RATE DISTORTION OPTIMIZATION FOR MOTION -PARAMETERS OF VIDEO CODING USING 3D-DWT

SHAIK JUMLESHA  
M.E.(CSE),(Ph.D),  
ASSOCIATE PROFESSOR IN CSE  
KKCW,PUTTUR,CHITTORE(dt)  
Ph: 09951747705

DR.CH.SATHYANARAYANA  
M.TECH(CSE),Ph.D  
ASSOCIATE PROFESSOR IN CSE & CE  
JNTUK- KAKINADA  
Ph:9177790000

ABSTRACT:

Motion Estimation is computationally the most complex part that is why it takes most of the time involved. For a good video system either the speed of the ME algorithm should be fast or the complexity should be reduced .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 (ME) and Compression 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 and MPEG-4 and H.261, H.263 and H.264 are the important standards. Motion Estimation provides Compression through Temporal redundancy removal for the video signal. Improvement in reducing the Computations overhead and achieves very good Peak Signal to Noise Ratio(PSNR) values, which makes the techniques more efficient than the conventional searching algorithms. Motion scalability is based on the simple concept that different decoding scenarios require different Motion Prediction qualities in the optimized Rate Distortion sense. To reduce the Motion Vector Overhead in Bidirectional frame prediction. We utilize the K-L Transform to obtain theoretical performance bounds at high bit-rates and compare to both optimum intra-frame coding of individual Motion-Compensated pictures and Single-Hypothesis Motion-Compensated predictive coding. The error drifting effect introduced from quantized motion is the first problem to face, followed by the interactive issue with other scalabilities, the embedded coding of scalable motion, and the Rate Distortion optimized estimation algorithm for motion parameters.

Key words: ROI, Motion Estimation, K-L Transform
I. Introduction
Video coding standards have steadily evolved during the past twenty years due to the need for efficient transmission and storage of digital video media. While the first standard, H.120 [1], set a crucial milestone for digital video coding in 1984, H.261 [6] introduced the hybrid video coding infrastructure, which became the basis of all modern video coding standards. Since H.261 was finalized in 1990, significant research has emerged to improve the coding efficiency of hybrid video compression. By 2003 when the state-of-the-art single layer video coding standard H.264/AVC [8] was completed, the performance of hybrid video coding reached an unprecedented summit. This achievement can be attributed to a comprehensive elaboration of the components, among which closed-loop motion prediction and rate-distortion optimized estimation play an important role.

A simplified chronology of video coding standards is shown in Fig.

![Chronology of international video coding standards](image)

The transmission of digital video signal requires a large bandwidth which is not practically possible. The solution to this problem is the compression of the digital video data. Many compression standards are available as already discussed. MPEG—stands for moving picture experts group, a body working on the compression standards. MPEG-1, MPEG-2 and MPEG-4 are the important standards provided by this body. ITU-T is the other body which has to its credit H.261 and H.263. H.264 is a compression standard which is the result of the combined effort of ISO/IEC’s MPEG and ITU-T’s VCEG named JVT. This standard is better in many respects than its predecessors. More computational complexity is required but the better compression performance and efficiency has made it more popular.

A video sequence can be considered to be a discredited three-dimensional projection of the real four-dimensional continuous space-time. The objects in the real world may move, rotate, or deform. The movements cannot be observed directly, but instead the light reflected from the object surfaces and projected onto an image. The light source can be moving, and the reflected light varies depending on the angle between a surface and alight source. There may be objects occluding the light rays and casting shadows. The objects may be transparent (so that several independent motions could be observed at the same location of an image) or there might be fog, rain or snow blurring the observed image. The discretization causes noise into the video sequence, from which the video encoder makes its motion estimations. There may also be noises in the image capture device (such as a video camera) or in the electrical transmission lines. A perfect motion model would take all the factors into account and find the motion that has the maximum likelihood from the observed video sequence. Changes between frames are mainly due to the movement of objects. Using a model of the motion of objects between frames, the encoder estimates the motion that occurred between the reference frame and the current frame. This
process is called motion estimation (ME) [4]. The encoder then uses this motion model and information to move the contents of the reference frame to provide a better prediction of the current frame. This process is known as motion compensation (MC), and the prediction so produced is called the motion-compensated prediction (MCP) or the displaced-frame (DF) [5]. In this case, the coded prediction error signal is called the displaced-frame difference (DFD). A block diagram of a motion-compensated coding system is illustrated in the reference frame employed for ME can occur temporally before or after the current frame. The two cases are known as forward prediction and backward prediction, respectively. In bidirectional prediction, however, two reference frames are employed and the two predictions are interpolated. The most commonly used ME method is the block-matching motion estimation (BMME) algorithm.

