논문 2011-48SP-6-7

프레임 호환 패킹 포맷으로 전송된 3D 스테레오 영상에 대한 내삽 방법

(Interpolation Method for 3D Stereo Images Transmitted by Frame-Compatible Packing Format)

레 앙 부*, 원 치 선**

(Anh Vu Le and Chee Sun Won)

요 약

3D TV를 위한 스테레오 영상은 기존의 디지털 TV와 호환성을 유지하기 위하여 프레임 호환 패킹 포맷으로 전송될 수 있다. 이 경우 수신단은 해상도가 절반으로 줄어든 스테레오 영상으로부터 원래 크기의 좌우 영상을 확대하여 복원할 필요가 있다. 이 논문은 삭제된 영상 라인의 픽셀값을 적응적으로 내삽하는 기법을 제안한다. 삭제되지 않은 주변 영상 라인의 정보로부터 수평 라인 기반 선형 필터와 NEDI6를 선택적으로 적용함으로써 기존의 Bilinear 내삽 기법에 비해 0.6dB 정도의 화질 향상을 달성할 수 있었다.

Abstract

Stereoscopic 3D video can be transmitted by frame-compatible packing format to fulfill the compatibility requirement with the existing digital TV. Then, the reduced stereo image needs to be expanded to the original size at the receiver. This paper proposes an adaptive interpolation method for the discarded image lines. The horizontal line-based linear filter and NEDI6 filter are used selectively for the interpolation of each pixel. Experimental results show that the NEDI6 combined with the horizontal line-based linear filter yields better image quality than the bilinear method by around 0.6dB.

Keywords : 3D TV, Stereoscopic image, Frame-compatible packing format, Image Interpolation

Ⅰ. Introduction

Recently, with increasing interest on 3D movies, 3DTV broadcasting services have started. However, since each stereoscopic 3D video frame consists of two frames with left and right sides, the amount of video data to be transmitted becomes double of the existing digital TV (DTV). So, to utilize existing transmission infrastructure including H.264/AVC, the compatibility of 3D video with conventional DTV transformation format is strongly required[1~4].

We can use the existing video codec and transmission infrastructure by converting the stereoscopic video format to a single frame of DTV. For example, the horizontal line sub-sampling can reduce the vertical resolution of the stereoscopic frames by half. Then, concatenating the frames in the form of a top-bottom packing format gives us a

* 학생회원, ** 정회원-교신저자, 동국대학교 전자전기공학부
(Division of electronics and electrical engineering, Dongguk University)

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원(20110025770) 및 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (NIPA-2011-(C1090-1111-0003)).
접수일자: 2011년9월5일, 수정완료일: 2011년9월27일

sequence of single frames with the same resolution of the existing DTV[1~4]. Also, other packing methods such as side-by-side, row interleaved, column interleaved, time-multiplexing, and checkerboard formats are available.

Using the packing methods, we can utilize the existing transmission infrastructure for the stereoscopic 3DTV. However, at a cost of the compatibility, the frame-compatible packing solutions suffer from the reduction of spatial or temporal resolution due to the decimation. The reduced resolution can be recovered at the decoder by using an interpolation technique. Here, it is important to devise an interpolation algorithm which utilizes the nature of the stereoscopic image structures. For example, for the top-bottom packing with one line-offset, the parallax estimation for each deleted line can be exploited to determine an appropriate interpolation mode[5]. Although this method[5] improves the video quality significantly, it is computationally demanding especially in the encoder. Also, it needs a substantial number of bits for side information (i.e., mode information) to be delivered to the decoder.

In this paper, we propose another interpolation method for the top-bottom frame-compatible packing format. This time, the main concern of our proposal is the reduction of computational complexity and the extra bits to be sent to the decoder. To achieve our goal, we combine the adaptive linear interpolation[6] with NEDI6 (New Edge Directed Interpolation 6)[7] method. That is, in our recent work, we found that the NEDI4 combined with adaptive linear interpolation method improves the interpolation performance[6]. Here. the NEDI4 method will be replaced by NEDI6 in this paper. Note that the NEDI6 is more suitable for the interpolation of horizontally sub-sampled frames (i.e., for the sub-sampling of the top-bottom packing).

