Adaptive Temporal Decimation Algorithm with Dynamic Time Window

Sai-Ho Kwok, *Member, IEEE*, Wan-Chi Siu, *Senior Member, IEEE*, and Anthony G. Constantinides, *Senior Member, IEEE*

***Abstract*— Decimation approaches for image processing have been widely used for various applications. For video processing, decimation refers to sampling the frame rate in order to reduce the number of processing frames. Most of these temporal decimation methods discard whole frames; as a result, some high-speed motions could be completely eliminated while some redundant frames might remain in the processing frames. Until recently, an adaptive temporal decimation approach has been successfully developed by Olstad to take both spatial and temporal information into consideration and is fully compatible with some existing discrete cosine transform (DCT)-based standards, such as MPEG and H.261. Moreover, it theoretically preserves all high activity motions and discards all low activity motions. However, we found that it is still not fully adaptive due to the confinement of the size of the time window. The discontinuity detection method is quite complex and, more importantly, the efficiency of coding block position maps is fairly low. In this paper, we propose to resolve the problem of the time window by a dynamic time window approach. By using variable sizes of the time window, the optimal number of remaining frames could be produced. It also enhances the visual quality of the resulting video while the compression is comparable with the conventional approach. Based on our proposed algorithm, a simple but efficient quantization process has been used to replace the highly complex temporal discontinuity detection. The conventional adaptive temporal decimation algorithm operates on the basis of block sequences, but our dynamic approach which can retain all high activity blocks operates on the spatiotemporal domain. This approach can reduce redundant planes with slow activity and give higher precision for blocks with high activity. Experimental results show that the proposed algorithm achieves the optimal number of remaining frames.**

***Index Terms*— Adaptive temporal decimation, dynamic time window, temporal discontinuity detection, video coding.**

I. INTRODUCTION

DECIMATION methods are common techniques that have been widely used in image processing. They have been used in conjunction with some existing video processing techniques such as with the motion vector estimation [1]–[3]. However, their major application is on sampling the frame rate in order to reduce the number of processing frames. Most of these temporal decimation methods discard some whole frames, hence, some high-speed motions could be missing and some redundant frames might remain.

The temporal decimation technique is regarded as a powerful tool to increase compression for video coding. A simple temporal decimation can be achieved by periodically dropping whole frames from the image sequence. However, dropping frames may cause objectionable temporal aliasing artifacts, especially in regions where the temporal activity is high. The approach became more mature recently and some good contributions were given by Olstad's adaptive temporal decimation [4]. Olstad's algorithm adaptively adjusts the amount of decimation in the time direction depending upon the local activity. The approach appears to be successful and it takes both spatial and temporal information into consideration while it is fully compatible with several existing discrete cosine transform (DCT)-based standards such as the MPEG and the H.261. Theoretically, it preserves all high activity motions and discards all low activity motions. However, we have found that the approach is still not fully adaptive for only using the size of the time window. Its discontinuity detection method is quite complex, and more importantly, the efficiency of coding the block position maps is fairly low. There are some prone problems with Olstad's algorithm as shown below.

1) *No adaptive control on the number of segments*: Blocks with no or little spatial activities are compulsively decimated into a specified number of segments, expecting that very few bits are needed to represent them. On the other hand, blocks with significant changes are confined to a limited number of segments so that more bits are required. Hence, distortion is inevitable. The number of segments in a block sequence is affected by the number of remaining frames, K. In other words, the bit rate heavily depends upon the input sequence.
2) *Not to be able to preserve all important blocks*: High activity blocks may be anywhere in the time window. In typical cases, continuous motions often occur in certain block sequences. They should all be retained in order to produce a faithful reconstruction. Adaptive temporal decimation algorithms fail to handle this situation. They result in temporal aliasing artifacts occurring in the reconstruction sequence.
3) *Portability*: There are no guidelines for determining the size of the time window N. The value of N varies for different video sequences.
4) *Ineffective coding of the block position map (BPM)*: The entropy of the BPM is rather high for any image sequence due to the fixed number of ones in the map

---

Manuscript received March 8, 1996; revised February 10, 1997. This paper was recommended by Associate Editor H. Gharavi. This work was supported by the British Council, Grant JRS 93/35.

S.-H. Kwok was with the Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BT, England. He is now with the Department of Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

W.-C. Siu is with the Department of Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

A. G. Constantinides is with the Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BT, U.K.

Publisher Item Identifier S 1051-8215(98)01551-1.

and a relatively small compression ratio for the variable length coding (VLC).

5) *Too many Intra frames*: Intra frames are much more expensive than Inter frames. Each time window needs one Intra frame for eliminating all propagating quantization errors across the window boundaries.