**Figure.** This is the most commonly used interframe coding method.

![Figure 2.1: Motion Compensated Video Coding](image)

**II. Current state of the Art:**

[1] Full-Search Block-Matching Algorithm (FSBMA) A search scheme which tries to find the best matching position of a 16x16 macro block (MB) of the current frame with a 16x16 block within a predetermined or adaptive search range in the previous frame gives what is called motion estimation. To determine the best motion vector position, the distance criterion is in most cases the SAD.

[2] Hierarchical and Multi resolution Fast Block-Matching The hierarchical and multi resolution fast block matching algorithms predict an approximate large scale MV in a coarse resolution representation and then to refine the predicted MV in a multiresolution fashion in order to obtain the MV in a finer resolution. On of the first algorithms of this class was propose by [Bier 8] and [BierB]. The hierarchical fast block-matching methods use the same image size but different block sizes at every hierarchical level for distance criterion. In some techniques different parameters and/or different algorithms at each level of hierarchy are applied. The research conducts includes: Multi grid sub sampling of 8x8, 4x4x, 2x2, 1x1 blocks on the luminance data for aH.263 coder was applied by [5]. HPDS (Hierarchical partial distortion search) is similar to the previous one and proposed by [3] Multiresolution fast block-matching. The multiresolution methods use different image resolutions with a smaller image size at a coarser level (i.e. of a pyramid form). Multiresolution representation may be traced in the following: Gaussian Pyramid, reduced pel resolution, Mean pyramid, MADM (Mean Absolute Difference of the means), BSPA (Block Sum Pyramid Algorithm). As far VLSI implementation is concerned it requires increased memory which is a drawback [58]. A multi-level motion intensity structure is the basis of an algorithm being proposed [59]. It determines the motion intensity at three levels...
and uses different estimation techniques to find motion estimation vector with faster speed. Table 2.4 summarizes the average video distortion and speedup performance of MVFAST, AMSED and FMEHMI compared with FS in terms of PSNR gain and number of searched points per MV search for the tested sequence. The performance of FMEHMI is closer to the performance of MVFAST and AMSED however FMEHMI has got more speed for the number of reference frames equal to one or five [59]. Most of the hierarchical block matching algorithms reduce computation by matching only some of the locations inside the search area. A novel hierarchical partial distortion search (HPDS) algorithm reduces the computation of each distortion measure instead of the number of checking points by using partial distortion measure. [60] Fast hierarchical algorithms such as 2D-logarithmic search, cross search, three-step search (3SS) and four step search have been proposed to reduce the computational complexity for block matching. A new search algorithm which is based on the following assumption: If a motion vector is the optimum in the full distortion measure, there is high probability that it will be one of the motion vectors with minimal values in the partial distortion measure. The HPDS algorithm divides the motion vector search into three levels. Lower level issues partial distortions (PDs) with higher decimation ratios. For each level, a number of candidate motion vectors (CMVs) with minimal PDs is selected to enter the next level. In order to reduce the encoder complexity a hierarchical block matching based motion estimation algorithm that uses a set of SAD computations for motion estimation of different block sizes is proposed. Hierarchical prediction and the median motion vector predictor of H.264 the algorithm defines a limited set of candidate vectors and optimal motion vectors for all partitions are Chases from this common set. Hierarchical estimation with the SAD rousing reduces the total computations by a factor of 17.6 with slight loss in coding efficiency [61]. A high performance and low cost hardware architecture for real-time implementation of a SAD reuse based hierarchical motion estimation algorithm .264/MPEG-4 part-10 video coding is presented [62]. A fast inter prediction algorithm based on hierarchical homogenous detection and cost analysis to select the best mode effectively is presented. For each macro block it is detected first whether it is spatial homogenous or not. For the non spatial homogenous macro block, 16x16 motion estimation is performed and examined whether the block is temporal homogenous or not. Once the homogenous macro blocks are detected, the best mode will be chosen as 16x16 modes. For the non-homogenous macro blocks the 8x8 motion estimation is executed [63].

