FAST ADAPTIVE BLOCK MATCHING FOR RAY-SPACE CODING IN FTV SYSTEM

Mei YU [1,3], Feng SHAO [1,2], Gangyi JIANG [1,3], Yu ZHOU [1]

1. Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China
2. Department of Information and Electronics Eng., Zhejiang University, Hangzhou 310027, China
3. National Key Laboratory of Machine Perception, Peking University, Beijing 100871, China

ABSTRACT

Ray-Space representation is the main approach to realizing free viewpoint television (FTV) with complicated scene. Data compression in ray-space is one of key technologies in ray-space based FTV systems. In ray-space based FTV system, block matching complication is the most important factor to influence coding efficiency. In this paper, a fast adaptive block matching algorithm is proposed by using starting search point prediction, still block determination and search stop criteria strategies. Experimental results show that the search speed is improved greatly as well as the coding efficiency.

1. INTRODUCTION

FTV is expected to be the next generation of visual application. It allows the users to change his/her viewing point and viewing direction freely and enjoy more photorealistic 3D images. The techniques in FTV can be classified into two categories, that is, model-based rendering (MBR)[1] and image-based rendering (IBR)[2]. Ray-Space representation is a newly developed method of IBR, used to describe 3D information by converting the original multi-view images to "ray" parameters[3]. Unlike other methods of IBR, the ray-space can generate an arbitrary viewpoint view without complicated analysis and rendering process. FTV scheme based on ray-space was adopted in the 3DAV draft presented by MPEG[4].

At particular time t_0, ray-space can be described with a 3D function $\{f(x, y, u)\}$[5]. In Fig.1, a ray-space section $\{f(x, y_k, u)\}$ is formed by horizontal segmentation in plane coordinate system. Obviously, ray distribution in ray-space is different from pixel distribution in 2D image. For a given $y=y_k$, a ray-space data slice $S(y_k)=\{f(x, y_k, u)\}$ is formed by transforming data in multi-view images with respect to y_k.

Because of enormous ray-space data, data compression becomes one of key technologies in the ray-space based FTV systems[6,7]. In this paper, inter-block correlations of ray-space data are analyzed. Then, a fast adaptive block matching algorithm for prediction coding of ray-space is proposed. Experimental results show that the searching speed of block matching in ray-space data compression is improved greatly and the coding efficiency is also improved resultantly compared with the previous 2D image based ray-space data compression methods.

2. CHARACTERISTIC ANALYSIS IN RAY-SPACE SLICE SEQUENCE

In order to analyze characteristic of ray-space data, distribution of matching vector of full search is analyzed, the statistic results are given in Fig.2(a). It is seen that the matching vectors have obvious center-biased characteristic and concentrates highly on the neighbor of (0,0). Then the matching vector direction is analyzed without taking into account center point and vectors with horizontal or vertical directions. The results are shown in Fig.2(b), from which it is seen that the vector mainly distributes in rectilinear direction.

Ray-space has strong inter-block correlations, and the high coding efficiency will be achieved if inter-block correlations prediction is utilized. Here, block matching vector error coefficient is defined by

$$\lambda=(SAD_{\text{pred}}-SAD_{\text{best}})/SAD_{\text{pred}}, \tag{1}$$

where SAD_{pred} is minimum SAD obtained by inter-block matching vector prediction, and SAD_{best} is minimum SAD obtained by full search prediction. λ reflects matching

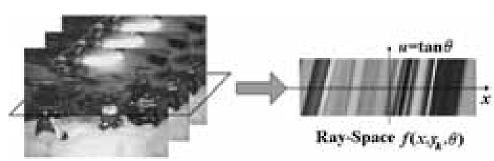


Fig.1 An example of ray-space slice $\{f(x,y_k,u)\}$.

