Lee *EURASIP Journal on Image and Video Processing* (2015) 2015:29
DOI 10.1186/s13640-015-0083-4

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Early search termination for fast motion estimation

Yun-Gu Lee

Abstract

This paper proposes a novel method for early termination of a motion search. It first introduces an estimator of the sum of absolute difference (SAD) between the current block and a search point. After analyzing the SAD estimator, it proposes the SAD condition to decide whether the current search point is around a minimum point or not. The counter condition is used to evaluate the current search point as an early search termination cue. After the current search point is considered as a minimum point, the search is stopped immediately. The proposed algorithm can easily be combined with most existing fast motion estimation algorithms to further reduce computational cost. While previous thresholding techniques have focused on the correlation of SAD values between a current block and its neighboring blocks, the proposed algorithm studies its own characteristics to predict the threshold. Hence, the proposed termination scheme can complement the previous early search termination techniques by merging the termination methods together, in order to further reduce computational cost. Experimental results demonstrate that the proposed algorithm successfully reduces the computational cost by combining the previous early search termination techniques.

Keywords: Fast motion estimation; Block matching; Early search termination; Adaptive threshold

1 Introduction

Recent video coding standards, such as H.264 [1] and HEVC [2], significantly improve coding efficiency, compared to previous video coding standards. Complexity is also increased along with improvement of the coding efficiency. The video coding standards have adopted the block matching algorithm (BMA) to reduce temporal redundancy between frames. In the BMA, the best matching block of each current block is found in the reference frame. Then, only the difference block between the matching block and the current block is coded with a motion vector (or a displacement). The straightforward method to find the best matching block is a full search block matching algorithm that checks all possible candidates in the search area, which requires a huge computational burden. Moreover, since recent video coding standards basically support variable block size and multiple reference frames, the number of candidates is dramatically increasing. Hence, motion estimation is the most time-consuming part among the video coding tools used in a video encoder. Motion estimation in video coding first finds the integer motion vector within the search area. Then, the subpixel motion vector is found around the integer motion vector. The computational burden for estimating the integer motion vector is much more than that for estimating the subpixel motion vector. So, this paper considers only integer motion estimation.

Fast motion estimation algorithms have been developed to reduce the heavy computational cost of the full search block matching algorithm, such as diamond search [3], MVFAST [4], PMVFAST [5], hexagon-based search [6], and hybrid unsymmetrical cross multihexagon-grid search [7]. In order to find the motion vector, these fast motion estimation algorithms check only limited search points using different shapes and sizes of search patterns, rather than examining all possible search points within a search area. Complex or large search patterns are robust to random motions, but they are expensive. Simple search patterns sometimes fail to find the global minimum point by falling into a local minimum one. Hence, advanced algorithms adaptively select shapes and sizes of search patterns. For example, the large and sparse search pattern is first adopted to prevent from falling into the local

Correspondence: harmony96@gmail.com
Department of Computer Science and Engineering, Kwangwoon University, 447-1, Wolgye-Dong, Nowon-Gu, 139-701 Seoul, South Korea

minimum point. Then, the small and dense search pattern is used to refine the search.

Early search termination techniques have been developed to further reduce the computational cost of fast motion estimation. Usually, a condition of early search termination is to compare the sum of absolute difference (SAD) value of a current search point with the threshold value. If the SAD value is less than the threshold value, the current search point is considered as the minimum point, and the search is immediately stopped without examining the remaining search points. Ismail mentioned that early stop search termination of a search process could reduce the number of computations [8]. In [4], a constant threshold was used for early search termination. Tourapis proposed an adaptive method in that the threshold value of the current block is determined among the minimum SAD values of the three adjacent blocks [5]. Ismail [9] proposed a dynamic early stop search termination (DESST) by considering a SAD value of an initial search center (ISC) of a current block and the average SAD values of ISCs in all the previous coded blocks. These methods determine the threshold value from SAD values of blocks spatially and temporally adjacent to the current block. Since motion fields of consecutive frames are smooth and gentle [9], they can efficiently judge a block to be stationary or not using the determined threshold. For the stationary blocks, they can terminate early the motion search. However, for the non-stationary blocks, the threshold value predicted from SAD values of blocks spatially and temporally neighboring to the current block is inadequate. Hence, while these methods significantly increase the performance, the determined threshold value may not be the best for every block.

In our previous work, a method to predict the distance between the optimal point (a position of the best matching block) and a current search point was proposed [10]. If the predicted distance between them is less than 0.5 pixel, the current search point is considered as the global minimum point, and the motion search was terminated early. However, since the method considered a one-dimensional image, the predicted distance may not be accurate. Also, there was no analysis of the distance measure in terms of performance and accuracy, so it was very difficult to combine the proposed distance measure with the existing motion estimation algorithms, in order to terminate a search early. Therefore, this paper proposes the new early search termination for fast motion estimation by extending our previous work to a two-dimensional space. It proposes a SAD estimator on the two-dimensional space and studies it to get a condition for early termination of a motion search. This paper also analyzes the proposed algorithm in terms of performance and accuracy.

While conventional early termination methods consider the SAD values of blocks spatially and temporally neighboring to a current block to determine termination conditions, the proposed algorithm takes into account a self-characteristic of the current block for a termination condition. Since the proposed algorithm utilizes different characteristics compared to the existing methods, the proposed termination method and the existing termination methods can complement each other, in order to further improve the speed. This paper provides how the proposed method complements previous thresholding techniques.

This paper is organized as follows. Section 2 describes the proposed algorithm. Experimental results are given in Section 3. Finally, conclusions are presented in Section 4.