Many of the above drawbacks could be reduced or eliminated in our temporal decimation algorithm.

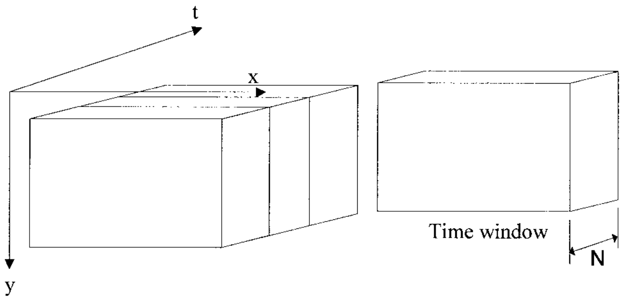
Let us give the organization of the rest of the paper. Section II of this paper defines some terms for our newly proposed algorithm. Section III introduces a visually lossless quantization algorithm for detecting temporal discontinuity. Section IV gives an overview of the dynamic time window approach. Section V illustrates the shuffling mechanism of the dynamic approach. Section VI examines the performance of the proposed algorithm as comparing it with the conventional algorithm. Section VII concludes the paper.

II. TERM DEFINITIONS

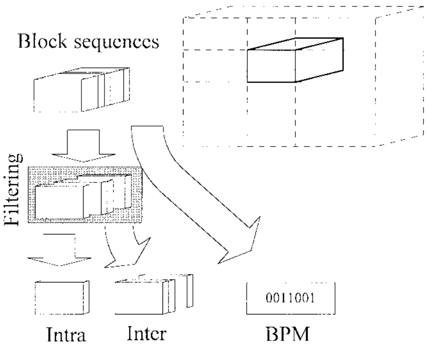
In this section, we define some key terminologies for the temporal decimation problem. Fig. 1 depicts a graphical representation of the temporal decimation algorithm. Fig. 1(a) indicates a time window while Fig. 1(b) explains the spatial temporal decimation approach.

A fixed number of image frames from the sequence is grouped as the initial *time windows* as shown in Fig. 1(a). The initial size N of the time window is predefined. This value is gradually squeezed to an optimal number K (which is equal to the total number of Inter frames) due to the temporal decimation. The value K, which is an unknown before the process, is the total number of remaining frames. Each time window is spatially subdivided into block sequences as shown in Fig. 1(b). The block size was selected as 8 × 8 pixels in our experimental work. Pixel intensities in each block sequence can be mathematically represented as $I(x, y, t)$ with $x, y \in \{0, \cdots, 7\}$ and $t \in \{0, \cdots, N-1\}$ where N denotes the initial number of frames in the current time window. For temporal decimation, a number of blocks within the block sequence has to be identified and discarded from the block sequence. The blocks to be eliminated contain relatively low activity in the block sequence and are known as *low activity blocks*. On the other hand, blocks remaining in the block sequence are *high activity blocks*. The number of removed blocks is $N - K$. After the elimination of the low activity blocks, block sequences can be considered to be composed of *segments*. Let us define a discontinuity plane to be the first block of a segment. Then all discontinuity segments are filtered with respect to their segments by an averaging filter which is defined as follows.

Let $I(x,y) = \{\cdots, I_r(x,y,1), I_r(x,y,2), I_e(x,y,3), I_e(x, y,4), I_r(x,y,5), I_r(x,y,6), I_r(x,y,7), \cdots\}$ be a possible sequence of intensity values at a certain position (x, y) in the temporal domain, where the subscript e stands for pixels in the eliminated blocks and r refers to pixels in the remaining blocks. The average filter will replace $I_r(x,y,2)$ by $I_r(x,y,2)'$, which is the average of $I_r(x,y,2)$, $I_e(x,y,3)$, and $I_e(x,y,4)$, for example.



(a)



(b)

Fig. 1. Graphical representation of temporal decimation algorithm. (a) Diagram showing time windows of a sequence. (b) Forming three prime components.

Three prime components, *Intra*, *Inter*, and *BPM*, are required for the reconstruction of the complete block sequence. The first block in the block sequence, which is called *Intra block*, acts as a reference for the rest of the blocks. Blocks other than the Intra block are *Inter blocks*. In order to reconstruct the complete block sequences, a mechanism (*BPM*) is required to indicate the positions of the discarded blocks. In our example in Fig. 1(b), zeros indicate the missing blocks.

III. VISUALLY LOSSLESS QUANTIZATION

For identifying blocks to remain in the block sequence, Olstad provided a temporal discontinuity detection. However, his method is computationally intensive. We propose a simple but efficient method to identify high activity blocks of block sequences in this section. The basic principle of our approach is quantization. We quantize block sequences into a smaller number of intensity levels, then apply the mean absolute difference (MAD) matching criterion to identify high activity blocks for continuous segments.