[4] Multiple Reference Frame for Motion Estimation

H.264/AVC performance is better than the other standards because of the adoption of variable block sizes, multiple reference frames and rate-distortion optimization within the codec. These features cast in encoder complexity. The H.264 encoder stores up to five previous frames for motion estimation and motion compensation. The processing time increases linearly with the number of reference frames, because motion estimation is performed for each reference frame. In the process the best coding result is obtained using Forward Domain Vector Selection (FDVS) and top-down splitting algorithms, a novel fast multiple reference-frame based motion estimation and mode decision algorithm for H.263-to-H.263 trans coder is proposed. Potential block-sizes reduction and early-termination schemes are also applied to make the processing faster [78]. The blocks with different sizes will be used when performing the inter prediction. Then each block must consider previous five frames when making motion estimation. The best reference frames of neighboring blocks to determine the best reference frame of the current block.
[5] Motion Feature Based Fast Multiple Reference Frame Motion Estimation: In H.264/AVC real-time encoding system, the ME engine executes the motion estimation MB by MB. If the motion feature of MB can be obtained and small search range on small motion MB is obtained, a lot of redundant search positions can be eliminated. The search range adjustment scheme is based on motion vectors of previous frames have been applied. The MVs of former frames can reflect the motion feature of current MB. However, necessity of extra memory storage for all MVs of the previous frames makes, these schemes hardware unfavorable [80]. On the basis of analysis it can be said, that the efficiency of multiple reference frames will be degraded by the relative motion between the camera and the objects, for the slow-moving MB, in spite of adopting the multiple reference frames but reduced search range. For the fast-moving MB, the first previous reference frame is used with the full search range during ME processing [81].

[6] ME Algorithms Similar to UMHexagonS Algorithms: Fast motion estimation is the solution of the FS algorithm. Fast motion estimation algorithms can be categorized in two ways. One type reduces the search points, like TSS (three step search) algorithm, four step search (4SS), diamond search algorithm (DS), hexagon-based search algorithm (HEXBS) etc. The other type of algorithms reduces the calculation for every search candidate. It uses sub-sampling method and partial calculation. Sub-sampling method is efficient but not accurate, while the partial calculation is accurate but not enough efficient. Fast motion estimation algorithms proposed for H.264 achieve high speed and good visual quality. UMHexagonS, (Unsymmetrical – cross Multi-Hexagon- grid Search) algorithm has been adapted by H.264 reference software. UMHexagonS algorithm gives a better speed up [82]. H.264 has accepted UMHexagonS algorithm because of its ability to reduce computational load for ME. The algorithm reduces the number of candidate blocks within a search window. The problem of local minimum is avoided and coding efficiency is increased by conducting it into two parts: initial search center prediction and use of the hybrid of integral pixel search. The spatial as well as temporal predictions constitute the main mechanisms for motion estimation. These mechanisms generate four types of predictions: Median Prediction, Up-Layer Prediction, Corresponding block Prediction and Neighboring Reference frame prediction. These prediction means are used to predict the initial search center [83]. The main steps to search motion vectors in the algorithm are as follows.

Step 1: The search begins with the unsymmetrical cross search taking a search range of 16. [84]. Step 2: The best match of step 1 gives the center point for step 2 search which is 5x5 full search. Step 3: The last search process uses extended hexagon-based search, composed of symmetrical-hexagon-grid search and a small diamond search until the center of the search pattern is the best candidate point. UMHexagonS algorithm uses the hierarchical and hybrid motion search strategies to find the optimum motion vector [84][85]. UMHexagonS uses Unsymmetrical Cross search because of the fact that the movement in the horizontal direction is more than that of the vertical direction and the distribution of the motion vector is zero centered. If the starting point is not good then the algorithm may fall into the local minimum. After the initial search start point is called the predicted motion vector. Symmetrical cross search is disposed at the predicted motion vector point. Then these points are tested to decide the search point with the minimum cost. The best matched point is the center point for the next search step. Multi square grid
search are disposed at the minimum cost point in step1. The best point serves as the center point for the next search step. In the last step small diamond search gives the best matched point as the final MV (Motion Vector) [86].The search patterns which are employed in another algorithm is like diamond pattern ,big-cross pattern, small cross pattern and mixed hexagon and small-cross pattern gives an improved motion estimation speed [87].The proposed algorithm is explained in 4.5.2, simulation and experimental work in

