Expanding Line Search for Panorama Motion Estimation

Ke Chen, Zhong Zhou, Ben Niu, Jingxiang Chen, Wei Wu
State Key Laboratory of Virtual Reality Technology and Systems
School of Computer Science and Engineering, Beihang University
Beijing, China
E-mail: chenke@vrlab.buaa.edu.cn

Abstract—**This paper describes an effective motion estimation algorithm for panoramic video. According to the characteristics of the block motion in panoramic video, the proposed algorithm extends the reference frames and constructs search lines for the Line Search. With the constructed search lines, the Line Search estimates the motion of the corresponding macro block. It starts at the block matched by the Line Search and estimates the motion of adjacent blocks. Experiment results show that the algorithm can estimate the motion of macro blocks in cubic panoramic video effectively. It is also shown that the algorithm improves the search speed with effective motion estimation.**

Keywords-panoramic video; motion estimation; Line Search

I. INTRODUCTION

Panoramic video covers 360°×180° view of scenes. Its huge amount of data makes it rather consuming to store or transmit, so encoding and compression are very important for panoramic video. Motion estimation is fundamental in video compression.

The simplest and most effective method of motion estimation is the FS algorithm, which searches all the candidates in the search range and finds the best-match position with the lowest distortion. Because all the candidates are required to be searched, the FS algorithm needs lots of computation. Then many fast motion estimation algorithm are proposed, such as the TSS (Three Step Search) algorithm [1], the CS (Cross Search) algorithm [2], the NTSS (New Three Step Search) algorithm [3], the FSS (Four Step Search) algorithm [4], the DS (Diamond Search) algorithm [5], the HS (Hexagon Search) algorithm [6] and the PLS (Predictive Line Search) algorithm [7]. The TSS and the CS algorithms use a large search step at the first search and the further searches are based on the first search. So, the two algorithms may easily fall into the local optimization rather than achieve the global optimization. According to the characteristic of motion vectors, the NTSS, FSS and DS algorithms reduce the search step and intensify the search of the area adjacent to the center in the search range. The HS algorithm searches fewer candidates than the DS algorithm in general to get the same motion vector. However, search accuracy of these fast algorithms is not as good as the FS, especially for the video sequences with big motion. In terms of the fast search algorithms, the PLS algorithm is closest to the FS in accuracy and the speed of the PLS is nearly 10 times faster than the FS, but slower than the DS and the HS.

A cubic panorama consists of six side images stitched together to provide viewers with a 360º horizontal view and the capability to look up to the ceiling and down to the floor. For the regular motor-mounted panorama capturing, the panoramic camera usually moves approximately in the horizontal direction along the road. Consequently, most parts of adjacent frames have similar content. For continuous panoramic frames, image blocks usually move from one side image to another side image of the cube than disappear at the edge of the image. The motion of the blocks in panoramic video is usually bigger than that in the video captured by usual prospective cameras. The fast motion estimation algorithms above failed to investigate the characteristic of the block motion in panoramic video, as limits the algorithm accuracy in the motion estimation of panoramic video.

The goal of this paper is to develop an effective motion estimation algorithm for cubic panoramic video. According to the characteristics of the block motion, each side image of the reference frames is extended with the image boundary consistency preserved. The proposed algorithm constructs search lines following the motion of blocks between adjacent frames and utilizes the Line Search to estimate the motion of the corresponding macro block. It starts from the macro block matched by the Line Search. After that, the adjacent macro blocks are estimated and the motion vectors are acquired. The details about the proposed algorithm are described in the next section, followed by experiment results and a conclusion.

II. PANORAMA MOTION ESTIMATION

A. Algorithm Motivation

Following the epipolar geometry of cubic panorama proposed in [8], the epipolar lines of the panoramic video of stationary scenes can be obtained, as shown in Fig. 1. The epipolar lines starts at the epipole on the front side image of the cube and ends at the epipole on the back side image. On the up, down, left and right side images of the cube, the epipolar lines move almost in the horizontal direction. The movement of the epipolar lines reflects the motion of the panoramic camera. Between adjacent frames, the motion of the camera is mostly horizontal.

