Simulated Annealing for Fast Motion Estimation Algorithm in H.264/AVC

Zhiru Shi, W.A.C. Fernando and A. Kondoz

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50974

1. Introduction

The promising video coding standard, H.264/AVC [1], is developed by the Join Video Team of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). By utilizing several new techniques, such as advanced intra predictions, variable block size ME, integer transformation, in-loop deblocking filter, H.264/AVC has achieved significant compression gain compared with previous video coding standards. It is now widely applied to many types of visual services, for example Digital Multimedia Broadcasting, Mobile Phone, and High Definition (HD) video delivery. In the near future, holography video and Super-HD video are expected to hit consumer market. These kinds of large sized video contents require higher coding efficiency while keeping the encoder complexity within an acceptable level. Therefore, new techniques are needed to reduce the computational complexity so that various real time video encoder and delivery services for the large sized video contents could be feasible.

In particular, Block-Matching Motion Estimation (BMME) with Full Search (FS) algorithm [2] is the main computational burden in H.264/AVC due to exhaustively search all possible blocks within the search window using Lagrangian multiplier. Although FS algorithm can obtain the optimum motion vector (MV) in most cases, it consumes more than 80% of the total computational complexity. Thus, a fast and efficient motion estimation algorithm is required for H.264/AVC. Recently, two major approaches were researched to overcome this problem. One employs fast mode decision algorithms to skip unnecessary block modes in variable block checking process [3, 4]. The other one utilizes Fast Motion Estimation (FME) searching algorithms to reduce unnecessary search points [5-11].

Various algorithms have been proposed to reduce search points for FME Search algorithm. Motion adaptive search (MAS) [5] utilized the motion activity information to adjust the search strategy. In Variable Step Search (VSS) algorithm [6], motion search range is determined by using the degree of correlation between neighbouring motion vectors. A Multi-Path Search (MPS) algorithm [7] has been proposed, in which all the eight neighbours around the origin of the search window were performed to find candidate points. This algorithm has good rate-distortion performance, but its computational complexity reduction is limited. To tackle this drawback, the directional gradient descent search (DGDS) algorithm [8] is developed. It searched on the error surface in eight directions by using directional gradient descent. The search patterns in each stage depend on the minima found in eight directions and thus the global minimum can be traced more efficiently.

The hybrid multi-hexagon-grid search (UMHexagonS) algorithm [9] was adopted in H.264/AVC reference software JM as its significant reduce the computational complexity with only little degradation in rate-distortion performance. UMHexagonS takes advantage of four kinds MV predictions to decided initial search point, i.e. the Median Prediction (MP), the Uplayer Prediction (UP), the Corresponding-block Prediction (CP) and the Neighbouring Reference-picture Prediction (NRP). After selecting the best initial point, it employs the unsymmetrical-cross search pattern and uneven-hexagon-grid search pattern, which are shown in Figure 1 as step2 and setp3-2. In these uneven search patterns, the number of horizontal search points is more than that of vertical points. This is mainly based on a common assumption, that the movement in the horizontal direction is higher than that in vertical direction. However, motion characteristic in each video sequence is unique. Also, the characteristic may change with the time. Therefore, with this horizontal-heavy pattern, UMHexagonS would lose accuracy and waste searching power.

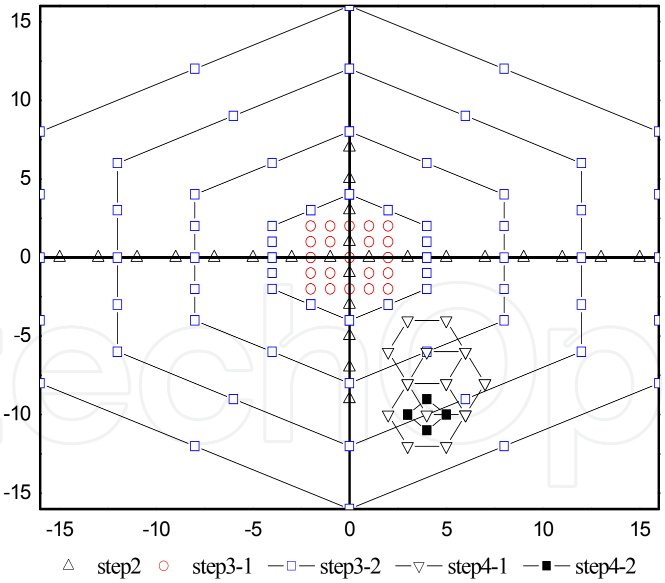


Figure 1. Search process of UMHexagonS algorithm

Predictive Intensive Direction searching (PIDS) algorithm was proposed in [10] to solve the problem caused by uneven search patterns by using a adaptive searching pattern. In PIDS algorithm, the correlation between predicted motion vector and optimal motion vector was investigated. The study revealed that the probability of predicted and optimum motion vector existing in the same directional region is at least 75%. Based on this statement, PIDS algorithm exploited the predicted MV direction to decide the intensive search direction. Thus, the intensive-direction search and coarse-direction search are selected adaptively for different regions. One example of search process of PIDS is depicted in Figure 2. As the uneven search pattern is changing according to the predicted motion vector of each block, it performs more precisely than UMHexagonS and achieves more computation reduction.

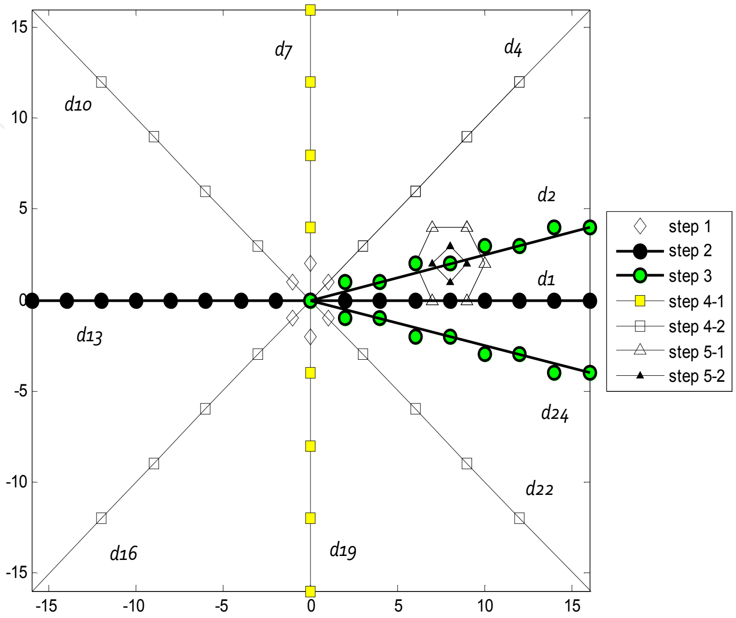


Figure 2. Search pattern of PIDS, example of intensive search in d_1

However, the PIDS algorithm's adaptive intensive search selection is limited in directional regions. With fixed number of search points in each direction, it cannot adjust the search range for different motion scenes. In study [11], a statistic analysis of MV distribution was carried out. A large number of global minima occupy near the search centre especially at the zero MV (0, 0) with a certain percentage of optimal MVs outside the radius of 10 pels. It indicated that most predicted and optimal MVs have high locality correlation. Meanwhile, some irregular MVs can hardly be well predicted due to poor correlation. In this chapter, direction and distance correlation between predicted MV and optimal MV are investigated MV correlation statistics information is calculated for each frame as its motion characteristic. With this information, the intensive and coarse search regions are adaptively changed for each block. The Simulated Annealing concept [12, 13] is employed to control searching process and to adaptively choose the intensive search region. After this Introduction the chapter is organized into five more sections as follows.

