it is useless in real-time encoding. There are several approaches to reducing the computational complexity. Fast search algorithms, the motion estimation process instead of full search, follows special pattern that checks less point number.

3.2 Three Step Search

Three step search is very simple and also robust. It searches for the best motion vectors in a coarse to fine search pattern. This algorithm is described in three steps. Initially, we have to choose the step size. In first step, eight blocks at a distance of step size from the centre around the centre block are picked for comparison. In the second step, the step size is halved. Now the centre is moved to the point with the minimum distortion & again eight blocks at a distance of step size from the centre around the centre block are picked for next comparison. Likewise steps are repeated till the step size becomes smaller than 1. A particular path for the convergence of this algorithm is shown in fig. It gives a flat reduction in computation by a factor of 9 to 10.



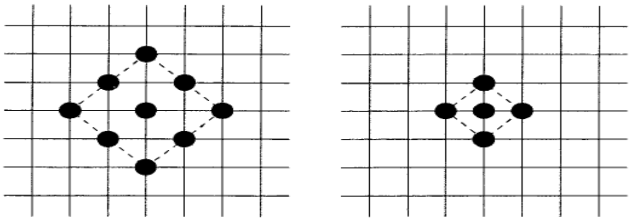
3.3 Diamond Search Algorithm

The DS algorithm is summarized as follows.

Step 1) The initial LDSP is centered at the origin of the search window, and the 9 checking points of LDSP are tested. If the MBD point calculated is located at the centre position, go to Step 3; otherwise, go to Step 2.

Step 2) The MBD point found in the previous search step is re-positioned as the center point to form a new LDSP. If the new MBD point obtained is located at the center position, go to Step 3; otherwise, recursively repeat this step.

Step 3) Switch the search pattern from LDSP to SDSP. The MBD point found in this step is the final solution of the motion vector which points to the best matching block.



4. Results

In this paper, we have implemented diamond search method & three step search method for motion estimation & we got the following results. We have more number of average computations & average PSNR using Diamond Search and three step search.

5. Conclusion

Three step search method is simple & near optimal performance. It searches for the best motion vectors in a coarse to fine search pattern. Computational complexity small motion estimation. Of this approach is almost 10 times less than full search method & PSNR is reduced only by one to two percent than full search method. Computational complexity and PSNR of diamond search is very less than three step search because the three step search uses a uniformly allocated checking point pattern in the first step, which becomes inefficient.

References

[1] Sang-jun Park, Gwanggil Jeon, adaptive partial block Matching Algorithm for fast motion estimation, 978-1-4244-4316-1/10/2010 IEEE

[2] Jik-Han Jung, Hwal-Suk Lee, Byung-Gyu Kim, Dong-Jo Park, Fast Block Matching Algorithm Using Spatial Intensity Distribution, 0-7695-2358-7/05 $20.00 © 2005 IEEE

[3] P.C.Shenolikar,S.P.Narote,Different approaches for motion estimation, international conference on control automation, communication and energy conservation -2009, 4th-6th June 2009

[4] Jong-Nam Kim and Tae Sun Choi, A fast three step search algorithm with minimum checking points using unimodel error surface assumption,IEEE transaction on consumer electronice,vol.44,no-3,august 1998

[5] Xuan Jing and Lap-Pui Chau, Senior Member, IEEE transactions on multimedia, An Efficient Three-Step Search Algorithm for Block Motion Estimation vol. 6, no. 3, June 2004

[6] Gustavo M.Callico Analysis of Fast Block Matching motion estimation algorithms for video super resolution systems, IEEE Transactions on consumer electronics, vol.54, n0.3, august 2008

[7] Aroh Barjatya Block Matching Algorithms for motion estimation, student member, IEEE

[8] Fast full-Search Block Matching, Michael Brunig and wolfgang Niehsen,IEEE transactions on circuits and systems for video technology, vol.11, no.2,February 2001

[9] Chi-Wai Lam,Lai-Man PO and chun ho chun, “A novel-kite-cross- diamond search algorithm for fast blockmatchingmo-tion estimation,” in Proc. 1997 Int. Conf. Information Communication and Signal Processing (ICICS), vol. 1, Sept. 9-12, 1997,pp.292-296.

[10] Yasser Ismail,Jason Mcneelly,Enhanced Efficient diamond search algorithm for fast block motion estimation.