For a typical b-bit gray-scale image representation, the number of intensity levels is 2^b; for b is usually in the range $6 \leq b \leq 8$. It is noteworthy that quantization levels less than 64 will introduce false contours or edges to the human visual system [5]. Therefore, a quantization process using 6-bit representation appears to be lossless to human visual system (HVS). The number of representation levels is not so important to our method because our approach is just to make use of this representation for the detection of temporal continuity within the block sequences. The process of uniform quantization and compact representation of the $u \times v$ image can be described

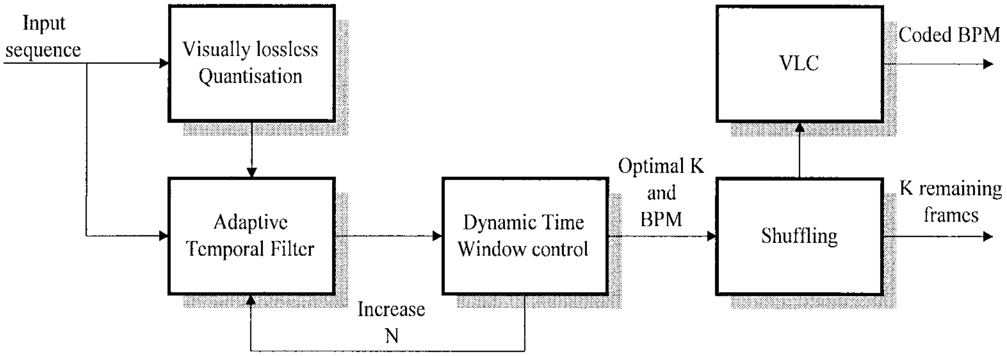


Fig. 2. Our proposed adaptive temporal decimation with dynamic time window.

by the following equation:

$$I_q(x,y,t) = \text{int}\left[\frac{I(x,y,t)}{S} + \frac{k}{2}\right] \quad \begin{array}{l} x = 0 \cdots u-1 \\ x = 0 \cdots v-1 \end{array}$$

where int$[a]$ extracts the integer portion of a, $I(x,y,t)$ is the pixel intensity, S is the corresponding quantizer step, and $I_q(x,y,t)$ is the resulting quantized intensity. The parameter k takes the value of one for quantization with rounding to the nearest integer and zero for quantization with truncation. For using a 6-bit representation, S is equal to four if the initial range of $I(x,y,t)$ is from 0 to 255.

Let us apply the above quantization process to the current time window. Then the MAD matching criterion is used to identify the discontinuous segments. Due to the fact that the dynamic range of the gray levels has been greatly reduced, a continuous segment in the block sequence has very small MAD with adjacent blocks in the temporal domain

$$\begin{aligned} \text{MAD}(m,n,t) = \sum_{y=0}^{7}\sum_{x=0}^{7} \big| I_q(m+x, n+y, t) \\ - I_q(m+x, n+y, t-1) \big| \\ t = 1 \cdots N-1 \end{aligned}$$

where m, n is the spatial location of the block

$$\text{MAD}(m,n,t) = \begin{cases} \text{continuous}, & \text{if } \text{MAD}(m,n,t) \le \textit{thr} \\ \text{discontinuous}, & \text{otherwise.} \end{cases}$$

The constant *thr* is a predefined threshold which is equal to one for the representation using 6 bits. All temporal activities are preserved by the corresponding segments. Blocks classified as continuous are low activity blocks, while blocks classified as discontinuous are high activity blocks. The topology of the temporal segments is kept in the BPM.

IV. DYNAMIC TIME WINDOW APPROACH FOR ADAPTIVE TEMPORAL DECIMATION

The MPEG standard [6] and the CCITT H.261 [7] are both based on block coding with the use of the DCT. Hence, it is natural that the design of the temporal decimation algorithm should be on block base in order to be compatible with these video coding standards. The adaptive temporal decimation algorithm makes use of the spatiotemporal information within the time window, which should retain all high activity blocks and remove all low activity blocks. The conventional algorithm fails to fulfill this mission. Let us elaborate our adaptive temporal decimation system, which is depicted in Fig. 2. It first extracts a number of frames from the original sequence and forms the first time window. Then the time window is spatially subdivided into block sequences. After the visually lossless quantization, all high activity and low activity blocks are identified in the block sequences. Every block sequence is then independently filtered with an adaptive temporal filter. The filtering process is to manipulate or to smooth the temporal signals into arbitrary piecewise constant functions. The temporal filter is a mapping from an original time evolution $I(x,y,t)$ to a piecewise constant function $g(x,y,t)$. It can be any linear or nonlinear decimation filter [2], [8]. The error between $I(x,y,t)$ and $g(x,y,t)$ could have already been minimized by the visually lossless quantization procedure. Each segment is represented by 8 × 8 image blocks that hold the temporal averages within the given time segment for each spatial location. The dynamic time window control in Fig. 2 is to increase N in order to achieve a desirable number of high activity blocks in the time window. All high activity blocks are then packed into K remaining frames by the shuffling process. The locations of removed blocks are recorded in the BPM. Eventually, the remaining frames are delivered to the standard coder such as the H.261, whereas the BPM is coded by VLC.