Section 2 statistical analyse MV direction and distance correlation characteristic. The block-matching motion estimation is described in this section. Section 3 gives an overview of simulated annealing and simulated quenching algorithm. Based upon analyses, the proposed SAAS algorithm is presented in section 4. The experimental results are given and illustrated in section 5. Finally, section 6 draws the final conclusion.

2. Statistical analysis of MV correlation characteristic

Because of the consistency of object and the consistency of motion, MVs have high correlation in both spatial and temporal domains. Thus MV prediction technique is adopted in H.264/AVC to improve ME efficiency. A predicted motion vector, mv_{pred}, is generated by previously coded neighbourhood motion vectors and MVD, the difference between the current vector and the predicted vector, is encoded and transmitted.

For the regions with smooth motion of a moving background and uniform motion of rigid objects, there normally exist very high correlations between predicted and optimal motion vectors. So that the BMME search algorithm only need to check a few points to obtain optimal position. While for the poor motion vector correlation scenario, like the complex and irregular motion, more candidate points are needed to be checked. Therefore, MV correlation characteristic will affect the searching strategy chosen in the BMME algorithm. In order to adaptively select an appropriate search pattern, MV correlation is statistical analysed in this section.

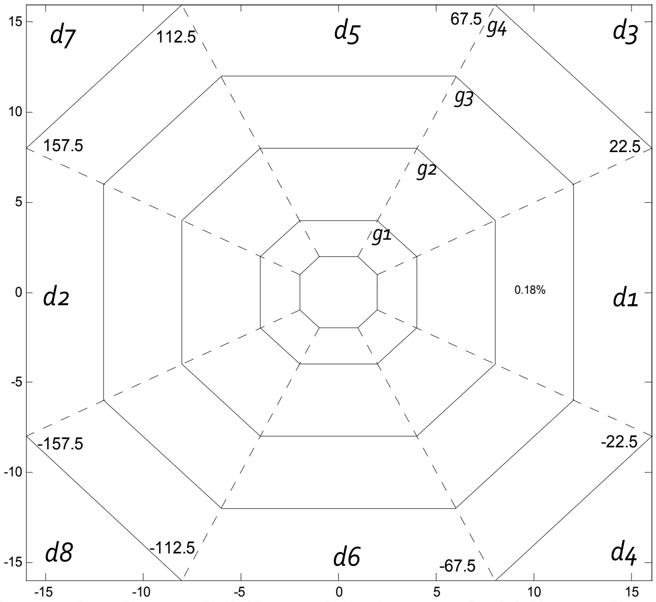


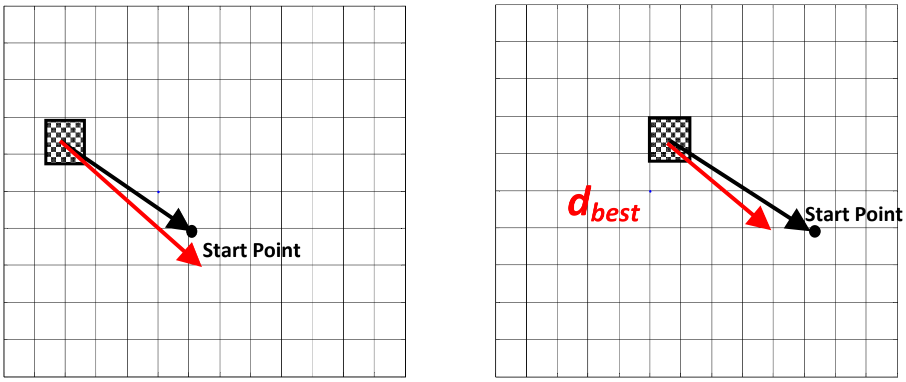
Figure 3. Direction and distance classification of MV correlation $p\left(MVC_{d_i,g_j}\right)$

To sufficient describe MV correlation characteristic, the MV correlation statistics are calculated in two aspects: motion vector directional correlation statistic and motion vector distance correlation statistic. Combining these two correlations together, the search window is divided into 8 direction regions d_i and a group of octagon grids g_j, as illustrated in Figure 3. Normally, the motion content of each video sequence is unique and the scene is changing with time. MVs correlation characteristic is changing with sequences and time. In this case, the analysis of MV correlation is frame based to improve the accuracy. The predicted and optimal motion vectors of previous frame are utilized for current coding frame.

MV directional difference d_{MVD} describes the directional similarity between predicted MV direction d_{pred} and optimal MV direction d_{best}, and is measured in degrees as follows

$$d_{MVD} = d_{best} - d_{pred} \tag{1}$$

where, $d_{MVD} \in [-180^{\circ}, 180^{\circ}]$ is classified into 8 classes, with boundary of $\pm 22.5°$, $\pm 67.5°$, $\pm 112.5°$ and $\pm 157.5°$ as illustrated in Figure 3. The statistical calculation is carried out by exploding the MV directional distribution in these classes. Then the distribution probabilities $p(MVC_{d_i})$ of MV directional correlation are obtained. If MV directional correlation is high, d_{best} locates in forward or backward of d_{pred}, as shown in Figure 4. Class $d_1 \in (-22.5°, 22.5°)$ and Class $d_2 \in (-180°, -157.5°) \cup (157.5°, 180°)$ indicate the forward and backward direction and normally have higher probabilities than other classes.



(a) Forward d_{pred} **(b) Backward** d_{pred}

Figure 4. Two situation of d_{best} locating along d_{pred}

MV distance correlation is measured by the distance between global minimum point and searching centre, which is known as the motion vector difference (MVD). Several Octagon grids g_j are utilized to categorise the MV distance correlation, as such circle-approximated pattern is more accurate to describe the MVD distribution. The interval between neighbour octagon grids is 4 pels, which is shown in figure 3. To evaluate the characteristic of MV distance correlation, MV distance correlation probabilities $p(MVC_{g_j})$ are calculated. Considering both directional and distance, MV correlation probabilities $p(MVC_{d_i, g_j})$ for the current coding frame are defined as:

$$p\left(MVC_{d_i, g_j}\right) = p\left(MVC_{d_i}\right) \times p\left(MVC_{g_j}\right) \tag{2}$$

where $d_i \in (1,8)$ is the directional classes and $g_j \in (1, sr/4)$ is the octagon grids within search range sr. The MV correlation $p(MVC_{d_i, g_j})$ represents the possibility of optimal MV obtained in the class (d_i, g_j).