Ⅱ. Frame-Compatible Top-bottom Packing

Our interpolation is based on the top-bottom frame-compatible packing format (see Fig. 1). So, every one of two consecutive horizontal lines will be dropped to reduce the vertical resolution by half for

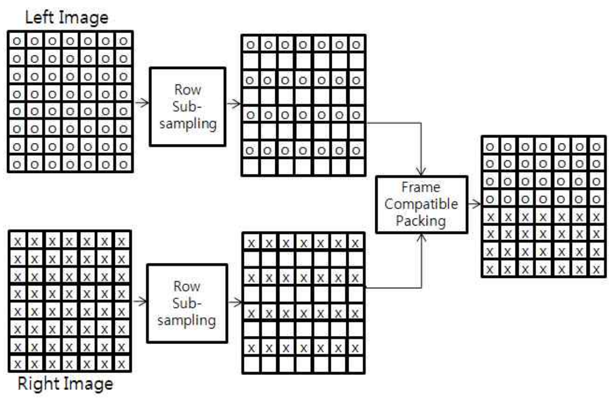

그림 1. 상-하 프레임 호환 패킹
Fig. 1. Top-bottom frame-compatible packing.

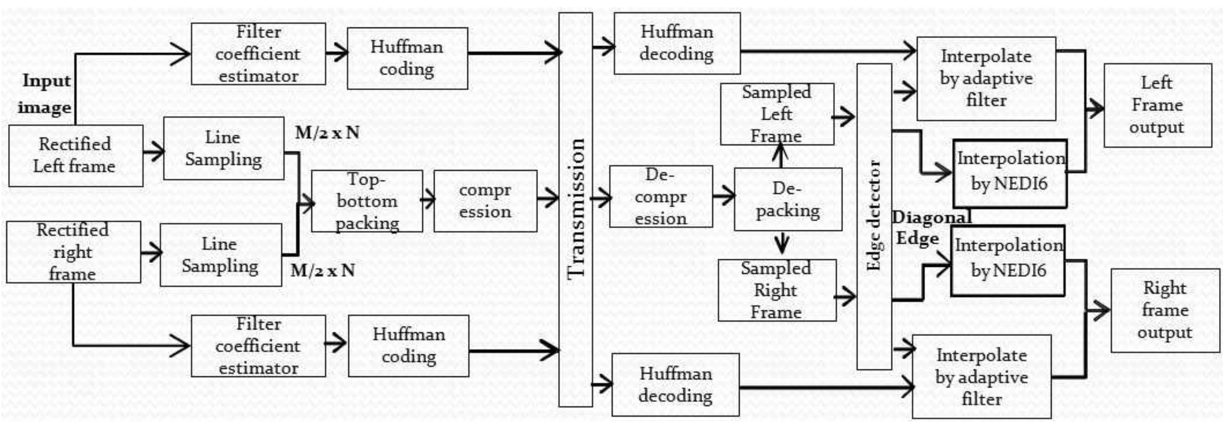

그림 2. 인코더-디코더 전체 블록도
Fig. 2. Block diagram for the encoder and the decoder.

both left and right frames. Then, the sub-sampled left and right images are combined as a single frame for compression and transmission (see Fig. 2 for the overall block-diagram of the transmission system). At the decoder the reduced left and right image frames are expanded by using an interpolation technique.

Recalling that the top-bottom packing format is based on the horizontal line deletion, we can utilize this line sampling structure as much as possible for better interpolation. For example, we can estimate the filter coefficient for each deleted line at the encoder using the original data in the vertical direction (see Fig 3). Also, the availability and the layout of the upper and lower line data for each deleted horizontal line lead us to consider 6 neighboring pixel values to be exploited for directional interpolations of the NEDI6 (see Fig 4). In the following sections, the above structures of the top-bottom packing format will be exploited for the development of our interpolation algorithm.