[7] Block Matching Algorithm : Figure 2.2 illustrates a process of block-matching algorithm. In a typical Block Matching Algorithm, each frame is divided into blocks, each of which consists of luminance and chrominance blocks. Usually, for coding efficiency, motion estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. These candidate blocks are just the displaced versions of original block. The best candidate block is found and its displacement (motion vector) is recorded. In a typical inter frame coder, the input frame is subtracted from the prediction of the reference frame. Consequently the motion vector and the resulting error can be transmitted instead of the original luminance block; thus inter frame redundancy is removed and data compression is achieved. At receiver end, the decoder builds the frame difference signal from the received data and adds it to there constructed reference frames. This algorithm is based on a translational model of the motion of objects between frames. It also assumes that all pels within a block undergo the same translational movement. There are many other ME methods, but BMME is normally preferred due to its simplicity and good compromise between prediction quality and motion overhead. This assumption is not strictly valid, since we capture 3-D scenes through the camera and objects do have more degrees of freedom than just the translational one. However, the assumptions are still reasonable, considering the practical movements of the objects over one frame and this makes our computations much simpler. There are many other approaches to motion estimation, some using the frequency or wavelet domains, and designers have considered scope to invent new methods since this process does not need to be specified in coding standards. The standards need only specify how the motion vectors should be interpreted by the decoder . Block Matching(BM) is the most common method of motion estimation. Typically each macro block (16x16 pels) in the new frame is compared with shifted regions of the same size from the previous decoded frame, and the shift which results in the minimum error is selected as the best motion vector for that macro block. The motion compensated prediction frameis then formed from all the shifted regions from the previous decoded frame [5].

[8] Block Matching Algorithm based on GA

Motion estimation is an essential component of many video encoding algorithms. The most popular method adopted to estimate the motion between frames is the block matching algorithm (BMA), in which a block of size M × N is compared with a corresponding block within a search area in the previous frame. Three main elements-match criterion, search area and search scheme-should be
considered in the BMA. The match scheme is the most important, which plays a crucial part in the performance of BMA. Essentially, the match scheme of BMA is an optimal problem. The full search BMA will always find the optimum motion vector by calculating the difference between every block in the search window from the previous frame and the current block. However, the price paid for this optimum performance is a high computational cost, which prevents it from being applied in most real-time systems. A number of fast search algorithms have been proposed to greatly reduce the computational complexity by finding a suboptimum motion. All these algorithms rely on an assumption that the difference measure decreases monotonically as the search position moves closer to the optimum position. Because this assumption usually does not hold, these fast algorithms may only find the local minimum and cannot find the global minimum. Thus, the quality of encoded video may become much worse. The genetic algorithm is an optimum-searching process based on the laws of natural selection and genetics. It adopts the coding set of parameters instead of the parameters themselves to operate and searches the optimum based on groups of points, not a single point. Moreover, it uses a random, instead of a definite rule to work on the searching process. All of these give it high robustness as well as parallelism, and enable it to be free of the limitation in continuity and single peak requirement. It is effective in solving the problems of searching global optimum, although the computational complexity of genetic algorithm is high [13]. To solve an actual problem with genetic algorithm, the parameters of the problem are coded firstly; and then a fitness function should be chosen for determining the winner. Later the initial populations are selected and begin to evolve, that is, are processed by the selection operator, crossover operator, mutation operator and other genetic operators.

[9] Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization tech-nique developed by Dr. Eberhart and Dr. Kennedy in 1995 [14], inspired by social behavior of bird flocking or fish schooling. It is widely accepted and focused by researchers due to its profound intelligence background and simple algorithm structure. Currently, PSO has been implemented in a wide range of research areas such as functional optimization, pattern recognition, neural network training, fuzzy system control etc. and obtained significant success. Like Genetic Algorithm (GA), PSO is also an evolutionary algorithm based on swarm intelligence. But, on the other side, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the solution space by following the current optimum particles. The original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. Through competitions and cooperation, particles follow the optimum points in the solution space to optimize the problem. Many proposals indicate that PSO is relatively more capable for global exploration and converges more quickly than many other heuristic algorithms. PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what’s the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization
problems [15]. In PSO, each single solution is a “bird” in the search space. We call it “particle”. All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. Particle Swarm has two primary operators: Velocity update and Position update. During each generation each particle is accelerated toward the particles previous best position and the global best position. At each iteration a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times, or until a minimum error is achieved.

[10] Clonal Particle Swarm optimization (CPSO) : Artificial immune system inspired by the principles of immune response is a simplified model of its natural coordinate for solving problems [21]. The essence of the conventional PSO is to use particles with best known positions to guide the swarm or the population to converge to a single optimum in the search space. Colonel expansion is probably a good way to guide or direct the conventional PSO escaping from local optima whilst searching for the global optima efficiently [22]. According to the clonal expansion process in natural immune system the clonal operator is to clone one particle as N same particles in the solution space according to its fitness function at first, then generate N new particles via clonal mutation, which are related to the concentration mechanisms used for antigens and antibodies in Natural immune system. The clonal operator is used to copy one point as N same points according to its fitness function, and then generate N new particles by undergoing mutation and selection processes.

