Error Concealment Techniques for Visual Quality Improving

화질 향상을 위한 오류 은폐 기법

서재원
충북대학교 전기전자컴퓨터공학부
Jae-Won Suh(sjwon@chungbuk.ac.kr)

요약

MPEG-2 비디오 압축열은 복잡한 부호화 알고리즘을 이용하여 압축하기 때문에 전송 오류에 매우 민감하다. 만약 패킷을 잃어버리거나 수신된 패킷에 오류가 있으면 현재 화면에 화질저하가 발생할 뿐만 아니라 화면수가 제한적이긴 하지만 뒤이어서 재생되는 화면에도 오류가 전파된다. 따라서 이런 전송오류의 영향을 막거나 최소화 하기위해서 다양한 오류 강인 부호화/복호화를 적용한다. 대표적인 오류 강인 방법이 오류 은폐 기법이다. 오류 은폐 기법은 손상된 비디오 데이터를 은폐하기 위해서 정상적으로 수신된 데이터의 공간적, 시간적 중복성을 이용한다. 손상된 데이터를 복원하기 위해 움직임 벡터를 추정하고 움직임 보상하는 것은 좋은 방법이다. 이 논문에서는 다양한 움직임 벡터 복원 방법에 기반한 오류 은폐 기법을 제안하고 일반적인 방법들과 성능을 비교한다.

■ 중심어 : | MPEG-2 | 동영상 압축 | 오류 은폐 기법 | 움직임 벡터 |

Abstract

The MPEG-2 video compressed bitstream is very sensitive to transmission errors due to the complex coding structure of the MPEG-2 video coding standard. If one packet is lost or received with errors, not only the current frame will be corrupted, but also errors will propagate to succeeding frames within a group of pictures. Therefore, we employ various error resilient coding/decoding techniques to protect and reduce the transmission error effects. Error concealment technique is one of them. Error concealment technique exploits spatial and temporal redundancies of the correctly received video data to conceal the corrupted video data. Motion vector recovery and compensation with the estimated motion vector is good approach to conceal the corrupted data. In this paper, we propose various error concealment algorithms based on motion vector recovery, and compare their performance to those of conventional error concealment methods.

■ keyword : | MPEG-2 | Video Coding | Error Concealment Technique | Motion Vector |

* This work was supported by the research grant of the Chungbuk National University in 2005.

접수번호 : #051212-001
접수일자 : 2005년 12월 12일
심사완료일 : 2006년 01월 23일
교신저자 : 서재원, e-mail : sjwon@chungbuk.ac.kr

I. Introduction

The digital television (DTV) standard describes a system designed to transmit high quality video and audio and ancillary data over a single 6 MHz channel. The need for video compression in a high definition TV (HDTV) system is apparent from the fact that the bit rate required to represent an HDTV signal in uncompressed digital form is about 1 Gbps and the bit rate that can reliably be transmitted within a standard 6 MHz TV channel is about 20 Mbps. This implies that we need to compress HDTV signals by 50:1 or greater. The MPEG-2 video coding standard [1] successfully achieves high compression ratio using a hybrid algorithm of motion compensation (MC) and discrete cosine transform(DCT).

The overall flow of the DTV signal consists of several steps to transport the compressed video data from transmitter to receiver. First of all, digitized video is compressed by a specific video compression algorithm adopted for the given application. The output of the encoder is segmented into fixed or variable length packets for easy transmission and multiplexed with other data types. The multiplexed packets are then sent to the transmission channel after channel encoding using forward error correction codes (FEC). The received transmission packets which may include transmission errors undergo channel decoding and demultiplexing to get unpacked bitstreams. Finally, the resulting bitstreams are entered to the video decoder to reconstruct the original video.

Due to the nature of broadcasting, it is nearly impossible to design a system to be totally error free. In addition, compressed bitstreams generated by the MPEG-2 video compression algorithm are very sensitive to channel disturbances. Even one bit error can degrade not only the current frame but also succeeding frames. Therefore, there has been a renewal of interest in error resilience coding techniques for real time video transmission over imperfect channels, since only channel coding and decoding could not provide the perfect solution for transmission errors [2][3]. When we use mathematically well structured FEC, we can detect and correct the transmission errors. However, because this mechanism reduces channel capacity comparatively, it is impossible to correct all transmission errors.