Fig. 3 illustrates the temporal interpolation process. It initially recovers the BPM and the remaining frames, then restores the resulting time window.

The dynamic time window approach can retain all identified high activity blocks and remove all identified low activity blocks within the time window since the number of discontinuity planes is not fixed similar to the situation of the Olstad's algorithm.

The block sequences in Fig. 4 are typical examples for head-and-shoulder image sequences. We use them to demonstrate the basic principle of our approach. Fig. 4 shows the comparison of the discontinuity segments assignment of the proposed algorithm and the conventional adaptive algorithm [4]. Three consecutive block sequences in a time window are used in this example. In addition, their corresponding BPM's are also shown on the right-hand side. For this example, there are eight blocks (N) in a block sequence. The first block (Intra block) is always used as a reference block. All high activity and low activity blocks are marked as shown in Fig. 4(a) and (b), and this identification has been made by using our visually

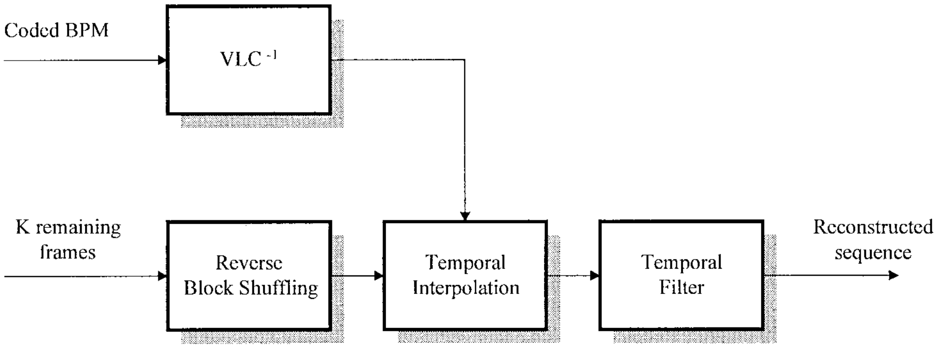


Fig. 3. Block diagram of temporal interpolation.

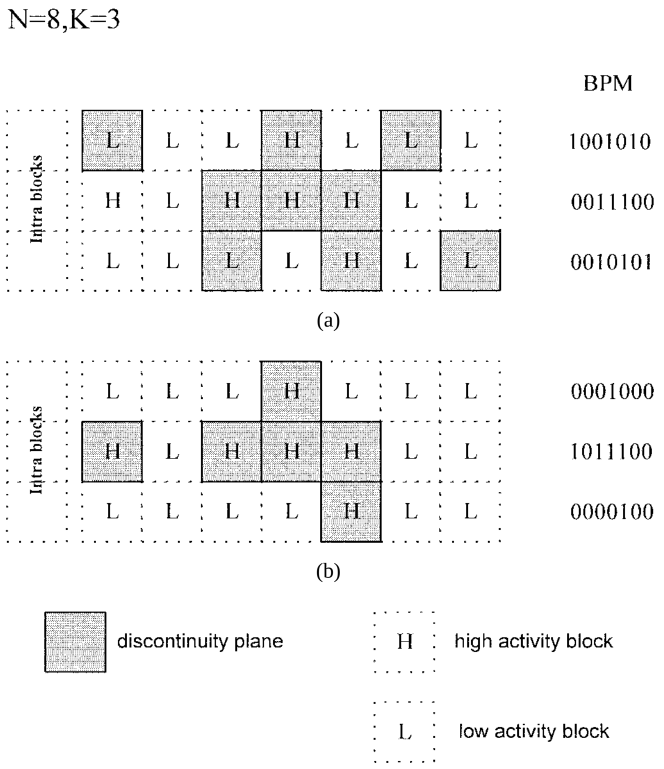


Fig. 4. The assignment of the discontinuous segments. (a) Olstad's algorithm and (b) dynamic time window.