2 Proposed method

2.1 SAD estimator

Let us assume that there are only translational motions between the previous and current frames, and all pixels in each block have the same translational motion. By assuming that pixels in the previous frame (or $p(i,j)$) can be reconstructed by linear interpolation of pixels in the current frame (or $c(i,j)$), a pixel in the previous frame is described as

$$\begin{aligned} p(i+a,j+b) = & (1-n)((1-m)c(i,j) + mc(i+1,j)) \\ & + n((1-m)c(i,j+1) \\ & + mc(i+1,j+1)) \end{aligned} \tag{1}$$

Here, $(a,\ b)$ and $(m,\ n)$ are integer and subpel components of the optimal motion vector, respectively (Fig. 1). From Eq. (1), a SAD value at the integer search point $(a,\ b)$ is as follows:

$$\begin{aligned} \mathrm{SAD}^{(a,b)}(m,n) = & \sum_{(i,j)\in \mathrm{block}} |c(i,j) - p(i+a,j+b)| \\ = & \sum_{(i,j)\in \mathrm{block}} |mG^{H}(i,j) + mn(G^{H}(i,j+1) \\ & - G^{H}(i,j)) + nG^{V}(i,j)| \end{aligned} \tag{2}$$

Here, $G^H(i,j)$ and $G^V(i,j)$ are $(c(i,j) - c(i+1,j))$ and $(c(i,j) - c(i,j+1))$, respectively. Equation (2) can be

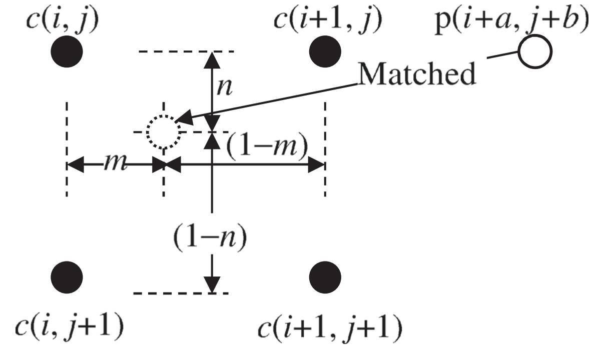

Fig. 1 Matching positions of a previous pixel in the current frame

interpreted as a SAD estimator according to (m, n). If either of m or n is zero, Eq. (2) becomes identical to the equation in the previous work [10], which considers an only one-dimensional case.

2.2 SAD contours

Let (d_x, d_y) be the distance along the x and y axes between the current search point and the optimal motion vector. In the following, this paper assumes that both d_x and d_y are always positive for simplicity. When both d_x and d_y are less than 1 pixel for a given integer search point, (d_x, d_y) is a subpel component of the optimal motion vector. Then, (d_x, d_y) is the same as (m, n), and the SAD value for a given (m, n) can be described by Eq. (2).

Figure 2a depicts contours where SAD values of search points on the same contour are the same. For example, since points A and B are on the same contour, their SAD values are the same. On the other hand, the contours are not always symmetric along $y = x$ as in Fig. 2a. If the gradient of a block is not the same along the x and y directions, the contours are not symmetric. Figure 2b shows a case in which the vertical gradient is much smaller than the horizontal gradient. In this case, the value of $G^V(i,j)$ is usually less than $G^H(i,j)$ in a block, and m is dominant to the value of Eq. (2).

2.3 Termination condition

Subpixel motion estimation finds the optimal motion vector with subpixel accuracy around the integer motion vector. Usually, the search range for the subpixel motion estimation is $[-1, 1]$ along the x and y directions. So if (d_x, d_y) for a current integer search point is less than 1 pixel along the x and y directions or the current integer search point is within the *rectangle*, the current search point should be the integer motion vector. Hence, the integer motion estimation can be stopped immediately, which is the early termination of a motion search. This subsection will study the condition wherein (d_x, d_y) is less than 1 pixel.

Let SAD values of points along *contour A* in Fig. 2a be 1000. If the SAD value of a current integer search point is less than 1000, the current search point should be inside of contour A. Otherwise, the current integer search point is outside of contour A. In this way, the relative position of the current integer search point can be roughly predicted by comparing the SAD values of the current integer search point and contours. If a contour is inside of the rectangle and the SAD value of a search point is less than that for the contour, (d_x, d_y) of the search point will be less than 1 pixel. However, although a contour is inside of the rectangle, if a SAD value is more than that for the contour, it is not certain that (d_x, d_y) is less than 1 pixel. For example, let *point C* be a current integer search point. In this case, the SAD value of the search point is more than that for contour A, and the search point is not inside of the rectangle. On the other hand, if *point D* is a current integer search point, although the SAD value of the search point is more than that for contour A, the search point is inside of the rectangle. Therefore, only if the SAD value of a search point is less than that for the contour that is inside of the rectangle, (d_x, d_y) for the search point is the inside of the rectangle.

Figure 3 depicts three types of the largest contours that satisfy the above condition. In the figure, $\mathrm{SAD}(0,1)$ and $\mathrm{SAD}(1,0)$ represent $\mathrm{SAD}^{(a,b)}(0,1)$ and $\mathrm{SAD}^{(a,b)}(1,0)$, respectively. From Eq. (2), they are

$$\mathrm{SAD}^{(a,b)}(0,1) = \sum_{(i,j)\in \mathrm{block}} |G^V(i,j)| \tag{3}$$

$$\mathrm{SAD}^{(a,b)}(1,0) = \sum_{(i,j)\in \mathrm{block}} |G^H(i,j)| \tag{4}$$

Each largest contour should cross one of (0, 1) and (1, 0) or both of them as in the figure. Let this condition be $\mathrm{Cond}_{\mathrm{MinSad}}$. The termination condition of $\mathrm{Cond}_{\mathrm{MinSad}}$ (or a SAD value for each contour) is simply the minimum one

