Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
6th Meeting: Awaji, Island, JP, 5-13 December, 2002

Document: JVT-F017
Filename: JVT-F017.doc

Title: **Fast Integer Pel and Fractional Pel Motion Estimation for JVT**

Status: Input Document to JVT

Purpose: Proposal

Author(s) or Contact (s):
Zhibo Chen, Peng Zhou, Yun He
Dept. of Electronic Engineering, Tsinghua University, Beijing, China
Yidong Chen
SVA Group, Co.Ltd, Shanghai, China

Tel: 86-10-62781413
Email: chenzhibo@yahoo.com
hey@tsinghua.edu.cn

Source: Tsinghua University, China and SVA Group, China

1. INTRODUCTION

Similar to former video standards such as H.261, MPEG-1, MPEG-2, H.263, and MPEG-4, H.264 is also based on hybrid coding framework, inside which motion estimation is the most important part in exploiting the high temporal redundancy between successive frames and is also the most time consuming part in the hybrid coding framework. Specifically multi prediction modes, multi reference frames, and higher motion vector resolution are adopted in H.264 to achieve more accurate prediction and higher compression efficiency. As a result, the complexity and computation load of motion estimation increase greatly in H.264 and our experiments demonstrate that motion estimation can consume 60% (1 reference frame) to 80% (5 reference frames) of the total encoding time of the H.264 codec and much higher proportion can be obtained if RD optimization or some other tools is invalid and larger search range (such as 48 or 64) is used.

Generally motion estimation is conducted into two steps: first is integer pel motion estimation, and the second is fractional pel motion estimation around the position obtained by the integer pel motion estimation (we name it the best integer pel position). For fractional pel motion estimation, 1/2-pel accuracy is frequently used (H.263, MPEG-1, MPEG-2, MPEG-4), higher resolution motion vector are adopted recently in MPEG-4 (1/4-pel accuracy)[1] and JVT (1/4, 1/8-pel accuracy)[2] to achieve more accurate motion description and higher compression efficiency.

Algorithms on fast motion estimation are always hot research spot, especially fast integer pel motion estimation has achieved much more attention because traditional fractional pel motion estimation (such as 1/2-pel) only take a very few proportion in the computation load of whole motion estimation. Many fast integer block-matching algorithms have been focused on the search strategies with different steps and search patterns in order to reduce the computation complexity and maintain the video quality at the same time. These typical fast block matching algorithms include three step search (TSS)[2] 2-D logarithmic search (2-D LOGS[3], , Block-based gradient decent search (BBGDS)[4], Four step search (FSS)[5], HEXBS(Hexagon-Based Search)[6], etc. These algorithms performed well in relatively small search range and picture size. But in many applications such as SDTV or HDTV, the picture size is large and the search range should be large enough for high coding efficiency. In some sequences like Bus or Stefan, the algorithms such as HEXBS are likely to drop into a local minimum in the early stages of search process. To solve this “local-minimum” problem, some previous works predicted the motion vector from the neighboring matching blocks or coordinate block and then searched from the predicted motion vector. Those method could avoid dropping into the local minimum with a better starting search point but they might also give a false predicted motion vector in some cases, for example, the neighboring matching blocks are in different moving objects. Experiment results show that these algorithms will cause nearly 1-2dB degradation in the image quality compared to Full Search (FS) algorithm in such sequences as Bus or Stefan.

In order to prevent the search points from getting into a local minimum in the earlier stages, global search is usually performed with more search points which can cover the overall search area. And search points were selected from sparser grids in the overall search window in order to reduce the number of search points in the first search step. For example, in two-step search strategies (2SS)[7] the grid size is two. After the best motion vector with the coarse grid is obtained, fine motion vector search is performed in a small area around it. The well-known algorithms such as TSS can be used in this step[8]. It can be easily seen that with grid size of 2, the computation complexity is still too high when large search window is used. Our experiments show that the PSNR quality will drop greatly as the grid size increases. Therefore fixed-grid search algorithm can't give a good compromise between computation complexity and video quality.

Based on our former proposed Horizontal and multi-grid search (HMGS) strategy[8] for MPEG-2 and the formal proposal for H.264[12][13][14], we propose a hybrid Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) algorithm for integer pel motion estimation in H.264 in this proposal according to the characteristics of multi prediction modes, multi reference frames in H.264. The proposed algorithm solves the above contradiction by Unsymmetrical-cross and Multi-Hexagon-grid Search strategies and hybrid search methods is used to be compatible with the multi prediction modes and multi reference frames characteristics in H.264.

And it's clear that with the development of integer pel fast motion estimation algorithm and the decreasing of integer motion search points, computation load of fractional pel motion estimation becomes more and more comparable to that of integer case. And our experiment results also demonstrate this phenomenon. Hence reducing the computation load for the fractional pel motion search is both necessary and significant.

Therefore we propose an UMHexagonS algorithm for integer pel search and a CBFPS (Center biased Fractional Pel Search) for fractional pel search in this proposal, which are the modified versions of the proposal in [12][13][14].

