BACKGROUND-FOREGROUND DIVISION BASED SEARCH FOR MOTION ESTIMATION IN SURVEILLANCE VIDEO CODING

Long Zhao[1,2], *Yonghong Tian*[2*], *Tiejun Huang*[2]

[1]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen 518055, Guangdong, China
[2]School of EE & CS, Peking University, Beijing 100871, China
*Corresponding to: yhtian@pku.edu.cn

ABSTRACT

Basically, motion search is very time-consuming in the process of video coding. For surveillance videos, however, there exist a large amount of static background regions whose motion vectors actually are equal to zero. By utilizing the background and foreground information of coding units, this paper proposes a background-foreground division based search algorithm (BFDS) to accelerate the motion search in surveillance video coding. The basic idea of BFDS is to classify a predicting unit (PU) into a background predicting unit (BPU) or a foreground predicting unit (FPU) and then adopt different search strategies respectively for BPUs and FPUs. That is, a zero motion vector biased search strategy is applied in BPUs to reduce the search complexity on a large scale while a precise global search strategy is applied in FPUs to get higher coding performance. Compared with the current TZ search algorithm used in HEVC, the proposed BFDS algorithm can reduce the number of search points by 57.73% while remaining the coding performance almost unchanged.

Index Terms—Motion Search, Surveillance Videos, Predicting Unit, BFDS, Search Complexity, HEVC

1. INTRODUCTION

In video coding, motion estimation plays an important role in reducing the temporal redundancy. In general, a good motion estimation method can improve video coding efficiency by increasing the prediction accuracy. In order to find a best-matching block in reference frames, however, the time-consuming motion search procedure is often needed. Thus to decrease the complexity of the search, it is desirable to develop robust and fast search algorithms.

Typically, there are two approaches to achieving this goal. The first one is to design the fast search patterns. In general, the 2-D logarithmic search [1] and three-step search [2] methods are suitable for catching large motion while the center-biased three-step search [3], four-step search [4] and diamond search [5] methods can better catch the small motion. It should be noted that the unsymmetrical-cross multi-hexagon grid search [6] method used in H.264 and the TZ search method [7] used in HEVC exactly combine the two kinds of search patterns to achieve more precise motion estimation.

Instead, the second approach is the early-termination of motion search. In [8], the zero-motion termination method was proposed to compare the sum of absolute difference value (SAD) with a threshold at the zero motion vector (MV) point to determine whether to stop the search procedure. An early-termination method for TZ search by testing whether the median predictor was the best search point among its neighbors was proposed in [9].

These methods, however, are not optimal for surveillance video coding since they do not make best use of the special characteristics of surveillance videos (e.g., relatively fixed background in a video captured by stationary cameras). For example, there always exist some static background regions in a surveillance video whose MVs are actually equal to zero. This motivates us that the background/foreground regions should have a strong relationship with the motion characteristic in the video. Thus by utilizing the background and foreground information (BFI), a background-foreground division based search algorithm (BFDS) is proposed in this paper. Basically, BFDS classifies a predicting unit (PU) into a background predicting unit (BPU) or a foreground predicting unit (FPU) and then adopts different search strategies respectively for BPUs and FPUs. That is, a zero motion vector biased search strategy is applied in BPUs to reduce the search complexity on a large scale while a precise global search strategy is applied in FPUs to get higher coding performance. As a result, the early-termination strategy can be more accurate in background regions while the search pattern is designed finer to catch both large and small motions in foreground regions. Moreover, the BFI in reference frames can also restrict the search area to reduce the number of search points.

BFDS can be used in various fast search methods over different video coding platforms since the motion search method of FPUs in BFDS is just modified from the existing one. In particular, BFDS is applied in the recent HEVC [10] platform in this paper. The quad-tree structure in HEVC makes the coding unit (CU) divided in different sizes from 64x64 to 8x8 and the PUs in different division patterns such as 2Nx2N, NxN, 2NxN, Nx2N and etc. The PUs of variable sizes can be adaptively classified into background regions and foreground regions such that a suitable algorithm can be designed to them so as to reduce the complexity of motion search while maintaining the coding performance.