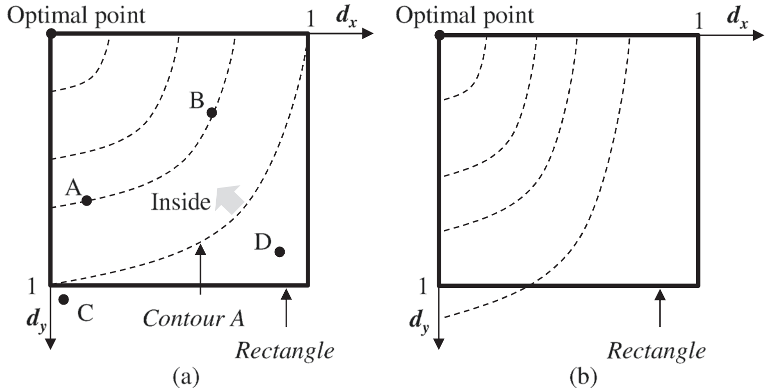

Fig. 2 SAD contours. **a** Symmetrical contours and **b** asymmetrical contours

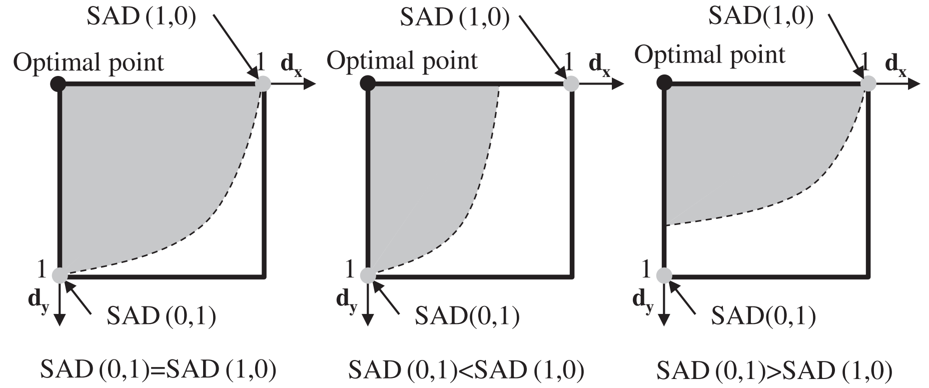

Fig. 3 Three types of the largest contours that satisfy an early termination condition

between $\mathrm{SAD}^{(a,b)}(0,1)$ and $\mathrm{SAD}^{(a,b)}(1,0)$ as follows:

$$\mathrm{Cond}_{\mathrm{MinSad}} : T = \mathrm{MIN}\left(\mathrm{SAD}^{(a,b)}(1,0), \mathrm{SAD}^{(a,b)}(0,1)\right) \tag{5}$$

Consequently, if the SAD value of the current search point is less than the threshold, integer motion estimation is terminated. It is noted here that $\mathrm{SAD}^{(a,b)}(0,1)$ and $\mathrm{SAD}^{(a,b)}(1,0)$ are calculated from only pixels in a current block.

Figure 4 illustrates two types of contours for the condition of $\mathrm{Cond}_{\mathrm{MaxSad}}$ defined as follows:

$$\mathrm{Cond}_{\mathrm{MaxSad}} : T = \mathrm{MAX}\left(\mathrm{SAD}^{(a,b)}(1,0), \mathrm{SAD}^{(a,b)}(0,1)\right) \tag{6}$$

Compared to Fig. 3, the sizes of the gray regions for cases with $\mathrm{SAD}(0,1) < \mathrm{SAD}(1,0)$ and $\mathrm{SAD}(0,1) > \mathrm{SAD}(1,0)$ are increased. Hence, the probability of early termination of a motion search will be increased. However, since parts of the gray regions are not included in the rectangular boxes, the search will be sometimes terminated at the improper position, and the quality performance may be degraded. Section 4 will present the performance degradation from wrong termination of a motion search by comparing the performances of $\mathrm{Cond}_{\mathrm{MinSad}}$ and $\mathrm{Cond}_{\mathrm{MaxSad}}$.

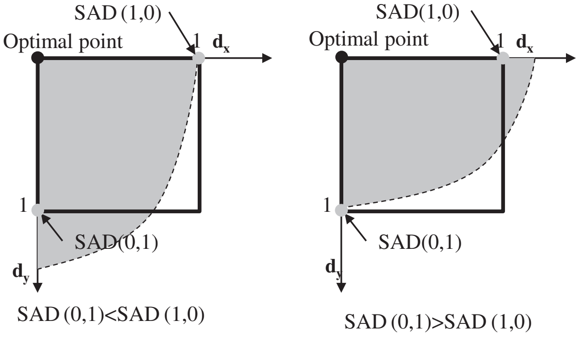

Fig. 4 Contours for a condition of $\mathrm{MAX}(\mathrm{SAD}^{(a,b)}(0,1), \mathrm{SAD}^{(a,b)}(1,0))$

The proposed method can be combined with the existing thresholding technique to further improve the search speed. $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ is a termination condition by combining $\mathrm{Cond}_{\mathrm{MinSad}}$ with a simple threshold technique, which was introduced in MVFAST [4], in order to further improve the speed. The termination condition is defined as follows:

$$\mathrm{Cond}_{\mathrm{MinSad,Sim}} : T = \mathrm{MAX}(2B_{\mathrm{W}}B_{\mathrm{H}}, \mathrm{MIN}\left(\mathrm{SAD}^{(a,b)}(1,0), \mathrm{SAD}^{(a,b)}(0,1)\right) \tag{7}$$

Here, B_{W} and B_{H} are the width and height of a block to be considered. $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ sets a constant threshold value to $2B_{\mathrm{W}}B_{\mathrm{H}}$ to support the variable block size mode. Note here that the threshold value for MVFAST [4] was set to 512. The performance of $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ will be given in the next section.

2.4 Implementation

The SAD condition for early termination of a motion search was proposed in the previous subsections. This subsection will describe how to implement the proposed method. The proposed algorithm cannot be running alone, but it should be merged with the existing fast motion estimation to further improve its search speed. To combine the proposed method with the fast motion estimation is simple as in Fig. 5c. After the SAD evaluation of each search point, its SAD value is compared with one of the proposed threshold T methods from the previous section. Only if the SAD value is less than the threshold, the current search point is considered as the minimum point, and the search is stopped immediately.

