The Study of Video Transcoding and Streaming System Based on Prediction Period

Seongho Park, Sungmin Kim and Hwasei Lee, *Member, KIMICS*

Abstract—Video transcoding is a technique used to convert a compressed input video stream with an arbitrary format, size, and bitrate into a different attribute video stream different attributes to provide a efficient video streaming service for the customers is dispersed in the heterogeneous networks. Specifically, frames deletion occur in a transcoding scheme that exploits the adjustment of frame rate, and at this time, the loss in temporal relation among frames due to frame deletion is compensated for the prediction of motion estimation by reusing motion vectors in the would-be deleted frames. But the processing time for transcoding don't have an improvement as much as our expectation because transcoding is done only within the transcoder.

So in this paper, we propose a new transcoding algorithm based on prediction period to improve transcoding-related processing time. For this, we also modify the existing encoder so as to adjust dynamically frame rate based on the prediction period and deletion period of frames.

To check how the proposed algorithm works nicely, we implement a video streaming system with the new transcoder and encoder to which it is applied. The result of the performance test shows that the streaming system with proposed algorithm improve 60% above in processing time and also PSNR have a good performance while the quality of pictures is preserved.

Index Terms— Transcoding, Streaming, Prediction period

I. INTRODUCTION

nsmission bandwidth that video compression is inevitable for efficient bandwidth management on both wired and wireless network environments. For example, an uncompressed video clip that plays 1 hour with 1024*768 sizes, 16 bit colors and 30 frames/sec requires 170 GB storage space and 48 MB/s transmission bandwidth. Considering the current network environment, it is almost impossible to service such video clips on the Internet in streaming mode. So, usually, video objects are compressed by using MPEG or H.26X standards for transmission. In these compression techniques, each frame is converted using DCT and then compressed based on quantization and variable length encoding. Subsequently, some redundancy is removed from temporally consecutive frames using motion estimation techniques[1,2,3]. However, with DCT, the quality of a picture gets worse as the compression rate gets higher that the motion estimation technique requires much computation overhead leading to high time complexity.

Moreover, interoperability among networks are getting more and more important, considering the diversity on the kinds of network facilities and video formats. Video transcoding is a technique with which a video is serviced effectively to a wide variety of clients, each of whom needs his own video stream suited to his own network environment[4,5,6]. A video transcoder is composed of a decoder and an encoder like fig. 1

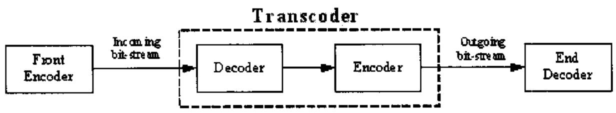


Fig. 1 The Structure of Video Transcoder

Specifically, to improve the quality of pictures in a frame-rate adjustment video transcoder, bit rate is adjusted by removing some frames of the original stream. In this transcoder, one of the most important issues is how to reconstruct the relation of prediction among temporally related frames. For this, in most of frame-rate adjustment video transcoding algorithms, motion estimation is done by reusing the motion vectors of removed frames. However, most algorithms are implemented only within the transcoder that their processing time is bound to the performance of the transcoder.

So, in this paper, we propose the algorithm that exploits the prediction period to reduce transcoding-related processing time. Furthermore, the proposed algorithm can be used to make a more lightweight streaming system in which the transcoder controls frame rate dynamically provided that the structure of a video stream is reconstructed at the server side.

This paper is as follows. In section 2, we introduce the related research about transcoding and transcoder and in

Manuscript received October 25, 2007.
Seongho Park is with the Information Technology Center, Pusan National University, Pusan, 609-735, Korea (Tel: +82-51-510-2785, Fax: +82-51-510-4692, Email: shpark@pusan.ac.kr)
Sungmin Kim is with Department of computer Engineering, Pusan National University, Pusan, 609-735, Korea
Hwasei Lee is with Department of Design, Pusan National University, Pusan, 609-735, Korea

section 3, we suggest the proposed algorithm: prediction period based transcoding algorithm. And in section4, we compare the proposed algorithm with the previous one and evaluate the performance of proposed video streaming system. Finally, we show a conclusion and refer to further research.