In the newest HEVC reference software HM-12.0, BFDS is added to the TZ search method. By combining the coding unit tree structure and PU patterns, BFDS is adaptive to different PU sizes. Compared to the original TZ search, BFDS can reduce the number of search points by averagely 57.73% while the degradation in video quality is negligible.

The rest of the paper is organized as follows. Sec.2 analyzes why to utilize BFI in the motion search algorithm. Sec.3 describes BFDS and the optimization of BFDS is presented in Sec.4. Experimental results are given in Sec.5 and the paper is concluded in Sec.6.

2. PROBLEM ANALYSIS

The background regions are static in surveillance videos so that the difference of blocks in the corresponding position of the current frame and the reference frame is small. The SAD distribution on the positions of searched blocks in the reference frame is shown in Fig.1, with the range of -16 to 16 in both horizontal and vertical directions. The scene of the experimental sequence is an overpass over a road, with slowly moving pedestrians and fast moving vehicles, which can be representative of motion characteristics of surveillance videos. From the Fig.1, it can be seen that the SAD in position (0,0) is the smallest, which proves that there tends to be no motion in the background regions.

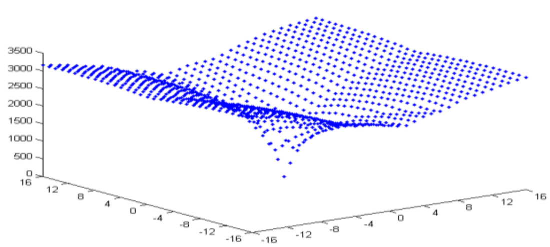


Fig.1 SAD distribution of positions of search blocks

Further on, the MV distribution characteristic is the most significant feature on the background and foreground regions for surveillance videos. The background regions are static so the PU can find the best matching block in the same position in the reference frame. The foreground regions, however, always contain moving objects which require MVs to find the best matching block in the reference frame.

Statistics on MV distribution is shown in Fig.2. It can be seen that for PUs classified into background regions, over 95% of MVs equal to zero and there nearly exist no MVs larger than 1 pixel. MVs of PUs which are classified into foreground regions, however, are central-biased distributed, namely a large proportion of MVs concentrate in the range less than 2 pixels while there are also about 20% of MVs larger than 5 pixels.

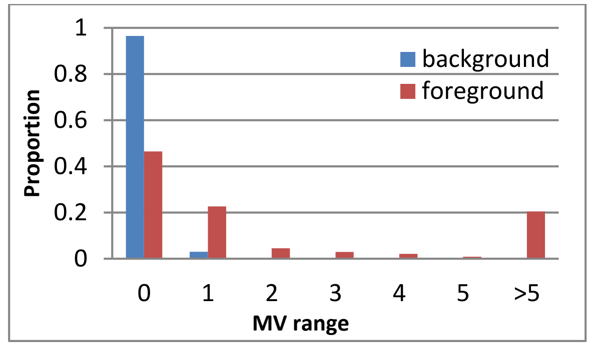


Fig.2 Statistics on MV distribution

It can be concluded that for background regions, a zero MV biased search strategy should be applied. The early termination in background regions can save the number of search points on a large scale. For foreground regions, a coarse-to-fine search strategy should be applied to catch the large motion to avoid the local optimum as well as to catch the small motion to achieve more accurate prediction.

3. THE BFDS ALGORITHM

Based on the MV distribution statistics, the background-foreground division based search algorithm (BFDS) is proposed to accelerate the motion search procedure.

3.1 The framework of BFDS

The framework of BFDS is shown in Fig.3. Firstly, the background frame is trained. PUs of the current frame are then classified into BPUs or FPUs. According to the PU type, the BPU and the FPU search strategies are applied respectively. By utilizing the BFI of the reference frame, an early termination algorithm is applied to skip the unnecessary search points in the FPU search strategy.