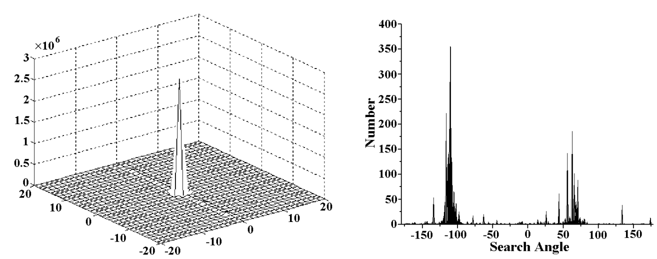
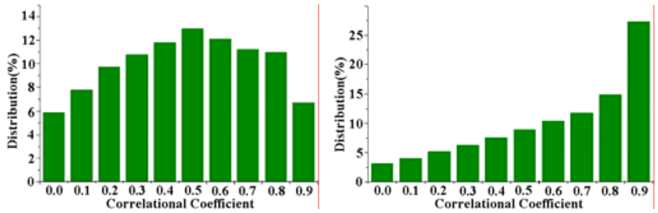


Fig.2(a) Statistic of matching vectors; (b) Directional distribution of matching vectors.

Table 1 Statistic of parameter λ.

Prediction method	0~0.10	0.10~0.20
Collocated block prediction	89.38%	1.57%
Adjacent block prediction	95.27%	1.70%



(a) Adjacent block correlation coefficient (b) Collocated block correlation coefficient

Fig.3 Inter-block correlation coefficient distribution.

Table 2 Markov parameters with respect to the left, top, top-right and collocated blocks.

	ω_1	ω_2	ω_3	ω_4
Background	0.0763	0.0701	0.0669	0.3816
Foreground	0.0964	0.1234	0.0253	0.1343

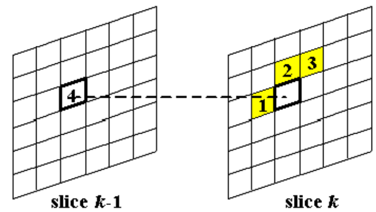


Fig.4 Prediction of starting search point.

degree between best matching vector and predictive matching vector. Two inter-block prediction schemes are adopted, one is collocated block prediction which predicts matching vector of current block from the vector of the block of the same location in previous slice, the other is adjacent prediction which predicts the matching vector of current block from adjacent blocks at the left, top, and top-right in current slice. Table 1 shows λ distribution in [0, 1] zone. Obviously, the current block is quite correlative with its adjacent blocks and the collocated block.

Further, inter-block correlation coefficient is analyzed. Let X[m, n] and Y[m, n] be two blocks, $\bar{X}$ and $\bar{Y}$ be their means respectively, inter-block correlation coefficient is defined as

$$R_B = \frac{\sum_{m=1}^{N}\sum_{n=1}^{N}(X[m,n]-\bar{X})(Y[m,n]-\bar{Y})}{\sqrt{\sum_{m=1}^{N}\sum_{n=1}^{N}(X[m,n]-\bar{X})^2}\sqrt{\sum_{m=1}^{N}\sum_{n=1}^{N}(Y[m,n]-\bar{Y})^2}}\ . \tag{2}$$

Let R_1, R_2 and R_3 be the block correlation coefficients corresponding to the left, top, and top-right blocks respectively, and R_4 denote the block correlation coefficient corresponding to collocated block. Then adjacent block correlation coefficient is defined by R=max{R_1, R_2, R_3} and collocated correlation coefficient is R=R_4. Fig.3 gives the correlation coefficients of the adjacent blocks and the collocated block. It is seen that inter-block correlation in ray-space is very strong and the collocated blocks correlation is even greater.

In some fast motion estimation algorithms, in order to reduce block matching calculation, it is usually supposed that the matching error increases monotonously along with searching location far away from global best motion vector. But in some cases it is not always satisfied and the search will always get into local optimization. If the motion vector field is smoothed, the local optimization problem will be easily solved. In this paper, starting search point prediction is performed. The prediction utilizes inter-block correlation as shown in Fig.4. Markov model is used to describe the correlations. Markov model can be expressed as $P(V_t|V_{t-1},v_t)$, the matching vector of current block is not only related to matching vectors of adjacent blocks in current slice, but also related to matching vector of collocated block in previous slice. Their correlations can be expressed as