II. RELATED RESEARCH

Most studies in transcoding are about the structure of transcoder or the motion prediction/estimation algorithms. This is the reason which the computation time and the quality of a picture depends on how the transcoder is constructed and the motion estimation algorithm is designned.

A. Transcoder-related research

In transcoding, the purpose of reduction of bit rate is to improve the computation time and get the high quality of pictures. CPDT(Cascaded Pixel-Domain Transcoder) [8] is one of the most widely known techniques that with this algorithm, the input stream is decoded entirely and then re-encoded with a new bit rate. This technique guarantees the high quality of pictures whereas it requires high computation time. Therefore it is hard to be applied in the real time streaming service.

DCT-Domain Transcoder is proposed to reduce computation time for motion vector estimation that it shows a simpler structure and shorter processing time than CPDT. However, CPTD has not motion estimation process, therefore, the inconsistency between the original decoded streams and the partially regenerated encoded streams is brought and it bring error propagation.

Recently, Dynamic Frame Skipping Transcoding[9] is proposed to improve such computation time while admitting the quantization error to the extent of human visual system.

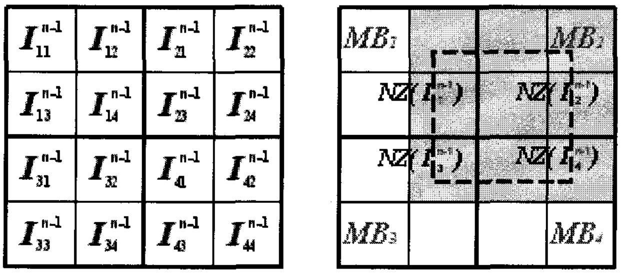
B. Motion estimation algorithm

In transcoding using Bilinear Interpolation, all motion vectors to be removed frames are stored to reduce computation time, and then motion vectors are interpolated in their reverse order. However, This algorithm requires memory to store motion vectors to be removed frames and brings about incorrectness due to composed motion vectors not considering the area occupied by four macro blocks.

With FDVS(Forward Dominant Vector Selection), the motion vectors of macro blocks occupying the widest area are selected if the currently referenced macro block exists over the macro blocks of the previous frames.

FDVS can be used easily in which case one or a group of consecutive frames are eliminated but it can not exploits the temporal relationship among frames because it simply reuses the selected motion vectors without motion estimation process. Moreover, it is difficult to select the motion vectors if the current macro block is overlapped over more than two macro blocks and each overlapped area in the blocks is the same in its size.

ADVS(Activity Dominant Vector Selection) shows us how to compute the number of non-zero DCT coefficient for action status information. See Fig. 2 to show the difference between FDVS and ADVS. The macro block MB1, which is composed of two blocks, is overlapped more narrowly than MB2, which is composed of four blocks. With ADVS, the motion vector of MB1 is selected if NZ(MB1) > NZ(MB2)(NZ means action status information and MB1 macro block 1).



$$NZ(I_1^{n-1}) = NZ(I_{12}^{n-1}) + NZ(I_{14}^{n-1})$$

$$NZ(I_2^{n-1}) = NZ(I_{21}^{n-1}) + NZ(I_{22}^{n-1}) + NZ(I_{23}^{n-1}) + NZ(I_{24}^{n-1})$$

$$NZ(I_3^{n-1}) = NZ(I_{32}^{n-1})$$

$$NZ(I_4^{n-1}) = NZ(I_{41}^{n-1}) + NZ(I_{42}^{n-1})$$

NZ(.) : number of nonzero quantized DCT coeffs

Fig. 2 Motion vector reconstruction with ADVS

III. UNITS PREDICTION PERIOD BASED TRANSCODING AND STREAMING

A. Prediction period based decoder

Most of transcoding algorithms exploit motion vectors to be removed frames for motion estimation. They are implemented within the transcoder while they do not show their best performance in computation time.