978-0-7695-3890-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ISM.2009.49

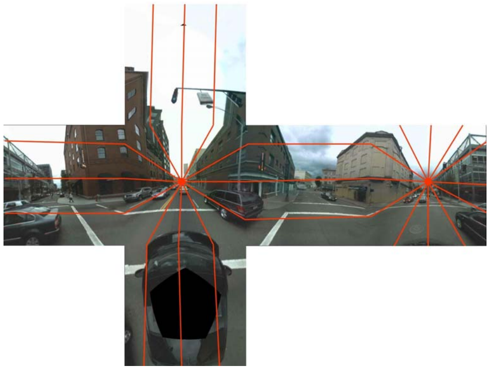


Figure 1. Epipolar lines of the cubic panoramic video

The movement of the epipolar lines also indicates the motion of the stationary scenes in the panoramic video. When the camera moves horizontally, the relations of the pixels between two adjacent panoramic frames are shown in Fig. 2. On the up, down, left and right side images of the cube, pixels of the area *S1* of the current frame are the same as the pixels of the area *S2* of the previous frame, as shown in Fig. 2(a). The difference between pixels of these two areas is only the horizontal relative movement in the frames. Fig. 2(b) and Fig. 2(c) describe the relations of the pixels on the front and back side images respectively. Pixels of the area *T1* of the current frame are similar to those of the area *T2* of the previous frame.

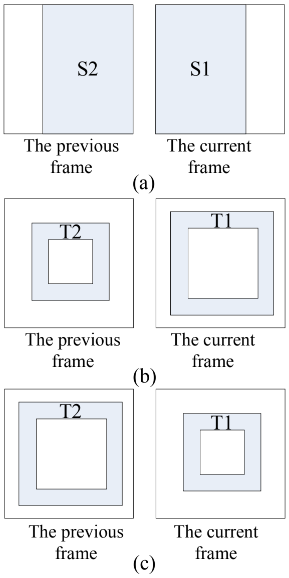


Figure 2. Relations of pixels between adjacent frames in panoramic video: (a) On the up, down, left and right side images of the cube (b) On the front side image of the cube (c) On the back side image of the cube

These characteristics of pixels between frames could benefit the search of macro blocks in motion estimation of panoramic video. In addition, adjacent macro blocks usually have the same or similar motion, as shown in Fig. 3. The motion of the current macro block (CMB) is correlated to the motion of the adjacent macro blocks, such as MB1, MB2 and MB3.

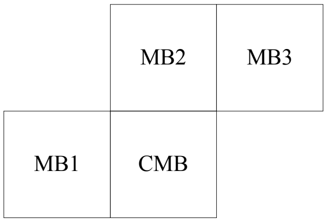


Figure 3. Spatial correlation of macro block motion

From the consideration of the relations of pixels between adjacent frames and the spatial correlation of macro block motion, we propose the PME (Panorama Motion Estimation) algorithm based on the Line Search to estimate the motion of adjacent macro blocks. The motion estimation algorithm for panoramic video is summarized as follows:

Step 1) Extend each side image of the reference frame with the adjacent images on the cube.

Step 2) According to the coordinates of the predictive frame, select the first block of the unsearched macro blocks and find the MBD (Minimum Block Distortion) point of the block through the Line Search algorithm. The MBD point found in this step is the solution of the motion vector of the selected macro block.

Step 3) The PME algorithm starts at the macro block which is matched by the Line Search. The adjacent macro blocks of the matched block are searched by the PME and the motion vectors of them are estimated.

Step 4) Check all the macro blocks of the predictive frame. If there are unsearched macro blocks, go to Step 2); otherwise, report the motion vectors of the macro blocks of the predictive frame.

B. Extension of the Reference Frame

Unlike the videos captured by traditional prospective cameras, cubic panoramic video covers 360°×180° view of scenes. For continuous panoramic frames, image blocks move from one side image of the cube to another rather than disappear at the edge of the image. Therefore, it is necessary for the motion estimation of panoramic video to take the movement of macro blocks between adjacent images into consideration. A method of block padding proposed in [9] is aimed to estimate the motion of blocks on the boundaries of the cube. However, our algorithm extends the reference frame rather than pad the blocks to solve the problem of the block movement from one image to another. According to the image boundary consistency of the images of the cubic panorama, each side images of the reference frame is extended by its adjacent images on the cube.