$p\left(MVC_{g_j}\right)$		g_1 [2,4]	$g2$ [4,8]	g_3 [8,12]	g_4[12,16]	$g5$[16,20]	$g6$[20,24]	$g7$[24,28]	$g8$[28,32]
		50.85%	40.09%	2.26%	0.10%	0.00%	1.65%	4.94%	0.10%
$d1$	8.03%	4.08	3.22	0.18	0.01	0.00	0.13	0.40	0.01
$d2$	35.72%	18.16	14.32	0.81	0.04	0.00	0.59	1.76	0.04
$d3$	6.79%	3.45	2.72	0.15	0.01	0.00	0.11	0.34	0.01
$d4$	0.41%	0.21	0.16	0.01	0.00	0.00	0.01	0.02	0.00
$d5$	17.81%	9.06	7.14	0.40	0.02	0.00	0.29	0.88	0.02
$d6$	23.93%	12.17	9.59	0.54	0.02	0.00	0.39	1.18	0.02
$d7$	6.28%	3.19	2.52	0.14	0.01	0.00	0.10	0.31	0.01
$d8$	1.03%	0.52	0.41	0.02	0.00	0.00	0.02	0.05	0.00

Table 1. MV correlation probabilities of video sequence coastguard, the 10th frame

One example is given in Table 1, which shows the MV correlation characteristic in the 10th frame of "coastguard" CIF video sequence. For better understanding, the 10th frame of the "coastguard" is given in Figure 5. It can be observed that the fast moving boats bring some fast and irregular motion, while camera panning generates smooth movement on background. According to Table 1, more than 35% of optimal MVs are detected in the directional class d_2 . In class d_5 and class d_6, there are also big percentage of optimal MVs appears, which implies the motion of this frame is directional irregular. While the distance correlation suggests that 90% of optimal MVs locate within the radius of 8 pels, which is quite stable when considering distance correlation. Considering both directional and distance correlation, there are only 3 partition regions, i.e. $\left(d_2, g_1\right)$, $\left(d_2, g_2\right)$ and $\left(d_6, g_1\right)$ with more than 10% probabilities to contain the optimal position. In the meanwhile, 21 of 64 regions' MV correlation probabilities are more than 0.1%. This suggests that intensive search is only needed to be performed in these regions. While the rest of regions, it is sufficient to be coarsely searched or even be totally skipped.

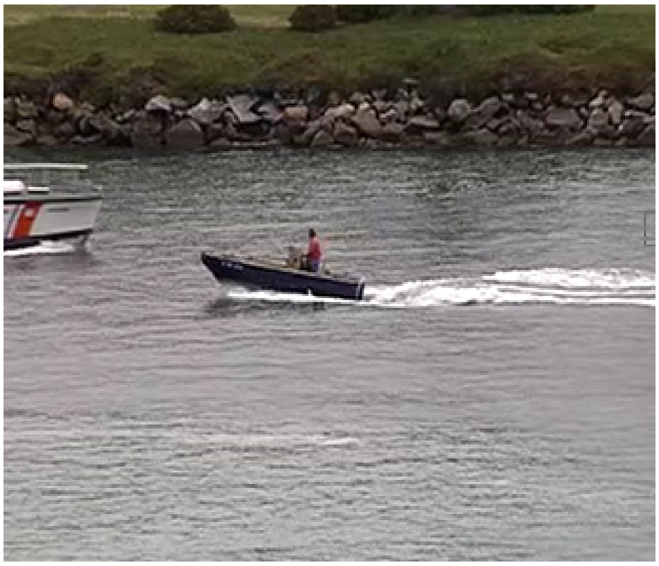


Figure 5. The 10th frame of CIF video sequence "coastguard".

Further illustration is demonstrated in Figure 6. The directional division describe the direction difference between the predicted motion vector and the optimal motion vector. The Octagon grid partitions represent the distance difference between two vectors. Such category division is not in image pels domain. It represents the unique motion character of each frame. Direction class d_1 covers $\pm 22.5^o$ of direction different between predicted motion vector and optimal motion vector. For each MB, the predicted motion vector determines the initial direction d_1 and then the division pattern is rotated accordingly. As the motion correlation for each frames are different, the division pattern is different among frames. For each macroblock, the predicted MVs are different, so that the search pattern is also adaptively changed.

Based on above satiric analysis, SAAS algorithm is proposed, which provides a more accurate approach to obtain optimal motion vector. Similar to PIDS algorithm, the number of search points in each division is adaptively adjusted. But more computational complexity can be saved as the intensive searching areas are more precisely divided with help of different Octagon grids.

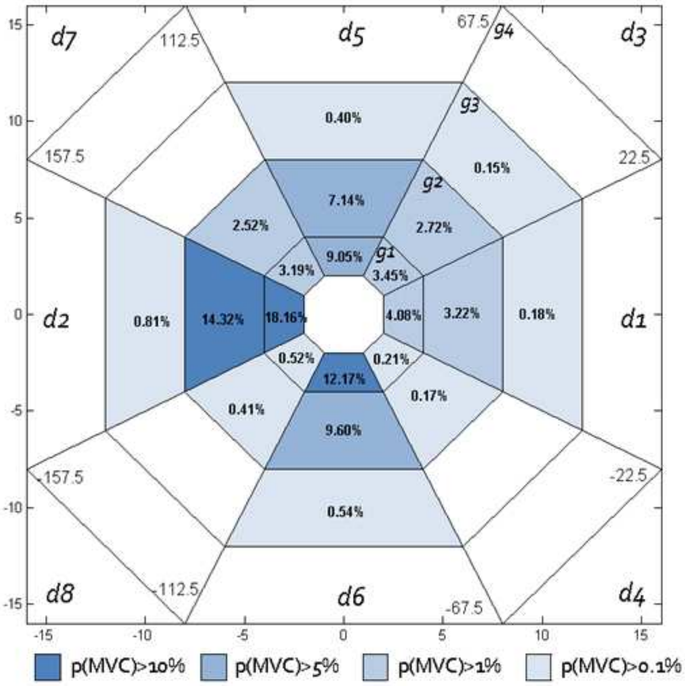


Figure 6. MV correlation statistics of coastguard, the 10th frame

3. Observations of simulated annealing and simulated quenching algorithm

3.1. Simulated annealing algorithm

Simulated annealing (SA) [13] is a probabilistic method for finding the global minimum of an optimization problem. It works by emulating the physical process where liquids are slowly cooled so that the atoms are often able to line themselves up and form a pure crystal.

The crystal can be seemed as the minimum energy state for this system. SA is especially suitable for the large scale problems with the global minimum hidden among several local minimum. The motion estimation is such kind of optimization problem that search for the optimal motion vector with minimum RD cost. However, most fast motion estimation search algorithms look for steepest descent for minimization and go downhill as far as they can go, as shown in Figure 7. Hence, these algorithms are easily trapped into a local minimum.