Different sequences from QCIF format to HD (High Definition) format with high motion degree are tested and our proposed fast integer pel and fractional pel motion estimation algorithm shows very good capability in keeping the rate distortion performance (largest distortion less than 0.1dB) and a great computation reduction can be achieved. And our algorithm is also simple with constant pattern and no need for additional memory, which is suitable for hardware implementation.

In section 2, UMHexagonS algorithm for integer pel motion estimation is described. In section 3, CBFPS algorithm for fractional pel motion estimation is given, and a half search stop techniques based on the detection of zero blocks is described in section 4, the experiment results are given in section 5 and the conclusion is shown in the last section.

2. Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) algorithm for integer pel search

The proposed UMHexagonS algorithm uses the hybrid and hierarchical motion search strategies. It's hybrid because it includes four steps with different kinds of search pattern:1) Predictor selection and prediction mode reordering; 2) Unsymmetrical-cross search; 3) Uneven multi-hexagon-grid search; 4) Extended hexagon based search; Fig.1 demonstrates a typical search procedure in a search window with search range equals 16 (it's assumed the start search point is (0,0) vector here).

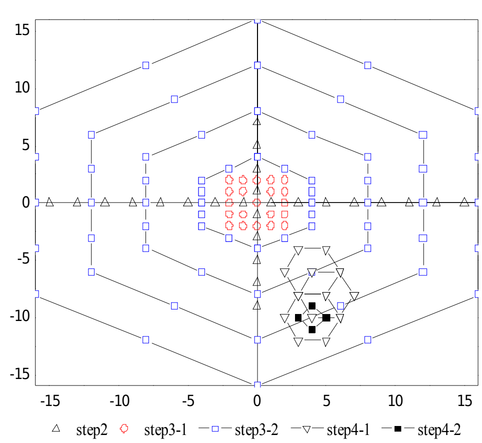


Fig.1 Search process of HUCMHGS algorithm, W=16

With the second and third step, the motion estimation accuracy can be nearly as high as that of FS. In order to reduce the computation load even more.

In the following discussion, the search range is supposed to be W.

2.1 Starting search point prediction

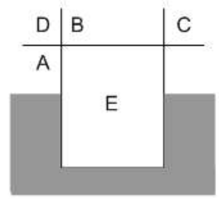


Fig.2 Reference block location for prediction of motion vectors

As Fig.2 shows, median predictor is used in median prediction of motion vectors, the median value of the adjacent blocks on the left, top, and top-right (or top-left) of the current block is used to predict the motion vector of the current block (as Fig.2 shows):

$$pred_{mv} = median(Mv_A, Mv_B, Mv_C) \qquad (1)$$

The predicted motion vector is (Pred_x, Pred_y). Some rules specify the predicted motion vector value has been defined in [2]: when block A lies outside the picture or GOB(Group of blocks) boundary, it is replaced by (0,0), when block C lies outside the picture or GOB boundary, it is replaced by motion vector of block D, when two blocks B and C lie outside, however, they are replaced by the motion vector of the third block.

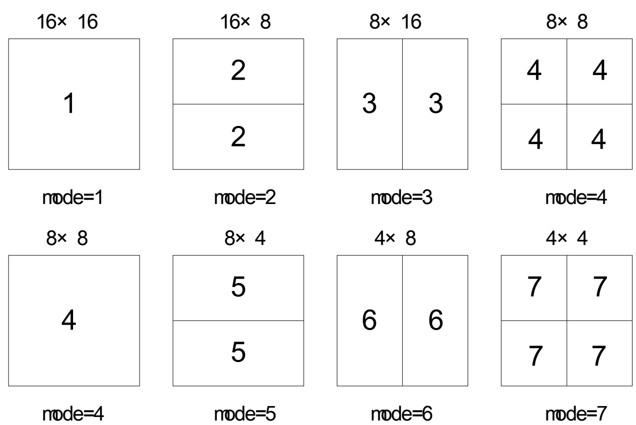


Fig3 Seven Prediction Modes in H.264

As Fig,3 shows, there are seven inter prediction modes defined in H.264. 8x8 modes(mode 4, 5, 6, 7) are first searched followed by 16x16 modes(mode 3, 2, 1) in current reference software[9]. Because such strategy is not beneficial in utilizing the motion relationship between different modes. Therefore the search order of the prediction modes here is changed according to the size of the block mode, a hierarchically search order from mode 1 to 7 is chosen as our prediction mode search order and the motion vector of the up layer block (for example, mode 5 or 6 is the up layer of mode 7, and mode 4 is the up layer of mode 5 or 6, etc.) is used as one of the prediction candidates of lower layer. Therefore for mode 1, the median prediction, the (0, 0) vector and the motion vectors of adjacent blocks on the left, top, and top-right are chosen as the prediction candidates; and for other modes, the median prediction, the (0, 0) vector and motion vector of the up layer are chosen as the vector prediction candidates.