[11] Bidirectional Motion Estimation Based on PSO

Generally, the most straightforward BMA called full search (FS) simply compares the given macro block (MB) in the anchor frame with all candidate MBs in the target frame exhaustively within a predefined search region. This is not fit for real-time applications because of its unacceptable computational cost. To speed up the search, various fast algorithms for block matching which reduce the number of search candidates have been developed. Well known examples are three-step search (TSS), Four Step Search (4SS), block-based gradient descent search (BBGDS) and diamond search (DS) have been pro-posed to reduce computational efforts, based on fixed search pattern and greedy search method. Over the last few years, promising computational intelligence methods, called evolutionary computing techniques such as genetic algorithm (GA), particle swarm optimization (PSO) have been successfully applied to solve motion estimation problem. Such approaches are suitable for achieving global optimal solution, which traditional fast BMA s are not able to obtain easily. The GA needs to set some key parameters such as population size, probability of mutation, probability of crossover, etc. If these parameters are not prefixed properly, efficiency of GA becomes lower and also it is time consuming process. So here we are adopted PSO procedure in order to do the bidirectional motion estimation. Motivated by the potential improvement attainable by switching from independent search to joint search for the motion vector estimation, and by the practical requirement of avoiding an excessively high search complexity, the proposed method is an iterative technique to jointly optimize the motion vectors by using particle
swarm optimization. Bidirectional ME forms a major computation bottleneck in video processing applications such as the detection of noise in image sequences, interpolation/prediction of missing data in image sequences and deinterlacing of image sequences.

III. CONCLUSION

Because of the Internet is more and more universal and the technology of multimedia has been progressed, the communication of the image data is a part in life. In order to employ effect in a limit transmission bandwidth, to convey the most, high quality user information. It is necessary to have more advanced compression method in image and data. Motion Estimation (ME) 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-2, MPEG-4. Full search Motion Estimation algorithm is not fit for real-time applications because of its unacceptable computational cost. Bidirectional ME forms a major computation bottleneck in video processing applications such as the detection of noise in image sequences, interpolation/prediction of missing data in image sequences and deinterlacing of image sequences. The computational complexity of motion estimation is 70 percent to 90 percent in video coding and processing systems, so the idea is to find a fast Motion Estimation algorithm to improve the operation. Traditional fast block matching algorithms are easily trapped into the local minima resulting in degradation on video quality to some extent after decoding. Since these Evolutionary Computing Techniques are suitable for achieving global optimal solution. The Bidirectional method is a novel iterative technique which reduces the number search point in a frame and a good reconstruction of frames making these proposed methods computationally fast and efficient techniques. The speed up of an algorithm can further be increased by optimizing the algorithm. A new method of using a few partitions of macro block instead of all partitions for H.264/AVC motion estimation is proposed. Performance analysis of proposed method shows significant reduction in motion estimation time for low motion activity video sequences. The subjective comparison with H.264/AVC motion estimation demonstrates that proposed methodology can be used without significant loss of subjective quality of video on the same lines of the method and shapes of the search area. In fact the number of search points has been reduced. Bidirectional ME forms a major computation bottleneck in video processing applications such as the detection of noise in image sequences, interpolation/prediction of missing data in image sequences and deinterlacing of image sequences. The bidirectional algorithm is giving less prediction error and the numbers of search point per each frame are less. In addition, skipping those static macro blocks from processing can reduce the computational cost of the algorithm.

IV. SCOPE FOR FUTURE WORK

Full search Motion Estimation algorithm is not fit for real-time applications because of its unacceptable computational cost. Bidirectional ME forms a major computation bottleneck in video processing applications such as the detection of noise in image sequences, interpolation/prediction of missing data in image sequences and deinterlacing of image sequences. The computational complexity of motion estimation is 70 percent to 90 percent in video coding and processing systems, so the idea is to find a fast Motion Estimation algorithm to improve the operation. From the simulation results it is observed that just consumes a few lines of codes due to its
simplicity which makes the ME algorithm based or hard-ware implementation. In the future, the global searching ability and to speed up the search and to avoid being trapped in local minima. In future other evolutionary computing techniques also can be tried for the better results. Three important factors Block size, search area, matching criteria can be varied such as Variable block size, large search area for complex motions and small search area for low complex motions and also by taking the initial particle positions also make a big change. It also assumes that all pels within a block undergo the same translational movement. There are many other ME methods, but Block matching ME is normally preferred due to its simplicity and good compromise between prediction quality and motion overhead. In bidirectional motion estimation also we can try to implement new techniques which will further reduce the complexity of MPEG video coding.

References


[12] Hung - Chih Lin; Yu-Jen Wang; Kai-Ting Cheng; Shang-Yu Yeh; Wei-NienChen; Chia -Yang Tsai; Tian - Sheuan Chang; Hsueh-Ming Hang;