lossless quantization. Three higher activity blocks $(K = 3)$ are to be identified from a block sequence in the conventional approach, while there is no constraint on the number of high activity blocks in a block sequence for our approach. Therefore, the total number of discontinuity or coded blocks in a block sequence could vary differently for different image sequences in our dynamic approach. From the coding point of view, the total numbers of coded blocks (which is 9 out of 21 as shown in Fig. 4(a) for the conventional adaptive decimation and is 6 out of 21 in Fig. 4(b) for our dynamic time window approach) are significant to very low bit-rate applications. We cannot predict the number of high activity blocks, but the number can be controlled by the threshold value of the visually lossless quantization. Our arrangement of block positions improves the VLC coding efficiency for BPM's and it requires a smaller number of bits according to the information theory. If high activity blocks are distributed equally all over the block sequence, the two approaches are not much different. However, practically, motion tends to be around the center of the picture; for instance this is the case of the head-and-shoulders type of images [9]. The proposed algorithm only needs to code one Intra frame and all changes within the block sequence can be handled by using the motion-compensation prediction errors [6], [7]. As a result, a faithful reconstruction can be achieved. Finally, all remaining frames are passed to the video coding stage with various coding techniques and standards, such as the H.261 and the MPEG.

The major advantage of our dynamic time window approach is that all the identified high activity blocks are included and delivered. As a result, the reconstructed sequence must be superior to the ones that were obtained by the conventional approach. Our adaptive temporal decimation system employs a simplified temporal discontinuity detection which could also speed up the processing time. The compression ratio for coding BPM's has been significantly improved due to the arrangement of the dynamic time window approach.

V. SHUFFLING MECHANISM FOR DYNAMIC TIME WINDOW APPROACH

Having illustrated our adaptive temporal decimation system in the previous section, let us introduce our approach giving an additional shuffling mechanism for the dynamic time window approach. After the quantization process, all high activity blocks can be extracted. The numbers of discontinuity planes are different for different block sequences over the entire time window. By using the conventional algorithm, several high activity blocks are ignored since its temporal discontinuity detection can only identify K (the total number of remaining frames) most different (discontinuity) planes and ignore other planes in the block sequence (see Fig. 4). This may result in serious aliasing artifacts at the reconstruction if the sequence contains a lot of motions. On the other hand, the conventional system may include some very low activity blocks in the remaining frames which should be abandoned ideally. Keeping all those high activity blocks may require N remaining frames for the worst case, i.e., $K = N$, the total number of frames in the time window. This is against the principle of the decimation.

Therefore we propose a dynamic time window approach to resolve these drawbacks of the conventional algorithm. Our dynamic approach basically identifies all high activity blocks by the visually lossless quantization and then packs them as remaining frames intelligently. As a result, a superior

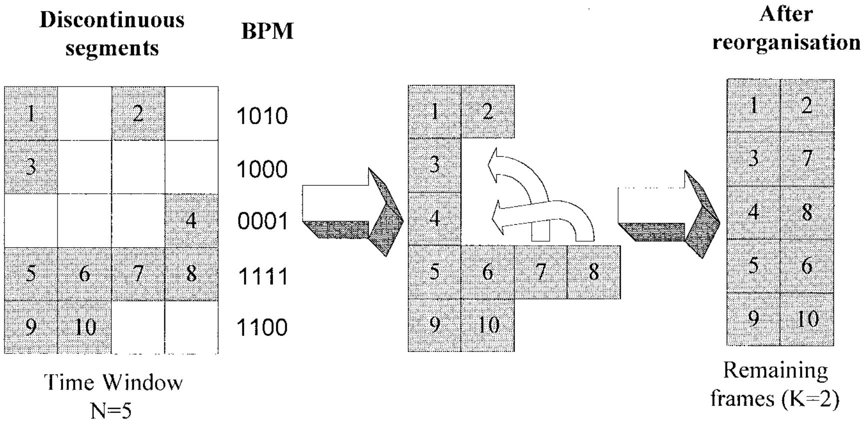


Fig. 5. A simple shuffling mechanism for additional discontinuity planes.

perceptual quality of the reconstructed sequence can always be maintained, especially for motion sequences. A feedback loop can be used to increase the size of the current time window (N) in order to fill up all possible spaces in remaining frames.

In order to perform our dynamic approach, a mechanism is needed to shuffle those discontinuity planes in such a way that the correlation between frames remains reasonably high and a reverse process exists during the reconstruction. This is a necessary process in order to maintain the compatibility to the H.261 and the MPEG standards. Before the shuffling process actually starts, the required number of remaining frames K has to be obtained by counting the total number of discontinuous segments or discontinuity planes within the time window. A simple and straightforward approach is to move all high activity blocks or discontinuity planes to the left-most positions and then to reorganize all additional discontinuity planes to fill up nonoccupied spaces in a top-down scanning order. The graphical representation is shown on Fig. 5.