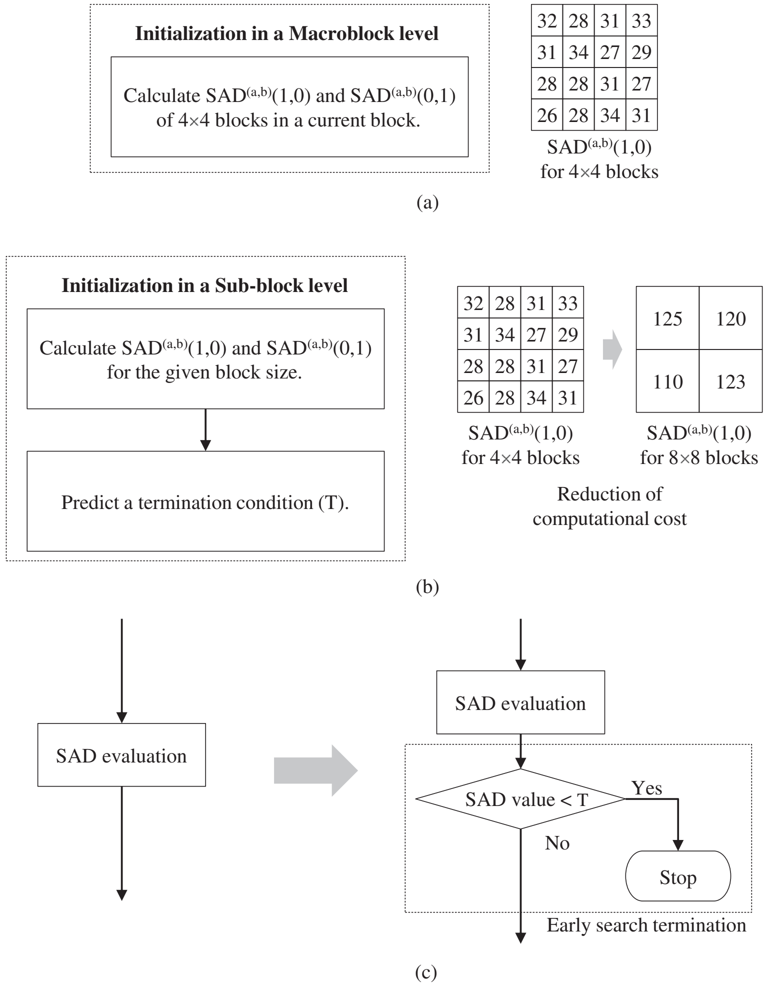
Initialization in a Macroblock level
Calculate SAD^(a,b)(1,0) and SAD^(a,b)(0,1) of 4×4 blocks in a current block.
32 28 31 33
31 34 27 29
28 28 31 27
26 28 34 31
SAD^(a,b)(1,0) for 4×4 blocks
(a)
Initialization in a Sub-block level
Calculate SAD^(a,b)(1,0) and SAD^(a,b)(0,1) for the given block size.
Predict a termination condition (T).
32 28 31 33
31 34 27 29
28 28 31 27
26 28 34 31
125 120
110 123
SAD^(a,b)(1,0) for 4×4 blocks
SAD^(a,b)(1,0) for 8×8 blocks
Reduction of computational cost
(b)
SAD evaluation
SAD evaluation
SAD value < T
Yes
No
Stop
Early search termination
(c)

Fig. 5 Implementation of the proposed method. **a** Initialization in a MB level, **b** initialization in each sub-block level, and **c** applying early search termination

To support variable block motion estimation, the threshold value for the termination condition does not need to be calculated at every block size. The computational cost is reduced by sharing the calculated data. First, the method calculates SAD(1,0) and SAD(0,1) of 4×4 blocks in a current block in Fig. 5a. Then, SAD(1,0) and SAD(0,1) for the given block size are calculated from the above 4×4 blocks (see Fig. 5b). Finally, the threshold is calculated according to the termination condition, such as $\text{Cond}_{\text{MinSad}}$, $\text{Cond}_{\text{MaxSad}}$, and $\text{Cond}_{\text{MinSad,Sim}}$.

The calculation of $\text{SAD}(0,1)$ and $\text{SAD}(1,0)$ requires additional computational cost. The computational load to calculate $\text{SAD}(0,1)$ or $\text{SAD}(1,0)$ is the same as the computation of one point search (or the SAD evaluation for one point). Hence, the overhead for the proposed algorithm corresponds to a two-point search in total. This computational cost can be further reduced using a subsampling technique [10] as follows:

$$\text{SAD}(0,1) = 4\sum_{j=0}^{N/2}\sum_{i=0}^{N/2}|c(2i,2j) - c(2i,2j+1)| \quad (8)$$

$$\text{SAD}(1,0) = 4\sum_{j=0}^{N/2}\sum_{i=0}^{N/2}|c(2i,2j) - c(2i+1,2j)| \quad (9)$$

Then, the overhead becomes a 0.5 point search.

Some assumptions, such as the pure translational motion, the reconstruction of a pixel using a simple bilinear interpolation, and convex contours of SAD, are not valid in general. Hence, the proposed algorithm may improve speed with a sacrifice of rate (or peak signal-to-noise ratio (PSNR)) performance. Therefore, experimental results in the next section will show the performance of the proposed algorithm in terms of speed and rate.

3 Experimental results

In order to verify the proposed algorithm, two types of experiments are performed. If a search point satisfies the proposed condition while the search point is not very close to the optimal point (or the integer motion vector), the proposed method terminates a motion search at the non-optimal point. Then, there is a chance of degrading the coding performance. Hence, the first experiment examines the accuracy and safety of the proposed termination condition. The second experiment verifies the performance of the proposed algorithm on a video encoder.

This paper includes experimental results for eight test video sequences including *BQTerrace*, *BasketballDrill*, *Kimono1*, *ChinaSpeed*, *ParkScene*, *FourPeople*, *KristenAndSara*, and *PeopleOnStreet*, which were used in the HEVC standard. The details are given in Table 1.

3.1 Accuracy test

In order to examine the accuracy and safety of the proposed termination condition of $\mathrm{Cond}_{\mathrm{MinSad}}$, SAD values for all possible search points for each block are examined and compared with the condition $\mathrm{Cond}_{\mathrm{MinSad}}$. When among all search points, only search points within an area close to the optimal point satisfy the termination condition, a block is marked as a *safe block*. Here, the distance between the optimal point (or the integer motion vector) and a search point within the close area should be equal to or less than 1 pixel. If one of the search points outside of the close area satisfies the termination condition, the block is marked as an *unsafe block*. If the current search point is out of the close area while the point satisfies the proposed termination condition, the search is stopped at the current search point, and the coding performance may be degraded. Hence, the number of safe blocks should be high, while the number of unsafe blocks is low. It should be noted here that the block marked as the unsafe block does not always degrade the coding efficiency. For example, although the block is the unsafe block, if the current search point is within the close area, the search is stopped immediately, and the coding performance will not be degraded. The proposed termination condition $\mathrm{Cond}_{\mathrm{MinSad}}$ for the accuracy test is given in Eq. (5).