The prediction with the minimum cost among these prediction candidates will be chosen as the starting search position of the next search step.

2.2 Unsymmetrical-cross search

Based on a common agreed conclusion that the movement in the horizontal direction is much heavier than that in the vertical direction for natural picture sequences, the optimum motion vector can be nearly accurately predicted by an unsymmetrical-cross search. As Fig.1 step2 indicates, an unsymmetrical-cross with the horizontal search range equals W and vertical search range equals W/2 is used. The unsymmetrical-cross search can be seen as a simple but efficient prediction method to give an accurate staring search point for the nest step. The distance between search points is chosen to be 2 in our strategy.

Of course for some special sequences with heavier vertical motion, the vertical search range can expanded to W.

The motion vector with the minimum cost will be chosen as search center of next search step.

2.3 Uneven Multi-Hexagon-Grid Search

There are two sub-steps in this search step: first a full search with search range equals 2 is carried out around the search center, as Fig.1 step3-1 shows. And then a Multi-Hexagon-grid search strategy is taken, as Fig.1 step3-2 shows. This search strategy is based on the consideration that the unsymmetrical-cross search will give an accurate starting search point and the uneven Multi-Hexagon-grid is used to deal with large and irregular motion cases. Considering the fact that horizontal motion is much heavier that vertical motion for natural video, a Sixteen Points Hexagon Pattern (16-HP) is used as our basic search pattern, just as Fig.4 shows, there are more search points in horizontal direction than in vertical direction in this pattern. And then the uneven Multi-Hexagon-grid is constructed by extend 16-HP with different scale factors (from 1 to W/4 in our proposed algorithm) and the search process starts from the inner hexagon to the outer hexagon.

The best motion vector derived in this step will be chosen as search center of the next search step.

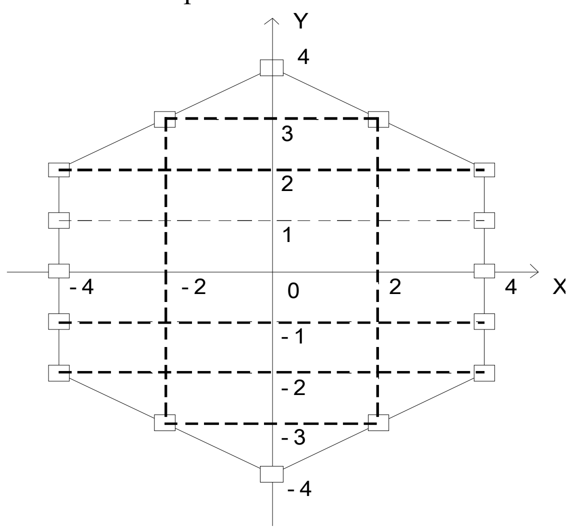


Fig.4 Sixteen Points Hexagon Pattern (16-HP) used in UMHexagonS

The best motion vector derived in this step will be chosen as search center of next search strategy

2.4 Extended Hexagon-based Search (EHS)

The multi-grid search may obtain optimum motion vectors with different accuracy according to the distance from the search window center to it. When the optimum motion vector in the previous step locates in the outer concentric area, the search result has relatively low accuracy and motion vector refinement by some center biased search method is adopted. And for small prediction mode (such as mode 7), the search strategy can go directly to this step by skipping unsymmetrical-cross search and uneven Multi-Hexagon-grid search, because the predicted motion vector for small prediction mode is accurate enough for small prediction mode.

It's proved that HEXBS can find a same motion vector with fewer search points than DS (Diamond Search) algorithm[6], therefore we use an EHS algorithm derived from HEXBS as our center biased search algorithm. The only difference between EHS and HEXBS is that when switching the search pattern from a larger to a smaller size of hexagon, the search will continue until the minimum block distortion (MBD) point is the center of the newly formed hexagon(as Fig.1 step-4 shows).

3. **Center biased Fractional Pel Search algorithm for fractional pel search**

We have proposed a Paraboloid Prediction based Fractional Pel Search (PPFPS) strategy [10] which combines paraboloid prediction based half pel search with a directional refinement algorithm to estimate the fractional pel motion vector. But implementation of this algorithm is constrained that the search model of the integer pixel motion estimation is diamond shape and the cost value of the four diamond vertex around the best integer pel position should all be available before fractional pel motion estimation [10].

In this proposal we will show that unimodal error surface assumption does hold true in most cases for the fractional pel error surface so that a Center Biased Fractional Pel Search (CBFPS) strategy is proposed.

The CBFPS algorithm is much faster than the algorithm provided in JVT and also faster than PPFPS especially in 1/8-pel case. There is no constraint on integer pel motion estimation.

