GOP Adaptation Coding of H.264/SVC Based on Precise Positions of Video Cuts

Yunpeng Liu, Renfang Wang, Huixia Xu, and Dechao Sun
College of Computer Science and Information Technology, Zhejiang Wanli University, Ningbo 315100, China
[e-mail: L35633@163.com]
*Corresponding author: Yunpeng Liu

Received January 14, 2014; revised May 17, 2014; accepted June 5, 2014; published July 29, 2014

Abstract

Hierarchical B-frame coding was introduced into H.264/SVC to provide temporal scalability and improve coding performance. A content analysis-based adaptive group of picture structure (AGS) can further improve the coding efficiency, but damages the inter-frame correlation and temporal scalability of hierarchical B-frame to different degrees. In this paper, we propose a group of pictures (GOP) adaptation coding method based on the positions of video cuts. First, the cut positions are accurately detected by the combination of motion coherence (MC) and mutual information (MI); then the GOP is adaptively and proportionately set by the analysis of MC in one scene. In addition, we propose a binary tree algorithm to achieve the temporal scalability of any size of GOP. The results for test sequences and real videos show that the proposed method reduces the bit rate by up to about 15%, achieves a performance gain of about 0.28–1.67 dB over a fixed GOP, and has the advantages of better transmission resilience and video summaries.

Keywords: H.264/SVC, hierarchical B-frame, adaptive GOP, cut detection

This work is supported in part by the NSFC (Grant No. 61073074, 61303144), Ningbo Natural Science Foundation (Grant No. 2013A610069), Zhejiang Science and Technology Program (Grant No. 2012C21004), Projects in Science and Technique of Ningbo Municipal (Grant No. 2012B82003), and Zhejiang Higher Education Reform Project (Grant No. jg2013135).

http://dx.doi.org/10.3837/tiis.2014.07.014

1. Introduction

H.264/SVC [1] (referred to as SVC in our work) provides the only scalable video stream to support a variety of devices and heterogeneous networks, wherein the temporal scalability is implemented by a hierarchical B-frame structure. Compared with the conventional group of pictures (GOP) structure with a fixed size, the adaptive GOP structure (AGS) can effectively improve coding efficiency by assigning a large GOP size to the low content variation and small GOP size to rapidly changing content.

There have been in-depth studies of AGS in H.264/AVC [2–5,7,8]. In [2], scene change is detected by the existing motion vectors and residual data, but only for P-frame prediction. The work in [3] uses the methodology of motion coherence (MC), which decides frame coding type according to motion acceleration information. Motion deviation is then used instead of motion magnitude to select the number of B frames. In [4, 5], the method detects scene change for the entire sequence using mutual information (MI) [6] or the similar two-dimensional entropy model that has a better accuracy but a higher computational complexity. To get a better rate-distortion performance, the frame most similar to the frames in one GOP is chosen as an ideal I-frame in [7] such that the least number of bits are needed to achieve the desired image quality. However, the complexity of the codec structure becomes very high, the accuracy of the background modeling directly affects the later encoding process, and the accuracy and noise resistance are both poor when using the sum of absolute difference (SAD) to detect scene change. In [8], shot cut is detected by comparing an actual frame with its motion-estimated prediction using an adaptive threshold. However, this detection effect performs well only for obvious and intense shot cuts; it performs poorly for more complex scenes. A block-based abrupt and gradual scene change detection algorithm was proposed in [9] that has a low computational complexity but poor accuracy for abrupt changes. Moreover, the paper focuses on the frame level-based parallel scheduling with adaptive GOP structure. The method in [10] adjusts the GOP lengths according to the channel condition instead of scene changes. Overall, in addition to the above problems, the above papers refer only to H.264/AVC and do not consider the scalable and predictive characteristics of SVC hierarchical B-frames.

The early AGS study [11] on SVC mainly focused on motion-compensated temporal filtering (MCTF), and a sub-GOP model was implemented in an early version of the reference software JSVM. Although a fast prediction algorithm [12] was used to improve the sub-GOP model and also applied to the hierarchical B-frame structure, this AGS algorithm was removed from the SVC standard because of its particularly high complexity. In [13], the gradient of visual rhythm is used to measure the content complexity. The variance of the gradient in a GOP is calculated with different GOP sizes such as 4, 8, 16, and 32 to decide the optimal GOP size. However, a high coding performance cannot be obtained because it cannot detect scene changes with I-frame insertions. In [14], the accumulated difference of luminance pixel components is utilized to set thresholds for scene change detection and adaptive GOP size selection, however, both the accuracy and threshold adaptation are poor. A proposal for a GOP size decision algorithm based on a linear support vector machine was given in [15], however, it applies to distributed video coding (DVC) instead of SVC. Overall, in addition to the above problems, the above papers do not resolve the problem of effective temporal scalability and proportionate GOP adaptation when an I-frame is inserted at the scene cut change position.

In this paper, we propose a flexible sized GOP adaptive implementation based on the positions of video cuts. Because the judgment of MC directly affects the coding efficiency for B-frame prediction, we combine this method with MI to detect scene cuts; then, the GOP is adaptively and proportionately set by the analysis of MC in a scene, and the temporal scalability of a flexible sized GOP is achieved by a proposed binary tree algorithm. The results for test sequences and real videos show that the proposed method can achieve much better coding performance over a fixed GOP and has the advantages of better transmission resilience and video summaries.

2. Video Cut Detection

In general, abrupt transitions are much more common than gradual transitions, accounting for over 99% of all transitions [16]. For this reason, we focus on abrupt cut detection. Scene cut generally is detected by the differences of pixels (grayscale) [4,5,7,9,14,19] or image characteristics [2,8,17,18] between successive frames, and the cut position is decided according to an empirical threshold. The accuracy of pixel difference [7,9,14] and histogram-based [18] methods is poor. The block-based motion estimation and prediction [2,8,17] method is sensitive to unusual cases when the background in consecutive frames changes rapidly in addition to the appearance and disappearance of multiple objects in the same scene.