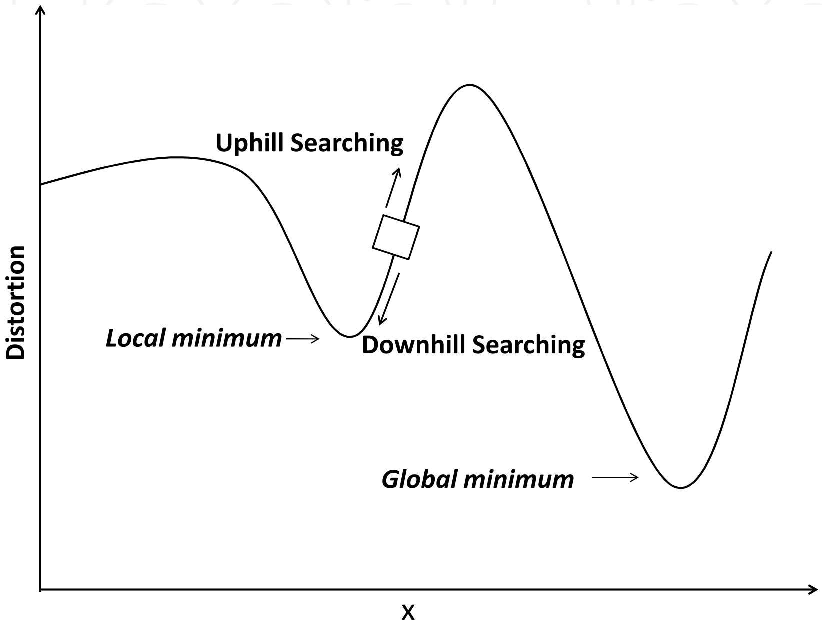


Figure 7. Uphill and downhill searching on rate-distortion surface

Avoiding the disadvantage stated above, SA algorithm can be viewed as a good solution to motion estimation search algorithm, in which occasional uphill moves will help the process escape from local minima. The so-called Boltzmann probability distribution as defined in equation (5),

$$Prob(E) \sim exp(-E/kT) \tag{3}$$

expresses that a system at temperature T has its energy probabilistically distributed among all different energy states. Even at low temperature, there is a chance for the system to get out of a local energy minimum. Therefore, the system sometimes goes uphill as well as downhill. But lower the temperature, less chances for any significant uphill to take place. The basic elements of simulated annealing are as follows:

- A finite solution space *S* (set of states).
- An objective function $E(s)$ (analogy of energy) at state s, whose minimization is the goal of the procedure**.**
- A Neighbourhood structure $N(s)$.
- A nonincreasing function *T* called cooling schedule, which controls the annealing procedure, and *T(t)* is called the temperature at time *t*.

Given the above elements, the process of SA searches for the minimum energy state s_0 is described as follows:

```
Select an initial solution $s_0 \in S$
Select an initial temperature $t_0 > 0$
Select a decreasing temperature function $T(t)$ as cooling schedule
Repeat
  Repeat
          Randomly generate neighbourhood point $s \in N(s_0)$;
     $\delta = E(s) - E(s_0)$;
     if $\delta < 0$, then $s_0 = s$;
     else, generated randomly $c$ uniformly in the range $(0, 1)$;
        If $c < \exp(-\delta/T(t))$ then $s_0 = s$;
    Until $counter = nrep$
    Decrease the temperature $t$ as per annealing schedule $T(t)$
 Until $T(t) < T_{min}$
$s_0$ is the approximation to the optimal solution.
```

3.2. Simulated Quenching algorithm

SA solution usually requires a large number of function evaluations to find the global minimum, which cause the speed of process is quite slow. That is the main disadvantage when using in fast motion estimation algorithm. To speed up the algorithm, a Simulated Quenching (SQ) methodology was proposed. Like SA, SQ algorithm also resembles the cooling process of molten metals through annealing. The analogy of the technique remains the same as that of SA except for quick temperature reduction annealing schedule. Thus the cooling rate becomes one of important parameters, which governs the successful working of SQ.

As in fast motion estimation algorithm, video contents and motion character are changing all the time, it's quite difficult to find a unique cooling scheme for such complicated application. In our proposed SAAS algorithm, we adaptive choose annealing schedule according to MV correlation probabilities information. For the frame with steady motion and high MV correlation, larger values of MV correlation probabilities are more easily to distribute in fewer divided regions. In this case, the faster anneal schedule will safely lead to global optimum. While a slower annealing schedule will be choosing when the frame with more irregular motion and MV correlation distribution is flat. The proposed SAAS algorithm with adaptive cooling scheme is specified in next section.

4. Proposed SAAS algorithm

The PIDS algorithm adaptive selects the intensive and coarse search regions in directional partition. However, with fixed number of search points in each direction area, it cannot adjust the search range for different motion scenes. To tackle this drawback, search pattern in SAAS algorithm is no longer restricted to certain directional regions, but is adaptively selected from more specific divided regions based on the MV correlation statistics. Flow chart of the SAAS algorithm is depicted in Figure 8 for better illustration. For each frame, ME search pattern is determined by MV correlation statistics. For each block, 24 directional candidates are employed to determine initial class d_1 as shown in Figure 9. Then, the search window division is carried out based on ME search pattern of the frame. One example of $d_1=c_4$ in the 10th frame of coastguards is demonstrated in Figure 10. In order to avoid trapping into a local minimum, Simulated Annealing based solution methodology is adopted to process the uphill and downhill searches, where MV correlation probabilities are set as the temperature parameter to control the annealing process adaptively.

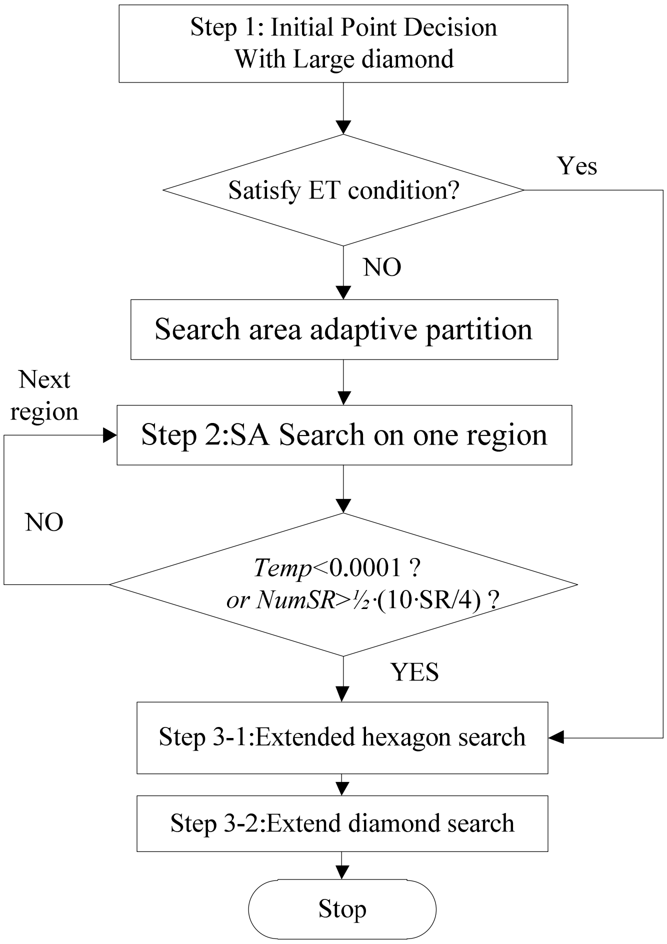

Step 1: Initial Point Decision
With Large diamond
Satisfy ET condition?
Yes
NO
Search area adaptive partition
Next
region
Step 2:SA Search on one region
NO
Temp<0.0001 ?
or NumSR>½·(10·SR/4) ?
YES
Step 3-1:Extended hexagon search
Step 3-2:Extend diamond search
Stop