Table 1 Test video sequences

Sequence	Resolution	Frame rate (fps)	Number of frames
BQTerrace (BQT)	1920 × 1080	60	100
BasketballDrive (BD)	1920 × 1080	50	100
Kimono1 (K)	1920 × 1080	24	100
ChinaSpeed (CS)	1024 × 768	30	100
ParkScene (P)	1920 × 1080	24	100
FourPeople (FP)	1280 × 720	60	100
KristenAndSara (KS)	1280 × 720	60	100
PeopleOnStreet (PS)	2560 × 1600	30	100

Table 2 represents the comparison of the early termination condition between the proposed condition $\mathrm{Cond}_{\mathrm{MinSad}}$ and Ismail's method [9], which is one of the state-of-the-art methods. The threshold for Ismail's [9] is as follows:

$$T_{\mathrm{S}} = \mathrm{MIN}[\mathrm{MAX}(\mathrm{SAD}_{\mathrm{ISCavg}},512),\mathrm{SAD}_{\mathrm{ISCcurrent}}] \times \epsilon_1 + \epsilon_2 \qquad (10)$$

where $\mathrm{SAD}_{\mathrm{ISCavg}}$ is the average $\mathrm{SAD}_{\mathrm{ISC}}$ value for all the previous coded blocks in the current frame whose best match motion vectors are the initial search centers (ISCs) themselves. $\mathrm{SAD}_{\mathrm{ISCcurrent}}$ is the value of SAD at the initial search center for the current block. Here, ϵ_1 and ϵ_2 are set to 0.75 and 128, respectively, which are the recommended values [9].

In the table, the percentages of the safe block of the proposed algorithm are similar to or higher than those of Ismail [9]. Meanwhile, the percentages of the unsafe block of the proposed algorithm are less. This means that the proposed algorithm may be slightly more accurate than Ismail's method.

The proposed algorithm can be combined with most existing fast motion estimation algorithms to reduce computational cost. Since the fast motion estimation algorithms adopt different shapes and sizes of search patterns, the overall speed will totally depend on the algorithms. Hence, "Total" in Table 2 is related with only the speed gain by combining the termination method. It represents

Table 2 Accuracy test

Sequence	Method	Safe block	Unsafe block	Total
BQT	$\mathrm{Cond}_{\mathrm{MinSad}}$	36.48 %	26.03 %	62.51 %
	Ismail [9]	3.30 %	28.68 %	31.98 %
BD	$\mathrm{Cond}_{\mathrm{MinSad}}$	15.03 %	21.78 %	36.81 %
	Ismail [9]	6.11 %	54.69 %	60.80 %
K	$\mathrm{Cond}_{\mathrm{MinSad}}$	25.84 %	6.57 %	32.41 %
	Ismail [9]	24.94 %	30.59 %	55.53 %
CS	$\mathrm{Cond}_{\mathrm{MinSad}}$	38.45 %	12.85 %	51.30 %
	Ismail [9]	17.95 %	46.07 %	64.02 %
P	$\mathrm{Cond}_{\mathrm{MinSad}}$	64.36 %	7.99 %	72.35 %
	Ismail [9]	21.50 %	22.85 %	44.35 %
FP	$\mathrm{Cond}_{\mathrm{MinSad}}$	48.21 %	7.28 %	55.49 %
	Ismail [9]	48.74 %	38.72 %	87.46 %
KS	$\mathrm{Cond}_{\mathrm{MinSad}}$	25.34 %	8.45 %	33.79 %
	Ismail [9]	30.70 %	54.05 %	84.75 %
PS	$\mathrm{Cond}_{\mathrm{MinSad}}$	38.17 %	5.43 %	43.60 %
	Ismail [9]	25.36 %	9.33 %	34.69 %
Avg	$\mathrm{Cond}_{\mathrm{MinSad}}$	32.49 %	12.05 %	48.54 %
	Ismail [9]	22.33 %	35.62 %	57.95 %

the percentages of blocks in that at least one search point satisfies the termination condition. There is a chance to terminate early a motion search at those blocks. The higher the percentages of "Total," the higher the speed gain is. Overall, the total percentage of Ismail's method is 9.41 % more than that of the proposed method, and Ismail's method provides the higher speed gain than the proposed algorithm . However, since the proposed algorithm utilizes a different feature to predict the termination condition, the proposed algorithm can be combined with Ismail's method to further improve the search speed. (The experimental results will be presented in the next subsection.) Hence, the proposed method is still meaningful.

3.2 Performance on a video encoder

3.2.1 Simulation environment

In order to evaluate the performance of the proposed termination method on a video encoder, it was implemented on the H.264 reference software JM 18.5 [11] and combined with 'Enhanced Predictive Zonal Search' (SearchMode = 3), which is one of fast motion estimation algorithms. For simulation, the number of reference frames is set to 1 without B frame coding. Rate control and rate-distortion optimized mode decision are turned off. The number of frames for simulation is set to 100. A variable block size mode is turned on. The quantity of ΔPSNR is a PSNR change compared to the conventional method. When the PSNR performance is degraded from

Table 3 Performance of the proposed algorithm by combining with a conventional motion estimation algorithm