3.1 Background

3.1.1 Hierarchical Fractional Pel Search algorithm in JVT

For fractional pel motion estimation, the undertaking is conducted within a so called fractional pel search window which is an area bounded by eight neighbor integer pels positions around the best integer pel position. As Fig.5 shows, for 1/4-pel case, the search range is three 1/4-pel units, and for 1/8-pel case, the search range is seven 1/8-pel units. In generation of these fractional pel positions, a 6 tap filter is used to produce the 1/2-pel positions, 1/4-pel positions is produced by linear interpolation; an 8-tap filters is used in providing 1/8-pel positions[1].

Fig.5 shows the typical Hierarchical Fractional Pel Search algorithm provided in JM test model. The HFPS is described by the following 4 steps:

Step 1. Check the eight 1/2-pel positions (X points) around the best integer pel position in order to find the best 1/2-pel motion vector;

Step 2. Check the eight 1/4-pel positions (Hexagon points) around the best 1/2-pel position in order to find the best 1/4-pel motion vector;

Step 3. If 1/8-pel motion vector accuracy is chosen, continue checking the eight 1/8-pel positions (Triangle points) around the best 1/4-pel position in order to find the best 1/8-pel motion vector;

Step 4. Select the motion vector and block-size pattern, which produces the lowest rate-distortion cost.

Therefore separately 16 or 24 fractional search points need to be checked to determine a 1/4-pel or 1/8-pel accuracy motion vector. Further more if Hadamard transform and ABT are used, the complexity of fractional pel motion estimation increases greatly. Thus the computation load of fractional pel motion estimation is becoming the bottleneck of improving the speed of motion estimation with the development of fast integer motion estimation.

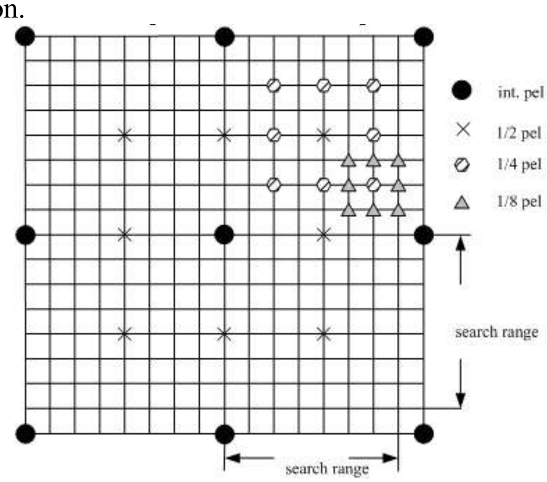


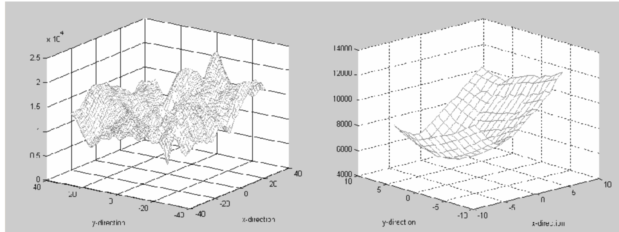
Fig.5 Hierarchical Fractional Pel Search algorithm

3.1.2 Possibility of faster search algorithm

For some fast integer pel motion estimation, they are all based on the assumptions that the matching error surface inside the search window is unimodal so that the matching error decreases monotonically as the searched point moves closer to the global optimum. However, this assumption is far away from the reality because of the complexity of video content, and the search would easily be trapped into a local minimum, just as Fig.6(a) shows.

It's easy to understand that the correlation inside a fractional pel search window is much higher than that of the integer pel search window, so unimodal error surface assumption does hold true in most cases for the fractional pel motion search[11] and our experiment also validates this. Fig.6(b) shows a typical matching error surface in a 1/8-pel motion search window(search range is 7) around the best integer pel position, which is a typical smooth unimodal surface.

Therefore based on the assumption of unimodal for the fractional pel error surface, a Center Biased Fractional Pel Search strategy is proposed, which is much faster than the HFPS algorithm and keeps the rate distortion performance at the same time.



(a) (b)

Fig.6 (a) Error surface of integer pel motion estimation (search range = 32); (b) Error surface of fractional pel motion estimation (1/8-pel case)

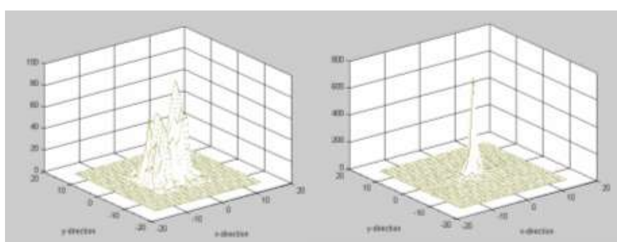
3.2 Center Biased Fractional Pel Search (CBFPS) algorithm