Compared with other methods, MI [4,5,19] has achieved greater success in a wider range of applications because of the accuracy of its matching and its robustness based on information theory [6]. It is calculated by

$$I_{t,t+1}^{K} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p_{t,t+1}^{K}(i,j)\log\frac{p_{t,t+1}^{K}(i,j)}{p_{t}^{K}(i)\,p_{t+1}^{K}(j)} \tag{1}$$

where *K* denotes the luminance value or one component of RGB; *N* denotes the gray levels, generally 256; *i* and *j* denote gray values; $p_t(i)$ and $p_{t+1}(j)$ denote the probability of occurrence of the pixel with gray values *i* and *j*, respectively; and $p_{t,t+1}(i, j)$ denotes the probability of occurrence of the gray pair *(i, j)* between images *t* and *t + 1*. If the MI value of adjacent images suddenly becomes smaller, indicating that the matching degree has deteriorated, this position often is a cut or significant change of image content.

Another effective cut detection method is MC [3], which measures motion deviation through the inter-frame acceleration value. In fact, the similarity not only includes that of pixel values, but also considers that of motion features. The skip mode of B frames is the typical tool used to exploit motion similarity or MC. Hence, the analysis of MC is even more important for the GOP choice of SVC. The acceleration value of the n^{th} frame is defined as

$$a(n) = |\,mv_x(n) - mv_x(n-1)\,| + |\,mv_y(n) - mv_y(n-1)\,| \tag{2}$$

where $mv_x(n)$ and $mv_y(n)$ are the horizontal and vertical components, respectively, of the feature motion vector (FMV) of frame *n*. FMV is the dominant motion vector in the frame. When no good match for a block is found during motion, a large value (999,999) is assigned. However, in the manner of MC, motion direction changes of the main object should also be considered. Therefore, *a(n)* is weighted by a factor *w(n)* to construct $a_w(n)$ as

$$a_w(n) = w(n) \times a(n) \tag{3}$$

For simplicity, *w(n)* is determined by

$$w(n) = \begin{cases} 4, \textit{direction change more than } \pi \\ 2, \textit{direction change between } \pi/6 \textit{ and } \pi \\ 1, \textit{direction change less than } \pi/6 \end{cases} \tag{4}$$

Two important performance indicators for cut detection are efficiency and accuracy. Accuracy includes two aspects: recall ratio = correct detections/(correct detections + missing detections) and precision ratio = correct detections/(correct detections + false detections). From the experimental results (at the end of this section) we determined that MI has a low efficiency but high precision ratio and MC has good efficiency and high recall but poor precision ratios. Accordingly, we combine the advantages of the two algorithms. The proposed algorithm is as follows:

Step 1: MC is used to detect a candidate cut position set P.

Step 2: If both $a_w(p\text{-}1)$ and $a_w(p)$ are greater than an empirically set threshold of 100 (here $p \in P$, hence the probability of cut change is very high based on experimental statistics), we identify the frame position as a real cut change position, otherwise we proceed to Step 3.

Step 3: In this case, the candidate position is likely caused by a relatively large motion. A sliding time window with width 10 is used to calculate the MI values of each frame.

Step 4: The inter frame absolute differences of all MI values are calculated, and if the difference of position p is a maxima, then the position is a real cut position.

In a new study on scene cut detection [20-23], although the detection accuracy is gradually improved by a more advanced algorithm, its complexity and low efficiency are not suitable for the analysis phase of video coding that needs real-time encoding in many scenarios. Furthermore, although there may be a slightly higher number of false and missing detections in the proposed method compared to the advanced algorithm (as shown in **Table 1**), from a global perspective of the video coding, the number is negligible and would not affect the overall coding efficiency for a significant number of long videos.

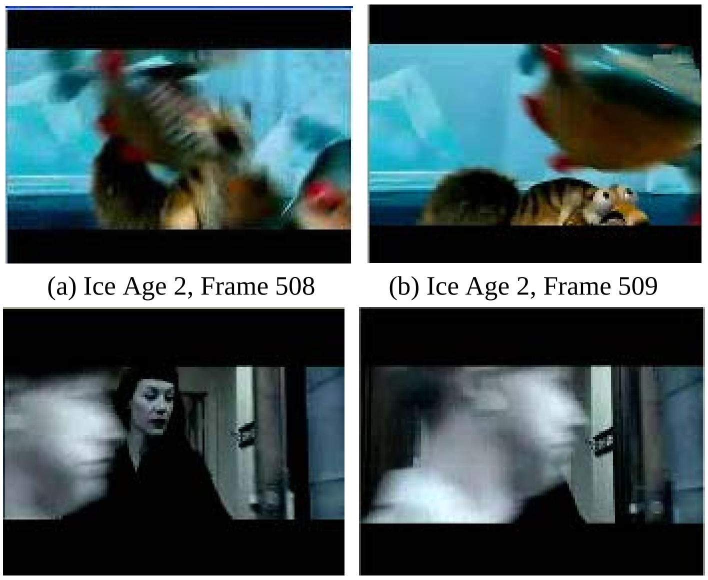
The following four methods (MI [19], MC [3], method in [23], and proposed) were tested and compared. TestA and TestB are combinations of standard sequences. TestA consisted of the sequences "stefan50," "akiyo100," "bus150," "bridge-far50," "bridge-close100," "hall150," "soccer50," and "mobile50," and TestB consisted of "foreman100," "flower50," "waterfall150," "bus100," "silent50," and "hall150." To illustrate the problem, news videos and other types of movie clips, including "The Curious Case of Benjamin Button" ("CCBB"), "James Bond-Casino Royale," and "Ice Age 2" were also tested. The results are shown in **Table 1**.

Table 1. Comparison of cut detection

Test videos	Methods	Correct detections	False detections	Missing detections	Recall ratio	Precision ratio
TestA	MI	7	0	0	100%	100%
	MC	7	0	0	100%	100%
	[23]	7	0	0	100%	100%
	Proposed	7	0	0	100%	100%
TestB	MI	5	0	0	100%	100%
	MC	5	0	0	100%	100%
	[23]	5	0	0	100%	100%
	Proposed	5	0	0	100%	100%

News	MI	70	3	5	93.33%	95.89%
	MC	74	5	1	98.66%	93.67%
	[23]	75	2	0	100%	97.40%
	Proposed	74	4	1	98.66%	94.87%
CCBB	MI	72	5	4	94.73%	93.51%
	MC	74	6	2	97.37%	92.50%
	[23]	75	2	1	98.68%	97.40%
	Proposed	74	3	2	97.37%	96.10%
James Bond	MI	74	5	9	89.16%	93.67%
	MC	81	9	2	97.59%	90.00%
	[23]	81	4	2	97.59%	95.29%
	Proposed	81	3	2	97.59%	96.43%
Ice Age 2	MI	82	3	7	92.13%	96.47%
	MC	87	8	2	97.75%	91.58%
	[23]	89	3	0	100%	96.74%
	Proposed	87	3	2	97.75%	96.67%

Because TestA and TestB comprise completely different sequences, the scene change is very obvious, and all cut change positions can be correctly detected. For news and real movies, MC has a lower precision ratio than MI. This is more obvious for “James Bond” and “Ice Age 2,” that contain more fast motion and dynamic content that is often mistaken for cut position. As shown in **Fig. 1**, frame 509 in “Ice Age 2” and frame 1075 in “CCBB” are false detections.