Ⅲ. Horizontal Line-based adaptive filter

The basic unit for the top-bottom packing is a horizontal line. That is, even numbered (or odd numbered) rows of left and right frames will be deleted to reduce the heights of the images by half. Note that since the original pixel values of dropped rows are available at the encoder, they can serve to provide a good cue to recover the original vertical resolution at the decoder. This leads us to apply the idea of side information to be transmitted to the decoder for the interpolation[7].

As shown in Fig 2, at the encoder, the optimal coefficient of linear filter for each deleted row is determined and transmitted to the decoder as a side information for interpolation. Specifically, as shown in Fig. 3, for the deleted line in the row of 2k+1, we employ a filter coefficient a_{2k+1}, which linearly approximates the deleted line pixels as follows

$$x'_{2k+1,j} = a_{2k+1}x_{2k,j} + (1-a_{2k+1})x_{2k+2,j} \quad (1)$$

where $x_{2k,j}$ and $x_{2k+2,j}$ are the pixels values of the

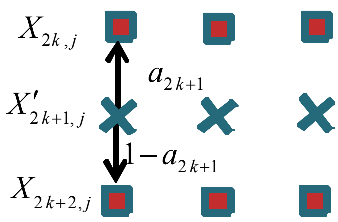

그림 3. 수평 라인 기반 1D 내삽
Fig. 3. Horizontal line-based 1D interpolation.

표 1. 필터 계수를 위한 Huffman 테이블.
Table 1. Huffman table for the filter coefficient.

00	0.5
011	0.4
101	0.6
110	0.7
111	0.3
0001	0.2
0010	0.9
1001	0.8
1010	0.1

undeleted upper line at 2k and the lower one at 2k+2. Also, $x'_{2k+1,j}$ represents the interpolated pixel value at (2k+1,j). For NxM image, the optimal filter coefficient a'_{2k+1} can be obtained by finding the value which minimizes the mean square error $\frac{1}{M}\sum_{j=1}^{M}(x_{2k+1,j} - x'_{2k+1,j})^2$ with respect to the coefficient a_{2k+1}. By taking the first derivative of this sum of squared errors with respect to the coefficient and set to zero, we have the following closed-form expression for the optimal coefficient estimation

$$a'_{2k+1} = \frac{\sum_{j=1}^{M}(x_{2k+1,j} - x_{2k+2,j})(x_{2k,j} - x_{2k+2,j})}{\sum_{j=1}^{M}(x_{2k,j} - x_{2k+2,j})^2} \quad (2)$$

Note that equation (2) should be done for each deleted line. So, for each deleted horizontal line 2k+1, the optimal coefficient a'_{2k+1} is calculated and coded for transmission. Since the estimated filter coefficient takes a value around 0.5 for most cases, we can reduce the number of bits to be sent by using the Huffman coding technique. Specifically, we can use the huffman coding table shown in Table 1 for the binary encoding[7].

Ⅳ. Interpolation by NEDI6

Note that the filter coefficient a_{2k+1} in equation (1) just takes into account the vertical continuity of the pixel values. Therefore, if there exists a diagonal edge at a pixel to be interpolated, then the linear interpolation may not be appropriate. To solve this problem, we need a special treatment for interpolating the pixels with diagonal edges. That is, if the pixel to be interpolated turns out to be classified as a diagonal edge, then we apply the NEDI6 interpolation method. Otherwise, the horizontal line-based adaptive filter in section Ⅲ will be used. This selective task requires an edge classifier at the decoder. For edge detection and classification, we can use a simple differential gradient method[8], where the gradients are calculated with the undeleted neighboring pixel values in the upper and lower lines. Since there are six pixels available around deleted edge pixel, NEDI6 is more appropriate than the NEDI4 of utilizing only four undeleted neighboring pixels. Also, NEDI6 turns out to give better visual quality than NEDI4[9].