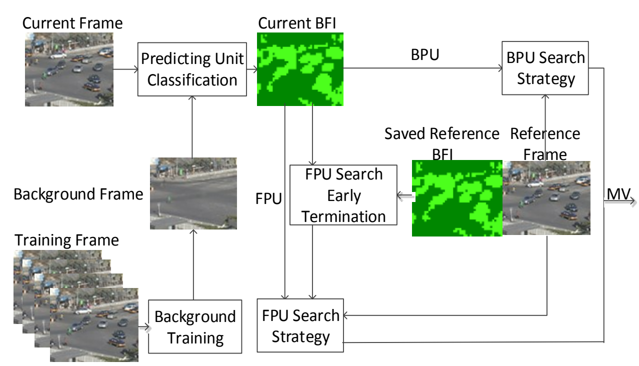


Fig.3 The framework of BFDS

3.2 The PU classification

In the PU classification, a background frame is firstly constructed. The running average background modeling method which uses the means and weights of the temporal pixels [11] is utilized in building the background frame for its low complexity and high performance. The constructed background frame is shown in Fig.4 (a).

With the constructed background frame, PUs in the current frame (Fig.4 (b)) are classified into BPUs and FPUs. Firstly, each 8x8 block is classified into a background block or a foreground block by comparing SAD of the 8x8 blocks in the current frame and the background frame with a threshold [12]. The BFI of the current frame in the precision of 8x8 is shown in Fig.4 (c). Then, a threshold $T_{classify}$ is introduced to classify the PU into a BPU or a FPU, which is calculated in the following equation,

$$T_{classify} = F_{classify} \times (W_{pu}/8) \times (H_{pu}/8) \quad (1)$$

where $F_{classify}$ is a constant factor, W_{pu} and H_{pu} are the width and height of the PU. Thirdly, the number of foreground 8x8 blocks the PU contains is calculated by utilizing the BFI. If the value is larger than $T_{classify}$, the PU is classified into a FPU, otherwise into a BPU.

(a)Background frame (b)Current frame (c)BFI

Fig.4 The background frame and the classification

3.3 The BPU search strategy

The BPU is identified as a static unit which has no motion. According to the PU classification criteria, the definition of a BPU has relationship with the size of PU. When the size of the PU is small, the BPU should have fewer foreground blocks to ensure that the PU has clear background region to provide precise prediction. When the size of the PU is large, however, the tolerance of foreground blocks can be larger. The reason is that although the best matching point for the large BPU is not in the zero MV point, the CU containing this BPU will tend to be divided into small CUs in the quad-tree way since the foreground region of the BPU could find better matching region in smaller PUs. As a result, the large PU can be classified into a BPU in relative loose criteria to avoid the unnecessary search points.

When a PU is classified into a BPU, the search strategy is simply setting the MV to zero since there should be no motion in the BPU according to the statistics on MV distribution and the BPU classification criteria. Experiments show that the zero MV strategy is most effective in consideration of both the search complexity and coding performance. The reason is that if a more complex search strategy is applied in BPUs, the best matching point is easy to be trapped into local optimum. By utilizing the static characteristic of background region in surveillance videos, the compulsive zero MV strategy has higher probability to get global optimum and thus has better search performance.

In the TZ search strategy in HEVC reference model HM-12.0, each PU should search for at least 24 points. As a result, the BPU search strategy can save at least 23 search points for each BPU.

3.4 The FPU search strategy

In balance of the coding performance and search complexity of the search procedure, a more precise search strategy is adopted to increase the coding performance for FPUs since the BPU search strategy has reduced the search complexity greatly. The FPU search strategy takes advantage of both a more precise search pattern and an early-termination algorithm to achieve better search performance.

3.4.1 The search pattern

The FPU search pattern is based on TZ search and modifies TZ search to a more precise level to achieve the goal mentioned above. The FPU search pattern is shown in Fig.5 and described as the following steps.

Step1: From the search center, conduct the multi-diamonds search. The radiuses of the diamonds are 1, 2, 4, 8, and 16.

Step2: If the minimum block distortion point (MBDP) is in the search center, go to Step 3; otherwise, set the MBDP as the new search center and go back to Step1.

Step3: Execute the 11x11 rectangular search with the search center got in Step2.

In this FPU search pattern, the diamond search in the radius of 32 and the zonal search are omitted compared to the original TZ search because the iterative search in Step 2 can also catch large MVs. Meanwhile, Step 3 is designed for catching MVs in a more precise level.