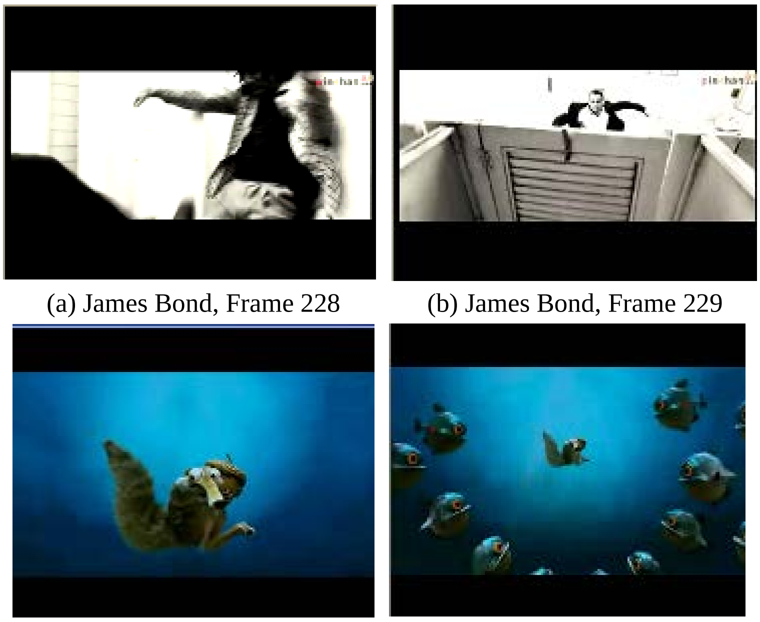
(a) Ice Age 2, Frame 508 (b) Ice Age 2, Frame 509

(c) CCBB, Frame 1074 (d) CCBB, Frame 1075

Fig. 1. MC false detections

MI has a lower recall ratio than MC. This is because in some large and similar contexts when the values of the luminance are relatively limited, although the overall distribution of the luminance and color is different, the proportion is close, and hence MI will regard it as a close match. As shown in **Fig. 2**, frame 229 in “James Bond” and frame 370 in “Ice Age 2” are missed detections. Further, as listed in **Table 1**, because of the combination of the advantages of MI and MC, the recall and precision ratio of the proposed algorithm are higher than either

MI or MC. Further, only a very small proportion of the frames are calculated by MI, and hence the low computational complexity makes the algorithm suitable for real-time processing.



(a) James Bond, Frame 228

(b) James Bond, Frame 229

(c) Ice Age 2, Frame 369

(d) Ice Age 2, Frame 370

Fig. 2. MI missing detections

3. Adaptive GOP Structure

The overall cut detection-based process for adaptive GOP is shown in **Fig. 3**.

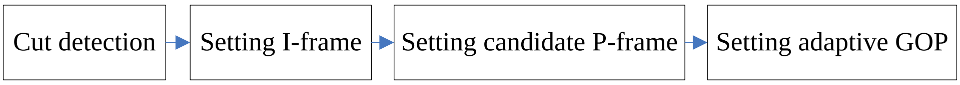


Fig. 3. Adaptive GOP implementation block diagram

3.1 Setting I-frames

Scene changes or large variations in one GOP are the main cause of hierarchical B-frame coding performance degradation. To solve this problem, generally an I-frame is inserted at the cut change position. Based on the statistics of a large number of cut change scenes, the duration of one scene normally is not less than 2 seconds, and since video frame rate changes from 25–30 fps, the duration of one scene is not less than 32 frames. Such a scene is called a valid encoding scene, and the frame in the cut change is set as an I-frame.

3.2 Setting candidate P-frames

(1) Setting key frame type 1

If the duration of one scene from a cut change is less than 32 frames, the frame in the cut change is determined not to be a I-frame but a key frame 1 (K1), later used for GOP partitioning. Because scenes of less than 32 frames are very rare, they have little impact on the overall coding efficiency. If there are many of these kind of scenes, frequent I-frame encoding will decrease the encoding efficiency.

(2) Setting key frame type 2

In our proposed cut detection algorithm above, the frames detected by MC but removed by MI are defined as key frames type 2 (K2), not cut positions but scenes containing relatively intense motion and introducing significant new content.

(3) Setting key frame type 3

The frames that destroy the motion coherence are defined as key frames type 3 (K3). A normalized curve after an MC detection for the "CCBB" clip is shown in **Fig. 4.** The local maxima that are not cut positions at frames 208–217 or near frame 262 are the candidate K3 frames. We can see that a motion change may last for several frames. To determine which frame is the most important of these, a function $a_{c,m}(n)$ that is a convolution of $a_w(n)$ with a time window, is introduced as

$$a_{c,m}(n) = \frac{1}{w}\sum_{\Delta t=-W/2}^{W/2} a_w(n+\Delta t),\ \ W = 2i+1,\ \ i = 0,1,... \qquad (5)$$

where W is the width of the time window. The frames corresponding to the peaks of $a_{c,m}(n)$ are then extracted as K3s.