Fig. 4 shows the basic idea of NEDI6, the deleted pixel is estimated according to the following linear equation:

$$x'_{2k+1,j} = \alpha_1 x_{2k,j-1} + \alpha_2 x_{2k,j} + \alpha_3 x_{2k,j+1} + \alpha_4 x_{2k+2,j-1} + \alpha_5 x_{2k+2,j} + \alpha_6 x_{2k+2,j+1} \quad (3)$$

In equation (3) we need 6 parameter values α_1, α_2, α_3, α_4, α_5, and α_6 for the interpolation. These parameter values can be estimated using the undeleted horizontal pixel values in the upper and lower lines. However, since the center pixel value $x_{2k+1,j}$ is not known, it is not possible to apply the MSE estimate at the decoder. Instead, as shown in Fig 4, we can

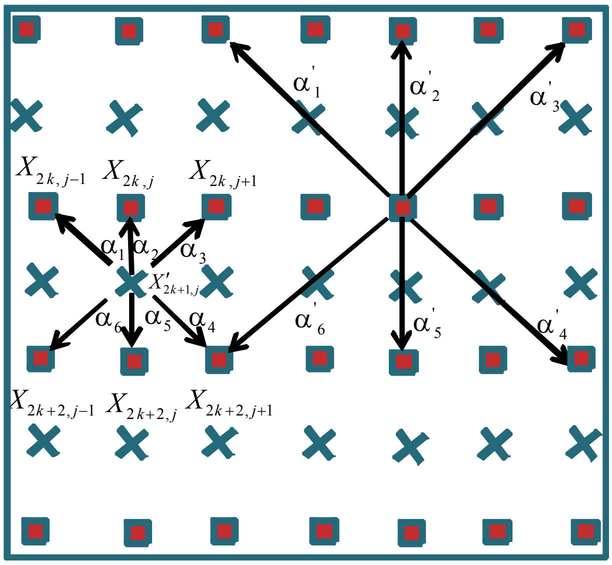

그림 4. NEDI6 기본 개념도
Fig. 4. Basic concept of NEDI6.

assume that the parameter values at high resolution (i.e., α_1, α_2, α_3, α_4, α_5, and α_6) are same as those of the undeleted low resolution (i.e., α'_1, α'_2, α'_3, α'_4, α'_5, and α'_6)[9]. Then, by estimating α'_1, α'_2, α'_3, α'_4, α'_5, and α'_6 with the optimal minimum MSE within WxW window for (2k+1, j), we can use the estimated high resolution parameters to interpolate the pixel value at (2k+1, j) at the decoder as follows

$$x'_{2k+1,j} = \alpha'_1 x_{2k,j-1} + \alpha'_2 x_{2k,j} + \alpha'_3 x_{2k,j+1} + \alpha'_4 x_{2k+2,j-1} + \alpha'_5 x_{2k+2,j} + \alpha'_6 x_{2k+2,j+1} \quad (4)$$

Ⅴ. Combined Line-based adaptive interpolation and NEDI6

There is no need to send side information (i.e., estimated filter coefficients) for NEDI6. As shown in Fig. 4, all filter coefficients can be estimated using the received undeleted image data at the decoder. However, estimating 6 parameter values for all pixels to be interpolated with the received image data in a WxW window is quite time-consuming. To reduce the computational complexity at the decoder, we can combine the horizontal line-based adaptive filter of section III with the NEDI6 of section IV. Then, by employing a simple edge classifier, if the current pixel to be interpolated is classified as a diagonal edge type, then we apply the NEDI6 by estimating the 6 parameters α'_1, α'_2, α'_3, α'_4, α'_5, and α'_6 for that particular pixel. Otherwise, we use the decoded parameter of (2) for the linear interpolation.

VI. Experimental Results

The NEDI6 alone can be used for the interpolation. This NEDI6-only and the NEDI6 combined with the horizontal line-based adaptive filter are compared with other existing interpolation methods in terms of PSNR and execution time in this section. Our experiments are executed in MATLAB version 7.8.0(R2009a) environment installed on PC with IntelcoreTMi3CPU 530@2.93GHz 4G Ram.