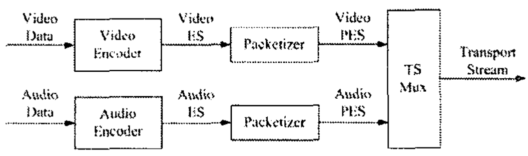
After channel decoding, we apply error concealment techniques for uncorrected errors. Error concealment techniques try to recover the corrupted data by exploiting the spatial and temporal redundancies of the video data. There are mainly two different types of error concealment techniques: spatial-domain error concealment and temporal-domain error concealment. The spatial-domain error concealment algorithms interpolate the lost area using spatially neighboring image data [4-7]. Because these algorithms consider an isolated lost macroblock (MB) by modification or simplification, they provide good subjective quality. However, if we consider the MPEG-2 video transmission for the DTV, we lose successive MBs from the erroneous MB to the beginning of the next resynchronization. Therefore, we cannot use interpolation algorithms directly and cannot expect good performance. On the other hand, temporal-domain error concealment schemes utilize previously decoded image data [8-12]. In these works, they estimate lost motion vectors (MVs) and compensate with the estimated MVs.

In this paper, we propose new MV recovery algorithms for temporal-domain error concealment. The organization of this paper is as follows. Section II presents a system configuration for test and evaluation of error concealment. In Section III, we summarize the MPEG-2 video encoding and decoding procedure and introduce error detection algorithm used in our simulation. In Section IV, we explain conventional and proposed temporal-domain error concealment schemes based on MV recovery. Section V presents simulation results and performance comparisons of different error concealment algorithms. Finally, we draw conclusions in section VI.

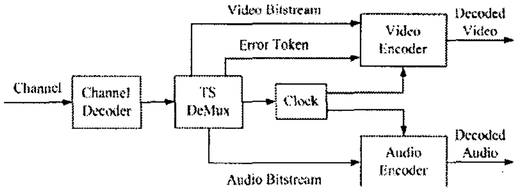
II. System Configuration

Since it is nearly impossible to design a complete DTV transmission system, the MPEG-2 system standard [13] introduces a general approach which defines transport stream (TS) packet transmission system considering noisy channels. [Fig 1] shows the basic multiplexing approach based on single video and audio elementary streams (ESs), which are outputs of video encoder and audio encoder, respectively. Each ES is packetized into packetized ES (PES) for proper transmission. After multiplexing, TS packets are transmitted over a noisy channel.

During the transmission of TS packets over a noisy channel, the TS packets can be corrupted by various noises. A channel decoder can detect and correct some parts of the transmission errors. If the channel decoder cannot detect the transmission error, the undetected TS packet error can be notified by TS syntax checking. As shown in [Fig 2], a TS packet consists of 188 bytes including 4 bytes header information.



(a) System Encoder



(b) System Decoder

Fig 1. MPEG-2 TS System for Noisy Channel

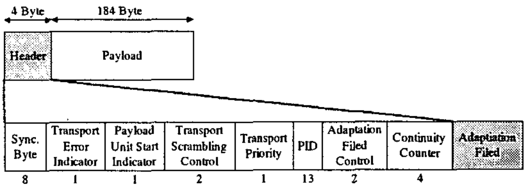


Fig 2. Transport Stream Syntax Diagram

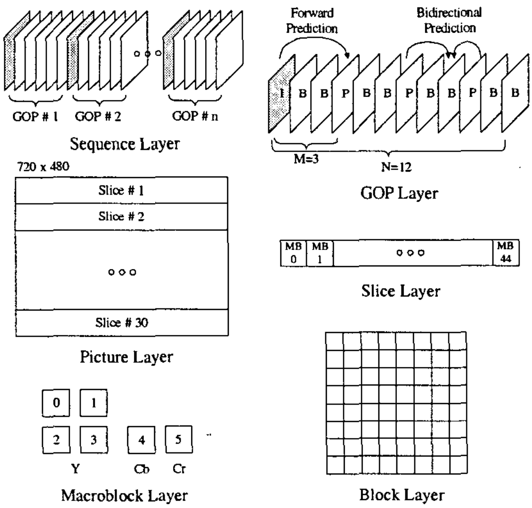


Fig 3. Hierarchical Layers of MPG-2 Video Coding

The TS packet header contains a 1 bit transport-error -indicator field, which notifies whether the received TS packet has one more uncorrectable bit errors or not. The indication of the TS packet error detected by TS DeMux to the audio and video decoder may be performed in various ways. In our system, TS DeMux sends a error token to the video decoder.