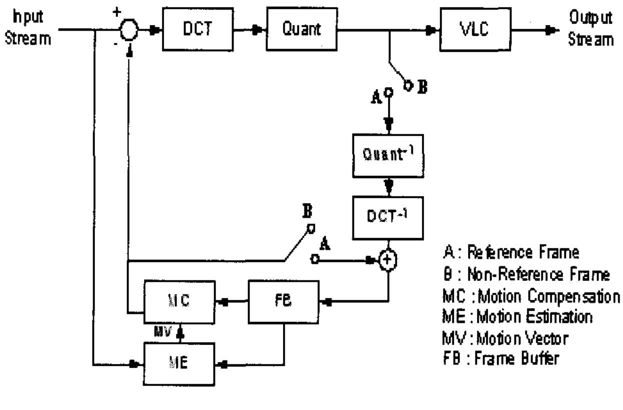


Fig. 3 Encoder in the server

So we propose a new transcoding-related algorithm that is implemented in the server. It exploits prediction period to improve the processing time to be removed frames and "prediction period" means the time interval of referenced frames that if it is applied to a video stream, the time to check the removed frames can be reduced because there are unreferenced frames among the succeeding ones.

Fig. 3 shows how the encoder in server is modified in the proposed algorithm. If a frame is referenced by a sequence of frames its switch is on A, otherwise its switch is on B. For proposed algorithm, the structure of a video stream has to be changed as well. The value of a prediction period can be changed to control frame rate dynamically. And Fig. 4 shows the structure of a frame when the prediction period is 2. Generally, The longer the interval between referenced frames, the bigger the estimation error of referenced frames. Considering this respect, we restricted that the value of the prediction period had one of the following values - 2, 3 or 4.

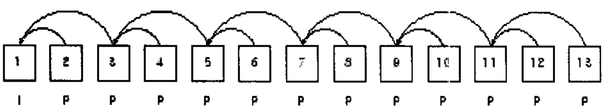


Fig. 4 The structure of a video stream with prediction period 2

B. Transcoder that controls frame rate dynamically

Dynamic Frame Skipping Transcoder[11] is transcoder to control frame rates dynamically, and each macro block has a different processing according to its decoded mode - Non-MC or MC. However, the quality of a picture could be degraded due to incorrect motion vectors because it runs FDVS for motion estimation. To overcome this, in our streaming system, we apply a motion vector composition scheme to Dynamic Frame Skipping Transcoder. So we call it MMAT(Motion Vector Mode Adaptive Transcoder).

Fig. 5 shows three examples of how frame rate is adjusted when prediction period is 2. If the video sequence has been changed through prediction period, transcoder adjusts the frame rate by applying "deletion period". A frame in which deletion period is applied will be given one of the following three attributes; NP(Non-Processing), P(Processing), NS(Non-Skipping) or S(Skipping). Here, NP is assigned to a frame which does not join transcoding processing while P is assigned to a frame which joins transcoding processing.

NS is assigned to a frame which is not deleted for transcoding whereas S is assigned to a frame which is deleted for transcoding

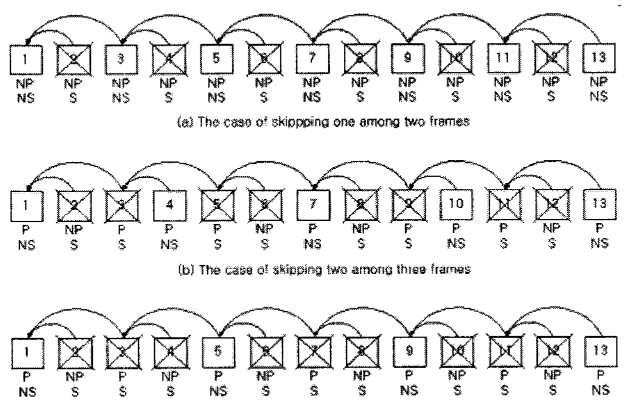


Fig. 6 deletion of frames with prediction period 2

In Fig. 6(a), we see that there is not any additional action because the would-be deleted frames are not referenced. In Fig. 6(b), though the second frame is deleted it is not referenced by any following frames. Therefore, there is no action on processing. But the third frame which will be deleted is referenced by the fourth frame that it has to join the transcoding processing. The first frame also join the transcoding process because it is referenced through motion processing and quantization processing by both the third and fourth frames(In the existing transcoding schemes, all frames have to join the transcoding process). Also the case of prediction period is either 3 or 4, the same procedure is applied.