Figure 8. Flow chart of the proposed SAAS algorithm

4.1. Dynamic update of MV correlation probability

In the SAAS algorithm, it is very important to keep MV correlation probabilities accurate. Not only because the MV correlation probabilities is the crucial element for search region partition and annealing schedule, but also motion characteristic of each video sequence is unique and the MV correlation probabilities are changing all the time. A pre-processing step is conducted to reveal the motion correlation characteristic for the each frame.

The MV correlation probabilities $p(MVC_{d_i,g_j})$ are calculated with equation (2) by MV directional and distance correlation probabilities. In order to get more accurate MV correlation characteristic, the first octagon grid is started from 2 pixels to get rid of the points near centre. This is mainly because of the MV directional difference is meaningless when optimal points are close to the centre point. After the calculation, the MV correlation probabilities are sorted by descending order with corresponding region$\left(d_i,g_j\right)$, which represents the region in direction d_i and in the g_j grid. A parameter $temp\left(d_i,g_j\right)$ that affects the annealing schedule as well as acceptance condition is also assigned by MV correlation probabilities $p(MVC_{d_i,g_j})$.

4.2. Step 1: Initial search point decision

The initial search point is selected from the four prediction models defined in the UMHexagonS. Based on the analysis in the last section, vectors around initial search point have a high probability to be the optimal MV. Therefore, we define large diamond search with 8 searching points around the start search point, which is similar shown in Figure 2 as step1. In contrast to the 25 point rectangular full search in UMHexagonS, this large diamond search reduces the computational requirement without sacrificing its accuracy. The point with the minimum rate-distortion cost is determined as the initial search point.

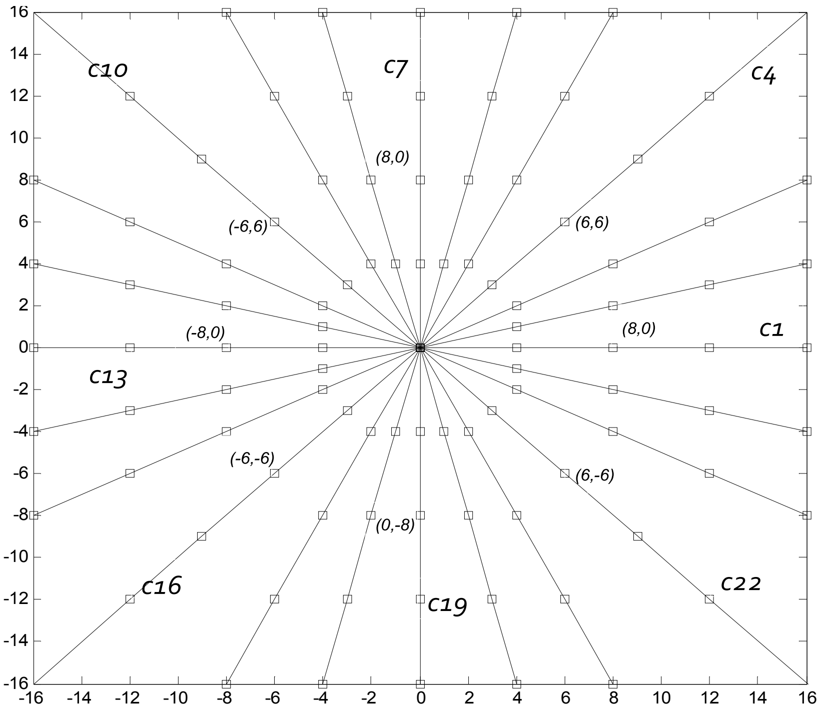


Figure 9. 24 candidate directions for *d₁* determination

4.3. Adaptive partition of search area

After obtaining the initial search point, search area need to be divided based on MV correlation probabilities and predicted MV. 24 candidate directions $\left(c_1,c_2\ldots,c_{24}\right)$, 3 times more than PIDS's, are employed which is indicated in Figure 9. The candidate direction c_i with minimum degree difference to d_{pred} is determined as initial search direction *d1*. The directional regions with boundary of ±22.5°, ±67.5°, ±112.5° and ±157.5° are spread according to the initial search direction *d1*. Then the octagon grids are utilized to divide the search window into regions, where is the search range. Based on that, the search window is adaptively partitioned. The coordinate of each region is represented by (d_i, g_j). Parameter $temp(d_i, g_j)$ are assigned to each region (d_i, g_j) as indexing. One more example is shown in Figure 10, which shows the search window divisions are adjusted when initial direction *d1=c4* in coastguards' 10th frame. Compared to the search window partition in figure 6, when *d1=c1,* the whole search pattern is changed as the difference of predicted MV. In next step, the simulated annealing search process will be conducted on different search region $r(d_i, g_j)$ with parameter $temp(d_i, g_j)$ as cooling scheme.

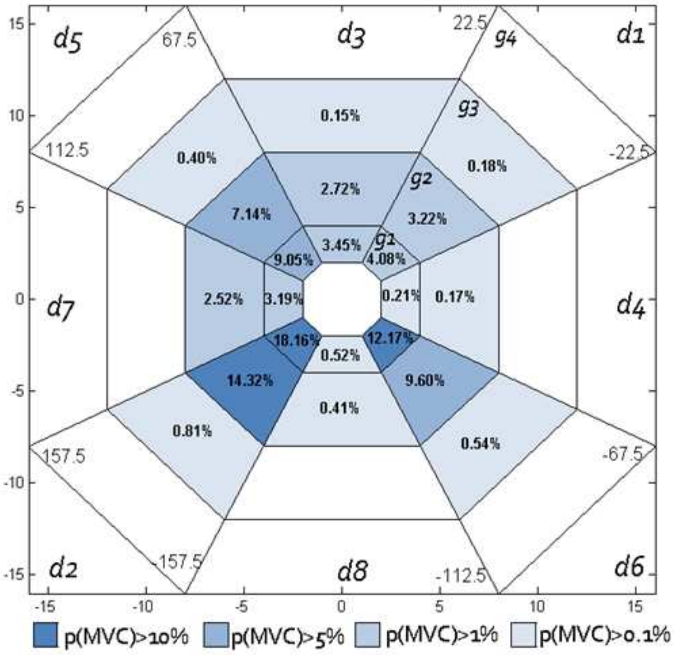


Figure 10. Search area division by directions and grids in SAAS, example of *d1=c4*, the 10th frame of coastguard