$$V(s)=\sum_r \omega_r V_1(s+r)+e(s)\ , \tag{3}$$

where s represents the current block, r denotes the four adjacent blocks of current block, marked as r_1, r_2, r_3 and r_4. $V(s)$ and $V_1(s+r)$ are the matching vectors of current block and the adjacent blocks, respectively. $e(s)$ is Gauss noise sequence with zero mean. Let Ω be the block set within one slice, $s\in\Omega$, and the block size is 8×8. Suppose $\boldsymbol{V}_s=[V_1(s+r_1),V_1(s+r_2),V_1(s+r_3),V_1(s+r_4)]$, by least square error estimation, the following formula can be obtained

$$\omega_r=[\sum_s \boldsymbol{V}_s\boldsymbol{V}_s^T]^{-1}[\sum_s \boldsymbol{V}_s V(s)]\ . \tag{4}$$

By using the above method, Markov parameter estimation is performed for two typical background and foreground ray-space slices, and the results are given in Table 2. It is seen that for background region slices, ω_4 is the largest which indicates that collocated block correlation is the strongest. For foreground region slices, the correlations between adjacent blocks and collocated block are comparatively equal, since ω_2 is close to ω_4.

For full searching, 44.8% of the matching vectors are (0,0), which indicates the best center-biased characteristic of ray-space. So a still block can be determined by a threshold Th_0, that is, if current block's SAD is lower than Th_0, it is regarded as a still block and skipped. Here, an adaptive threshold Th_0 is defined by

$$Th_0=\begin{cases}\text{median}(SAD_1,SAD_2,SAD_3), & \text{if reference frame is I frame}\\ \min\{\text{median}(SAD_1,SAD_2,SAD_3),SAD_4\}, & \text{if reference frame is P frame}\end{cases} \tag{5}$$

In this paper, the combination of gradient diamond search mode and rectilinear direction search mode is adopted. In the gradient diamond search, the search mode with step size 2 is called as the large diamond search (LDS), while the search mode with step size 1 is called as the small diamond search (SDS). Based on ray-space characteristic analysis, ray-space matching vectors have significant rectilinear direction, so rectilinear direction strategy is commended.

In some fast algorithms, SAD threshold has been used as the stop criteria to speed up the search. That is, the search process stops when the current minimum *SAD* of a block is less than a given threshold. But most of algorithms use a constant threshold, leading to inadaptability. Since there are strong correlations between current block and adjacent blocks, two adaptive thresholds SAD_{thrh1} and SAD_{thrh2}, are used, which is similar to threshold of early termination proposed in [8], defined by $SAD_{\mathrm{thrh}}=(1+\beta)SAD_{\mathrm{pred}}$, where SAD_{pred} is the *SAD* corresponding to the predicted matching vector, and β is a modulated factor which plays a key role in keeping a tradeoff between reconstructed quality and search speed.

According to above analysis, a fast adaptive block matching algorithm is proposed in this paper. Its main procedure is described as follows:

Step 1: Let SAD_0 be the SAD with respect to the zero matching vector (0,0), and SAD_i (i=1,2,3,4) correspond to SAD associated with the matching vectors of the three adjacent blocks and the collocated block. If reference frame of current block is I frame, $SAD_{\mathrm{min}}=\mathrm{median}(SAD_1,SAD_2,SAD_3)$; if reference frame is P frame, $SAD_{\mathrm{min}}=\min\{\mathrm{median}(SAD_1,SAD_2,SAD_3),SAD_4\}$;

Step 2: Still block determination: If $SAD_0<SAD_{\mathrm{min}}$, then current block is considered as still block, so skip it to the next block.

Step 3: If $SAD_0=SAD_{\mathrm{min}}$, let the point indicated by zero matching vector as the search center, repeat SDS and modify the SAD_{min} until the best-matched point lies in the center of the diamond; If $SAD_0>SAD_{\mathrm{min}}$, let the point indicated by the matching vector MV_{pred} with respect to SAD_{min} as the search center, and turn to Step 4;

Step 4: If the two components V_x and V_y of the matching vector MV_{pred} are larger or smaller than 0 simultaneously, and satisfy $(V_x)^2+(V_y)^2<T$, the point indicated by MV_{pred} is considered to be located within the rectilinear search region, so search the next point in the rectilinear direction indicated by V_y/V_x. If the *SAD* of the new point is the minimum *SAD*, moreover, satisfy $SAD_{\mathrm{min}}<SAD_{\mathrm{thrh1}}$, then turn to step 6, otherwise, if the second condition is not satisfied then turn to step 5; If the *SAD* of the new point is not the minimum, then continue to search the next point in the rectilinear direction.