If the video decoder receives a damaged packetized bitstream with an error token, there is no way to notify how many data within the packetized bitstreams are usable. Therefore, for a practical purpose, the damaged bitstream is thrown out and treated as a lost one. As a result, all subsequently received bitstream become useless until synchronization is reestablished. According to the MPEG-2 video coding algorithm, because the smallest synchronization unit is the slice header, we lose horizontal strips of MBs from the erroneous first MB to the beginning of the next MB slice.

III. Modified MPEG-2 Video Codec

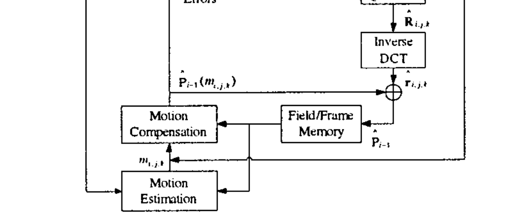
1. Hierarchical Structure of Video Sequence

A hierarchical coding structure of MPEG-2 video sequence is shown in [Fig 3]. The overall bitstream syntax is composed of several layers, and each layer performs a different logical function. The outermost layer is video sequence layer, which contains global parameters, such as the size of the video frames, the frame rate, the bit rate, and so on. A group of pictures (GOP) layer supports for random access, fast search, and editing. GOP contains an intra coded frame (I-frame), forward predictive coded frames (P-frames), and bidirectionally predictive coded frames (B-frames). I-frame is coded by itself, which is used as an anchor frame for forward and backward prediction of neighboring frames. P-frame is coded using forward motion compensated prediction from the past I-frame or P-frame. B-frame is coded using bidirectional motion compensated prediction from the past and the future I-frame and P-frame.

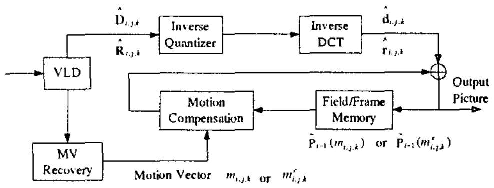
The picture layer contains information for the frame type and the display order. The picture layer is composed of slice layers, and each slice layer consists of several consecutive MB layers. Slice layer is used for resynchronization during the decoding of a frame or in the event of errors. One MB contains 16 x 16 luminance pixels and corresponding 8 x 8 two chrominance blocks, which is ME and MC unit. The block layer is 8 x 8 basic unit of DCT.

2. MPEG-2 Video Compression Algorithm

[Fig 4](a) shows a block diagram of the MPEG-2 video encoder. In order to encode I-frame, the input frame is partitioned into 8 x 8 pixel blocks. The 8 x 8 two-dimensional DCT is applied to the each block, and the DCT coefficients are quantized. Finally, the quantized DCT coefficients are entropy encoded using variable length coding (VLC). In succeeding frames, temporal redundancies are removed by prediction structure. Residual errors are difference between current MB and motion compensated MB obtained by motion estimation (ME) and MC. ME is the process of determining the best matching block in the anchor frame with the MB to be encoded of current frame. MV is the displacement between the MB to be encoded and the best matching block. Using this MV, we can reconstruct motion compensated MB which is similar to the current MB of input frame, which is known as MC. Consequently, the residual errors are encoded and transmitted like an intra frame encoding procedure. The MV describing the direction and amount of motion of the MB is transmitted as part of the bitstream.



(a) MPEG-2 Video Encoder



(b) Modified MPEG-2 Video Decoder

Fig 4. Modified MPEG-2 Video Coding

[Fig 4](b) shows a block diagram of a modified MPEG-2 video decoder. Except for the error concealment block, the configuration is the same as the MPEG-2 decoder. Decoding procedure is opposite of the encoder. If the decoder receives correct MV information and residual errors, we can obtain well reconstructed outputs. However, if the decoder receives corrupted data due to the transmission errors, it cannot know which area of the anchor frame will be used for MC using MV and cannot sum the residual errors with this motion compensated prediction to obtain the output. In order to reduce the effects of the transmission errors, we estimates lost MV in the MV Recovery block and compensate with the estimated MV.