3.2.1 Prediction of Fractional Pel Motion Vector:

We proposed a Center Biased Fractional Pel Search (CBFPS) algorithm for fast fractional pel motion estimation. Using formula (1), $pred_{mv}$ here is defined as the fractional pel unit, so it includes the information of predicted integer pel motion vector and predicted fractional pel motion vector. Therefore we could extract the predicted fractional pel motion vector by using this formula:

$$frac_{pred_{mv}} = (pred_{mv} - mv)\beta \qquad (2)$$

where mv is the integer pel motion vector of block E, and here mv is also in fractional pel unit, % is the mode operation, $\beta=4$ in 1/4-pel case and $\beta=8$ in 1/8-pel case. Fig.7 validates the strong spatial correlation between fractional pel motion vectors. Fig.7(a) shows the fractional pel motion vector distribution using the FFPS algorithm versus the (0,0) motion vector, and Fig.7(b) shows the fractional pel motion vector distribution using the FFPS algorithm versus the median predictor motion vector. It's clear that the difference between the predicted motion vectors versus the vectors generated by FFPS congregate around the (0,0) point, therefore it's much easier to find the motion vector around the predicted motion vector.



(a) (b)

Fig.7 (a) Fractional pel motion vector distribution using the Full Search algorithm versus the (0,0) MV (b) Fractional pel motion vector distribution using the Full Search algorithm versus the median predictor.

3.2.2 Description of CBFPS algorithm

Diamond search pattern has been proved to be simple and efficient in many fast integer pel motion estimation algorithms. Here we also choose the diamond search pattern in our CBFPS algorithm.

Fig.8 illustrates the implementation of CBFPS algorithm, the whole algorithm can be described in following steps:

Step 1. Predict the motion vector of the current block by equation (1) and (2), the predicted motion vector is (Pred_x, Pred_y);

Step 2. Cost of the original search center (0, 0) and (Pred_x, Pred_y) are compared. The point with the minimum matching error is chosen as the search center;

Step 3. If the MBD (Minimum Block Distortion) point is located at the center, go to step 4; otherwise choose the MBD point in this step as the center of next search, then iterate this search step;

Step 4. Choose the MBD point as the motion vector.

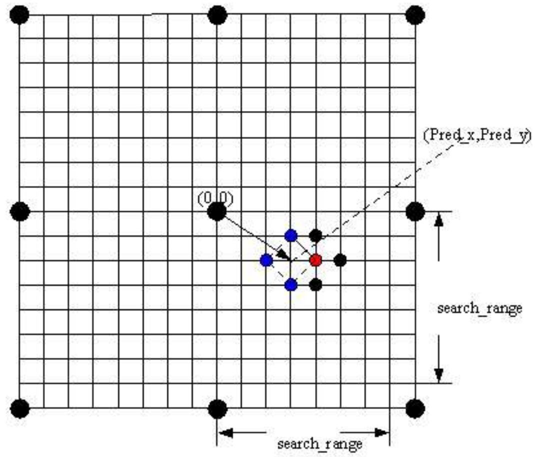


Fig.8 Implementation of CBFPS algorithm

Further improvement can also be achieved by using more accurate prediction and half-stop techniques in our proposed CBFPS algorithm.

4. **Early Termination by Detection of Zero Blocks**

We propose an early termination technique based on early detection of zero blocks. Here we mean zero blocks as those blocks whose transform coefficients will all be zero after quantization. If this kind of block is searched in motion estimation, then there is no need to search more accurate matching block because there should be no gain in coding efficiency. And for zero blocks, transform and quantization steps can be eliminated for speed up.

4x4 integer transform is used in JVT, is based on the DCT but with some fundamental differences [1]:

It is an integer transform (all operations can be carried out with integer arithmetic, without loss of accuracy);

The inverse transform is fully specified in the H.264 standard and if this specification is followed correctly, mismatch between encoders and decoders should not occur.

The core part of the transform is multiply-free, i.e. it only requires additions and shifts.

A scaling multiplication (part of the complete transform) is integrated into the quantizer (reducing the total number of multiplications).

The entire process of transform and quantization can be carried out using 16-bit integer arithmetic and only a single multiply per coefficient, without any loss of accuracy.

4x4 DCT transform of an input array X is given by:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \left[\quad \mathbf{X} \quad \right] \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \qquad (3)$$

where $a = 1/2$, $b = \sqrt{\frac{1}{2}}\cos(\frac{\pi}{8})$, $c = \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8})$.

Then the matrix multiplication can be factorized to the following equivalent form:

$$\mathbf{Y} = \left(\mathbf{C}\mathbf{X}\mathbf{C}^T\right) \otimes \mathbf{E} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \left[\quad \mathbf{X} \quad \right] \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \qquad (4)$$

CXC^T is a "core" 2-D transform. **E** is a matrix of scaling factors and the symbol $\otimes$ indicates that each element of CXC^T is multiplied by the scaling factor in the same position in matrix **E** (scalar multiplication rather than matrix multiplication). The constants *a* and *b* are as before; *d* is *c/b* (approximately 0.414). To simplify the implementation of the transform, *d* is approximated by 0.5. To ensure that the transform remains orthogonal, *b* also needs to be modified so that: $a = 1/2$, $b = \sqrt{\frac{2}{5}}$, $d = 1/2$.