Five 3D left and right video sequences[10]: Hand, Alt_Moabit, Professor, Car, Horse are used for our experiments. The original stereoscopic sequences are down-sampled for the top-bottom packing. Then we apply H.264/AVC compression standard with various compression ratios. The compressed video sequences are decompressed at the decoder and unpacked to have left and right frames. They are up-sampled by using various interpolation methods including bilinear, 6-tap interpolation filter of H.264/AVC, a combination of bilinear and NEDI6, NEDI6-only, and the combination of the horizontal line-based adaptive filter and NEDI6 (denoted as "proposed method") for comparisons. Compression parameters for H.264/AVC are set such that bit-rate ranges [500-2000 kbps], GOP=15, baseline-profile, and compression structure with IPBPBPB.

Fig. 5 shows average PSNR values of the five test videos for different bit-rates. As one can see, among all tested methods the NEDI6 and the proposed method yield the highest PSNR values. This demonstrate the NEDI6-based interpolation methods are superior to other non-NEDI6 interpolations. Also, the combination of NEDI6 with the horizontal line-based adaptive filter is better than the combination of NEDI6 with the bilinear

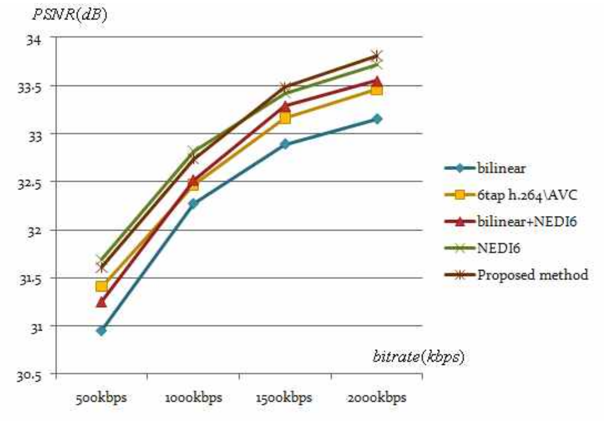

그림 5. PSNR 비교
Fig. 5. PSNR comparisons.

표 2. H.264/AVC 압축을 제외한 인코더와 디코더의 수행시간 비교.
Table 2. Comparisons of execution time for the encoder and the decoder except the H.264/AVC compression processes.

	Bilinear	6tap	NEDI6	Bilinear+NEDI-6	Adaptive+NEDI-6
hand	0.15	0.51	8.51	1.43	1.84
outdoor	0.25	0.51	9.21	1.72	2.45
professor	0.10	0.95	7.95	1.22	1.51
car	0.15	0.71	8.71	1.13	1.54
horse	0.15	0.2	7.2	1.24	1.39
Average	**0.16**	**0.576**	**8.316**	**1.348**	**1.746**

filter. The proposed method gives around 0.6dB higher PSNR values for all bit-rates comparing to the bilinear method.

The NEDI6 and the proposed method yield similar PSNR values. However, as shown in Table 2, the comparisons of time-consumption including all processing times except compression and decompression (i.e., H.264/AVC encoder and decoder) show that the NEDI6-only needs about 5 times more computations than the proposed NEDI6 combination method.

Because the proposed method (i.e., the NEDI6 combined with the horizontal line-based interpolation) needs extra steps at the encoder for estimating the filter coefficient for each deleted row, it will consume more CPU time than the NEDI6 combined with the bilinear method. However, as shown in Fig. 5, the proposed method yields higher PSNR values. As one can see in