Sequence	QP	Conventional method			$\text{Cond}_{\text{MinSad}}$			$\text{Cond}_{\text{MaxSad}}$		
		PSNR (dB)	BR (Mbps)	ME time (ms)	ΔPSNR (dB)	ΔBR (%)	RR	ΔPSNR (dB)	ΔBR (%)	RR
BQT	12	50.34	380.91	20.81	0.01	0.08	0.48	0.03	0.18	0.75
	18	45.12	216.47	19.95	0.00	0.23	0.47	0.01	0.50	0.74
	24	39.21	90.54	18.05	0.00	0.59	0.49	0.00	1.06	0.74
	30	35.09	20.84	15.57	0.00	1.22	0.44	−0.01	1.92	0.71
BD	12	50.08	283.53	27.91	0.01	−0.01	0.22	0.03	−0.02	0.52
	18	44.31	138.69	27.05	0.01	0.15	0.22	0.01	0.37	0.51
	24	39.65	35.65	24.31	0.00	0.22	0.25	−0.01	0.85	0.51
	30	37.49	12.01	21.96	0.00	0.15	0.14	0.00	1.21	0.43
K	12	49.37	109.12	29.94	0.00	0.02	0.16	0.00	0.06	0.38
	18	43.94	36.06	29.51	−0.01	0.12	0.17	−0.01	0.28	0.39
	24	41.86	10.89	27.57	0.00	0.06	0.16	−0.01	0.34	0.35
	30	39.95	4.85	24.86	0.00	−0.02	0.07	−0.01	0.30	0.24
CP	12	51.44	32.19	10.87	0.01	0.52	0.10	0.03	1.75	0.33
	18	47.21	18.24	10.59	0.00	0.64	0.10	−0.01	2.58	0.32
	24	42.81	9.93	10.16	0.00	0.80	0.10	−0.02	3.44	0.32
	30	38.26	5.11	9.39	−0.01	0.79	0.09	−0.03	3.88	0.31
PS	12	49.54	124.18	20.78	0.00	0.02	0.47	0.00	0.06	0.63
	18	43.66	49.82	20.27	0.00	0.05	0.47	0.00	0.20	0.65
	24	39.87	15.23	18.72	0.00	0.12	0.46	0.00	0.60	0.64
	30	36.56	5.61	18.36	0.00	0.05	0.33	−0.01	0.71	0.56
FP	12	49.44	76.49	6.50	0.00	0.07	0.20	0.00	0.09	0.55
	18	45.01	23.02	5.40	0.00	0.07	0.22	0.00	0.25	0.57
	24	42.28	4.48	3.93	0.00	0.16	0.21	0.00	0.80	0.49
	30	39.37	1.39	4.03	0.01	0.29	0.15	0.00	0.73	0.45
KS	12	49.50	66.53	6.55	0.01	0.07	0.09	0.01	0.12	0.53
	18	45.56	19.26	5.05	0.01	0.11	0.08	0.00	0.37	0.54
	24	43.02	3.98	3.96	0.00	0.23	0.06	0.00	0.95	0.46
	30	40.11	1.27	4.04	0.01	0.24	0.04	0.00	1.43	0.37
PS	12	49.96	281.31	72.57	0.00	0.06	0.22	0.01	0.29	0.49
	18	45.09	138.68	70.08	0.00	0.10	0.20	0.01	0.60	0.47
	24	40.91	58.89	65.52	0.00	0.12	0.17	0.00	1.03	0.47
	30	37.50	25.09	60.40	−0.01	0.10	0.11	−0.01	1.20	0.40
Average					0.00	0.23	0.22	0.00	0.88	0.49

^{0}*BR* bitrate, *RR* reduction ratio, *ME* motion estimation the conventional method, ΔPSNR becomes a negative value and vice versa. The quantity of ΔBR is a bitrate change from the conventional method in terms of percentage. When the bitrate is increased from the conventional one, it has a positive value. RR represents a reduction rate as follows:

$$\mathrm{RR} = \frac{\mathrm{METime}_{\mathrm{CONV}} - \mathrm{METime}_{\mathrm{NEW}}}{\mathrm{METime}_{\mathrm{CONV}}} \tag{11}$$

Here, $\mathrm{METime}_{\mathrm{CONV}}$ and $\mathrm{METime}_{\mathrm{NEW}}$ represent motion estimation times for conventional and newly implemented methods, respectively.

3.2.2 Results

Table 3 shows the performances of the conventional method and the proposed algorithms of $\mathrm{Cond}_{\mathrm{MinSad}}$ and $\mathrm{Cond}_{\mathrm{MaxSad}}$. In the table, the conventional method represents the original Enhanced Predictive Zonal Search implemented on the JM 18.5. In $\mathrm{Cond}_{\mathrm{MinSad}}$, a termination condition of $\mathrm{Cond}_{\mathrm{MinSad}}$ in Eq. (5) is combined with the Enhanced Predictive Zonal Search to further reduce motion estimation time. As in the table, the $\mathrm{Cond}_{\mathrm{MinSad}}$ successively reduces the motion estimation time of the original Enhanced Predictive Zonal Search, and the average reduction ratio is 0.22. This simulation explains that the proposed termination condition can further reduce the computational cost of fast motion estimation. As mentioned in Section 2.4, some assumptions in this paper are not valid in general. Hence, while the search speed is improved, the PSNR performance is degraded.

In $\mathrm{Cond}_{\mathrm{MaxSad}}$, a termination condition of $\mathrm{Cond}_{\mathrm{MaxSad}}$ in Eq. (6) is combined with the original Enhanced Predictive Zonal Search. Compared to Fig. 3, the sizes of the gray regions in Fig. 4 are increased. Hence, RR of the $\mathrm{Cond}_{\mathrm{MaxSad}}$ is more than that of the $\mathrm{Cond}_{\mathrm{MinSad}}$. However, since parts of the gray regions are outside the rectangular boxes, the search with $\mathrm{Cond}_{\mathrm{MaxSad}}$ will be sometimes terminated at the improper position, and ΔBR is increased compared to $\mathrm{Cond}_{\mathrm{MinSad}}$ or the conventional method. Table 3 shows the above trade-off between *speed* and *accuracy*. Nowadays, prices for storage devices become cheap. Accordingly, in some applications, an increase of 0.88 % in bitrate is not critical while the reduction of computational cost is important. $\mathrm{Cond}_{\mathrm{MaxSad}}$ will be beneficial for those applications.

Table 4 Performance of the proposed algorithm by combining with a simple threshold technique [4]