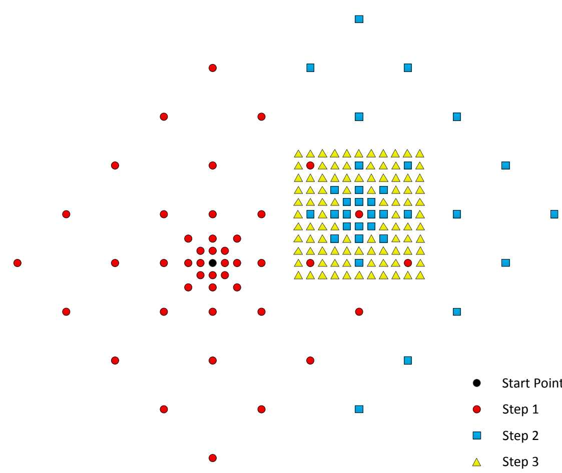


Fig.5 FPU search pattern

3.4.2 The early-termination algorithm

Traditionally, the motion search area in the reference frame is restricted in a rectangular window. For the FPU motion search, BFI in the reference frame can be utilized to restrict the motion search area.

As shown in Fig.6, both the current frame and the reference frame have BFI. For the FPU in the current frame, the search is firstly restricted in the rectangular search window in the reference frame. Since the FPU represents an object in the current frame and the object also occurs as foreground in the reference frame, the best matching block should be in the foreground region in the reference frame. Therefore, the motion search conducted in the background regions in the reference frame should be skipped directly.

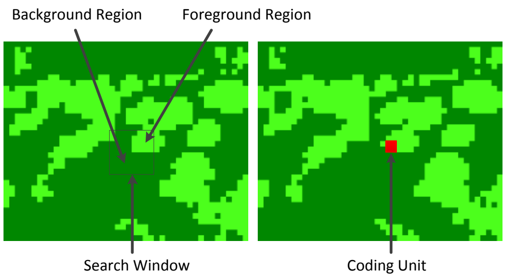


Fig.6 BFI of the reference frame and current frame

Under the restriction of BFI in the reference frame, a number of points can be saved in the FPU search category. To classify a searched block into a background block or a foreground block in low complexity, a block matching method in low complexity is proposed to judge whether the block should be searched or not. Four corner points in both the reference block and the current PU are selected and the BFI of the four point pairs is examined to test whether they are matched. If the number of matching points is larger than a threshold T_{match}, the block is searched; otherwise, the block is skipped.

By adopting the early-termination algorithm, the complexity of the FPU search strategy can be reduced.

In this chapter, the proposed BFDS algorithm is described. PUs are firstly classified into BPUs and FPUs. The zero MV search strategy is then adopted for BPUs and a modification of the TZ search is adopted for FPUs in balance of the coding performance and search complexity. Further on, an early-termination method is proposed in the FPU search strategy to reduce the motion search complexity.

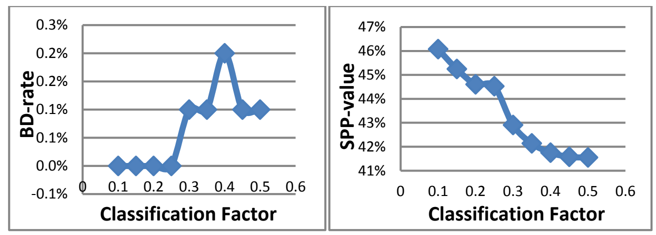
4. THRESHOLDS OPTIMIZATION

There are two thresholds in the proposed BFDS algorithm to be optimized. The first is $T_{classify}$, the threshold to classify a PU into a BPU or FPU. The second is T_{match}, the threshold to judge whether a block in the reference frame should be searched or skipped.

4.1 $T_{classify}$ optimization

$F_{classify}$ is trained to get the optimum $T_{classify}$. The range of $F_{classify}$ is set from 0.1 to 0.5, with a step of 0.05. The search performance measurement is a combination of the BD-rate [13] and the Search-Points-Proportion value (SPP-value). BD-rate represents the coding efficiency and SPP-value is the ratio of the number of the search points in the BFDS divided by that in the TZ search in HM 12.0. The results are shown in Fig.7 (a) and Fig.7 (b).