4.4. Step 2: Simulate annealing search

4.4.1. Objective function and solution space

In order to employ simulated annealing search in BMME algorithm, the SA elements are defined combing the concept of motion estimation in this section. The procedure for optimal MV searching is performed using predicted MV as centre of the search window. To optimally select the least rate-distortion cost, Lagrangian multiplier tool [14] are defined as follow:

$$J(mv, \lambda_M) = SAD(s, c(mv)) + \lambda_M \cdot R(mv - mv_{pred}) \tag{4}$$

where mv is the candidate motion vector, mv_{pred} is the predicted the motion vector from neighbour blocks. s and c are the source video and the reconstructed video, respectively. SAD represents sum of absolute difference between the block in current frame and the block in the reference frame. $R()$ represents the bits used to encode the motion information computed by a table-lookup and λ_M is the Lagrangian multiplier set according to the quantization parameter (QP), which is given by

$$\lambda_M = \sqrt{0.85 \times 2^{QP/3}} \tag{5}$$

This rate-constrained function has achieved good RD performance for motion estimation. In SA search step, RD cost function $J(mv, \lambda_M)$ is employed as Objective function E .

$$E = J(mv, \lambda_M) \tag{6}$$

The divided regions $r(d_i, g_j)$ in search window are denoted as solution space, which is indexed by MV correlation probabilities rather than spatial neighbour region. The order of regions with decreasing MV correlation probabilities for the 10th frame of sequence *coastguard* is shown in Table 2. This mechanism can be seemed as a randomly selection from solution space. Compared to the simple downhill search in continuous space, this scheme intensively searches the regions with higher MV correlations probabilities first. For the regions with lower probabilities, coarse search or early terminal will be applied.

order	*MV Correlation Probability %*	*Region Coordinate*	*order*	*MV Correlation Probability %*	*Region Coordinate*
1	18.16	(d_2 ,g_1)	**16**	0.81	(d_2,g_3)
2	14.32	(d_2 ,g_2)	**17**	0.59	(d_2,g_6)
3	12.17	(d_6 ,g_1)	**18**	0.54	(d_6,g_3)
4	9.60	(d_6 ,g_2)	**19**	0.52	(d_8,g_1)
5	9.05	(d_5,g_1)	**20**	0.41	(d_8,g_2)
6	7.14	(d_5,g_2)	**21**	0.40	(d_5,g_3)
7	4.08	(d_1,g_1)	**22**	0.40	(d_1,g_7)
8	3.45	(d_3,g_1)	**23**	0.39	(d_6,g_6)
9	3.22	(d_1,g_2)	**24**	0.34	(d_3,g_7)
10	3.19	(d_7,g_1)	**25**	0.31	(d_7,g_7)
11	2.72	(d_3,g_2)	**26**	0.29	(d_5,g_6)
12	2.52	(d_7,g_2)	**27**	0.21	(d_4,g_1)
13	1.76	(d_2,g_7)	**28**	0.18	(d_1,g_3)
14	1.18	(d_6,g_7)	**29**	0.17	(d_4,g_2)
15	0.88	(d_5,g_7)	**30**	0.15	(d_3,g_3)

Table 2. Mv search region order in the 10th frame of sequence *coastguard*

4.4.2. Annealing schedule

The annealing schedule is one of crucial parameter for the SA process. If the temperature in the system dropping too fast, the advantage of SA, which converge to the global optimum, is defeated. However the too slow cooling process might affect the efficiency of our fast searching algorithm. Moreover, it quite difficult to set a fixed annealing schedule for the changeable video contents. In SAAS algorithm, the sorted MV correlation probabilities $p(MVC_{d_i,g_j})$ are assigned to corresponding annealing parameters $temp\left(d_i,g_j\right)$ for region $r\left(d_i,g_j\right)$ to control the annealing schedule.

$$temp(d_i,g_j) = p(MVC_{d_{i,}g_j}) = p(MVC_{d_i}) \times p(MVC_{g_j}) \tag{7}$$

$temp\left(d_i,g_j\right)$ is a set of parameter in pixel domain for particular block, while $p\left(MVC_{d_i,g_j}\right)$ is a relative parameter in frame level. By using this adaptive annealing schedule, the cooling speed is changing with video content and motion correlation, while governs the successful working of the SA procedure.

To improve the searching efficiency, SAAS performs different number of iterations at different temperature status. Inside each region $r\left(d_i,g_j\right)$, mv search is randomly performed along the direction d_i in the range of $[g_{j-1},g_j]$. The number of search points ($NumS(d_i,g_j)$) in division $r(d_i,g_j)$ is determined by a pair of thresholds, *temp_high* and *temp_low*.

$$NumS\left(d_i,g_j\right) = \begin{cases} 4, & temp\left(d_i,g_j\right) > temp\_high \\ 2, & temp\_low < temp\left(d_i,g_j\right) < temp\_high \\ 1, & temp(d_i,g_j) < temp\_low \end{cases} \tag{8}$$

After several experiments with more than 50 different sequences, we empirically determined temp_high = 0.3 and temp_low = 0.15. These thresholds provide satisfying performance on different motion senrou. By utilizing this mechanism, SAAS exploits intensive search in the regions with high MV correlation, and selects fewer search points in less correlation region automatically.

4.4.3. Minimum accepted condition

The minimum accepted condition in SA is based on Boltzmann probability distribution. Referring to equation (3), there is a high probability to perform uphill search when difference of cost function E is smaller and the temperature T is higher. By using Boltzmann concept, SAAS utilizes the following SA Condition.

$$\frac{E\left(best\right) - E\left(r\left(d_i,g_j\right)\right)}{E\left(best\right)} < \rho \tag{9}$$

Where, is current global minimum and is a threshold controlled by annealing parameter, and is given by:

$$\rho = \frac{-1}{2 \cdot \log\left(temp\left(d_i, g_j\right)\right)} \tag{10}$$

With ρ, the SA condition is directly proportionate to $temp\left(d_i, g_j\right)$ and inversely proportional to difference of cost function E. If $E(best) > E(r(d_i, g_j))$, region $r\left(d_i, g_j\right)$ is directly identified as the current global optimal. Otherwise, the SA condition still provides occasional upward moves. As ρ is controlled by $temp\left(d_i, g_j\right)$, for the division with lower $temp\left(d_i, g_j\right)$, the chance to conduct upward moves is smaller.

4.4.4. Termination condition

It is impossible to conduct SA search on all search partition, as there are regions partitioned in search window. Moreover, the majority of regions contain low MV correlation probabilities, as shown in Table 1. For these reasons, it is appropriated to limit the total number of search regions (*NumSR*) and have an early termination condition. Two termination conditions are given, one is the temperature status and the other is the number of searched regions.

$$\begin{aligned} &1)\ temp\left(d_i, j\right) < 0.0001 \\ &2)\ NumSR > \frac{1}{2}\left(10 \cdot SR / 4\right) \end{aligned} \tag{11}$$

If one of these termination conditions is satisfied, the SA search will stop and go to the Extended Hexagon Search step which is introduced in UMHexagonS, to refine the local optimum. Otherwise, SA search will proceed to next region by the indexed of the decreasing parameter temp.