The extension of reference frames is illustrated in Fig. 4, taking the top side image for example. In Fig. 4(a), the top side image and its four adjacent images are laid out into a plane. The extended image, the dashed square in Fig. 4(a), is padded with the overlaid pixels. Unfortunately, the blank area at each corner of the square is unpadded by any pixels. Just taking the blank area *ABCO* at the upper left corner for example, the method to fill the blank area at the corner is illustrated in Fig. 4(b). According to the image boundary consistency of the cube, the padding of the area *ABCO* should preserved the consistency of pixel intensity on the padding boundaries *AO* and *CO*. The left and up images are respectively divide by into four parts by the diagonals. Following the consistency, the NO.1 part of the left image is stitched to the edge *FO*, and similarly the NO.4 part of the up image is stitched to the edge *EO*. Consequently, the blank area *ABCO* is padded by the overlaid parts of the left and up images. Specifically, the area *ABO* is filled with the image of the area *CEO* and the area *BCO* is filled with the image of the area *DAO*.

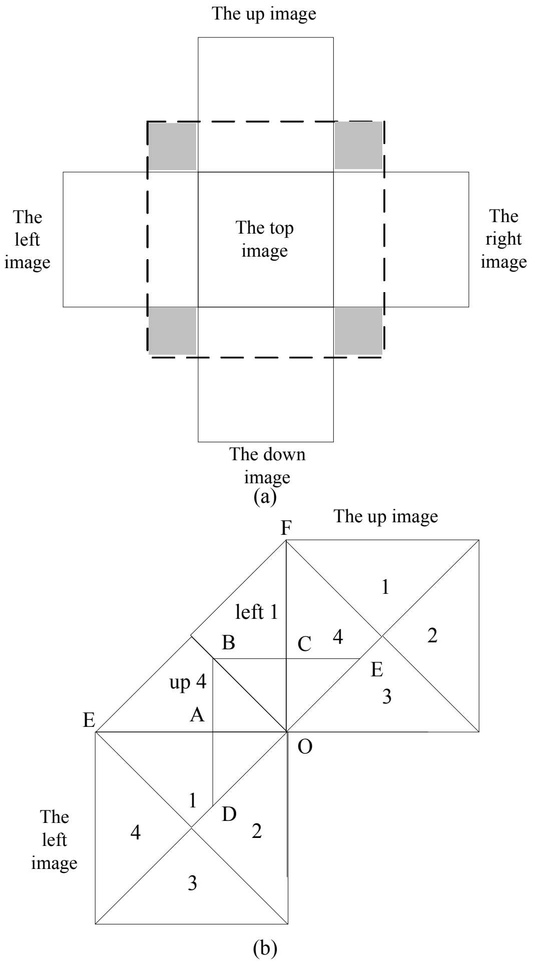


Figure 4. Extension of the reference frame: (a) Extension of the top side image (b) Padding the blank area at the upper left corner

C. Initialization to Expand the Line Search

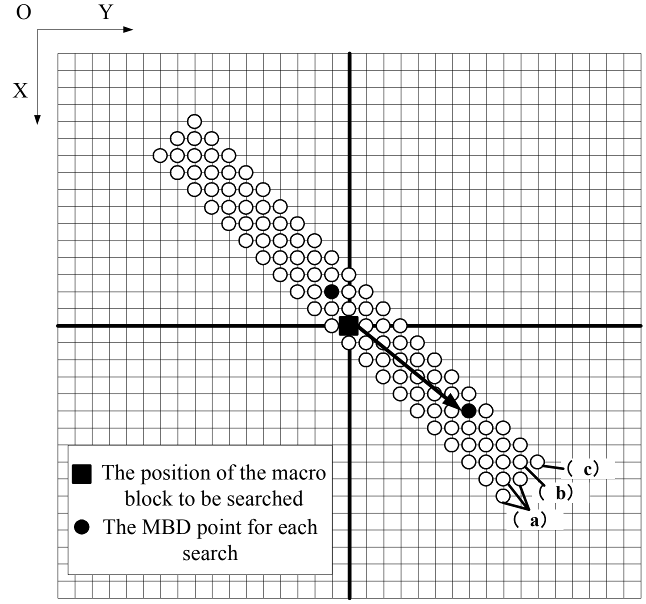
The Line Search starts from searching all points in the three lines, and then searches additional lines in the direction of descending distortion. The process stops when the MBD point is not on the boundary of searched lines [7]. Because of its effectiveness, the Line Search is applied as the initialization of the next PME algorithm. In our algorithm, search lines of the Line Search are depended on the movement of the macro blocks. According to the epipolar geometry of cubic panorama, when the camera moves horizontally between adjacent frames, the major movement of macro blocks is: on the up, down, left and right side images of the cube, the macro blocks move mainly in the horizontal direction; on the front and back side images, the macro blocks diffuse from the epipole of the front image and converge at the epipole of the back image. Therefore, the initial equation of search line is *x=ky+p*, and the parameters *k* and *p* are calculated by