From the results, it can be seen when $F_{classify}$ falls in the range 0.1 to 0.25, there is no loss in coding performance. The SPP-value has a decreasing trend as $F_{classify}$ increases. The reason is when $F_{classify}$ is larger, more PUs are classified into BPUs and the BPU search strategy saves more search points. Meanwhile, the coding performance decreases because when PUs containing some foreground regions are classified into BPUs improperly, the search procedure is unable to find a good matching block. When $F_{classify}$ is small, however, the coding performance will not decrease since the BPUs have very little foreground region and the matching position will depend greatly on the background region, thus the zero MV will always be the best choice.



(a)Coding performance (b) Search complexity

Fig.7 Search performance on different values of $F_{classify}$

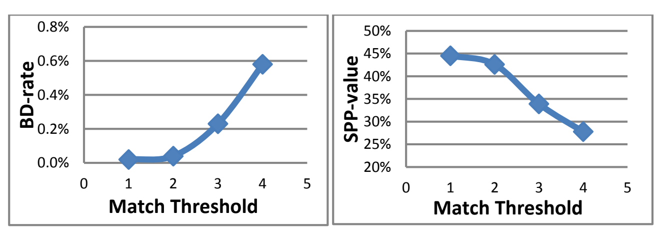
Restricted by the coding performance, 0.25 is selected for $F_{classify}$ to achieve the lowest search complexity. The values of $T_{classify}$ for various PU sizes are shown in Tab.1, which cope with the assumption that the tolerance of foreground region in a BPU increases as the size of PU becomes larger.

Table 1. $T_{classify}$ values for various PU sizes

PU size	4x4 4x8 8x4	8x8 8x16 16x8	16x16	16x32 32x16	32x32	32x64 64x32	64x64
$T_{classify}$	0	0	1	2	4	8	16

4.2 T_{match} optimization

In the FPU early-termination algorithm, T_{match} equal to 0 means that the fast algorithm is prohibited and T_{match} equal to 5 means that no search is done in FPU. Thus the meaningful values for T_{match} are 1, 2, 3 and 4. BD-rate and SPP-value are also selected as the measurement of the early-termination algorithm performance. The result is shown in Fig.8 (a) and Fig.8 (b).



(a)Coding performance (b) Search complexity

Fig.8 Search performance on different values of T_{match}

From the results, it can be seen that the BD-rate increases when T_{match} becomes larger. The reason is that when T_{match} becomes larger, the search block must match the current PU better to continue the search procedure, thus having more chance to early termination. As a result, the coding performance will become worse and the search complexity will decrease. Taking the coding performance as a precondition, the value of 1 is set to T_{match} since there is no loss in the coding performance at this value.

In this chapter, parameters for the PU classification criteria and the early-termination criteria are respectively optimized. With the modified thresholds, the BFDS algorithm can get higher motion search performance.

5. EXPERIMENTS

5.1 Experimental settings

Experiments were conducted on HM-12.0 and the search range is set to 64. Eight surveillance videos from the AVS workshop [14] are selected, including six SD (720x576) videos and two HD (1600x1200) videos, which are shown in Fig.9. The scenes of the videos include dark and bright (DA/BR), indoor and outdoor (ID/OD), large foreground and small foreground (LF/SF), fast motion and slow motion (FM/SM). 50 frames are tested in the experiment.

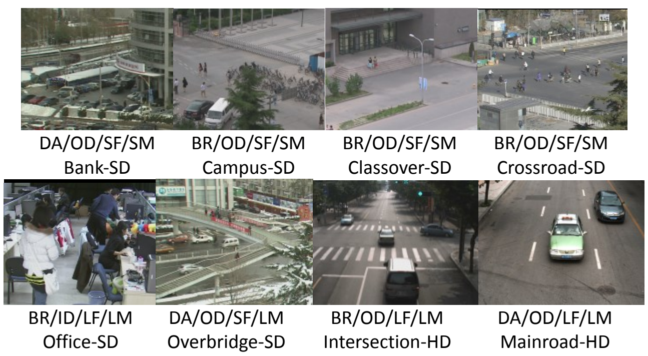


Fig.9 Scenes of the surveillance videos

The PU classification threshold $T_{classify}$ is set according to Tab.1, determined by the size of PU. The block matching threshold T_{match} is set to 1 to get the best search performance. The TZ search in HM-12.0 is selected as the anchor and the early-termination TZ search algorithm (ETTZ) proposed in [9] is tested for comparison. BD-rate and SPP-value are adopted as the evaluation criteria for motion search performance.