Table 1 The comparison of frame rates with different prediction rate and deletion period

deletion period \ prediction period	2	3	4
2(deletion of 1 frame every 2 frames)	0%	33%	0%
3(deletion of 2 frame every 3 frames)	67%	0%	42%
4(deletion of 3 frame every 4 frames)	50%	50%	0%

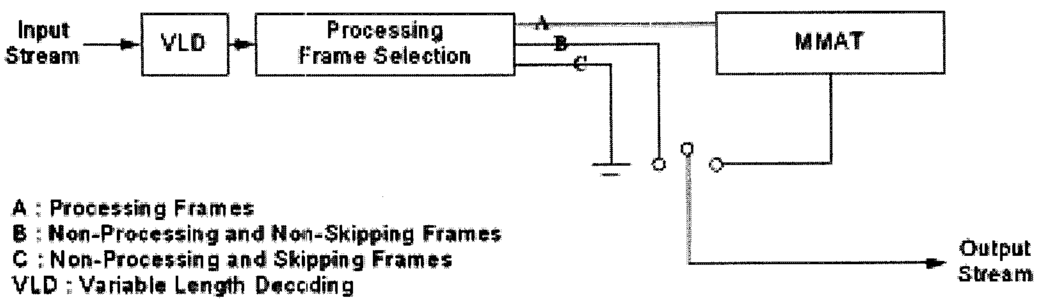


Fig. 5 Structure of transcoder

```
Procedure Processing_Frame_Selection()
Begin Procedur      // decide deletion period as frame rate
nsp = deletion period
pp = prediction period
If (pp == nsp * alpha)          // alpha = 1, 2, 3,
  Exit Procedure          // Bypass
End If
lcm = LCM(pp, nsp) // the least common multiple of pp nsp
// check the last prediction period frame's deletion
last_pp_skip=1
            // 1: deleted frame, 0: remaining frame
k = 1       // frame number
 For ( k <= lcm )
  If((k-1) % pp == 0)
    Processing Frame       // processed frame
  Else
    If(pp <nsp)      // if deletion period > prediction period
      If((k-1) % nsp == 0)
        Processing Frame       // processed frame
      End If
    Else         // deletion period < prediction period
      If(last_pp_skip == 1 && (k-1) % nsp == 0)
        Processing Frame       // processed framed
      End If
    End If
  End If
  If((k-1) % pp==0 && (k-1) % nsp==0)
    // check the last prediction period frame's deletion
    last_pp_skip = 0
    End If
    k++;            // next frame
End For
End Procedure
```

Fig. 7 Frame selection algorithm for dynamic frame rate control

Fig. 7 is the detailed algorithm about how a frame is selected or deleted for transcoding in Fig. 6.

C. The composition of motion vectors

ADVS, as FDVS, has difficulty in selecting the proper motion vectors when the current macro block is overlapped over more than precedence macro blocks and action status information of each macro block is the same.

The action status information is measured by the number of non-zero DCT coefficient of 8X8 blocks. In other words, DCT energy from some overlapped blocks of the previous macro block is regarded as if it is got from all of four blocks of the previous one. This means the remaining non-overlapped blocks may include more action status information and such can degrade the quality of pictures.

To overcome this, we propose a new motion vector composition scheme based on the new weight. The new weight is computed by considering both the size of overlapped area and action status information. It subsequentially is applied to the motion vectors of the overlapped macro blocks. And then, a new motion vector is created(The new weight stems from the overlapped area and action status information). The overlapped area is computed as the unit of pixel and The action status information is not measured by 8X8 blocks as with ADVS but is measured by macro block unit as the case of motion vectors.

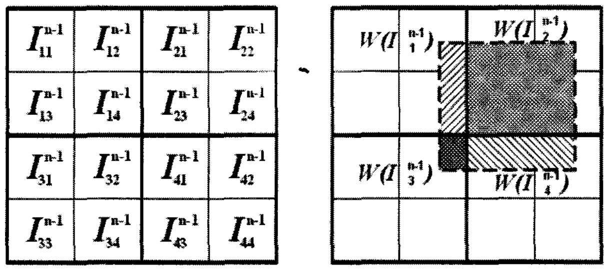


Fig.8 Computation of weight in four neighbored macro blocks

Fig. 8 shows the weight that stems from the four neighbored macro blocks. The computation of weight is as follows.