$$k = \begin{cases} \dfrac{x_c - x_m}{y_c - y_m} & \text{for the front and back side images} \\ 0 & \text{for the other side images} \end{cases} \qquad (1)$$

$$p = x_m - ky_m$$

where (x_m, y_m) indicates the position of the block to be searched and (x_c, y_c) is the center of the image.

The search procedure is demonstrated by an example as shown in Fig. 5. Assume that the position of the macro block to be searched is (16, 16) on the front image of the cube and the true motion vector is (5, 7). According to (1), this macro block is in line *x=y* and all candidates in line *x=y+1*, *x=y* and *x=y-1* are searched (a). The MBD point in this step is at (14, 15), which is on boundary of searched lines so an additional line *x=y-2* is searched (b). The MBD point is at (21, 23), therefore, *x=y-3* is also searched (c). Finally, because no candidate points in line *x=y-3* has lower distortion than position (21, 23), the procedure stops and the motion vector of (5, 7) is obtained.



O
Y
X
(c)
(b)
(a)
■ The position of the macro block to be searched
● The MBD point for each search

Figure 5. The procedure of the line search

The Line Search algorithm which utilizes the characteristic of the block motion to construct search lines is proposed to estimate the motion of the start macro block of the PME algorithm. The proposed Line Search algorithm is described as follows:

Step1) Search three consecutive lines of candidates. If the macro block to be searched locates in line *x=ky+p*, then all points in line *x=ky+p+1*, *x=ky+p*, *x=ky+p-1* are tested. If the MBD point is located in line *x=ky+p+1*, go to Step2); if the MBD point locates in line *x=ky+p-1*, go to Step3); otherwise go to Step4).

Step2) Let *p=p+1*, then test all points in line *x=ky+p+1*. If the MBD point is in line *x=ky+p*, go to Step4), otherwise repeat the current step.

Step3) Let *p=p-1*, then test all points in line *x=ky+p-1*. If the MBD point is in line *x=ky+p*, go to Step4), otherwise repeat the current step.

Step4) Report the MBD point and calculate the corresponding motion vector.

D. Search of Adjacent Macro Blocks

The spatial correlation of macro block motion describes that adjacent macro blocks usually have the same or the similar motion. Generally speaking, the motion vector of a macro block is correlated to the motion vector of the adjacent macro blocks, such as the left, the up and the upper right adjacent macro blocks. Therefore, based on the macro block matched by the Line Search, the motion of the adjacent blocks is estimated by the PME algorithm. Compared to the Line Search, the PME makes use of the spatial correlation of macro block motion to decreases the candidate points to be searched.

In the PME, a macro block is located by the macro block coordinates (*i, j*). In terms of a macro block, the transition from the pixel coordinates (*x, y*) to (*i, j*) is defined as

$$\begin{cases} i = x/(2r) \\ j = y/(2r) \end{cases}, \text{ r is the radius of the macro block} \quad (2)$$

To present three different types of macro blocks at the *k*th round, the PME uses three macro block sets, D_k, M_k and N_k, where D_k is the set of macro blocks to be searched, M_k is the set of matched macro blocks and N_k is the set of unmatched macro blocks. For a macro block (*i, j*) matched by the Line Search, the three sets at the beginning of the PME is initialized as

$$\begin{aligned} D_0 &= \phi \\ M_0 &= \{m(i,j)\} \\ N_0 &= \phi \end{aligned} \quad (3)$$

where the block *m*(*i*, *j*) included by M_0 is the block matched by the Line Search.