Note that K is a variable since the total number of discontinuous planes is video dependent. The percentage of the occupation of remaining frames cannot be determined in advance. In order to keep a higher percentage of the occupation, the size of the time window N is arranged to be dynamic. An increase in N can yield a larger number of high activity blocks to fill up nonunoccupied spaces. When the size of the time window is optimized and all additional discontinuity planes are accommodated in K remaining frames, it may result in the situation of not having high correlation between remaining frames.

Due to the simplification of the complexity of the temporal discontinuity detection, this additional shuffling process has not increased much the processing time of the overall decimation; furthermore, for some cases such as dealing with larger time windows, our proposed algorithm runs much faster.

For the receiving end, an additional reverse process is required to reorganize the discontinuity planes into a proper order. It is the reverse operation as depicted in Fig. 5. The overheads of this process are very little since the reverse operation can be included in the temporal interpolation stage.

VI. Experimental Results

We consider that the Olstad's adaptive temporal decimation is the closest as compared to our algorithm. In addition, the most trivial decimation technique by dropping frames periodically is another candidate for our performance comparison. A number of motion sequences have been used for the comparison. Let us use 30 frames of the image sequence "Miss America" of common intermediate format (CIF) to illustrate the results, which are found to be typical.

TABLE I
Performance Comparison of Various Temporal Decimation Algorithms

	Periodically dropping frames	Olstad's algorithm	The proposed algorithm
High activity preservation	Poor	Good	Very good
Usability for all sequences with good quality	No	No	Yes
Speed in decimation and Interpolation	Very fast	Fast	relatively slower
Memory requirement	Very little	More	Most
Perceptual image quality (for motion sequences)	Poor	Better	Very good

The performance comparison between the three temporal decimation algorithms is summarized in Table I. The proposed algorithm has been found to be able to process video sequences of different natures. For any image sequences, it can gradually increase the number of remaining frames, K, but Olstad's algorithm has to be tuned manually. Although the memory requirement for our proposed algorithm appears to be higher due to the storage requirement of the dynamic time window mechanism, and a longer time is required for decimation and interpolation, it gives much better perceptual quality and uses.

The second experiment is to evaluate the efficiency of coding the BPM's for Olstad's algorithm and our algorithm. The threshold value for our visually lossless quantization is one, and all BPM's were coded by the arithmetic coder [10], [11] in our experiments. Table II shows that 18 frames of the "Miss America" sequence have been used. They are equally divided into three groups and are fed to the two adaptive temporal decimation algorithms. The proposed algorithm greatly reduces the total number of bits for coding the BPM's. Moreover, the numbers of remaining frames K are also smaller, in the ratio of one to three for all three time windows

TABLE II
EFFICIENCY OF CODING BPM

	Miss America Sequence (N = 6)	Coded BPM (bits)	Number of discontinuity planes
Proposed algorithm	Frame 1 - 6 (K = 1)	2448	1269
Olstad's algorithm	(K = 3)	7728	1320
The proposed algorithm	Frame 7 - 12 (K = 1)	2072	1004
Olstad's algorithm	(K = 3)	7744	1320
The proposed algorithm	Frame 13 - 18 (K = 1)	2608	1181
Olstad's algorithm	(K = 3)	7760	1320

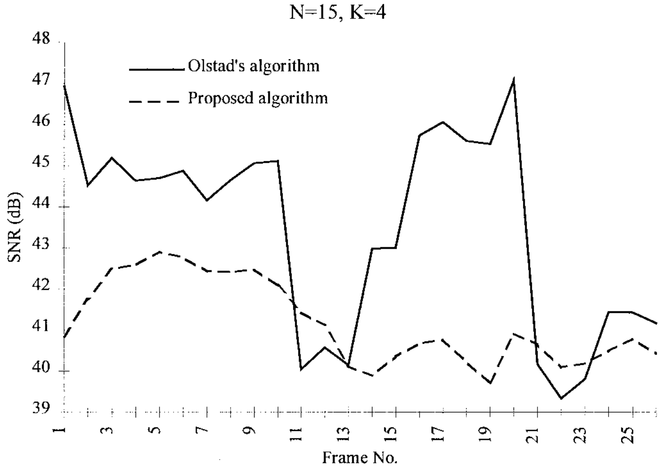


Fig. 6. The SNR of the proposed algorithm and Olstad's algorithm.

over the 18-frame sequence. The number of remaining frames K varies adaptively according to the temporal activity of the sequence. The size of time windows N has not been fixed to six, but it happened to be this value for all three time windows by chance in the simulation.