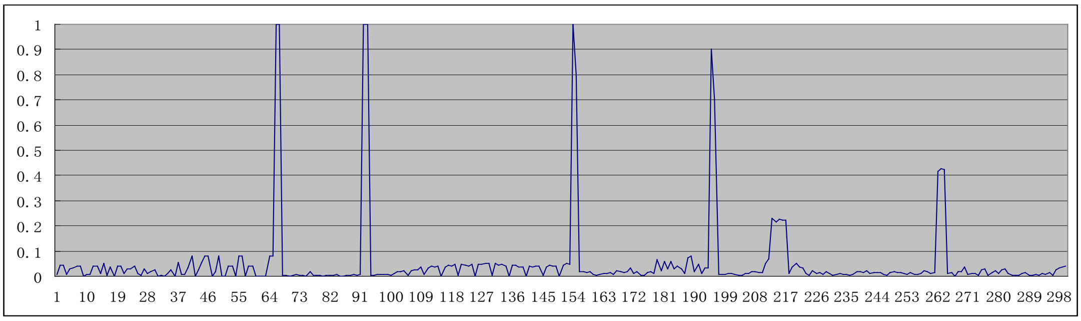


Fig. 4. $a_w(n)$ normalized curve of the "CCBB" clip

The frames belonging to the union of K1, K2, and K3 are determined to be the candidate P-frames of GOPs in one scene.

3.3 Setting the adaptive GOP

After the identification of I-frames and candidate P-frames, the adaptive algorithm proceeds as follows:

Step 1: The mean and approximate variance of $a_w(n)$ between two I-frames are calculated, and if both values are greater than threshold T1, the motion is relatively intense and the basic size (BS) of the GOP is set 4 or 8, otherwise it is set to 16 or 32. The empirical value of T1 is 1 for the QCIF format and 2 for the CIF and QVGA formats.

Step 2: If there are no candidate P-frames between two consecutive I-frames, go to Step 3, otherwise go to Step 5.

Step 3: First, the GOP is divided according to the value of the BS from Step 1. The remaining frames are then evenly distributed in order. For example, if BS = 32 and the frame number in one scene is 104, the GOP structure is I.32*3.4.2.1 by conventional AGS, but I.34*2.35 by our proposed method, which not only enhances the inter-frame correlation in one GOP, but also tries to ensure that GOPs with similar motion to proportionate sizes. Hence, temporal scalability is balanced, where low temporal levels represent more main video content and a good rhythm of images and a lower bit rate can represent more scene content in a constrained transmission environment. When a large sized GOP appears, e.g., 34 and larger, go to Step 4.

Step 4: A unified and standard method must be used to achieve the temporal levels, predictive structure, and sequence of the hierarchical B-frame according to GOP size. The proposed binary tree-based method can effectively solve this problem.

(1) A node of the binary tree is defined as *Node(beginValue, endValue)*, where *endValue* > *beginValue*, and *GOPSize* denotes the currently processed GOP size.

(2) The creation of the binary tree is shown in **Fig. 5**. A recursive algorithm is used. First, *Node(0, GOPSize)* is created as the root node, then the left and right nodes are created by the rules and recursively set as descendants of the root. In order to illustrate the rules clearly, examples are shown in **Fig. 6**. The temporal scalability and predictive structure of GOP structure I.5.6 are shown in **Fig. 7**.

(3) The height of the binary tree is equal to the total temporal levels of the GOP. In one GOP, *CodingIndex* denotes the actual coding index, *FrameId* denotes the display index, and *TemporalId* denotes the temporal level of each frame.

(4) When encoding each frame in one GOP, *CodingIndex* is known, but *FrameId* and *TemporalId* must be calculated from the binary tree. Because the preorder traversal sequence of the binary tree is the same as the encoding sequence, the corresponding *FrameId* can be calculated from the known *CodingIndex*.

(5) When the *endValue* of the left node or *firstValue* of the right node is equal to *FrameId* during a preorder traversal, the node is found. The level of the found node becomes the *TemporalId*.

(6) After obtaining the correct parameters, the correct predictive sequence and temporal scalability can be found for one GOP.

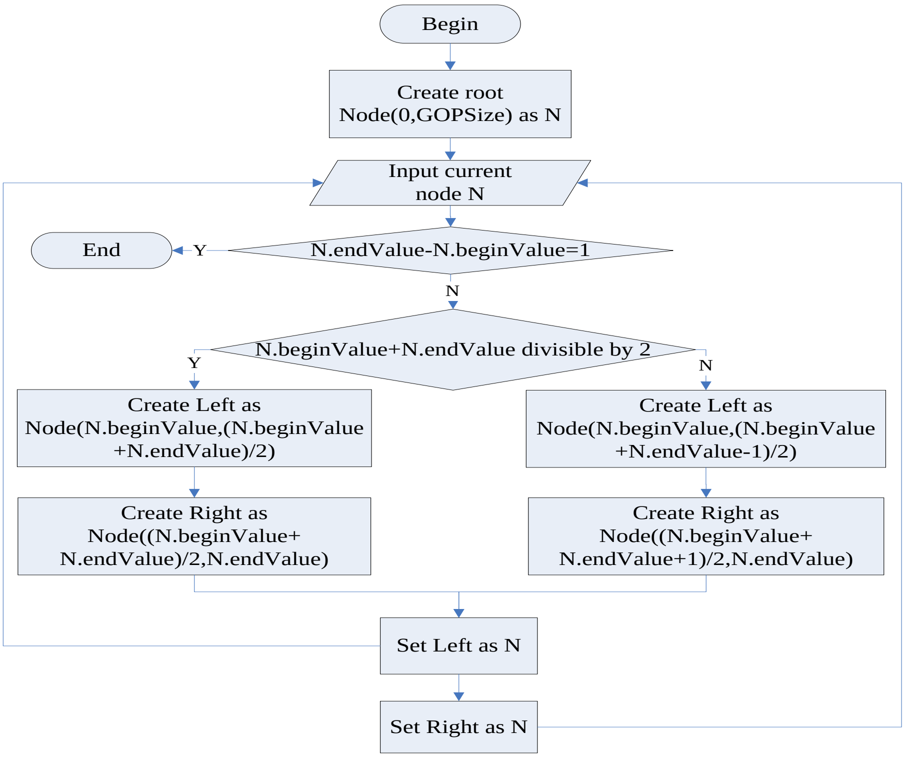
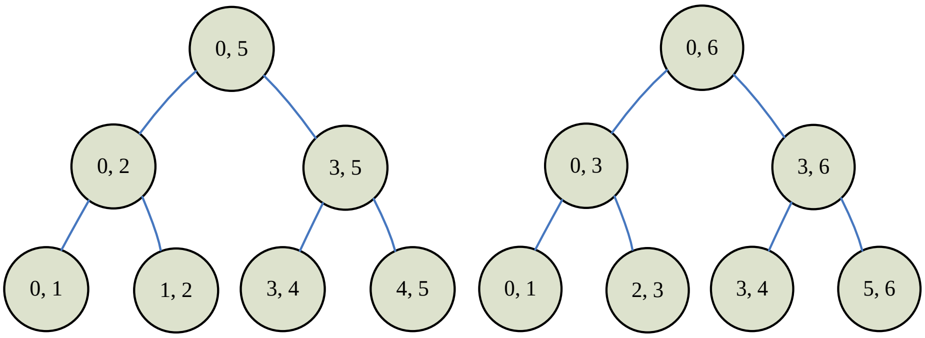