In the process of the PME algorithm, D_k, M_k and N_k at the *k*th round could be calculated from D_{k-1}, M_{k-1} and N_{k-1} at the *k-1*th round. On account of the spatial correlation of macro block motion, the motion of the macro block to be searched would be estimated based on the motion vectors of the adjacent matched blocks. Thus, D_k, the set of macro blocks to be searched, would be generated from M_{k-1} which is the set of match macro blocks at the *k-1*th round. The generation of D_k from M_{k-1} is factorized as

$$\begin{aligned} D_k = &\{d(p, j+k) | m(p, j+k-1) \in M_{k-1}\} \\ &\cup \{d(i+k, q) | m(i+k-1, q) \in M_{k-1}\} \\ &\cup \{d(i+k, j+k) | m(i+k-1, j+k-1) \in M_{k-1}\} \end{aligned} \quad (4)$$

In accordance with the motion vector reference relationship *R* in (5), each macro block *d(p, q)* of D_k is searched by the HS algorithm based on the motion vector of the corresponding macro block in M_{k-1}. The MBD point of each macro block in D_k is obtained by HS and the block whose MBD is lower than the threshold *TSS* belongs to the matched macro blocks. At the *k*th round, M_k is defined by (6).

$$R(d(p,q)) = \begin{cases} m(p, j+k-1), & q=j+k, m(p, j+k-1) \in M_{k-1} \\ m(i+k-1, q), & p=i+k, m(i+k-1, q) \in M_{k-1} \\ m(i+k-1, j+k-1), & p=i+k, q=j+k, m(i+k-1, j+k-1) \in M_{k-1} \end{cases} \quad (5)$$

$$M_k = \{m(p,q) | d(p,q) \in D_k \wedge MBD(p,q) \le TSS\} \quad (6)$$

The unmatched macro blocks at the *k-1*th round causes that some macro blocks in the *k*th round have no motion vectors for reference. Naturally, the macro blocks with no reference motion vectors are considered as the unmatched macro blocks at the *k*th round. In addition, the block whose MBD is larger than the threshold TSS is also classified into the set of unmatched macro blocks. Therefore, the set N_k of the *k*th round is calculated by

$$\begin{aligned} N_k = &\{n(p, j+k) | n(p, j+k-1) \in N_{k-1}\} \\ &\cup \{n(i+k, q) | n(i+k-1, q) \in N_{k-1}\} \\ &\cup \{n(i+k, j+k) | n(i+k-1, j+k-1) \in N_{k-1}\} \\ &\cup \{D_k\text{-}M_k\} \end{aligned} \quad (7)$$

The PME algorithm ends when $M_k = \phi$ and the motion vectors of blocks in the sets of matched macro blocks are reported, such as the blocks in M_1, $M_2 \ldots M_{k-2}$ and M_{k-1}. For a macro block (*i*, *j*) matched by the Line Search, the procedure of the PME is summarized as follows:

Step 1) Let *k=0*, then according to (3), initialize the set D_0 of macro blocks to be searched, the set M_0 of matched macro blocks and the set N_0 of unmatched macro blocks.

Step 2) Let *k=k+1*, then according to (4), calculate the set D_k of macro blocks to be searched.

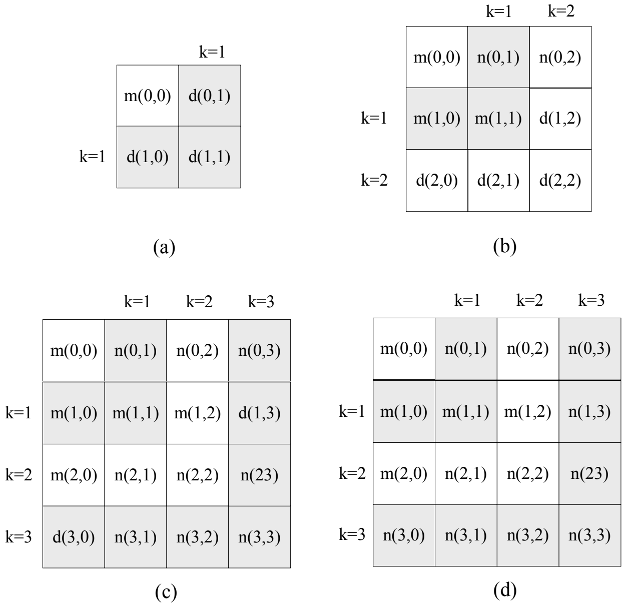
Step 3) Following (5), each block in D_k is searched by the HS algorithm based on the motion vector of the corresponding macro block in M_{k-1}. Then, according to (6), calculate the set M_k of matched macro blocks.