From Table II, we also see that our algorithm needs to handle a smaller number of discontinuity planes at the video coding stage (MPEG or H.261), while the number of discontinuity planes is fixed to 1320 for Olstad's algorithm with the current settings of N and K. Our approach could be beneficial to Inter frame coding.

Let us evaluate the perceptual quality of the reconstructed sequence using our proposed algorithm and Olstad's algorithm. Fig. 6 shows the corresponding SNR of the reconstructed sequences for both algorithms when there is no video codec in between the decimation and the interpolation. We try to demonstrate that the reduction in the number of remaining frames means a degradation for an objective measurement. In fact, there could not be much visual difference between them since our HVS can usually perceive very good quality images when the SNR is above 40 dB. The lowest SNR for the reconstructed sequence using our approach is 39.713 dB at frame 19 while an SNR of 45.536 dB is given by Olstad's algorithm. The actual reconstructed images of frame 19 are given in Fig. 7(b) and (c) while Fig. 7(a) illustrates the original image of frame 19. It appears that there is no visual difference among the original, our picture, and Olstad's picture, even though there is 6-dB objective difference between Figs. 7(b) and (c).


(a)

(b)

(c)

Fig. 7. Comparison of actual nineteenth frames.(a) Original nineteenth frame, (b) produced by Olstad's algorithm, and (c) produced by our proposed algorithm.

The objective degradation is due to the fact that a smaller number of discontinuity planes as well as remaining frames were coded in our dynamic approach. The perceptual quality of images produced by our approach is better, however it is not easy to make a demonstration since the way to objectively measure is to compare pixel by pixel. It is a matter of fact that the visual quality of the resulting sequence generated by our dynamic time window approach can always be guaranteed, it always meets the HVS criterion when the 6-bit representation and the threshold value $thr = 1$ are maintained at the visually lossless quantization stage. Olstad's algorithm is not able to deliver a good quality image when motions occur in a particular point of the time window; for instance, this is typical for head-and-shoulder sequences that motions tend to appear in the center since some high activity blocks at the center are removed. However, our approach can retain all these high activity blocks.

VII. CONCLUSION

A dynamic time window is suggested for the adaptive temporal decimation algorithm in this paper. This can be regarded as a refined version of Olstad's adaptive temporal decimation algorithm. Although the objective measures of both our proposed algorithm and the Olstad's algorithm are comparable, our approach can greatly improve the overall compression as compared with Olstad's algorithm, due to the reduction in the number of remaining frames (the number of discontinuity planes) and the reduction in coding the BPM. In fact, our dynamic approach can guarantee an adequate visual quality for reconstructed sequences when the 6-bit image representation and a threshold value of $thr = 1$ are maintained at the visually lossless quantization stage. Experimental results have also proven that our dynamic time window approach outperforms the conventional adaptive decimation algorithm in many ways; such as the usability for different sequences, the ability to preserve high activity, and so on. We also introduce a visually lossless quantization, which has a simple structure, to replace the temporal discontinuity detection used in Olstad's algorithm. However, an interesting problem of our approach is that a more intelligent scheme is desirable at the shuffling stage, while at the present moment it is not intelligent enough to allocate additional discontinuity planes to sensible positions. Without such a scheme, a lower correlation between frames might be produced. However, the design of a better shuffling method is an open question.

The proposed algorithm could also be improved further by a feedback control to the quantization step b. This compensation mechanism is designed to have a further adjustment on the number of remaining frames K in order to meet the requirement of the bit rate. This is a fruitful direction of further investigation.

REFERENCES

[1] Y. L. Chan and W. C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 113–118, Feb. 1996.

[2] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, Apr. 1993.

[3] K. R. Rao and P. Yip, *Discrete Cosine Transform—Algorithms, Advantages and Applications*. New York: Academic, 1990.

[4] B. Olstad, "Adaptive temporal decimation for video compression algorithms," *J. Electron. Imaging*, vol. 2, no. 1, pp. 5–18, Jan. 1993.

[5] R. J. Schalkoff, *Digital Image Processing and Computer Vision*. New York: Wiley, pp. 156–158, 1989.

[6] C. T. Chen, "Video compression: Standards and applications," *J. Visual Commun. Image Representation*, vol. 4, no. 2, pp. 103–111, 1993.

[7] "Draft revision of recommendation H.261: Video codec for audiovisual services at $p \times 64$ kbit/s," *CCITT Doc. COM XV-R 17-E* (also in *Signal Processing: Image Commun.*, vol. 2, no. 2, pp. 221–239, Aug. 1990.)