In other cases, that is, the point with respect to MV_{pred} is not within the rectilinear search region, then repeat LDS and modify the SAD_{min} until the best-matched point lies in the center of the diamond, and then turn to Step 5.

Note that if $SAD_{\mathrm{min}}<SAD_{\mathrm{thrh1}}$ at anytime during the searching, then turn to Step 6, or if $SAD_{\mathrm{thrh1}}<SAD_{\mathrm{min}}<SAD_{\mathrm{thrh2}}$, then turn to Step 5;

Step 5: Do SDS once, and go to Step 6;

(a) The most left and right images of "Xmas"

(b) The most left and right images of "Cup"

Fig.5 Test multi-view images for Ray-space compression.

Step 6: Finish the searching process, and obtain the matching vector and the corresponding SAD_{min}.

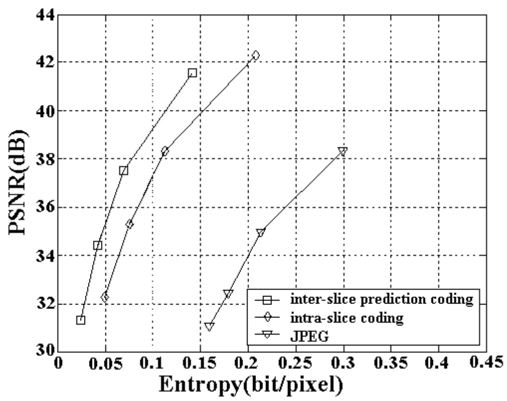
3. EXPERIMENTAL RESULTS

In the experiments, two sets of grayscale multi-view images of real scene with the size of 640×480, are used for the testing purposes, the most left and the most right image of the multi-view images are given in Fig.5. "Cup" has 144 multi-view images, forming 480 slices with size of 640×144, and "Xmas" has 96 multi-view images, forming 480 slices with size of 640×96.

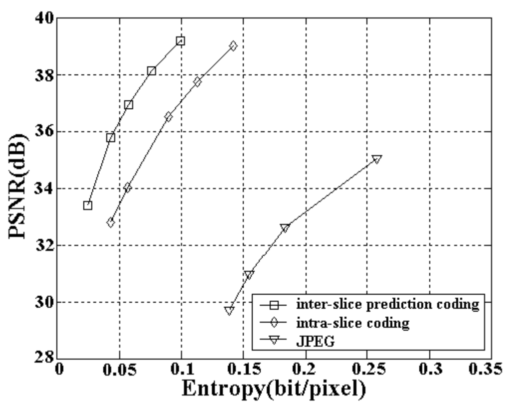
First, three coding methods for ray-space data compression are compared, and the coding performances are given in Fig.6. The intensity of rays from the same point on an object does not drastically change in different viewing direction, thus the ray-space slice looks like many overlapping or parallel rays with different luminances. Ray-space coding methods proposed by Fujii et al only utilize these 2D intra-slice redundancies, similar to still image coding technique, leading to low coding efficiency. It is clear that the correlations between adjacent slices are very strong and high compression efficiency can be obtained by using these correlations.

Then, the proposed fast adaptive block matching algorithm (FABMA) is compared with full search (FS), novel three step search (NTSS), four step search (FSS), block-based gradient descent search (BBGDS). Table 3 lists the comparison between the above methods. The items contain average PSNR, δ_{PSNR} (difference from the PSNR of FS), check points that implies block matching time, and the speedup ratio compared with FS. The test sequences are still "Xmas" and "Cup", and the search range is 16. The results demonstrate that the proposed FABMA not only can achieve very similar search accuracy to FS, but also speed up the matching estimation largely. On the average, it runs 73% faster than BBGDS, 200% faster than FSS, and 500% faster than NTSS.