Begin
Create root Node(0,GOPSize) as N
Input current node N
End
Y
N.endValue-N.beginValue=1
N
N.beginValue+N.endValue divisible by 2
Y
N
Create Left as Node(N.beginValue,(N.beginValue +N.endValue)/2)
Create Left as Node(N.beginValue,(N.beginValue +N.endValue-1)/2)
Create Right as Node((N.beginValue+ N.endValue)/2,N.endValue)
Create Right as Node((N.beginValue+ N.endValue+1)/2,N.endValue)
Set Left as N
Set Right as N

Fig. 5. Creation flow chart for the binary tree



(a) GOPSize = 5 (b) GOPSize = 6

Fig. 6. Examples of binary tree creation

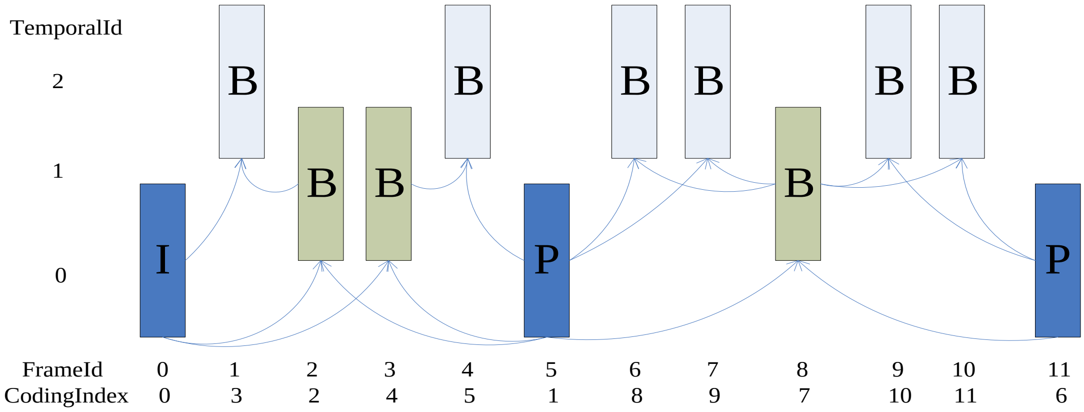
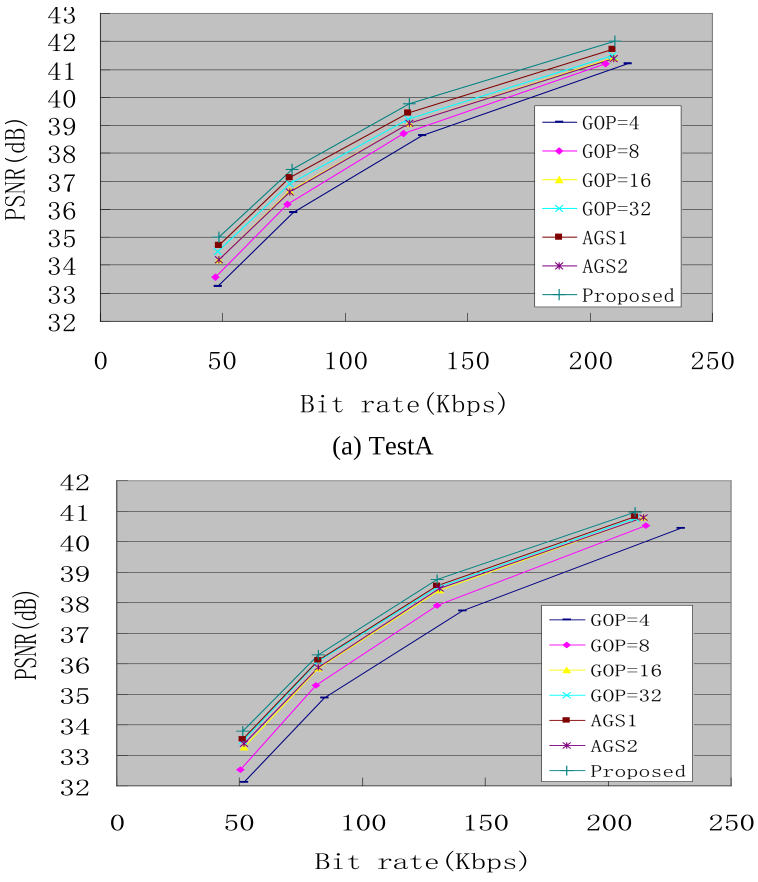

TemporalId
2
1
0
I
B
B
B
B
P
B
B
B
B
B
P
FrameId 0 1 2 3 4 5 6 7 8 9 10 11
CodingIndex 0 3 2 4 5 1 8 9 7 10 11 6

Fig. 7. Predictive structure of hierarchical B-frame for flexible sized GOP

Step 5: If the distance between the current candidate P-frame and previous I-frame or P-frame is less than four frames, the candidate P-frame is canceled. The remainder are the actual P-frames, and hence the GOP division between an I-frame and P-frame or two P-frames is the same as the algorithm in Step 3.

4. Experimental Results

To illustrate the effectiveness of the algorithm, different standard sequences were randomly combined to form the scene cut changes. The TestA and TestB sequences were the same as those used in Section 2. The video codec used was the SVC reference software JSVM9.19.14; the resolution was QCIF; QP was respectively taken at 24, 28, 32, and 36; and the compared fix GOP respectively was 4, 8, 16, and 32. The algorithms in [14] and [8] are referred to as AGS1 and AGS2, respectively. The computer configuration were: Pentium(R) Dual-Core E5200 CPU 2.50 GHz, with 2 G DDR2 running Windows XP. Comparisons of the rate-distortion curves are shown in **Fig. 8**, and the comparisons of PSNR (dB) and bit rate (Kpbs) for QP = 28 are listed in **Table 2**.