Step 1. Compute w_i, the weight from each overlapped macro block *i*.

$$W_i = (NZ(I_i)\times OA_i)/\sum_{i=1}^{4}(NZ(I_i)\times OA_i)$$

where $NZ(I_i)$: action status information, OAi : a size of overlapped area in macro block i respectively.

Step 2. Compose MV, the motion vector of current macro block

$$\text{Composed MV} = \sum_{i=1}^{4} MV_i \times W_i$$

IV. PERFORMANCE ANALYSIS

A. Decoder at clients' side

Generally, most people use media player such as Window Media Player[12], RealPlayer[14] and QuickTime[13]. But a video stream, which is encoded with our encoding algorithm in the server, may not be played properly unless it is not transcoded with our transcoding scheme. So we designed a decoder that has an additional switch, which is used to check whether a video stream is transcoded with our proposed algorithm or not(This does not need much additional cost as compared with the existing decoder).

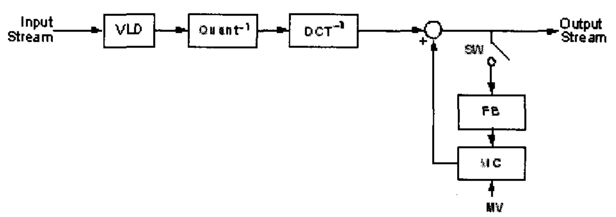


Fig. 8 Decoder at client's side

B. Experiments

To check how proposed algorithm operates, we use some performance metrics. For this, in terms of the transcoding time and the quality of transcoded output pictures, we compare between the proposed encoder-decoder system and the ones in [11]. The four video files - Suzie, Football, two music video clips - in this tests is H.263 format. Each of video files includes non-zero motion vectors of 44%, 82%, 61% and 52% respectively. The trancoder used in tests of proposed algorithm is implemented by modifying H.263 codec in FFmpeg[15]. Table 2 and 3 show detailed information in the environment of tests.

Table 2 Input Video Streams

Sequence	Quantization parameter	Video Format	Frame rate (fps)	# of frames
Football	5	QCIF	30	257
Suzie				149
Music Video-1				300
Music Video-2				300

Table 3 Test environment

system specification	CPU & Memory	Pentium IV 1.2 Ghz CPU, 256M
	OS	Redhat Linux 9.0
video type	H.263	"IPPP.."
video codec	FFmpeg	Version 0.4.8
output frame rate	15, 10, 7.5	2~4(deletion period)
output quantization parameter	5	

Fig. 10 - 13 show computation time(μs) and video quality(dB) when the prediction period of proposed transcoder is 2,3 or 4 and the deletion period is 2.