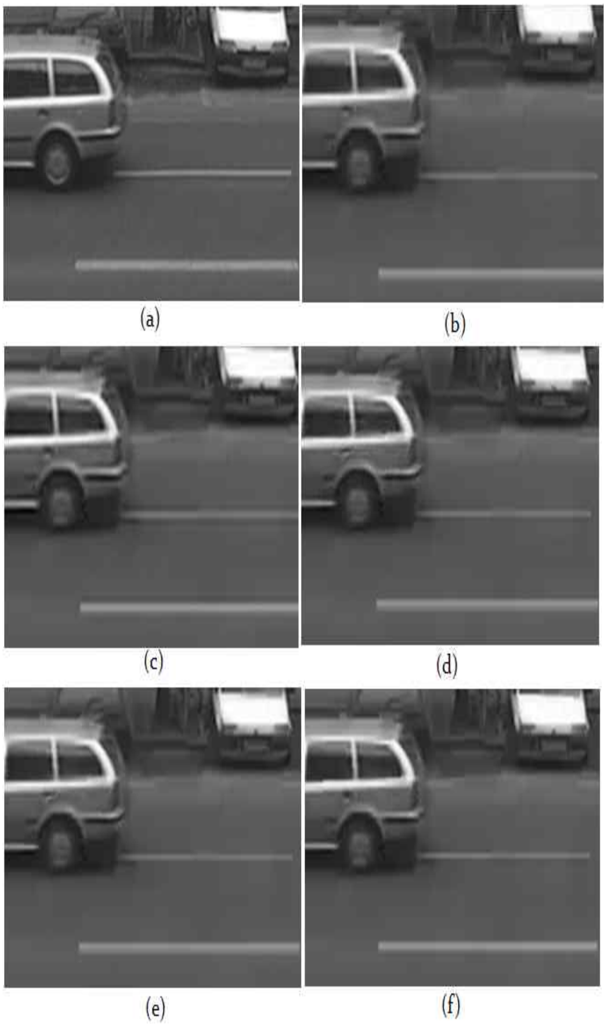

그림 6. 2000kbps에서 내삽 결과 비교: (a) 원영상, (b) bilinear, (c) 6-tap H.264/AVC 필터, (d) NEDI6, (e) NEDI6+bilinear, (f) 제안방법 (NEDI6+수평 라인 기반 적응 필터)

Fig. 6. Comparisons after the interpolations at 2000kbps: (a) Original image, (b) bilinear, (c) 6-tap H.264/AVC filter, (d) NEDI6, (e) NEDI6+bilinear, (f) the proposed method (NEDI6+horizontal line-based adaptive filter).

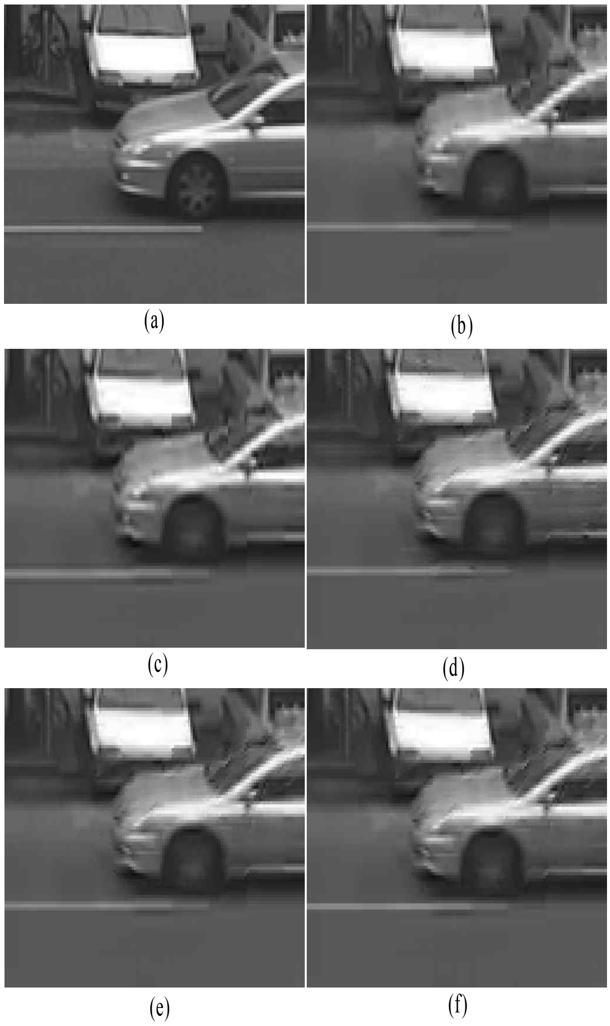

그림 7. 500kbps에서 내삽 결과 비교: (a) 원영상, (b) bilinear, (c) 6-tap H.264/AVC 필터, (d) NEDI6, (e) NEDI6+bilinear, (f) 제안방법 (NEDI6+수평 라인 기반 적응 필터)

Fig. 7. Comparisons after the interpolations at 500kbps: (a) Original image, (b) bilinear, (c) 6-tap H.264/AVC filter, (d) NEDI6, (e) NEDI6+bilinear, (f) the proposed method (NEDI6+horizontal line-based adaptive filter).