(a) TestA

(b) TestB

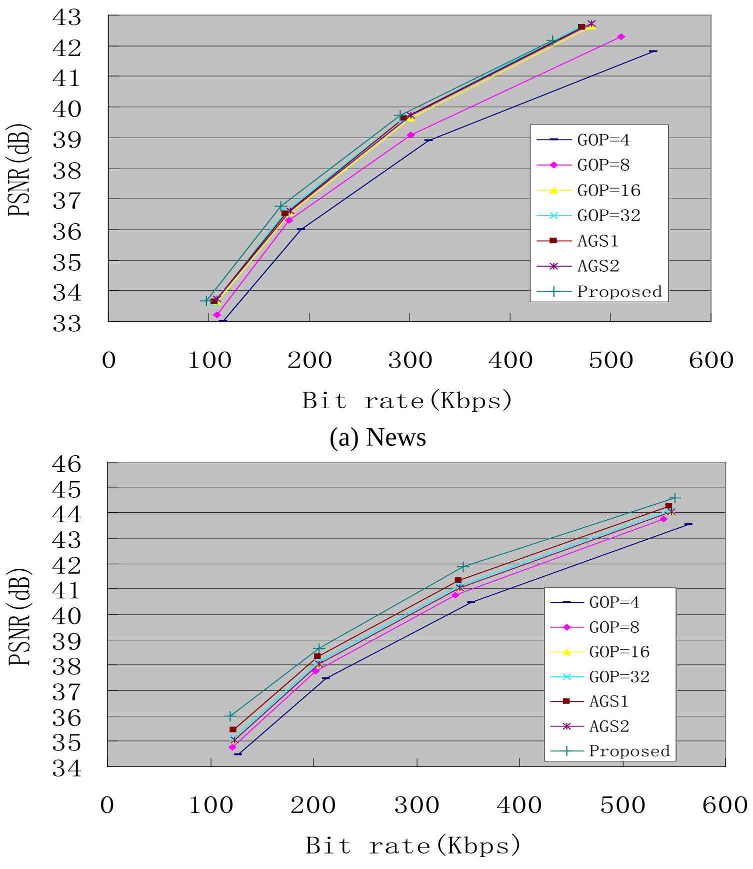
Fig. 8. Comparisons of rate-distortion curves

Table 2. Comparisons of PSNR and bit rate of combined sequences

QP28	**TestA**		**TestB**	
	PSNR(dB)	Bit rate(Kbps)	PSNR(dB)	Bit rate(Kbps)
GOP4	38.62 (-1.14)	131.1 (3.74%)	37.73 (-1.05)	140.5 (7.33%)
GOP8	38.69 (-1.07)	123.8 (-1.94%)	37.90 (-0.88)	130.3 (0.08%)
GOP16	39.09 (-0.67)	126.1 (-0.08%)	38.41 (-0.37)	131.5 (0.99%)
GOP32	39.24 (-0.52)	125.6 (-0.45%)	38.50 (-0.28)	130.3 (0.08%)
AGS1	39.46 (-0.3)	125.8 (-0.32%)	38.55 (-0.23)	130.3 (0.08%)
AGS2	39.10 (-0.66)	126.1 (-0.08%)	38.44 (-0.34)	130.8 (0.46%)
Proposed	39.76	126.2	38.78	130.2

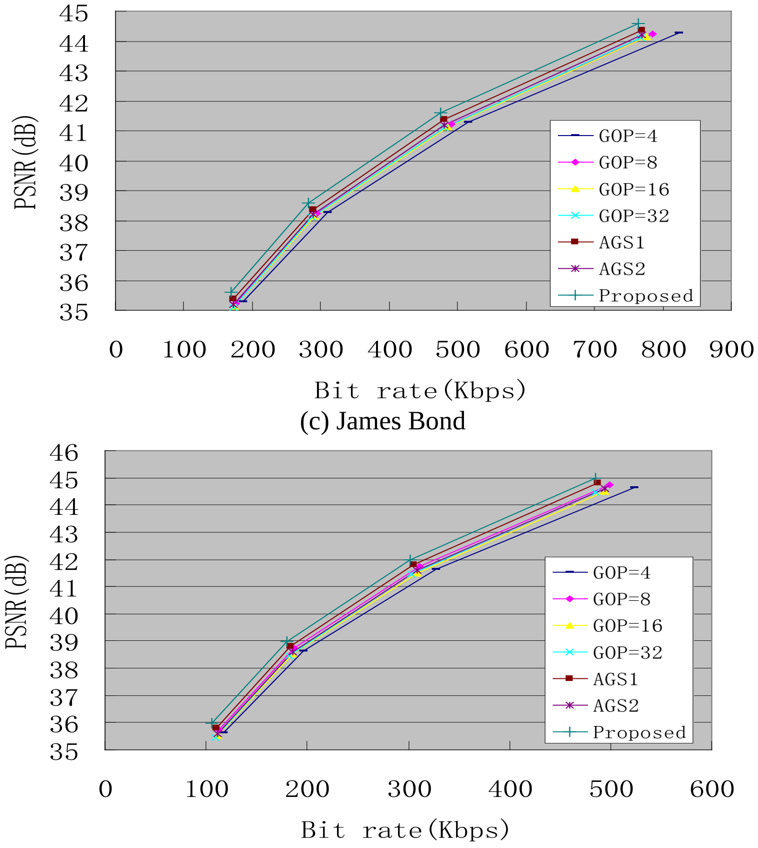
TestA comprises sequences with more relatively intense motions. Therefore, as shown in **Fig. 8(a),** there is not much variation among the PSNR for GOPs with fixed sizes of 4, 8, 16, and 32 frames. The coding efficiency of AGS1 slightly outperforms GOP32. On the other hand, as shown in **Fig. 8(b),** TestB comprises sequences with small motions, and in this case, GOPs with a fixed size of 32 show the best coding efficiency. Furthermore, the performance of AGS1 is almost the same as GOP32. The performance of AGS2 is a little better than GOP16. Regardless of whether a fixed-size GOP, AGS1, or AGS2 is used, our proposed method shows the best coding efficiency. From further analysis, as shown in **Table 2**, when QP is fixed, GOP4 has the worst rate-distortion performance compared to the proposed method, TestA and TestB respectively have 1.14 and 1.05 dB PSNR reductions, as well as 3.74% and 7.33% bit rate increases. Compared to GOP8, GOP16, GOP32, AGS1, and AGS2, our proposed method has a similar bit rate, but 1.07–0.3 dB and 0.88–0.23 dB PSNR gains for TestA and TestB, respectively. Because more frames with similar MC are assigned to the same GOP, the number of GOPs is reduced, a better balance and proportion is maintained, and hence the bidirectional prediction correlation of hierarchical B-frame is enhanced.