Sequence	QP	$\mathrm{Cond}_{\mathrm{MinSad}}$			$\mathrm{Cond}_{\mathrm{MinSad,Sim}}$		
		ΔPSNR (dB)	ΔBR (%)	RR	ΔPSNR (dB)	ΔBR (%)	RR
BQT	12	0.01	0.08	0.48	0.01	0.17	0.58
	18	0.00	0.23	0.47	0.00	0.37	0.58
	24	0.00	0.59	0.49	0.00	0.77	0.59
	30	0.00	1.22	0.44	0.00	1.32	0.48
BD	12	0.01	−0.01	0.22	0.01	0.00	0.25
	18	0.01	0.15	0.22	0.01	0.19	0.25
	24	0.00	0.22	0.25	0.00	0.36	0.28
	30	0.00	0.15	0.14	0.00	0.29	0.15
K	12	0.00	0.02	0.16	0.00	0.04	0.24
	18	−0.01	0.12	0.17	−0.01	0.16	0.25
	24	0.00	0.06	0.16	0.00	0.20	0.22
	30	0.00	−0.02	0.07	0.00	0.02	0.12
CP	12	0.01	0.52	0.10	0.01	1.11	0.18
	18	0.00	0.64	0.10	0.00	1.40	0.18
	24	0.00	0.80	0.10	0.00	1.69	0.18
	30	−0.01	0.79	0.09	−0.01	1.42	0.16
PS	12	0.00	0.02	0.47	0.00	0.02	0.50
	18	0.00	0.05	0.47	0.00	0.06	0.52
	24	0.00	0.12	0.46	0.00	0.16	0.50
	30	0.00	0.05	0.33	0.00	0.05	0.33
FP	12	0.00	0.07	0.20	0.00	0.01	0.50
	18	0.00	0.07	0.22	0.00	0.06	0.52
	24	0.00	0.16	0.21	0.00	0.25	0.50
	30	0.01	0.29	0.15	0.01	0.14	0.33
KS	12	0.01	0.07	0.09	0.01	0.04	0.64
	18	0.01	0.11	0.08	0.01	0.08	0.56
	24	0.00	0.23	0.06	0.00	0.57	0.42
	30	0.01	0.24	0.04	0.01	0.53	0.30
PS	12	0.00	0.06	0.22	0.00	0.11	0.30
	18	0.00	0.10	0.20	0.00	0.22	0.28
	24	0.00	0.12	0.17	0.00	0.31	0.24
	30	−0.01	0.10	0.11	−0.01	0.20	0.16
Average		0.00	0.23	0.22	0.00	0.39	0.36

[0] *BR* bitrate, *RR* reduction ratio, *ME* motion estimation

Table 4 represents the performance of the proposed method by combining it with a simple threshold technique, which was first introduced in MVFAST [4]. The termination condition of $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ is described in Eq. (7). The proposed algorithm sets a threshold value to $2B_{\mathrm{W}}B_{\mathrm{H}}$ to support the variable block size mode. Note here that the threshold value for MVFAST [4] was set to 512. In the table, while the coding efficiency is degraded by 0.16 %, the reduction rate is 14 % (or 0.14). This simulation shows that the proposed termination method can be merged with the existing early search termination to further reduce the computational cost. For the sequences of KS, the speed of $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ is much faster than that of $\mathrm{Con}_{\mathrm{MinSad}}$. Meanwhile, the RR of $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ for the sequence of PS is almost the same as that of $\mathrm{Cond}_{\mathrm{MinSad}}$. The main reason is that the proposed method utilizes the self-characteristic of a current block, which is different from the constant thresholding technique. In other words, the gains of two termination algorithms are not correlated.

Table 5 represents the comparison of early termination conditions between the proposed method and Ismail's

Table 5 Performance comparison

Sequence	QP	Ismail[9]			$\mathrm{Cond}_{\mathrm{MinSad,Sim}}$			Ismail + $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$		
		ΔPSNR (dB)	ΔBR (%)	RR	ΔPSNR (dB)	ΔBR (%)	RR	ΔPSNR (dB)	ΔBR (%)	RR
BQT	12	0.02	0.05	0.29	0.01	0.17	0.58	0.02	0.15	0.67
	18	0.00	0.14	0.29	0.00	0.37	0.58	0.00	0.39	0.67
	24	−0.01	0.21	0.27	0.00	0.77	0.59	−0.01	0.79	0.66
	30	0.00	0.11	0.21	0.00	1.32	0.48	−0.01	1.32	0.59
BD	12	0.02	−0.11	0.28	0.01	0.00	0.25	0.03	−0.07	0.39
	18	0.01	0.13	0.28	0.01	0.19	0.25	0.01	0.25	0.38
	24	−0.01	0.03	0.28	0.00	0.36	0.28	−0.01	0.36	0.38
	30	0.00	0.04	0.20	0.00	0.29	0.15	0.00	0.35	0.27
K	12	0.00	−0.02	0.31	0.00	0.04	0.24	0.00	0.01	0.37
	18	−0.01	0.08	0.33	−0.01	0.16	0.25	−0.01	0.15	0.38
	24	−0.01	−0.08	0.29	0.00	0.20	0.22	−0.01	0.08	0.33
	30	0.00	−0.02	0.22	0.00	0.02	0.12	0.00	0.06	0.26
CS	12	0.01	0.43	0.12	0.01	1.11	0.18	0.02	1.32	0.25
	18	−0.01	0.59	0.12	0.00	1.40	0.18	0.00	1.75	0.24
	24	−0.01	0.41	0.13	0.00	1.69	0.18	−0.01	1.82	0.24
	30	0.00	0.09	0.10	−0.01	1.42	0.16	−0.01	1.48	0.22
P	12	0.00	−0.01	0.40	0.00	0.02	0.50	0.00	0.01	0.60
	18	0.00	0.01	0.40	0.00	0.06	0.52	0.00	0.07	0.60
	24	0.00	0.09	0.32	0.00	0.16	0.50	0.00	0.22	0.57
	30	0.00	0.08	0.24	0.00	0.05	0.33	0.00	0.20	0.48
FP	12	0.00	0.00	0.59	0.00	0.01	0.50	0.00	0.01	0.67
	18	−0.01	−0.01	0.54	0.00	0.06	0.52	0.00	0.04	0.61
	24	0.00	0.16	0.39	0.00	0.25	0.50	0.00	0.32	0.48
	30	0.00	0.01	0.29	0.01	0.14	0.33	0.01	0.27	0.44
KS	12	0.01	0.01	0.64	0.01	0.04	0.64	0.01	0.06	0.70
	18	0.00	0.03	0.56	0.01	0.08	0.56	0.00	0.12	0.62
	24	0.00	0.13	0.44	0.00	0.57	0.42	0.00	0.75	0.51
	30	0.00	0.35	0.32	0.01	0.53	0.30	0.00	0.50	0.44
PS	12	0.01	0.05	0.22	0.00	0.11	0.30	0.01	0.16	0.38
	18	0.00	0.17	0.20	0.00	0.22	0.28	0.00	0.36	0.35
	24	−0.01	0.19	0.16	0.00	0.31	0.24	−0.01	0.48	0.30
	30	−0.01	0.10	0.11	−0.01	0.20	0.16	−0.01	0.31	0.22
Average		0.00	0.11	0.30	0.00	0.39	0.36	0.00	0.44	0.45