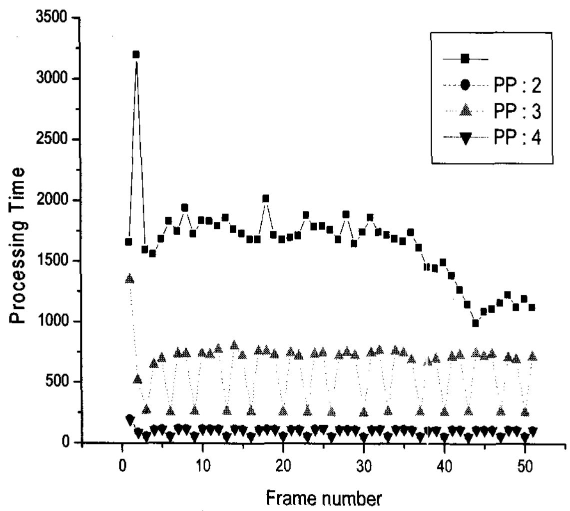


Fig. 10 Computation time per prediction period in football

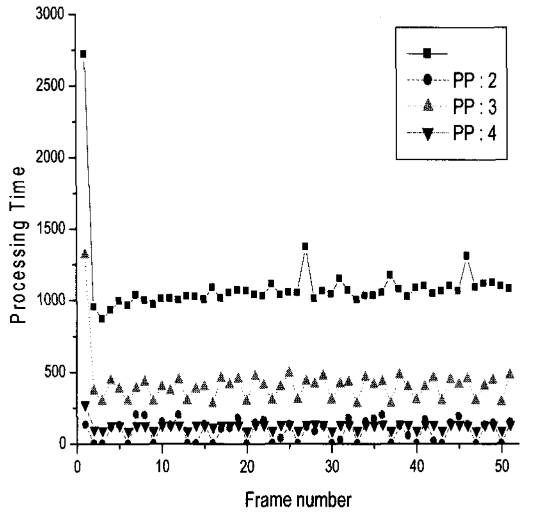


Fig. 11 Computation time per prediction period in music video 1

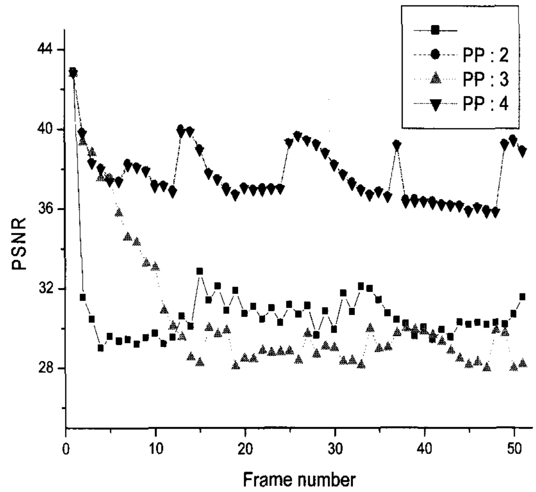


Fig. 12 Video quality per prediction period in football

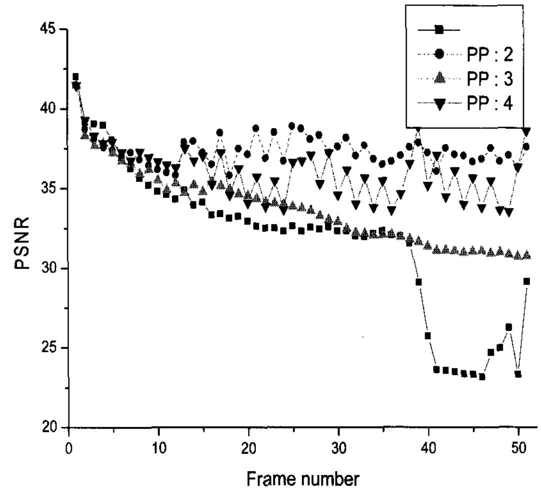


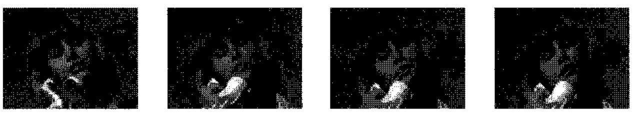
Fig. 13 Video quality per prediction period in music video 1

Table 4 shows the average PSNR(Peak Signal to Noise Rate) and the PT(processing time) in four video streams.