To further illustrate the reliability and practicality of the algorithm, we also tested the real news and movie clips used in the experiment of Section 2. Each video presented different characteristics of motion and illumination. The resolution was QVGA, QP was 28, and the number of tested frames was 2000. Comparisons of the rate-distortion curves are shown in **Fig. 9**. **Table 3** compares the PSNR (dB) and bit rate (Kpbs) for QP = 28.



(a) News

(b) CCBB



(c) James Bond

(d) Ice Age 2

Fig. 9. Comparisons of rate-distortion curves for news and real movies

Table 3. Comparisons of PSNR and bit rate for news and real movies

QP28	**News**		**CCBB**		**James Bond**		**Ice Age 2**	
	PSNR (dB)	Bitrate (Kbps)	PSNR (dB)	Bitrate (Kbps)	PSNR (dB)	Bitrate (Kbps)	PSNR (dB)	Bitrate (Kbps)
GOP4	39.41 (-1.67)	319.1 (15.26%)	40.66 (-1.47)	352.5 (6.21%)	41.27 (-0.32)	515.0 (8.64%)	41.64 (-0.34)	326.9 (6.42%)
GOP8	39.60 (-1.48)	300.8 (10.11%)	40.76 (-1.37)	337.4 (2.02%)	41.23 (-0.36)	490.8 (4.14%)	41.73 (-0.25)	311.6 (1.83%)
GOP16	39.74 (-1.34)	301.0 (10.17%)	41.06 (-1.07)	342.2 (3.39%)	41.13 (-0.46)	485.5 (3.09%)	41.51 (-0.47)	308.9 (0.97%)
GOP32	39.62 (-1.46)	293.4 (7.84%)	41.13 (-1.00)	341.4 (3.16%)	41.10 (-0.49)	480.1 (2.00%)	41.47 (-0.51)	303.1 (-0.92%)
AGS1	40.13 (-0.95)	294.7 (8.25%)	41.55 (-0.58)	341.0 (3.05%)	41.37 (-0.22)	480.7 (2.12%)	41.80 (-0.18)	304.9 (-0.33%)
AGS2	39.90 (-1.18)	300.5 (10.01)	41.08 (-1.05)	342.0 (3.33%)	41.20 (-0.39)	490.0 (3.98%)	41.71 (-0.27)	312.0 (1.95%)
Proposed	41.08	270.4	42.13	330.6	41.59	470.5	41.98	305.9

As can be seen in **Fig. 9(a),** because there are rarely fast motions in “News,” the variation of the rate distortion curves is more obvious. It is clear that GOP4 shows the worst performance. GOP16, GOP32, AGS1, AGS2, and the proposed method have similar performances. Compared to “News,” “CCBB” has more motions, and the differences among curves begins to decrease, as shown in **Fig. 9(b)**. “James Bond” and “Ice Age 2” have many intense motions, hence there is not much difference among the curves, as shown in **Fig. 9(c)** and **Fig. 9(d)**. To summarize, the proposed algorithm maintains an optimal rate distortion performance across the four different types of test videos.

From the further analysis listed in **Table 3**, the cut change is evident in video “News,” where the content is very different between adjacent scenes and there are only small motions. Therefore, the corresponding coding performance of the proposed algorithm is significantly improved, achieving a 15.26% bit rate reduction and 1.67 dB PSNR gains, 7.84% bit rate reduction and 1.46 dB PSNR gains, and a 8.25% bit rate reduction and 0.95 dB PSNR gains when compared to GOP4, GOP32, and AGS1, respectively. There are a certain number of intense motions in “CCBB,” but the majority of its motions are small. Hence, the performance of the proposed algorithm is somewhat poorer than for “News.” Its bit rate reduction varies from 3.05–6.21%, and the corresponding PSNR gains vary from 0.58–1.47 dB. “Ice Age 2” is an animation, hence it and “James Bond” have more high motion intensive scenes that are not favored by large GOPs. Various methods have similar performance results for the two movies. GOP32 has the worst PSNR, and GOP4 has the highest bit rate. Compared to GOP4, the proposed algorithm has a 8.64% (James Bond) and 6.42% (Ice Age 2) bit rate reduction and about 0.3 dB PSNR gains for both movies. Compared to GOP32 and AGS1, it has a similar bit rate, and respectively about 0.5 and 0.2 dB PSNR gains for both movies. Because AGS2 has worse shot cut detection, it has a similar performance to GOP16 in “News” and “CCBB,” and a similar performance to GOP8 in “James Bond” and “Ice Age 2”.

5. Conclusion

To improve coding efficiency and fast retrieval for H.264/SVC, cut positions are first accurately detected by a combination of MC and MI, and I-frames are inserted at cut positions. This destroys the prediction structure of hierarchical B-frame such that inter-frame correlation and temporal scalability are impacted. Hence, the flexible sized GOP adaptive algorithm was proposed to minimize this impact. The test results show that our proposed algorithm can further improve the coding efficiency and favor the formation of video summaries compared with fixed GOPs and other advanced AGS algorithms.