Step 4) The set N_k of unmatched macro blocks is calculated by (7). If $M_k = \phi$, go to Step 5); otherwise go to Step 2).

Step 5) Report the motion vectors of blocks in the sets of matched macro blocks, such as M_1, $M_2 \ldots M_{k-2}$ and M_{k-1}.

The search procedure is demonstrated by an example as shown in Fig. 6. *d* represents the macro block to be searched, *m* represents the matched macro block and *n* represents the unmatched macro block. Assume that the matched block by the Line Search is the macro block (0, 0) and the initial set of matched macro blocks is M_0 = {*m*(0, 0)}. For *k=1*, the set of macro blocks to be searched is D_1 = {*d*(0, 1), *d*(1, 0), *d*(1, 1)}. The blocks in D_1 are searched based on the motion vector of *m*(0, 0), as shown in Fig. 6(a). Assume the macro block *d*(0, 1) is unmatched. As a result, the set of matched macro blocks for *k=1* is M_1 = {*m*(1, 0), *m*(1, 1)} and the set of unmatched macro blocks for *k=1* is N_1 = {*n*(0, 1)}. Based on N_1, the macro block (0, 2) is considered as the unmatched block and the set of macro blocks to be searched for *k=2* is

D_2 = {d(2, 0), d(2, 1), d(1, 2), d(2, 2)}, as shown in Fig. 6(b). Following the motion vector reference relationship R, the Blocks in D_2 are searched based on the motion vectors of the corresponding macro blocks. Assume the macro block d(2, 1) and d(2, 2) are unmatched. Consequently, the set of matched macro blocks for $k=2$ is M_2 = {m(2, 0), m(2, 1)} and the set of unmatched macro blocks for $k=2$ is N_2 = {n(2, 1), n(0, 2), n(2, 2)}. When $k=3$, the set of macro blocks to be searched is D_3 = {d(3, 0), d(1, 3)}, as shown in Fig. 6(c). Unfortunately, both the blocks in D_3 are unmatched. As shown in Fig. 6(d), the set of unmatched macro blocks for k=3 is N_3 = {n(3, 0), n(3, 1), n(3, 2), n(0, 3), n(1, 3), n(2, 3), n(3, 3)}and the set of matched macro blocks for $k=3$ is $M_3 = \phi$. For $M_3 = \phi$, the search terminates and the motion vectors of blocks in M_1 and M_2 are found.



		k=1
	m(0,0)	d(0,1)
k=1	d(1,0)	d(1,1)

(a)

		k=1	k=2
	m(0,0)	n(0,1)	n(0,2)
k=1	m(1,0)	m(1,1)	d(1,2)
k=2	d(2,0)	d(2,1)	d(2,2)

(b)

		k=1	k=2	k=3
	m(0,0)	n(0,1)	n(0,2)	n(0,3)
k=1	m(1,0)	m(1,1)	m(1,2)	d(1,3)
k=2	m(2,0)	n(2,1)	n(2,2)	n(23)
k=3	d(3,0)	n(3,1)	n(3,2)	n(3,3)

(c)

		k=1	k=2	k=3
	m(0,0)	n(0,1)	n(0,2)	n(0,3)
k=1	m(1,0)	m(1,1)	m(1,2)	n(1,3)
k=2	m(2,0)	n(2,1)	n(2,2)	n(23)
k=3	n(3,0)	n(3,1)	n(3,2)	n(3,3)

(d)

Figure 6. The procedure of the PME of the adjacent macro blocks: (a) For k=1 round (b) For k=2 round (c) For k=3 round (d) The end of the PME

III. EXPERIMENT RESULTS

In this section, experiment results of the proposed algorithm applied to real panoramic video are presented. The algorithm was performed on a personal computer (Pentium4-2.4 GHz).