Fig. 6, there are some noticeable differences between the non-NEDI6 methods (i.e., (b) and (c)) and the NEDI6-based methods (i.e., (d), (e), and (f)) especially in the hatchback of the car. Even among the NEDI6-based methods, some differences are noticeable (see the doorknob of the car). For a low bit-rate (i.e., a high compression ratio) video with some compression artifacts, similar subjective differences are observable (see Fig 7).

The overhead of the additional bits for the filter coefficients to be sent to the decoder is from 2 to 4 bits per deleted line (see the Huffman table in Table 1), which increases the bit per pixel (bpp) only about 0.0005.

Ⅶ. Conclusions

In this paper, the NEDI6 interpolation method and its combination with the adaptive linear interpolation method are proposed for possible interpolation methods of the frame-compatible top-bottom packing format in 3DTV. Experimental results show that both the NEDI6 interpolation and its combination with the horizontal line-based adaptive interpolation method yield the highest PSNR values. In terms of computational complexity, however, the NEDI6 combined with the horizontal line-based adaptive filter is 5 times faster than the NEDI6 alone.

References

[1] 원치선, “3DTV 방송을 위한 프레임 호환 패킹기술,” 전자공학회 논문지, 제38권 제2호, 151-156쪽, 2011년 2월

[2] A. Vetro, W.-H. Lou and M. Flynn, “TV Architecture Supporting Multiple 3D Services,” IEEE ICCE, 2010.

[3] ISO/IEC JTC1/SC29/WG11, “Use cases and requirements for the enhancement of MPEG frame-compatible stereoscopic 3D video,” N11681, October, 2010.

[4] A. Vetro, T. Wiegand, and G. J. Sullivan, “Overview of the Stereo and Multiview Video Coding Extensions of the H.264/AVC Standard,” Proc, of the IEEE, vol 99 Issue 4, pp. 626 – 642 2011.

[5] C. S. Won, “Adaptive Interpolation for 3D Stereoscopic Video in Frame-Compatible Top-Bottom Packing”, Consumer Electronics (ICCE), 2011 IEEE International Conference, 2011.

[6] A. V. Le, H. Kim, and C. S. Won, “Fast Interpolation for Line Pruned Images,” J. Electron. Imaging 20, 033010, 2011.

[7] D. T. Vo, J. Sole, P. Yin, C. Gomila, and T. Q. Nguyen, “Selective Data Pruning-Based Compression Using High-Order Edge-Directed Interpolation,” IEEE Tr. On Image Processing, 19(2), oo, 399-409, 2010.

[8] M.-J. Chen, C.-H. Huang, W.-L. Lee, “A fast edge-oriented algorithm for image interpolation” Image and Vision Computing, Springer, 23, 791-798, 2005.

[9] X. Li and M. T. Orchard, “New edge-directed interpolation,” IEEE Tr. Image Processing, 10(10), 1521-1527, 2001.

[10] http://sp.cs.tut.fi/mobile3dtv/stereo-video/

저 자 소 개

Anh Vu Le(학생회원)
2007년 Hanoi University of Technology 학사
2011년 동국대학교 전자전기공학부 석사 과정
<주관심분야 : 영상신호처리, 영상필터>

원 치 선(정회원)-교신저자
1982년 고려대학교 전자공학과 학사
1986년 University of Massachusetts /Amherst 석사
1990년 University of Massachusetts /Amherst 박사
1989년~1992년 (주) 금성사 선임연구원
1992년~현재 동국대학교 전자전기공학부 교수
<주관심분야 : 영상신호처리, 영상검색, 컴퓨터비젼>