3. Error Detection

Since error concealment operation is applied to the corrupted MBs, error concealment techniques are largely dependent on the capability of error detection. There are several ways for detecting bitstream errors at the video decoder. The video decoder can detect errors when there is no proper entry in the VLC table for a certain codeword. However, if a corrupted codeword is mapped into another valid codeword, the video decoder cannot detect the error and this error will be propagated to next codewords. Therefore, even though the error can be detected in the VLD layer, we cannot tell where errors are located in the bitstream and how many data are corrupted. To detect more exact location of the error, we need other techniques such as a backtracking algorithm [14].

In this paper, we can obtain the MB position of error occurrence by checking the MB address (MBA) which defines the absolute position of the MB. In our system, when we decode the correctively received packetized bitstreams, we continuously store the last decoded MBA. If we receive the error token from the TS DeMux, the address (recorded MBA + 1) is the beginning position of the erroneous MBs. As early described, since the smallest synchronization unit is slice header, we lose consecutive MBs until synchronization is reestablished in the starting of next slice.

IV. Error Concealment

1. Conventional Motion Vector Recovery

We are here concerned with the previous works of the MV recovery. One of the difficult problems for error concealment techniques is the limitation of available information. In order to conceal the lost MB, we can just use only upper and lower MBs of the lost MB because of considering correlation. In addition, if the corrupted area is wider than a single row of MBs, the problem becomes much more serious. A simple estimate value for the lost MV is zero with an assumption that no motion has occurred between the previous reference frame and current frame. The use of zero MV produces a reasonably good approximation in a small and slow motion area. However, we can not expect same results in the large and fast motion area.

To conceal the lost MB, motion compensated error concealment algorithms have been generated. First, we can exploit neighboring MVs of the lost MB. The MV of the lost MB can be obtained by taking the average value of MVs of the vertically adjacent MBs (AVG)

[8]. In this scheme, if vertically neighboring MBs have MVs, we can obtain reasonably good reconstruction quality for the lost MB. However, if only one or none of the vertical neighbors has a valid MV, quality of the reconstructed image is not satisfactory. To solve this problem, we propose modified average algorithms, which will be explained in the next subsection.

Other conventional error concealment methods use boundary pixels of the lost MB to estimate the lost MV. The boundary matching algorithm (BMA) [9] uses one pixel boundary line of above, below, and left of the lost MB to recover MV. In the beginning, they calculate the squared sum of differences between boundary pixels of currently being decoded frame and pixels of previous decoded frame. The BMA chooses the MV, which produces the smallest total squared sum of differences. Decoder motion vector estimation algorithm (DMVE) [12] uses several boundary pixel lines (two to eight) of the lost MB. However, these algorithms have a significant limitation which left pixels are not available for matching when video data is received with errors. Nevertheless, if left pixels are used for MV recovery, they already include error concealment mismatch.

2. Proposed Motion Vector Recovery

2.1 Modified Average Algorithm

As early described, when vertically adjacent MBs have MVs, we can get good performance. In order to satisfy that condition, we propose a modified average algorithm (MAVG). First, we define 16 x 8 target blocks (TBs) for intra MB like a [Fig 5](a) and estimate MV for the target block using block matching algorithm. Then, we can recover the lost MV by taking average of the estimated MV of the intra MB and the original MV of the inter MB.

We develop above idea a little further. In order to get more accurate MV for the lost MB, we separate 16 x 8 TB into two 8 x 8 small target blocks (STBs) as shown in [Fig 5](b). The similar MV estimation procedure applies to this scheme. Although we can get good performance, these two methods require high computation time due to the ME. To reduce the processing time, we define alternative STBs as shown in [Fig 5](c). With this modification, we can reduce a half of the computational complexity.

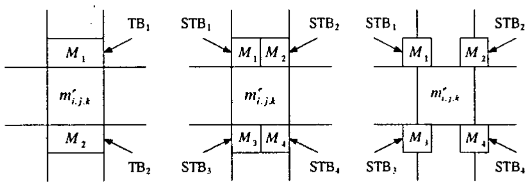


Fig 5. Modified Average Algorithm

2.2 Extension Matching Algorithm with EMA

We now introduce another newly proposed extension matching algorithm (EMA) which exploits boundary pixels of the lost MB like a BMA [11]. As shown in [Fig 6], we form an extended MB. The lost MV is determined by minimizing the sum of absolute error (SAE) between extended pixels in the current frame and the previous reference frame. In this algorithm, the extended width (EW) and the search range (SR) are very important parameters. If the EW and the SR are increased, the computational time is increased. Therefore, this EMA entails a considerable amount of processing complexity at the decoder.