[0] *BR* bitrate, *RR* reduction ratio, *ME* motion estimation

method [9], which is one of the state-of-the-art methods. The early termination of Ismail [9] in Eq. (6) is also implemented on the JM reference software 18.5 by combining it with Enhanced Predictive Zonal Search. In order to support the variable block size mode, Eq. (6) is modified as follows:

$$T_{\mathrm{S}} = \mathrm{MIN}[\mathrm{MAX}(\mathrm{SAD}_{\mathrm{ISCavg}}, 2B_{\mathrm{W}}B_{\mathrm{H}}), \mathrm{SAD}_{\mathrm{ISCcur}}] \times \epsilon_1 + \epsilon_2 \tag{12}$$

Here, B_{W} and B_{H} are the width and height of a block to be considered. As in the table, $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ successively reduces the motion estimation time, and the average reduction ratio is 0.36. While the average reduction ratio of $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ is better than that of Ismail, the PSNR performance of $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$ is worse than that of Ismail. However, although the proposed method does not outperform Ismail's method, the proposed algorithm can be used to complement Ismail's method, in order to further improve its search speed. While Ismail's method considers the SAD values of blocks temporally and spatially neighboring to determine the termination condition, the proposed algorithm takes into account the self-characteristic of a current block for calculating the termination condition. Since their termination conditions are exclusive, the merged algorithm further accelerates the search speed. Two termination conditions of Ismail's and proposed methods are simultaneously considered in Ismail + $\mathrm{Cond}_{\mathrm{MinSad,Sim}}$, in order to show how the

Table 6 Performance comparisons of Cond$_{\mathrm{MaxSad}}$, Cond$_{\mathrm{MinSad}}$, Cond$_{\mathrm{MinSad,Sim}}$, Ismail [9], and Ismail + Cond$_{\mathrm{MinSad,Sim}}$

	ΔPSNR (dB)	ΔBR (%)	RR
Cond$_{\mathrm{MaxSad}}$	0.00	0.88	0.49
Cond$_{\mathrm{MinSad}}$	0.00	0.23	0.22
Cond$_{\mathrm{MinSad,Sim}}$	0.00	0.39	0.36
Ismail [9]	0.00	0.11	0.30
Ismail + Cond$_{\mathrm{MinSad,Sim}}$	0.00	0.44	0.45

[0] *BR* bitrate, *RR* reduction ratio, *ME* motion estimation

proposed algorithm complements Ismail's method. The method chooses the maximal threshold value between Ismail's and Cond$_{\mathrm{MinSad,Sim}}$ to combine two algorithms together. The reduction ratio of Ismail + Cond$_{\mathrm{MinSad,Sim}}$ is more than those of Cond$_{\mathrm{MinSad,Sim}}$ and Ismail's method. This simulation results show that the proposed algorithm also complements Ismail's method. Hence, although the proposed algorithm does not outperform the existing methods, it can be used to accelerate the search speed, and it makes the proposed algorithm meaningful.

Table 6 shows the overall performance comparisons of Cond$_{\mathrm{MaxSad}}$, Cond$_{\mathrm{MinSad}}$, Cond$_{\mathrm{MinSad,Sim}}$, Ismail [9], and Ismail + Cond$_{\mathrm{MinSad,Sim}}$.

4 Conclusions

This paper presents a method to terminate early a motion search for a video encoder. Unlike the previous work, this paper introduces a SAD estimator by considering a two-dimensional space and studies the estimator to get a condition for terminating a motion search early. The proposed algorithm can easily be combined with most of the methods of fast motion estimation to reduce computational cost. While previous thresholding techniques consider SAD values of blocks neighboring a current block, the proposed algorithm takes into account the characteristic of the current block to predict the threshold. Hence, the proposed termination scheme can complement the previous early search termination techniques.

Competing interests

The author declares no competing interests.

Acknowledgements

This present research has been conducted through the Research Grant of Kwangwoon University in 2014. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2014R1A1A2054105).

Received: 13 February 2015 Accepted: 11 August 2015
Published online: 02 September 2015

References

1. T Wiegand, GJ Sullivan, G Bjøntegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circ. Syst. Video Technol. **13**(7), 560–576 (2003)
2. GJ Sullivan, J Ohm, WJ Han, T Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1649–1668 (2012)
3. S Zhu, K-K Ma, A new diamond search algorithm for fast block matching motion estimation. IEEE Trans. Image Process. **9**(2), 287–290 (2000)
4. PI Hosur, KK Ma, in *Proc. 2nd Int. Conf. Information Communications and Signal Processing (ICICS '99)*. Motion vector field adaptive fast motion estimation (Singapore, Dec. 1999), pp. 7-10
5. AM Tourapis, OC Au, ML Liou, Highly efficient predictive zonal algorithms for fast block-matching motion estimation. IEEE Trans. Circ. Syst. Video Technol. **12**(10), 934–947 (2002)
6. C Zhu, X Lin, LP Chau, KP Lim, HA Ang, CY Ong, in *Proc. ICASSP*. A novel hexagon-based search algorithm for fast block motion estimation (Utah, USA, 2001), pp. 1593–1596. 7-11 May 2001
7. ZB Chen, P Zhou, Y He, in *JVT-F017, 6th Meeting, Awaji*. Fast integer pel and fractional pel motion estimation for JVT (Awaji, Japan, Dec. 2002)
8. Y Ismail, M Shaaban, M Bayoumi, in *Proc. IEEE ISCAS*. An adaptive block size phase correlation motion estimation using adaptive early search termination technique (New Orleans, USA, 2007), pp. 3423–3426. 27-30 May 2007
9. Y Ismail, JB McNeely, M Shaaban, H Mahmoud, MA Bayoumi, Fast motion estimation system using dynamic models for H.264/AVC video coding. IEEE Trans. Circ. Syst. Video Technol. **22**(1), 28–42 (2012)
10. Y Lee, JB Ra, Fast motion estimation robust to random motions based on a distance prediction. IEEE Trans. Circ. Syst. Video Technol. **16**(7), 869–875 (2006)
11. JM reference software (Aug 1 2014). http://iphome.hhi.de/suehring/tml/