The above results can be explained by the following aspects. First, the smoother matching vector field obtained by matching vector prediction may avoid the searching falling in local optimization. Secondly, FABMA begins its search from the matching vector of one of the intra-slice or inter-slice adjacent blocks, which usually enable it to find an appropriate point earlier. Thirdly, adaptive still block determination can reduce unnecessary search, so search speed is accelerated. At last, the proposed adaptive stop criterion assures that the final point is good enough.



(a) Coding results of "Xmas" slice sequence



(b) Coding results of "Cup" slice sequence

Fig.6 Comparison between three compression methods in rate-distortion efficiency.

Table 3 Comparison of the proposed algorithm and other block matching algorithms.

Ray-space data	Search algorithm	PSNR (dB)	δ_{PSNR} (dB)	Check points	Speedup ratio
Xmas	FS	37.17	0.0	1089	1.00
	BBGDS	37.10	-0.07	9.9	110.00
	FSS	37.13	-0.04	17.1	63.68
	NTSS	37.16	-0.01	32.9	33.10
	FABMA	37.13	-0.04	5.7	191.05
Cup	FS	36.98	0.0	1089	1.00
	BBGDS	36.90	-0.08	9.4	115.85
	FSS	36.91	-0.07	17.3	62.94
	NTSS	36.92	-0.06	33.0	33.00
	FABMA	36.91	-0.07	5.4	201.67

4. CONCLUSIONS

Ray-space is an effective technology to realize FTV system with complicated scene. Since ray-space records ray data captured by multi-viewpoint system, the data quantity is extremely larger than normal video sequence. Therefore, effective compression comes to be a key technology to be solved in ray space based FTV system, and it is also associated with the complexity of the FTV system and the quality of the rendered free viewpoint images. To speed up the search of matching vector, an adaptive block matching scheme is proposed according to the characteristics of ray space correlations, and as a result, the coding efficiency is improved while the coding time is reduced. In the future work, 4D ray space compression will be investigated so as to make use of temporal correlation within ray space to achieve higher coding efficiency.

5. REFERENCES

[1] T.Fujii, T.Kimoto, M.Tanimoto, "Ray Space Coding for 3D Visual Communication", Proceedings Picture Coding Symposium, IEEE Press, 1996, 2: 447-451.

[2] P. Nabangchang, T. Fujii and M. Tanimoto, "Experimental System of Free Viewpoint Television", SPIE 2003 Stereoscopic Displays and Virtual Reality Systems, 2003, 5006:554-563.

[3] M.Levoy, P.Hanrahan, "Light field rendering", Proceeding of SIGGRAPH'96, ACM Press, 1996, 31-42.

[4] M.Tanimoto, T.Fujii, FTV ISO/IEC JTC1/SC29/WG11, M8595, Jul.2002.

[5] T.Fujii T.Komoto, M.Tanmoto, "Ray Space Coding Based on Arbitrary-Shaped DCT", Stereoscopic Display and Visual Reality Systems, SPIE, 1998, 3295: 258-268

[6] T.Fujii, T.Kimoto, M.Tanimoto, "Free-Viewpoint TV System Based on Ray-Space Representation", SPIE 2002 Three-Dimensional TV, Video, and Display, 2002, 4864-22: 175-189.

[7] SHAO Feng, JIANG Gangyi, YU Mei, et al, "Analysis of Inter-slice Correlation in Ray Space and Its Application to Data Compression", 7th International Conference on Signal Processing Proceeding, IEEE Press,2004, vol.2: 1219-1222.

[8] Zhibo Chen, Peng Zhou, Yun He, "Fast Motion Estimation for JVT", 7th Meeting: Pattaya II, Thailand, 7-14 March, 2003

ACKNOWLEDGE

This work was supported by the Natural Science Foundation of China (grant 60472100), Natural Science Foundation of Zhejiang Province (grant RC01057, Y105577), Zhejiang Science and Technology Project of China (grant 2004C31105), and Ningbo Science and Technology Project of China (grant 2003A61001, 2004A610001, 2004A630002).