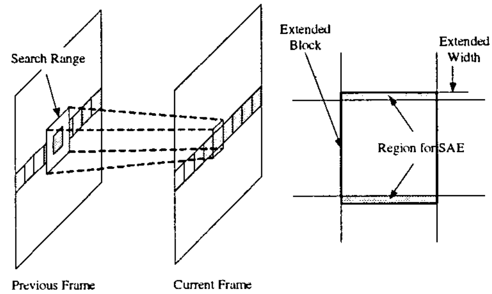


Fig 6. Extension Matching Algorithm (EMA)

In order to reduce the computational complexity of EMA, we use an initial MV as shown in [Fig 7]. First, we set the initial MV for the lost MB by AVG. The initial MV establishes a starting point of the SR, and it enables to reduce the SR. If none of the vertical neighbors has a valid MV, we use normal SR the same as that of EMA. As a result, we can effectively reduce the processing time.

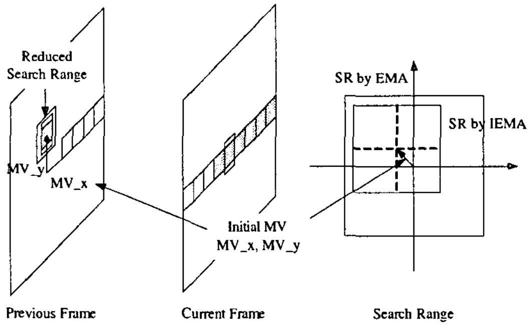


Fig 7. EMA with Initial Motion Vector (IEMA)

2.3 Optical Flow Algorithm

Since optical flow is very similar to the true motion, we propose a new motion vector recovery algorithm based on optical flow fields. Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow arises from relative motion of objects and the viewer. Consequently, optical flow can be corresponded to the motion. To find optical flow is to minimize the error in the optical flow constraint equation and the measure of departure from smoothness as Eq (1) [15].

$$e^2 = \int\int[(E_x u + E_y v + E_t)^2 + \alpha^2(u_x^2 + u_y^2 + v_x^2 + v_y^2)^2]dxdy \quad (1)$$

where u and v are the x and y components of the optical flow, E_x, E_y, and E_t means partial derivatives of image brightness with respect to x, y and t, respectively. The minimization is to be accomplished by finding suitable values for the optical flow velocity (u, v). Using the calculus of variation and the approximation of Laplacian, Eq (1) can be written as

$$(\alpha^2 + E_x^2 + E_y^2)(u - \bar{u}) = -E_x(E_x\bar{u} + E_y\bar{v} + E_t)$$
$$(\alpha^2 + E_x^2 + E_y^2)(v - \bar{v}) = -E_y(E_x\bar{u} + E_y\bar{v} + E_t) \quad (2)$$

α^2 plays a significant role only for areas where the brightness gradient is small, preventing haphazard adjustments to the estimated flow velocity occasioned by noise in the estimated derivatives. This parameter should be roughly equal to the expected noise in the estimated of $E_x^2 + E_y^2$. Optical flow can be computed by a new set of velocity estimates (u^{n+1}, v^{n+1}) from the estimated derivatives and the average of the previous velocity estimates (u^n, v^n) by

$$u^{n+1} = \bar{u}^n - E_x(E_x\bar{u}^n + E_y\bar{v}^n + E_t)/(\alpha^2 + E_x^2 + E_y^2)$$
$$v^{n+1} = \bar{v}^2 - E_y(E_x\bar{u}^n + E_y\bar{v}^n + E_t)/(\alpha^2 + E_x^2 + E_y^2) \quad (3)$$

where n is iteration number, and $\bar{u}$ and $\bar{v}$ are local average of velocity [15].