The 2nd and 4th rows of matrix **C** and the 2nd and 4th columns of matrix C^T are scaled by a factor of 2 and the post-scaling matrix **E** is scaled down to compensate. (This avoids multiplications by ½ in the "core" transform CXC^T which would result in loss of accuracy using integer arithmetic). The final forward transform becomes:

$$\mathbf{Y} = \mathbf{C}_f \mathbf{X} \mathbf{C}_f^T \otimes \mathbf{E}_f = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \left[\quad \mathbf{X} \quad \right] \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \qquad (5)$$

H.264 uses a scalar quantizer. The definition and implementation are complicated by the requirements to avoid division and/or floating point arithmetic and incorporate the post- and pre-scaling matrices.

The basic forward quantizer operation is as follows:

$$Z_{ij} = round(Y_{ij} / QpStep) \qquad (6)$$

where Y_{ij} is a coefficient of the transform described above, Qstep is a quantizer step size and Z_{ij} is a quantized coefficient. A total of 52 values of Qstep are supported by the standard and these are indexed by a Quantization Parameter, QP. It should be noted that Qstep doubles in size for every increment of 6 in QP; Qstep increases by 12.5% for each increment of 1 in QP. The wide range of quantizer step sizes makes it possible for an encoder to accurately and flexibly control the trade-off between bit rate and quality.

It can be assumed that if the DC value of transform coefficient is zero, then all other AC coefficients are zero. For a 4x4 block, the quantized DC value is:

$$DC = \sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y) / ((2^{qbits} - f) / QE[q_{rem}][0][0]) \qquad (7)$$

While $\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y) < \sum_{x=0}^{3}\sum_{y=0}^{3} |f(x,y)|$, where $\sum_{x=0}^{3}\sum_{y=0}^{3} |f(x,y)|$ is the Sum of Absolute Difference (SAD) of the 4x4 block. Therefore if (8) is satisfied, the quantized DC value would be zero:

$$\sum_{x=0}^{3}\sum_{y=0}^{3} |f(x,y)| < ((2^{qbits} - f) / QE[q_{rem}][0][0]) \qquad (8)$$

Where $qbits = 15 + QP/6$, and $q_{rem} = QP\%6$, $f = (1 << qbits)/6$, QE is the defined quantization coefficient table and QP is the input quantization parameter.

Thus the threshold for 4x4 block (prediction mode 7) T_7 can be defined as :

$$T_7 = (2^{qbits} - f) / QE[q_{rem}][0][0] < TH_7 \, ? (2^{qbits} - f) / QE[q_{rem}][0][0] : TH_7 \qquad (9)$$

If the SAD of a 4x4 block is small than T_7, it can be assumed to be a zero block and then the motion estimation process can stop if it search a block with SAD smaller than T_7.

For other types of prediction modes (mode 1,2,…,6), the corresponded threshold can be defined as:

$$T_i = T_7 * \beta_i < TH_i \, ? T_7 * \beta_i : TH_i \qquad i = 1, \ldots, 6 \qquad (10)$$

where β_i is the scale factor to get the threshold for larger prediction mode from 4x4 prediction mode and TH_i is used as the maximum threshold to reduce the possibility of erroneous and inadequate early termination for different prediction modes.

5. Experiments results

Our proposed UMHexagonS algorithm (combined with CBFPS and early termination technique) was integrated within version 5.0 of the JVT software [9]. Apart from the test conditions specified as common condition in [15], we have also added some additional more difficult sequences of heavier motions to demonstrate the efficacy of our implementation. Average results computation defined in [16] are also included in our comparison. And these results were generated using an excel equivalent in our results due to its convenience and the result is very similar to that of [16].

Timing statistics generated from the encoder is used, and an accurate timer function QueryPerformanceCounter() is used for calculating part running time of integer pel motion estimation and fractional pel motion estimation. Our experiments show that this calculation is stable and accurate enough for timing statistics collection.

Our experiments results are all in JVT-F017.xls and a short description is given here.

Main parameters used in our experiments:

Hadamard transform is used, searchRange is set as 32 and OutFile Mode is 0, RD Optimization is used.

And for each test sequences, reference frame number equals 1 or 5 are tested separately and entropy coding equals CAVLC or CABAC are also tested separately. In

For the performance of the UMHexagonS scheme compared the Fast Full Search scheme in JM5.0, in the following table, "Total" means the time saving in total time, "Integer" means time saving in integer pel motion estimation part, and "Fractional " means timesaving in fractional pel motion estimation part. "Avg.bits" represents the average bit saving and "Avg.PSNR" represents the average PSNR loss.

5.1 Experiment results on sequences defined in common condition:

Tabel 1. Results of sequences in common condition

		Total	Integer	Fractional	Avg. bits	Avg. PSNR(dB)
Ref =1	CAVLC	46.81%	91.23%	33.35%	+0.8%	-0.04
	CABAC	46.43%	90.77%	33.54%	+0.65%	-0.031
Ref =5	CAVLC	70.26%	90.30%	33.06%	+0.42%	-0.023
	CABAC	70.03%	90.09%	32.64%	+0.69%	-0.037

5.2 Experiment results on sequences with high motion:

We choose Stefan and Bus sequence as the typical high motion sequences in this test.