Table 4 Performance comparisons per prediction and deletion period

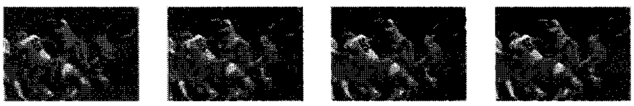
comparison item			football	suzie	MV-1	MV-2
transcoder in [11] (deletion period :2)		avg. PSNR	30.72	38.21	31.75	33.06
		PT(μs)	198985	71786	182946	182629
prediction period : 2	deletion period:2	avg. PSNR	37.88	39.66	37.30	38.56
		PT(μs)	11481	4735	16873	18155
	deletion period:3	avg. PSNR	30.18	37.23	30.96	33.28
		PT(μs)	131149	46958	122765	124872
	deletion period:4	avg. PSNR	30.01	36.53	30.17	32.22
		PT(μs)	117645	38997	92813	93447
prediction period : 3	deletion period:2	avg. PSNR	30.94	36.29	33.64	34.89
		PT(μs)	67782	25182	62735	61234
	deletion period:3	avg. PSNR	37.80	39.10	36.54	37.82
		PT(μs)	13039	4745	16848	17595
	deletion period:4	avg. PSNR	29.83	36.08	30.09	31.74
		PT(μs)	101618	37217	93637	93796
prediction period : 4	deletion period:2	avg. PSNR	37.72	39.02	36.19	37.34
		PT(μs)	11223	4958	18633	19387
	deletion period:3	avg. PSNR	29.68	34.38	30.66	32.2
		PT(μs)	83198	31198	78693	78315
	deletion period:4	avg. PSNR	37.72	39.02	36.19	37.34
		PT(μs)	11739	4435	15792	11743

Fig. 14 - 17 compare the quality of a picture per output stream in a prediction period. In Figure 12, (a) is made by using the existing transcoder[1], while each (b), (c), (d) is made using our proposed transcoder with prediction period 2,3 and 4 respectively.



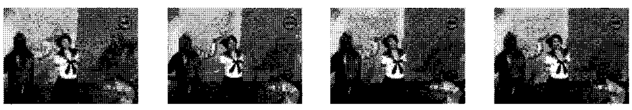
(a) (b) (c) (d)

Fig. 14 Comparison of picture quality per prediction period in Suzie(stream #72)



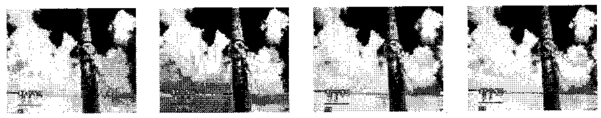
(a) (b) (c) (d)

Fig. 15 Comparison of picture quality per prediction period in Football(stream#24)



(a) (b) (c) (d)

Fig. 16 Comparison of picture quality per prediction period in Music Video 1(stream #252)



(a) (b) (c) (d)

Fig. 17 Comparison of picture quality per prediction period in Music Video 2(stream #264)

As shown in the figures, proposed algorithm shows the same or a little better PSNR than the existing one. However, in terms of the processing time, proposed algorithm needs only half processing time of the existing one.

In Table 5, we also measured average PSNR and processing time per stream to check how a decoder has effect on the overall transcoding performance. As shown in Table 5, the longer the prediction period is, the smaller PSNR processing time is. This means that the overall performance is heavily dependent on how the transcoder operates.

Table 5 Performance comparison per prediction period in the encoder

Comparison		football	suzie	Music Video-1	Music Video-2
decoder [11]	avg. PSNR	38.63	39.91	38.77	38.84
	PT(μs)	775348	594388	897886	940065
our decoder with prediction period 2	avg. PSNR	37.88	39.66	37.30	38.56
	PT(μs)	697813	534949	808097	846059
our decoder with prediction period 3	avg. PSNR	37.80	39.10	36.54	37.82
	PT(μs)	682306	523061	790140	827257
our decoder with prediction period 4	avg. PSNR	37.72	39.02	36.19	37.34
	PT(μs)	666799	511174	772182	808456

V. CONCLUSIONS AND FURTHER RESEARCH

In this paper, we proposed a video streaming system that the encoder at the server side is modified to exploit the prediction period to reduce processing time for frame deletion. We also designed a transcoder that dynamically control frame rate to reduce processing time and get better quality of pictures.

The result of performance test shows that the proposed transcoder requires only 60% of processing time compared with the existing transcoder[11]. Moreover, with proposed scheme, The quality of pictures are improved more than 6% compared with the existing one. Furthermore, the more movement a video has, the higher the quality of pictures is.

So far, the proposed scheme works without interactive mode. So we are doing research about how the proposed scheme works in interactive mode.

ACKNOWLEDGMENT

This work was supported for two years by Pusan National University Research Grant

REFERENCES

[1] Video Coding for Low Bit Rate Communication, *ITU-T Recommendation H.263*, Jan. 1998.

[2] "MPEG-2 video", *ITU-T Recommendation H.262-ISO/IEC 13818-2*, Jan. 1995.