5.2 Experimental results

The results are shown in Tab.2 and Tab.3. From the experimental results, it can be concluded that there is no loss in the coding performance of the videos when adopting BFDS compared to the original TZ search algorithm. The motion search complexity is reduced on a large scale, achieving 55.47% for SD videos and 64.51% for HD videos respectively. Besides, BFDS outperforms ETTZ in both the search complexity and coding performance. The experimental results show that the BFDS algorithm is able to reduce the motion search complexity while maintaining the coding performance.

Table 2. Coding performance of BFDS and ETTZ

Sequence		BD-rate	
		BFDS	**ETTZ**
Bank-SD		0.0%	0.0%
Campus-SD		0.1%	0.0%
Classover-SD		0.0%	-0.2%
Crossroad-SD		0.1%	0.1%
Office-SD		-0.1%	0.3%
Overbridge-SD		0.1%	0.0%
Intersection-HD		0.2%	0.1%
Mainroad-HD		-0.2%	-0.1%
Average	**SD**	**0.0%**	**0.1%**
	HD	**0.0%**	**0.0%**
	ALL	**0.0%**	**0.0%**

Table 3. Motion search complexity of BFDS and ETTZ

Sequence		BFDS		ETTZ	
		SPP	**Save**	**SPP**	**Save**
Bank-SD		31.48%	68.52%	59.48%	40.52%
Campus-SD		43.02%	56.98%	60.66%	39.34%
Classover-SD		21.18%	78.82%	56.62%	43.38%
Crossroad-SD		60.56%	39.44%	68.05%	31.95%
Office-SD		69.82%	30.18%	68.06%	31.94%
Overbridge-SD		41.12%	58.88%	64.11%	35.89%
Intersection-HD		51.43%	48.57%	61.04%	38.96%
Mainroad-HD		19.55%	80.45%	64.60%	35.40%
Average	**SD**	**44.53%**	**55.47%**	**62.83%**	**37.17%**
	HD	**35.49%**	**64.51%**	**62.82%**	**37.18%**
	ALL	**42.27%**	**57.73%**	**62.83%**	**37.17%**

5.3 Experimental analysis

The whole BFDS framework adds few computations in the background modeling and the PU classification. The background modeling method needs only three add operations and one multiplication operation for each pixel in the training frame [11], while the PU classification needs one subtraction operation and one add operation for each pixel [12]. Thus the complexity increase introduced by these two modules is fairly low.

As for the search complexity reduction performance, four variables are introduced to represent the characteristic of surveillance videos. The first is the background proportion p_b, which represents the foreground density of the scene. The second is the proportion of the zero MVs in the foreground regions $p_{fgZeroMV}$, which represents the static characteristic of foreground regions. The third is the proportion of the large MVs in foreground regions $p_{fgLargeMV}$, which represents the large motion degree of the foreground objects. Further on, $p_{allZeroMV}$ represents the static characteristic of the whole regions of the video, and it can be calculated in the following equation,

$$p_{allZeroMV} = (1 - p_b)p_{fgZeroMV} + p_b \qquad (2)$$

According to the search pattern in BFDS and TZ search, the SPP-value can be approximately calculated in the following equation,

$$P(p_b, p_{fgZeroMV}, p_{fgLargeMV}) = \frac{p_b + (1 - p_b)(123 + 44p_{fgZeroMV} + 99(1 - p_{fgzeroMV}) + 88p_{fgLargeMV}}{122 - 100\left((1 - p_b)p_{fgZeroMV} + p_b\right) + 853p_{fgLaregeMV}} \qquad (3)$$

From Eq.3, we can see that if p_b and $p_{fgZeroMV}$ are larger and $p_{fgLargeMV}$ is smaller, the SPP-value will be smaller, which proves that the BFDS algorithm works better when the background regions of the video are more exposed and the foreground motion is smaller. The comparison of the experimental and the calculation results are shown in Tab.4. There is only an average of 5.83% difference between the experimental and the calculation results, which proves that the calculation can estimate the search complexity performance approximately.