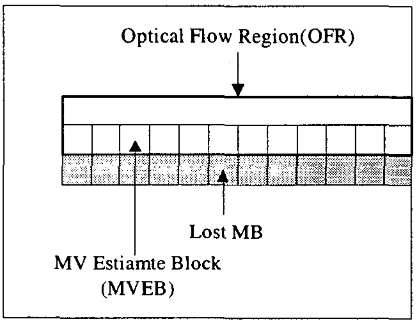


Fig 8. Optical Flow Algorithm

The proposed MV recovery algorithm uses the optical flows of correctly decoded neighboring data as shown in [Fig 8]. As a beginning, we obtain optical flows of the optical flow region (OFR) by Eq (3). The computations are based on just two frames which are the current frame and the previously decoded reference frame. In the second place, we take average of optical flows within the MV estimate block (MVEB) that come in touch with the lost MB. The average value is used as the MV for the lost MB. In this algorithm, how to define the OFR and the MVEB is very important. If the OFR is increased, the computation time is increased. The MVEB is closely related to the accuracy of estimated MV for the lost MB.

V. Simulation Results

In order to evaluate the performance of the error concealment algorithms, three different 4:2:0 CCIR 601 test sequences have been used: Football, Ballet, Bicycle, Train, and Flower Garden. These have been coded by an MPEG-2 encoder at 5 Mbs at 30 frames/sec (NTSC). We use restricted slice structure, N = 12 (number of frames in a GOP), and M = 3 (number of frames between successive I- and P- or P- and P-frames). GOP structure implies that if some error occurs in the I-frame, it can propagate through all other frames within the GOP including the corrupted frame. Similarly, an error of the P-frame may affect neighboring P-frames and B-frames, while errors of the B-frame can be isolated. Therefore, it is desirable to develop error concealment algorithms to prevent error propagation within the GOP and improve reconstructed picture quality. In order to compare the performance of error concealment techniques, we assume that we lost one TS packet in the first P-frame. [Table 1] shows the average peak signal to noise ratio (PSNR) values of the reconstructed frames within the GOP.

Table 1. Performance Comparison

	FOOTBALL	BALLET	BICYCLE	TRAIN	FG
Original	32.59	29.12	26.57	24.88	26.36
AVG	30.62	28.41	24.55	22.95	24.90
MAVG	31.50	28.76	25.05	23.00	24.50
BMA	31.25	28.53	23.37	23.39	25.11
DMVE	31.23	28.52	23.30	23.38	25.12
IEMA	32.21	29.01	25.42	23.44	25.04
OFA	32.59	28.87	25.21	23.14	25.36

Computer simulations have been performed to compare described error concealment algorithms: three conventional algorithms which include AVG [8], BMA [9], and DMVE [12] and proposed algorithms including MAVG, EMA, IEMA, and OFA. In order to estimate the MV of the lost MB, every algorithm uses [-25, 24] SR for motion search with block matching algorithm. While BMA takes one pixel boundary line, DMVE can exploit variable pixel boundary lines from one to eight. From the simulation, we can find that DMVE produces best results when it takes two pixel boundary lines. MAVG use alternative STB structure to reduce computation time and STB size is 8 x 8. EMA generates best performance when EW is 1 pixel. Finally, IEMA has [-5, 4] reduced SR and [-25, 24] normal SR. In the OFA, the width of the OFR is 32 pixels and the size of the MVEB is the same as that of MB.

Table 2. Comparison of Computational Complexity

	(-) Operation	(+) Operation	(×) Operation
BMA	50×50×16×3 = 120000	50×50×15×3 = 112500	50×50×16×3 = 120000
IEMA (worst)	50×50×16×2 = 80000	50×50×15×2 = 75000	50×50×16×2 = 80000
IEMA (best)	10×10×16×2 = 3200	10×10×15×2 = 3000	10×10×16×2 = 3200
OFA	32×16×3×4 + 32×2 = 6208	32×16×3+ 32×16×4 + 32×6×2 = 23296	32×16×3×3 + 32×16×3×12 + 32×4×2 = 3968

[Table 2] compares the computational complexity of different error concealment algorithms. We count the number of (-), (+), and (x) operations to estimate MV of the lost MB. IEMAworst means that we cannot use the initial MV and use the normal SR. IEMAbest is obtained using the reduced SR.