Tabel 2. Results of sequences with high motion

		Total	Integer	Fractional	Avg. bits	Avg. PSNR(dB)
Ref =1	CAVLC	41.83%	87.90%	29.48%	+0.65%	-0.031
	CABAC	42.23%	87.36%	30.65%	+0.65%	-0.033
Ref =5	CAVLC	65.78%	86.21%	26.89%	+0.38%	-0.019
	CABAC	66.55%	86.18%	30.45%	+0.51%	-0.025

5.3 Experiment results on B frames:

We choose Stefan and Bus sequence as the test sequence and 2 B frames are inserted between two P frames or I frame and P frame. Here in our experiment only the first frame is I frame.

Tabel 3. Results on B frame

		Total	Integer	Fractional	Avg. bits	Avg. PSNR(dB)
Ref =1	CAVLC	54.06%	87.86%	31.41%	-1.42%	+0.076
	CABAC	53.85%	87.77%	31.94%	-0.40%	+0.023
Ref =5	CAVLC	67.09%	85.84%	30.30%	-0.78%	+0.044
	CABAC	67.48%	85.84%	31.37%	-0.25%	+0.016

It's showed there is even an average PSNR increment by using our proposed UMHexagonS algorithm. This is because of a gain up to 0.190dB can be achieved by using UMHexagonS for Stefan sequence in this case.

5.4 Experiment results on HD sequence:

Jet is a HD sequence with group of jets taking off for flight show [17] and we cut the 100 frames (from 450f to 550f) with the most violent motion from the whole sequence to verify the algorithm. Our UMHexagonS is integrated to the JM5.0 with rate control and the bit rate is set to 6Mbits/s and the search range is set to 48 to test UMHexagonS under larger search range case. Experiment result shows that a PSNR drop of 0.02dB and 4.1 times speed up in total encoding time can be achieved by comparing the UMHexagonS scheme with the Fast Full Search scheme in JM5.0.

6. Conclusion

In this proposal, we propose a new fast motion estimation algorithm called UMHexagonS for fast integer pel motion estimation in H.264. UMHexagonS is a hierarchical and composed of four main steps. Firstly, the starting search point prediction utilizes the motion vector relationship in spatial domain and between different prediction modes. Secondly an unsymmetrical-cross search is performed to give an initial search point. After that an uneven multi-hexagon-grid search is used to keep search from dropping into local minimum. At last an extended hexagon based search is performed for the motion vector refinement. And Under the assumption of unimodal error surface inside the fractional pel search, a Center Biased Fractional Pel Search (CBFPS) strategy with a diamond search pattern is also proposed to combine with UMHexagonS algorithm. And an early termination method based on detection of zero blocks is also first proposed for H.264 in out paper for speed up.

Different sequences from QCIF format to HD (High Definition) format with high motion degree are tested and our proposed fast integer pel and fractional pel motion estimation algorithm shows very good capability in keeping the rate distortion performance (largest distortion less than 0.1dB) and a great computation reduction can be achieved.

Further improvement can be achieved by adaptively choose different step and parameters inside UMHexagonS according to different motion statistics, different prediction mode, and different reference frame number.

7. Reference

[1] "Editor's Proposed Draft Text Modifications for Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Draft 2", JVT-E022d2, Geneva, Switzerland, 9-17 October, 2002

[2] T. Koga, K. Iinuma, A. Hirano, Y. Lijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proceedings Nat. Telecommunications Conf. 81, New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.

[3] J.Jain, A.Jain, "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, vol.COM-29, pp.1799-1806, Dec. 1981.

[4] L.Liu, E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Transactions on CSVT, vol.6, no.4, pp.419-422, June 1996.

[5] L.M.Po, W.C.Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Transactions on CSVT, vol.6, no.3, pp.313-317, June 1996.

[6] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Patten for Fast Block Motion Estimation", IEEE Trans. on CSVT, pp.349-355, Vol.12, No.5, May, 2002.