The test sequences in our experiment are the demo panoramic videos from the Immersive Media Cooperation. Each test sequence includes 50 frames and the size of each side image of a frame is 512×512. In the experiment, FS, DS, HS and the proposed algorithm are applied to estimate the motion of the video. Just taking the urban street sequence for example, the comparison of the compensated images of several algorithms is illustrated in Fig. 7. Fig. 7(a) shows the six side images of the original 6th frame. Fig. 7(b) shows the compensated images of the 6th frame generated by the proposed algorithm. Fig. 7(c), Fig. 7(d) and Fig. 7(e) show the compensated images of the 6th frame generated by the FS, DS and HS algorithms respectively. From the images of Fig. 7, the compensated images of the proposed algorithm are closer to the images generated by the FS than those of the DS and the HS.

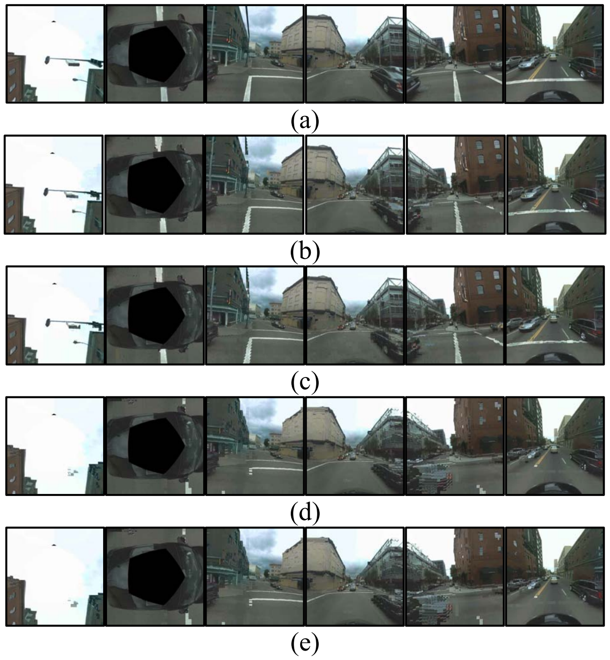


Figure 7. Comparison of the compensated images of several motion estimation algorithms: (a) The images of the original 6th frame (b) The compensated images generated by the proposed algorithm (c) The compensated images generated by the FS algorithm (d) The compensated images generated by the DS algorithm (e) The compensated images generated by the HS algorithm

To evaluate the performance of the proposed algorithm, we choose PSNR (Peak Signal to Noise Ratio) as the main metrics. PSNR reflects the distortion of an image. Generally speaking, the larger the PSNR, the less the distortion of the image and therefore the more effective the motion estimation algorithm is. For FS, DS, HS and the proposed algorithm, Table I shows the average PSNR of the compensated frames of each test sequence. Also, the frame-by-frame comparison of PSNR are illustrated in Fig. 8, after applying FS, DS, HS and the proposed algorithm to the urban street sequence. Compared to the DS and HS algorithms, the performance of the proposed algorithm is more approximate to the FS algorithm. The average PSNR value of the proposed algorithm is 5dBs higher than those of DS and HS.

TABLE I. PSNR PERFORMANCE COMPARISON

Video Sequences	PSNR(dB)			
	The proposed algorithm	FS	DS	HS
Urban Street	28.74	29.23	23.51	23.47
Expressway	23.91	24.39	20.24	20.16
Garden	31.23	31.64	26.38	26.31
Pavement	33.62	33.97	27.14	27.1
Skating Rink	30.21	30.61	25.36	25.31
Average	29.54	29.98	24.53	24.47