References

[1] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, September, 2007. Article (CrossRef Link)

[2] Ding JunRen, Lin JiKun and Yang JarFerr, “Motion-based adaptive GOP algorithms for efficient H.264/AVC compression,” in *Proc. of the 9th Joint Conference on Information Sciences*, pp. 1-4, Octorber 8-11, 2006. Article (CrossRef Link)

[3] Ma Yanzhuo, Wan Shuai, Chang Yilin, Yang Fuzheng and Wang Xiaoyu, “Adaptive GOP structure based on motion coherence,” in *Proc. of SPIE*, pp. 74550T-1-74550T-8, August 4-5, 2009. Article (CrossRef Link)

[4] B. Zatt, M. Porto, J. Scharcanski and S. Bampi, “GOP structure adaptive to the video content for efficient H.264/AVC encoding,” in *Proc. of 17th International Conference on Image Processing*,

pp. 3053-3056, September 26-29, 2010. Article (CrossRef Link)

[5] Lenka Krulikovská, "A novel method of adaptive GOP structure based on the positions of video cuts," in *Proc. of 53rd International Symposium ELMAR*, pp. 67-70, September 14-16, 2011.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theroy*, Wiley, New York, 1991. Article (CrossRef Link)

[7] Paul Manoranjan, Lin Weisi, Lau Chiew Tong and Lee Bu Sung, "Explore and model better I-frames for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1242-1254, September, 2011. Article (CrossRef Link)

[8] Lenka Krulikovsk´a and Jaroslav Polec, "GOP structure adaptable to the location of shot cuts," *International Journal of Electronics and Telecommunications*, vol. 58, no. 2, pp. 129-134, June, 2012. Article (CrossRef Link)

[9] Hsiao HsuFeng and Wu ChenTsang, "Balanced parallel scheduling for video encoding with adaptive GOP structure," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2355-2364, December, 2013.Article (CrossRef Link)

[10] Ling Zi, Jiang XingJun and Liu JianJun, "Efficiency of dynamic GOP length in video stream," *Advanced Materials Research*, vol. 765-767, pp. 879-884, May, 2013. Article (CrossRef Link)

[11] M. W. Park, G. H. Park, S. Jeong, D. Y. Suh and K. Kim, "Adaptive GOP structure for joint scalable video coding," *IEICE Transactions on Communications*, vol. E90-B, no. 2, pp. 431-434, February, 2007. Article (CrossRef Link)

[12] YiHau Chen, ChiaHua Lin, ChingYeh Chen and LiangGee Chen, "Fast prediction algorithm of adaptive GOP structure for SVC," in *Proc. of SPIE-IS&T Electronic Imaging*, pp. 65080U-1-65080U-9, January 30 - February 1, 2007. Article (CrossRef Link)

[13] Chen HungWei, Yeh ChiaHung, Chi MingChieh, Hsu ChingTing and Chen MeiJuan, "Adaptive GOP structure determination in hierarchical B picture coding for the extension of H.264/AVC," in *Proc. of International Conference on Communications, Circuits and Systems*, pp. 697-701, May 25-27, 2008. Article (CrossRef Link)

[14] Tian Song, Shinpei Matsuoka, Yoshitaka Morigami and Takashi Shimamoto, "Coding efficiency improvement with adaptive GOP selection for H.264/SVC," *International Journal of Innovative Computing, Information and Control*, vol. 5, no.11, pp. 4155-4165, November, 2009.

[15] Masala Enrico, Yu Yanmei and He Xiaohai, "Content-based group-of-picture size control in distributed video coding," *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 332-344, March, 2014. Article (CrossRef Link)

[16] S Paschalakis and D Simmons, "Detection of gradual transitions in video sequences," *http://www.wipo.int/pctdb/en/wo.jsp?WO=2008046748&IA=EP2007060594&DISPLAY=STATUS*, 2008.

[17] Yu Zhenyu and Lin Zhiping, "Scene change detection using motion vectors and DC components of prediction residual in H.264 compressed videos," in *Proc. of IEEE Conference on Industrial Electronics and Applications*, pp. 990-995, July 18-20, 2012. Article (CrossRef Link)

[18] I. Radwan Nisreen, M. Salem Nancy, I. El Adawy Mohamed, "Histogram correlation for video scene change detection," *Advances in Intelligent and Soft Computing*, vol. 166, no. 1, pp. 765-773, May, 2012. Article (CrossRef Link)

[19] Angadi Shanmukhappa and Naik Vilas, "Shot boundary detection and key frame extraction for sports video summarization based on spectral entropy and mutual information," *Lecture Notes in Electrical Engineering*, vol. 221, no. 1, pp. 81-97, January, 2013. Article (CrossRef Link)

[20] Nourani Vatani Navid, Borges Paulo Vinicius Koerich, Roberts Jonathan and Srinivasan Mandyam , "On the use of optical flow for scene change detection and description," *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2013. Article (CrossRef Link)

[21] Hamidreza Rashidy Kanan and Roghayeh Dadashi, "AVCD-FRA: A novel solution to automatic video cut detection using fuzzy-rule-based approach," *Computer Vision and Image Understanding*, vol. 117, no. 7, pp. 807-817, July, 2013. Article (CrossRef Link)

[22] Viral B Thakar and Sarman K Hadia, "An adaptive novel feature based approach for automatic video shot boundary detection," in *Proc. of International Conference on Intelligent Systems and*

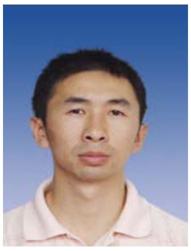
Signal Processing, pp. 145-149, March 1-2, 2013. Article (CrossRef Link)

[23] Birinci Murat and Kiranyaz Serkan, “A perceptual scheme for fully automatic video shot boundary detection,” *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 410-423, March, 2014. Article (CrossRef Link)

Yunpeng Liu received the Ph.D. from the School of Computer Science and Technology, Zhejiang University, in 2013, and is currently an associate professor at the School of Computer Science and Information, Zhejiang Wanli University, China. His research interests include video analysis, video coding , pattern recognition, and machine learning.

Renfang Wang received the Ph.D. from the School of Computer Science and Technology, Zhejiang University, in 2008, and is currently a full professor at the School of Computer Science and Information, Zhejiang Wanli University, China. His research interests include computer graphics and digital image processing.

Huixia Xu received her PhD from Department of Mathematics, Zhejiang University, in 2008, and is currently an associate professor at Institute of Mathematics, Zhejiang Wanli University, China. Her research interests include computer-aided geometric design and digital geometry processing.



Dechao Sun is a Ph.D student of Signal and Information Processing, Ningbo University, and is also an senior engineer at the School of Computer Science and Information, Zhejiang Wanli University, China. His research interests include digital geometry processing and embedded system.