[7]Y.Y.Chung, M.T.Wong, N.W.Bergmann, "Fast search block-matching motion estimation algorithm using FPGA," Proceedings of SPIE, VCIP'2000, vol.4067, pp.913-921, June 2000

[8] Danian Gong, Yun He,"Fast motion estimation algorithm using unequal-distributed multi-grid search strategy", Proceeding of PCS2001, pp. 362-365, Seoul, April, 2001

[9] JM5.0, Reference Software of H.264, ftp://ftp.imtc-files.org

[10] Zhibo Chen, Cheng Du, JinghuaWang, Yun He, "PPFPS- A Paraboloid Prediction based Fractional Pel Search Strategy for H.26L", IEEE ISCAS 2002, USA

[11] Cheng Du, Yun He, Junli Zheng, "A paraboloid prediction based fast half pixel motion estimation" Journal of Tsinghua University- Science and Technology, Vol.41(1), pp.1-4, Jan. 2001

[12] Zhibo Chen, Yun He, etc. "Fast Integer and Fractional Pel Motion estimation", JVT-E045.doc, 5th Meeting: Geneva, Switzerland, 9-17 October, 2002

[13] Zhibo Chen, Yun He, etc. "Prediction based Directional Refinement (PDR) algorithm for Fractional Pixel Motion Search Strategy", JVT-D069.doc, 4th Meeting: Klagenfurt, Austria, 22-26 July, 2002

[14] Zhibo Chen, Yun He,etc, "A novel fast fractional-pel motion estimation", 3rd Meeting: Fairfax, Virginia, USA, 6-10 May, 2002

[15]G. Sullivan, "Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material", document VCEG-N81, Sep. 2001.

[16]G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves", document VCEG-M33, Mar. 2001

[17]"Description of new High Definition Video Test Sequences", ISO/IEC JTC1/SC29/WG11 MPEG02/ N5125 October 2002, Shanghai

(Append for Proposal Documents)

JVT Patent Disclosure Form

International Telecommunication Union
Telecommunication Standardization Sector

International Organization for Standardization

International Electrotechnical Commission

Joint Video Coding Experts Group - *Patent Disclosure Form*

(Typically one per contribution and one per Standard | Recommendation)

Please send to:

JVT Rapporteur Gary Sullivan, Microsoft Corp., One Microsoft Way, Bldg. 9, Redmond WA 98052-6399, USA
Email (preferred): Gary.Sullivan@itu.int Fax: +1 425 706 7329 (+1 425 70MSFAX)

This form provides the ITU-T | ISO/IEC Joint Video Coding Experts Group (JVT) with information about the patent status of techniques used in or proposed for incorporation in a Recommendation | Standard. JVT requires that all technical contributions be accompanied with this form. *Anyone* with knowledge of any patent affecting the use of JVT work, of their own or of any other entity ("third parties"), is strongly encouraged to submit this form as well.

This information will be maintained in a "living list" by JVT during the progress of their work, on a best effort basis. If a given technical proposal is not incorporated in a Recommendation | Standard, the relevant patent information will be removed from the "living list". The intent is that the JVT experts should know in advance of any patent issues with particular proposals or techniques, so that these may be addressed well before final approval.

This is not a binding legal document; it is provided to JVT for information only, on a best effort, good faith basis. Please submit corrected or updated forms if your knowledge or situation changes.

This form is *not* a substitute for the *ITU ISO IEC Patent Statement and Licensing Declaration*, which should be submitted by Patent Holders to the ITU TSB Director and ISO Secretary General before final approval.

Submitting Organization or Person:	
Organization name	**Tsinghua University, China and SVA Group, China**
Mailing address	**100084 425# East mail building, Tsinghua University, Beijing, China**
Country	**China**
Contact person	**Zhibo Chen, Yun He**
Telephone	+86-10-62781413
Fax	+86-10-62770317
Email	chenzb@video.mdc.tsinghua.edu.cn, hey@tsinghua.edu.cn
Place and date of submission	11/30/2002
Relevant Recommendation | Standard and, if applicable, Contribution:	
Name (ex: "JVT")	**JVT**
Title	**Fast Integer Pel and Fractional Pel Motion Estimation for JVT**
Contribution number	**JVT-F017**

(Form continues on next page)

Disclosure information – Submitting Organization/Person (choose one box)

☐ 2.0 The submitter is not aware of having any granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

or,

The submitter (Patent Holder) has granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution. In which case,

☐ 2.1 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a free license to an unrestricted number of applicants on a worldwide, non-discriminatory basis to manufacture, use and/or sell implementations of the above Recommendation | Standard.

☐ 2.2 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a license to an unrestricted number of applicants on a worldwide, non-discriminatory basis and on reasonable terms and conditions to manufacture, use and/ or sell implementations of the above Recommendation | Standard.

Such negotiations are left to the parties concerned and are performed outside the ITU | ISO/IEC.

☒ 2.2.1 The same as box 2.2 above, but in addition the Patent Holder is prepared to grant a "royalty-free" license to anyone on condition that all other patent holders do the same.

☐ 2.3 The Patent Holder is unwilling to grant licenses according to the provisions of either 2.1, 2.2, or 2.2.1 above. In this case, the following information must be provided as part of this declaration:

- patent registration/application number;
- an indication of which portions of the Recommendation | Standard are affected.
- a description of the patent claims covering the Recommendation | Standard;

In the case of any box ***other than 2.0*** *above, please provide the following:*

Patent number(s)/status ____________________

Inventor(s)/Assignee(s) ____________________

Relevance to JVT ____________________

Any other remarks: ____________________

(please provide attachments if more space is needed)

(form continues on next page)