[8] P. Saint-Marc, J. Chen, and G. Medioni, "Adaptive smooting: A general tool for early vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 514–529, June 1991.

[9] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," in *Proc. IEEE*, vol. 83, pp. 705–740, May 1995.

[10] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," in *Proc. IEEE*, vol. 82, pp. 857–865, June 1994.

[11] M. R. Nelson, "Arithmetic coding and statistical modeling," *Dr. Dobb's J.*, pp. 16–29, Feb. 1991.

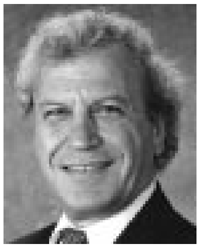
Sai-Ho Kwok (S'95–M'96) holds the B.Eng. (Hons.) degree in electronic and communications engineering (1992) from the University of North London, U.K., and the Ph.D. degree in digital image processing (1997) from the Imperial College of Science, Technology and Medicine, University of London.

He is currently an Assistant Professor of Information and Systems Management at the Hong Kong University of Science and Technology (HKUST). He was a visiting scholar at the Hong Kong Polytechnic University (1994–1995). While pursuing the Ph.D. degree, he taught MIS at HKUST. He has published in IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Graphical Models and Image Processing*, as well as in numerous refereed conference proceedings such as ICNNSP and IEEE SICSPCS. His research interests include image segmentation, segmented coding, image database system, and object oriented database.

Wan-Chi Siu (S'77–M'77–SM'90) received the Associateship (AP(HK)) in electronic engineering from the Hong Kong Polytechnic University (formerly called Hong Kong Polytechnic), the M.Phil. degree in electronics from the Chinese University of Hong Kong, and the Ph.D. degree in signal processing from the Imperial College of Science, Technology and Medicine, London, in 1975, 1977, and 1984, respectively.

Between 1975 and 1980 he was with the Chinese University of Hong Kong where he was an Electronic Engineer before he left the Department of Electronics. He then joined the Hong Kong Polytechnic University in 1980 as a Lecturer and became a Senior Lecturer in 1985, a Principal Lecturer in 1987, and a Reader in 1989. He subsequently became Chair Professor and Associate Dean of Engineering Faculty in 1992, and has been Chair Professor and Head of Department of Electronic Engineering of the same university since 1994. He has published over 160 research papers. His research interests include digital signal processing, fast computational algorithms, transforms, high-performance signal processors, video coding, computational aspects of image processing and pattern recognition, and neural networks.

Dr. Siu is now an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS and Member of the Editorial Board of the *Journal of VLSI Signal Processing Systems* for Signal, Image and Video Technology. He was the General Chairman of the International Symposium on Neural Networks, Image and Speech Processing (ISSIPNN'94) which was held in Hong Kong, April 1994, while he is also a Co-Chair of the Technical Program Committee of the IEEE International Symposium on Circuits and Systems (ISCAS'97) to be held in Hong Kong, June 1997. Between 1991 and 1995, he was a member of the Physical Sciences and Engineering Panel of the Research Grants Council (RGC), Hong Kong Government, and in 1994 he chaired the first Engineering and Information Technology Panel to assess the research quality of 19 Cost Centers (departments) from all universities in Hong Kong. He is a Chartered Engineer and a Fellow of the IEE. He has also been listed in Marquis Who's Who in the World.



Anthony G. Constantinides (SM'78) is a Professor of Signal Processing and the head of the Signal Processing and Digital Systems Section of the Department of Electrical Engineering at Imperial College, London. He has been actively involved with research in various aspects of digital filter design, digital signal processing, and communications for a number of years. His research spans a wide range of digital signal processing both from the theoretical as well as the practical points of view. His recent work has been directed toward the demanding signal processing problems arising from the area of telecommunications. He has published a range of books and papers in learned journals in the area of digital signal processing and its applications.

In 1985, Dr. Constantinides was awarded the Honor of Chevalier, Palmes Academiques, by the French Government and in 1996, he was promoted to Officer, Palmes Academiques. He holds several visiting professorships and other fellowships and honors around the world. He presently is serving as a member of the Signal Processing Society Technical Committee on Neural Networks for Signal Processing of the IEEE Signal Processing Society. He served as the first president of the European Association for Signal Processing (EURASIP) and has contributed in this capacity to the establishment of the *European Journal for Signal Processing*. He has been on, and is currently serving as, a member of many technical program committees of IEEE international conferences. He organized the first ever international series of meetings on digital signal processing in London in 1967 and in Florence (with Vito Cappellini) since 1972.