4.5. Step 3: Extended hexagon-based search

A large hexagon search pattern and a small diamond search pattern are employed in this step, which is modified from UMHexagonS. The large diamond pattern has six search locations, while the small diamond search pattern has four points. The large hexagon pattern in the step 3-1 is recursively used and its centre recursively moved until the location with the minimum rate-distortion cost lies in the centre of the hexagon. After this, a small diamond pattern in the step 3-2 is recursively utilized until the location with the minimum rate-distortion cost is at the centre of this pattern. Finally, this point is determined as the point of motion vector for the current block. But our Extended Hexagon-based search process is only limited within the one search region $r\left(d_i, g_j\right)$, which contains the optimal MV. Compared to UMHexagonS, this centre basis optimal MV refinement approach can obtain optimal MV with fewer search points.

Sequence	ME TIME (sec)					Average Search Points				
	UMH	PIDS	Gain	SAAS	Gain	UMH	PIDS	Gain	SAAS	Gain
Bus	725.6	618.3	14.79%	558.4	23.04%	30.61	19.43	37.52%	13.15	57.03%
Coastguard	742.5	622.4	16.17%	569.6	23.28%	32.11	19.62	38.90%	13.29	58.62%
Crew	655.5	587.1	10.43%	544.2	16.98%	20.79	14.35	30.99%	10.72	48.45%
Harbour	711.8	580.8	18.40%	554.8	22.04%	33.23	18.12	45.46%	13.09	60.61%
Mobile	648.7	538.7	16.97%	501.5	22.70%	28.85	16.14	44.06%	10.65	63.08%
Stefan	568.8	489.5	13.94%	444.9	21.79%	25.38	15.43	39.22%	9.99	60.63%
Template	597.3	512.2	14.25%	487.5	18.38%	22.40	12.53	44.07%	8.99	59.85%
Average			14.99%		21.17%			40.03%		58.32%

Table 3. Results of proposed SAAS comparing to that of UMHexagonS and PIDS in terms of average search points reduction (%) and motion estimation time reduction (sec) (QP=28)

5. Experimental results

In this section numerous experiments with H.264/AVC reference Joint Model (JM) software version 16.1 were conducted. We compared the proposed SAAS algorithm against the FS, PIDS and UMHexagonS algorithms, in terms of computational complexity (speed measured by ME time and average search points (ASP)) and Rate-distortion performance (PSNR and bit rate). Several commonly used sequences, covering a wide range of motion characteristics, are taken into consideration.

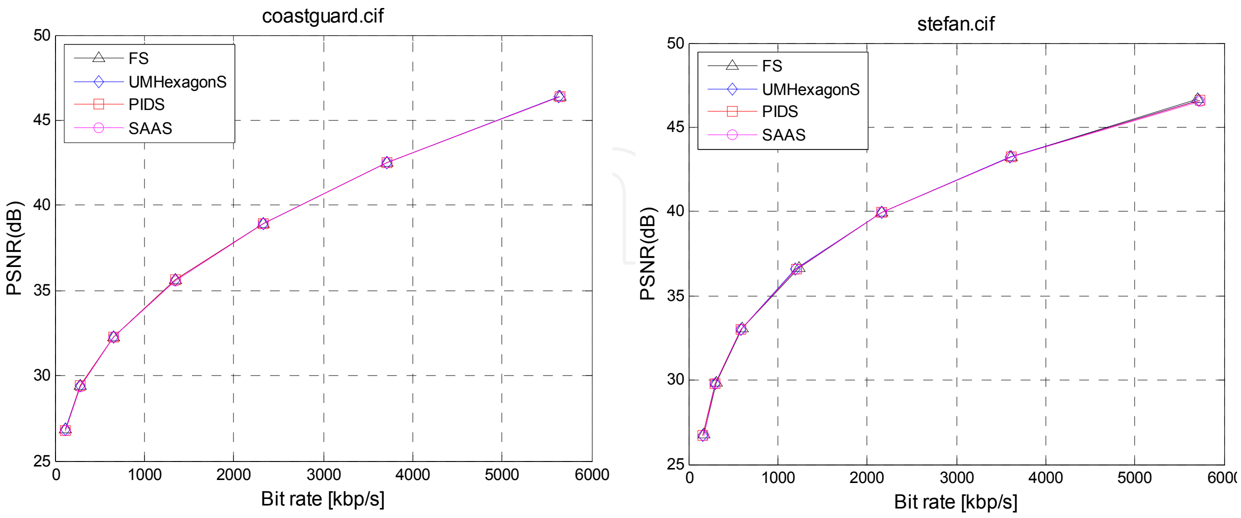


Figure 11. Rate-Distortion performance comparison of FS, UMHexagonS, PIDS and SAAS at various QPs

The group of picture (GOP) structure was IPPP, in which only first frame has been coded as I frame and first P frame has been coded by UMHexagonS. The sequences are tested at 30fps (frames per second). The Content Adaptive Variable Length Coding (CAVLC) entropy coder is used for all the simulations, with 5 reference frames. A search range of 32 and the quantization parameter of 28 are used. The simulation platform in our experiments is done with a PC of 2.44 GHz CPU and 8G RAM.

For complexity comparisons, the proposed algorithm is compared to the hybrid UMHexagonS adopted by the H.264/AVC reference software. Two different measurements are used to calculate the computational efficiency, average search points requirement and encoding time. Results are presented in Table 3. As shown in the Table 3, SAAS needs 48-63% less search points than UMHexagonS and saves average of 21% encoding time. Since it performs more precise search pattern adjustment, SAAS requires average 45% less search points than PIDS.

Sequence	*PSNR(dB)*				*Bit-rate (kb/s)*			
	FS	*UMH*	*PIDS*	*SAAS*	*FS*	*UMH*	*PIDS*	*SAAS*
Bus	35.792	35.786	35.809	35.800	1225.29	1240.05	1262.27	1249.88
	Gain	-0.006	0.017	0.008	*Degrade*	1.20%	3.02%	2.01%
Coastguard	35.610	35.600	35.602	35.598	1342.64	1344.28	1345.30	1348.56
	Gain	-0.010	-0.008	-0.012	*Degrade*	0.12%	0.20%	0.44%
Crew	37.895	37.870	37.870	37.869	680.590	672.72	671.94	678.053
	Gain	-0.025	-0.025	-0.026	*Degrade*	-1.16%	-1.27%	-0.37%
Harbour	35.623	35.627	35.627	35.625	1572.30	1571.14	1569.25	1572.59
	Gain	0.004	0.004	0.002	*Degrade*	-0.07%	-0.19%	0.02%
Mobile	35.376	35.370	35.377	35.383	1843.16	1843.08	1844.19	1848.52
	Gain	-0.006	0.001	0.007	*Degrade*	0.00%	0.06%	0.29%
Stefan	36.632	36.607	36.602	36.607	1189.09	1202.54	1206.90	1209.64
	Gain	-0.025	-0.030	-0.025	*Degrade*	1.13%	1.50%	1.73%
Template	35.608	35.595	35.587	35.585	1159.96	1162.89	1162.81	1164.11
	Gain	-0.013	-0.021	-0.023	*Degrade*	0.25%	0.24%	0.36%
Average		-0.012	-0.009	-0.010		0.21%	0.51%	0.64%