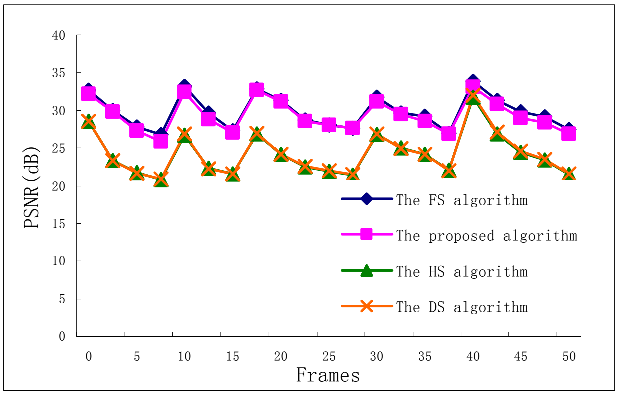


Figure 8. PSNR comparison of each frame for the urban street sequence

Table □ shows the complexity comparison of FS, DS, HS and the proposed algorithm for a 32×32 pixels search range. For macro blocks on six side images of the cube, the average number of candidates searched by the proposed algorithm is 83.4. Compared to the FS algorithm, which searches 1024 points in the search area, the speedup of the proposed algorithm is about 12 times faster. The average searched candidates of the DS algorithm and the HS algorithm are 21.6 and 17.8 respectively, faster than the proposed algorithm. However, for motion estimation of panoramic video, the average PSNR of the DS and the HS are 5dBs lower than the proposed algorithm. Therefore, as far as the search effectiveness is concerned, the proposed algorithm is better than the DS and the HS.

TABLE II. SEARCHED CANDIDATES COMPARISON

Video Sequences	Searched Candidates per Macro Block			
	The proposed algorithm	FS	DS	HS
Urban Street	86	1024	23	17
Expressway	90	1024	24	21
Garden	81	1024	20	17
Cross Road	77	1024	19	16
Skating Rink	83	1024	22	18
Average	83.4	1024	21.6	17.8

IV. CONCLUSION

An effective motion estimation algorithm for cubic panoramic video is described in this paper. The main features of the proposed algorithm are the construction of search lines for the Line Search and the PME algorithm for adjacent macro blocks. Search lines of a macro block are decided according to the characteristic of the block motion in the panoramic video. With the constructed search lines, the Line Search algorithm estimates the motion of the corresponding macro block and the block matched by the Line Search is the start of the PME. The adjacent macro blocks are estimated by the PME because spatial correlation exists between neighboring motion vectors. From the experiment results, the PSNR performance of the proposed algorithm is very close to that of the FS and the speed of it is 12 times faster. It is also shown that PSNR performance of the proposed algorithm is higher than those of the DS and the HS.

ACKNOWLEDGMENT

This work is supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2009CB320805 and the Industry-Academy-Research Program of Guangdong Province & the Ministry of Education under Grant No. 2008A090400020.

REFERENCES

[1] T. Koga, K. Linuma, A. Hirano, Y. Iijima and T.Ishiguro, “Motion-compensated interframe coding for video conferencing,” Proc. Nat. Telecommunication Conference, New Orleans, Dec. 1981, pp. G5.3.1-G5.3.5.

[2] M. Ghanbari, “The cross-search algorithm for motion estimation,” IEEE Transactions Communications, vol. 38, no. 7, pp. 950-953, Jul. 1990, doi:10.1109/26.57512.

[3] R Li, B Zeng, ML Liou, “A new three-step search algorithm for block motion estimation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 4, pp. 438-442, Aug. 1994, doi:10.1109/76.313138.

[4] LM Po, WC Ma, “A novel four-step search algorithm for fast block motion estimation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313-317, Jun. 1996, doi:10.1109/76.499840.

[5] S Zhu, KK Ma, “A new diamond search algorithm for fast block-matching motion estimation,” IEEE Transactions on Image Processing, vol. 9, no. 2, pp. 287-290, Feb. 2000, doi:10.1109/83.821744.

[6] C Zhu,X Lin, LP Chau, KP Lim, HA Ang, CY Ong, “A novel hexagon-based search algorithm for fast block motion estimation,” Proc. 2001 IEEE International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, May 2001, pp. 1593-1596, doi:10.1109/ICASSP.2001.941239.

[7] YW Huang, SY Ma, CF Shen, LG Chen, “Predictive Line Search: an efficient motion estimation algorithm for MPEG-4 encoding systems on multimedia Processors,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 1, pp. 111-117, Jan. 2003, doi:10.1109/TCSVT.2002.808093.

[8] F Kangni, R Laganière, “Epipolar Geometry for the Rectification of Cubic Panoramas,” Proc. the 3rd Canadian Conference on Computer and Robot Vision, Quebec City, Jun. 2006, pp. 70-77, doi:10.1109/CRV.2006.29.

[9] KH Jiang, E Dubois, “Compression of Cubic Panorama Datasets with Spatially Consistent Representation,” Proc. IEEE International Workshop on Haptic Audio Visual Environments and their Applications, Ottawa, Nov. 2006, pp. 111-116, doi:10.1109/HAVE.2006.283774.