VI. Conclusions

In this paper, we have described the basic concept of error concealment technique, overall configuration of transmission system, and brief MPEG-2 coding algorithm. We have reviewed the merits and demerits of the conventional MV recovery algorithms and proposed MV recovery algorithms to improve the performance of the error concealment. The simulation results indicate that the proposed IEMA generates the best performance. However, we should not overlook the fact that OFA requires the smallest computation time among the introduced algorithms. While MV recovery methods based on motion estimation have a big burden in the decoder, OFA requires only 32 iterations to obtain optical flow of the OFR. These results lead to the conclusion that proposed IEMA and OFA can be solution for error concealment. We respect that proposed error concealment algorithms will provide greatly improved DTV visual quality.

참고문헌

[1] ISO/IEC IS 13818-2 (MPEG-2 Video): Information Technology - Generic Coding of Moving Pictures and Associated Audio Information, April., 1996.

[2] Y. Wang and Q. F. Zhu, "Error Control and Concealment for Video Communication: A Review," Proc. IEEE, Vol.8, pp.422-434, Aug., 1998.

[3] E. Ayanoglu, P. Pancha, A. R. Reibman, and S. Talwar, "Forward Error Control for MPEG-2 Video Transport in a Wireless ATM LAN," Proc. ICIP, Vol.2, pp.833-836, Sep., 1996.

[4] M. Ghanbari and V. Seferidis, "Cell-loss Concealment in ATM Video Codecs," IEEE Trans. on Circuits and Systems for Video Technology, Vol.3, No.3, pp.238-247, June., 1993.

[5] W. Kwok and H. Sun, "Multi-Directional Interpolation for Spatial Error Concealment," IEEE Trans. on Consumer Electronics, Vol.39, No.3, pp.455-460, Aug., 1993.

[6] S. Aign and K. Fazel, "Temporal and Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec," Proc. ICC, Vol.3, pp.1778-1783, June., 1995.

[7] J. W. Suh and Y. S. Ho, "Error Concealment Based on Directional Interpolation," IEEE Trans. on Consumer Electronics, Vol.43, No.3, pp.295-302, Aug., 1997.

[8] H. Sun, K. Challapali, and J. Zdepski, "Error Concealment in Digital Simulcast AD-HDTV Decoder," IEEE Trans. on Consumer Electronics, Vol.38, No.3, pp.108-116, Aug., 1992.

[9] W. M. Lam, A. R. Reilbman, and B. Liu, "Recovery of Lost or Erroneously Received Motion Vectors," Proc. ICASSP, pp.V417-V420, Apr., 1993.

[10] J. Feng, K. Lo, H. Mehropour, and A. E. Karbowiak, "Error Concealment for MPEG Video Transmissions," IEEE Trans. on Consumer Electronics, Vol.43, pp.183-187, May., 1997.

[11] J. W. Suh and Y. S. Ho, "Motion Vector Recovery for Error Concealment," SPIE Visual Communication and Image Processing, pp.667-676, Jan., 1999.

[12] J. Zhang, J. F. Arnold, and M. R. Frater, "A Cell-Loss Concealment Technique for MPEG-2

Coded Video," IEEE Trans. on Circuits and Systems for Video Technology, Vol.10, No.4, pp.659-665, June., 2000.

[13] ISO/IEC IS 13818-1 (MPEG-2 System): Information Technology - Generic Coding of Moving Pictures and associated Audio Information, April., 1996.

[14] W. J. Chu and J. J. Leou, "Detection and Concealment of Transmission Errors in H.261 Images," IEEE Trans. on Circuits and Systems for Video Technology, Vol.8, No.1, pp.74-84, Feb., 1998.

[15] B. K. P. Horn and B. G. Schunch, "Determining Optical Flow," Artificial Intelligence, Vol.17, pp.185-203, 1981.

저 자 소 개

서 재 원(Jae-Won Suh) 정회원

- 1995년 2월 : 충북대학교 전자공학과(공학사)
- 1997년 2월 : 광주과학기술원 정보통신공학과(공학석사)
- 2003년 2월 : 광주과학기술원 정보통신공학과(공학박사)
- 2003년 4월~2004년 8월 : LG전자기술원 MM연 선임연구원
- 2004년 9월~현재 : 충북대학교 전기전자컴퓨터공학부 전임강사

<관심분야> : 디지털 영상신호처리, 동영상 표준화, 생체신호처리