Table 4. Comparison of the experimental and calculation results

Sequence		Experiment	Calculation	Difference
Bank-SD		31.48%	31.04%	0.44%
Campus-SD		43.02%	45.79%	2.77%
Classover-SD		21.18%	20.31%	0.87%
Crossroad-SD		60.56%	65.24%	4.68%
Office-SD		69.82%	98.50%	28.68%
Overbridge-SD		41.12%	42.46%	1.34%
Intersection-HD		51.43%	46.69%	4.74%
Mainroad-HD		19.55%	22.70%	3.15%
Average	**SD**	**44.53%**	**50.56%**	**6.46%**
	HD	**35.49%**	**34.70%**	**3.95%**
	ALL	**42.27%**	**46.59%**	**5.83%**

6. CONCLUSION

To accelerate the motion search in surveillance video coding, this paper proposed a background-foreground division based search algorithm (BFDS). By utilizing the background and foreground information (BFI) of coding units and the statistics of motion vectors in surveillance videos, BFDS adopts different search strategies for background regions and foreground regions. Moreover, BFDS can be added to various search methods over different video coding platforms. Especially in this paper, the BFDS algorithm is added to the TZ search in HEVC reference software by combining the BFI and the coding unit tree structure. The experimental results show that BFDS can reduce 57.73% of the search complexity while maintaining the comparable coding performance. In the future work, we will explore a more precise search strategy in foreground regions and an adaptive search strategy in background regions in the case that the camera shakes.

7. ACKNOLEDGEMENT

This work is partially supported by grants from the National Natural Science Foundation of China under contract No. 61390515, No. 61035001 and No. 61121002.

8. REFERENCES

[1] Jain, Jaswant, and Anil Jain. "Displacement measurement and its application in interframe image coding." IEEE Trans. Communication 29(12), pp. 1799-1808, 1981.

[2] Koga, T. "Motion-compensated interframe coding for video conferencing." Proc. NTC'81, 1981.

[3] Li, Reoxiang, Bing Zeng, and Ming L. Liou. "A new three-step search algorithm for block motion estimation." IEEE Trans. Circuits Syst. Video Tech. 4(4), pp. 438-442, 1994.

[4] Po, Lai-Man, and Wing-Chung Ma. "A novel four-step search algorithm for fast block motion estimation." IEEE Trans. Circuits Syst. Video Tech. 6(3), pp. 313-317, 1996.

[5] Zhu, Shan, and Kai-Kuang Ma. "A new diamond search algorithm for fast block-matching motion estimation." IEEE Trans. Image Proc. 9(2), pp. 287-290, 2000.

[6] Chen, Zhibo, P. Zhou, and Y. He. "Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H. 264." Proc. Picture Coding Sym. 2003.

[7] JVT of ISO/IEC MPEG, ITU-T VCEG, MVC software Reference Manual-JMVC 8.2, May 2010.

[8] Yang, Jar-Ferr, Shih-Cheng Chang, and Chin-Yun Chen. "Computation reduction for motion search in low rate video coders." IEEE Trans. Circuits Syst. Video Tech. 12(10) , pp. 948-951, 2002.

[9] Zhaoqing Pan, et al, “Early Termination for TZSearch in HEVC Motion Estimation.” IEEE Conf. Acoustics, Speech, Signal Proc. 2013.

[10] Bross, B., et al. "High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003." Joint Collaborative Team on Video Coding (JCT-VC), Stockholm, Sweden, 2012.

[11] Zhang, Xianguo, et al. "Low-complexity and high-efficiency background modeling for surveillance video coding." IEEE Conf. Visual Com. Image Proc. 2012.

[12] Zhao, Long, et al. "A background proportion adaptive Lagrange multiplier selection method for surveillance video on HEVC." IEEE Conf. Multimedia Expo, 2013.

[13] Bjontegard, Gisle. "Calculation of average PSNR differences between RD-curves." ITU-T VCEG-M33, 2001.

[14] ftp://124.207.250.92/public/seqs/video.