[3] Coding of audio-visual Objects, Part 2 Visual, Amendment 4: *Streaming Video Profile, ISO/IEC 14496-2/FPDAM4*, July 2000

[4] Hong, YongTaek; Lee, KyungHo; Kim, Jingsang; Cho, Won-Kyung, "High Speed Architecture for MPEG-2/H.264 Video Transcoding," *Communications and Information Technologies, 2006. ISCIT 06. International Symposium* pp. 674 - 678, Oct. 2006.

[5] N. Bjork and C. Christopoulos, "Video Transcoding for Universal multimedia Access," *ACM Workshop on Multimedia Standards, Interoperability and Practice (MM2000 Workshop), Proceedings of ACM Multimedia 2000*, pp. 75-79, November 4, 2000.

[6] Henig, A. Raz, D., "Efficient management of transcoding and multicasting multimedia streams," *Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International Symposium*, pp. 425 - 438, May 2005.

[7] M. J. Chen, M. C. Chu and C. W. Pan, "Efficient motion estimation algorithm for reduced frame-rate video transcoder," *IEEE Trans. Circuits syst. Video Technol.*, vol. 12, pp. 269-275, Apr. 2002.

[8] J. Youn, M. T. Sun and C. W. Lin, "Motion vector refinement for high performance transcoding," *IEEE Trans. Multimedia*, vol. 1, pp. 30-40, Mar. 1999.

[9] A. Vetro, C. Christopoulos, Sun Huifang, "Video transcoding architectures and techniques: an overview," *Signal Processing Magazine, IEEE*, vol 20, Issue 2, pp. 18 - 29 March 2003.

[10] A. Vetro and C. Christopoulos and H. Sun, "Video Transcoding Architectures and Techniques: An Overview," *IEEE signal Processing Magazine, ISSN: 1053-5888*, Vol. 20, Issue 2, pp. 18-29, March 2003.

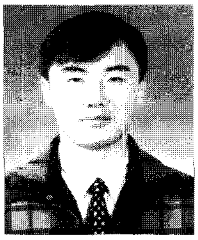
[11] Kai-Tat Fung Yui-Lam Chan, Wan-Chi Siu, "New architecture for dynamic frame-skipping transcoder," *Image Processing, IEEE Transactions* on Volume 11, Issue 8, pp. 886 - 900, Aug. 2002

[12] Microsoft Windows Media, Microsoft Corporation Inc. [Online]. Available: http://www.microsoft.com/windows/windowsmedia/

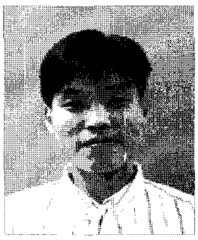
[13] Apple QuickTime Player, Apple Corporation Inc. [Online]. Available: http://www.apple.com/quictime/

[14] Real Networks RealPlayer [Online]. Available: http://www.real.com/

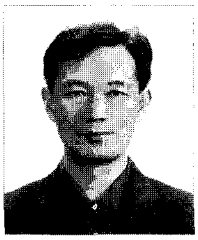
[15] FFmpeg multimedia system. [Online]. Available: http://www.ffmpeg.org



Seongho Park
He received the B.S, M.S. and Ph.D. degrees in Computer Science Pusan National University, in 1996, 1998 and 2002, respectively. Since 2002 he has been a faculty member of Pusan National University, where he is now associate professor in Information Technology Center. His research interests are in the areas of video transcoding, multimedia communication, the system of multimedia.



Sungmin Kim
He received the B.S. degree in computer science in 2001, and M.S. degree in 2003 from Pusan National University, Pusan, Korea. He is currently pursuing the Ph.D. degree in computer engineering at Pusan National University. His research topics include transcoding, video communication, and video compression.



Hwasei Lee
He received B.S. degree from Pusan National University in 1985 and M.S. degree from Pusan National University in 1987, and Ph.D. degree from Pusan National University, at Busan, Korea, in 2004. From 1991 to 2005, he was a professor of Miryang National University, Miryang, Korea. Since 2006, he has been a professor in the Department of Design of Pusan National University, Busan, Korea.