Table 4. Results of proposed SAAS comparing to that of FS, UMHexagonS and PIDS in terms of PSNR gain (dB) and bit-rate degradation (%) (QP=28)

Considering the large reduction in the computational complexity, the quality degradation is very small. Rate-Distortion performances are presented in Table 4. In all cases, the FS outperforms the others in image quality and bit-rate. Compared to FS, average PSNR degradation of the proposed algorithm is only 0.010. In term of bit-rate, SAAS has a slightly higher degradation than PIDS and UMHexagonS. SAAS has bit-rate decreasing of 0.64% in average. Further information can be obtained in Figure 9, which compares the rate-distortion performance among FS, UMHexagonS, PIDS and SAAS against different QPs (16, 20, 24, 28, 32, 36 and 40). Figure 10 compares the simulation results versus frame number of video sequences *Coastguard*. It is clearly reveal the superiority of SAAS to UMHexagonS in computational reduction, which more than 50% of search points are saved while the PSNR and bit-rate performance are very similar. From the results above, it can be confirmed that the SAAS algorithm has the capability to dramatically reduce the computational burden with negligible degradation in the RD performance.

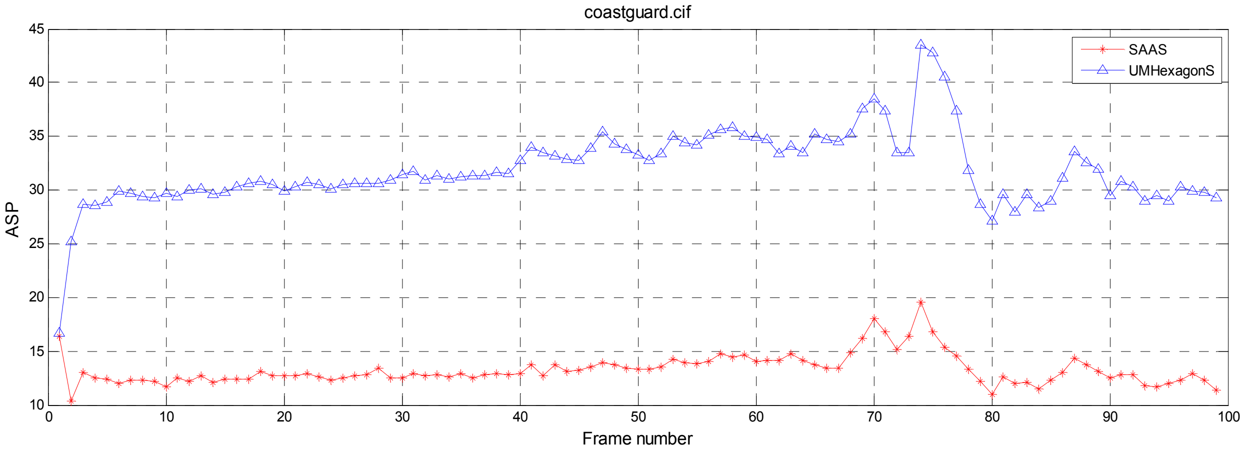


Figure 12. Rate-Distortion performance and complexity cost comparison versus frame

6. Conclusion

This chapter presents a novel fast motion estimation algorithm, Simulated Annealing Adaptive Search algorithm. As mv field has heavy correlation, the proposed algorithm takes the advantage of MV correlation information, which is statistically calculated and plays a significant role in SAAS process. In the SA search step, the search region is adaptively divided and the divisions are searched indexed by MV correlation probabilities in descending order. Furthermore, by utilizing Boltzmann probability concept, the minima acceptation or rejection condition of each SA search is controlled by this correlation information. The experimental results demonstrate that more than 48% of ASP and 21% of ME time can be saved, while maintaining a similar bit-rate without losing the picture quality.

Author details

Zhiru Shi, W.A.C. Fernando and A. Kondoz
I-Lab, CVSSP, University of Surrey, Guildford, United Kingdom

7. References

[1] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG (2004) Text of ISO/IEC 14496 10 Advanced Video Coding 3rd Edition.

[2] Richardson I. E. G. (2003) H. 264 and MPEG-4 video compression. The Atrium, Southern Gate, Chichester, West Sussex, England: John Wiley & Sons.

[3] Jianfeng R, Kehtarnavaz N, and Budagavi M(2008) Computationally efficient mode selection in H.264/AVC video coding. IEEE Trans. Consumer Electronics. 54: 877-886.

[4] Knesebeck M, Nasiopoulos P(2009) An efficient early-termination mode decision algorithm for H.264. IEEE Trans. Consumer Electronics. 55: 1501-1510.

[5] Hosur P. I. (2003) Motion adaptive search for fast motion estimation. IEEE Trans. Consumer Electronics. 49: 1330-1340.

[6] Ki Beom K, Young J, Min-Cheol H(2008) Variable step search fast motion estimation for H.264/AVC video coder IEEE Trans. Consumer Electronics.54: 1281-1286.

[7] Goel S, Bayoumi M. A(2006) Multi-Path Search Algorithm for Block-Based Motion Estimation. IEEE Int. Conf Image Processing. pp. 2373-2376.

[8] Lai-Man P, Ka-Ho N, Kwok-Wai C, Ka-Man W, Uddin Y, Chi-Wang T(2009) Novel Directional Gradient Descent Searches for Fast Block Motion Estimation. IEEE Trans. Circuits and Syst. Video Technol. 19:1189-1195.

[9] Zhibo C, Zhou P, Yun H, Chen Y(2002) Fast integer pel and fractional pel motion estimation. JVT 6th Meeting. JVT-F017.

[10] Zhiru S, Fernando W. A. C, Silva D. V. S. X(2010) A motion estimation algorithm based on Predictive Intensive Direction Search for H.264/AVC. in (ICME) IEEE Int. Conf. Multimedia and Expo. pp. 667-672.

[11] Lin C. C, Lin Y, Hsieh H. J(2009) Multi-direction search algorithm for block motion estimation in H.264/AVC. Image Processing IET 3: 88-99.

[12] Metropolis N, Rosenbluth A. W, Rosenbluth M. N, Teller A. H, Teller E, (1953)Equation of state calculations by fast computing machines. Chemical Physics J. 21:1087.

[13] Kirkpatrick S, Gelatt C. D, Vecchi M. P(1983) Optimization by simulated annealing Science. 220: 671.

[14] Sullivan G. J, Wiegand T(1998) Rate-distortion optimization for video compression. IEEE Magazine Signal Processing 15:74-90.