대용량 파노라마 비디오 스트리밍의 성능개선

Performance improvement for Streaming of High Capacity Panoramic Video

김 영 백* 김 태 호** 이 대 규*** 김 재 준****
Young Back Kim Tae-Ho Kim Dae-Gyu Lee Jaejoon Kim

요 약

인터넷, 이동통신 및 방송분야에 걸쳐 고화질 파노라마 비디오를 제공할 경우 고압축률과 임의 접근 기능 모두를 만족하는 적절한 비디오 코덱이 요구된다. 사용자 측면에서 고압축률은 대용량 파노라마 데이터의 비디오 스트리밍을 하기 위해 반드시 필요하다. 임의 접근 기능은 시점과 방향을 사용자가 자유롭게 이동할 수 있게 해 준다. 본 논문에서는 압축 효율이 높은 H.264/AVC를 기반으로 셀 단위 병렬 처리 기법을 통해 10Mbps 대역폭에서 대화면 파노라마 영상의 스트리밍 서비스를 위한 성능 개선 방안을 제안한다. 제안된 방안은 전체 화면을 256 x 256 이하 크기의 셀로 나누어 인코딩하고 현재 뷰(view)에 포함된 셀들만 디코딩한다. 이때 인코딩/디코딩은 셀 단위로 병렬 처리한다. 또한 현재 뷰에 포함된 셀들만 전송함으로써 블록 추출 없이 임의 접근 기능이 가능함을 실험을 통해 보여준다.

ABSTRACT

When providing high quality panoramic video across the Internet, mobile communications, and broadcasting areas, it requires a suitable video codec that satisfies both high-compression efficiency and random access functionality. The users must have high-compression efficiency in order to enable video streaming of high-volume panoramic data. Random access allows the user to move the viewpoint and direction freely. In this paper, we propose the parallel processing scheme under cell units in order to improve the performance of streaming service for large screen panoramic video in 10Mbps bandwidths based on H.264/AVC with high compression rate. This improved algorithm divides a screen composed of cells less than 256 x 256 in size, encodes it, and decodes it with the cells in the present view. At this point, encoding/decoding is parallel processed by the present cell units. Also, since the cells only included in the present view are packed and transmitted, the possible processing of not extricating blocks is proven by experiment.

☞ KeyWords : Panoramic Video Streaming, cell, Multiprocess, Data Transmission, PAV Package Format, H.264/AVC
파노라마 비디오 스트리밍, 셀, 다중프로세스, 데이터 전송, PAV 패키지 포맷, H.264/AVC

1. Introduction

In the past, users acquired multimedia services passively, but the trend has changed into an interactive multimedia services environment. The biggest advantage of interactive multimedia service is that users can get the information they want in real time from the service provider. The panoramic video based application is the well defined system for the interactive multimedia environment. The QuickTime VR system [1] has shown that the interactive multimedia video application can be provided in a network environment. A panoramic video is one that unwraps a 360° video image into a seamless, distortion free horizontal image in real

* 준 회 원: 대구가톨릭대학교 산학협력교수
realtech@hanafos.com
** 준 회 원: 대구대학교
crewx1@nate.com
*** 준 회 원: 대구대학교
gyu1211@nate.com
**** 정 회 원: 대구대학교 정보통신공학부 교수
jjkimisu@daegu.ac.kr(교신저자)

[2009/11/26 투고 - 2009/12/15 심사(2010/02/09 2차) - 2010/02/16 심사완료]

time. As shown in Fig. 1, the images generated by multiple adjacent cameras were processed by image stitching or photo stitching to combine multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image [2-6]. The typical applications are VOD (Video-on-Demand) service, IPTV (Internet Protocol Television), 3DTV and etc. Due to the high volume of streaming required for a panoramic video, however, more research is necessary on compression and transmission [7][8].

In general, the size of panoramic video ranges from 1K x 1K to 4K x 4K. The test sample named LittleCity.avi shown in Fig. 1 is encoded by motion-JPEG with a size of 1920 x 960, 9 seconds of replaying time and amounts of 32Mbytes [9]. Comparing this to the 600 – 900 Mbytes for one movie film with 90 minutes replaying time, it becomes clear that the storage volume of panoramic video is likely to be very large. In order to provide application service with large volume panoramic video, the compression ratio should be pushed to the limit. Panoramic video service is provided on demand; the service provider displays the user's required picture once the user makes a request to the service provider. In other words, the user can see a part of an entire panoramic video and access the region of interest by dragging the mouse or changing a view point randomly.

(Figure 1) An example of panoramic video (LittleCity.AVI)

Thus far, panoramic video streaming services based on a standard video compression algorithm are unknown. In this paper, we suggest using cell unit parallel processing based on H.264/AVC to improve streaming of panoramic video service from a 10Mbps bandwidth. This proposed algorithm divides a screen composed of cells less than 256 x 256 in size, encodes and decodes it with the cells in the present view. At this point, encoding/decoding is parallel processed by the present cell units. Furthermore, since the cells only included in the present view are packed and transmitted, the possible processing of not extricating blocks is proven by experiment.

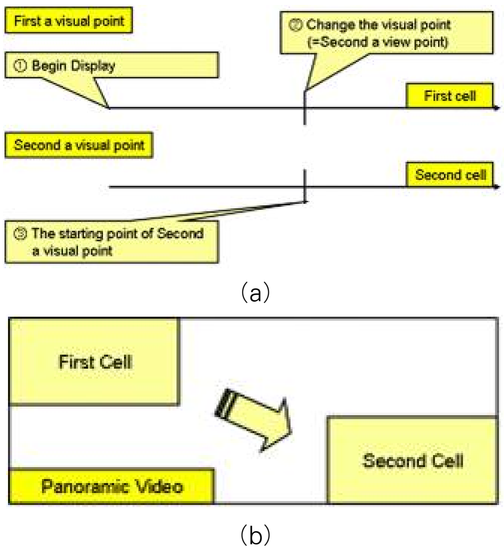
The rest of this paper is organized as follows. First, we describe the introductory review for random access and compression of the panoramic video in section 2. We then propose a new scheme for streaming of high capacity panoramic video in section 3. In Section 4, we analyze our experimental results. Conclusions are drawn in Section 5.

2. Review on Random Access and Compression

On encoding/decoding panoramic video, random access and compression should be taken into account as described in section 1 [10]. Panoramic video has been shown in sizes of over 1k x 1k supported by MPEG (Moving Picture Expert Group) and characterized by high resolution and large volume. This means that there are difficulties to provide the service without an efficient compression process. Therefore, it is evident that the compression efficiency plays a key role in the communication of panoramic video. Random access functionality is another critical concern in providing panoramic video. Random access functionality based on MPEG video compression has delivered for weak results. Most video compression techniques utilize the interframe prediction method to compress, which relies on using the prediction from the previous frame. This indicates that it should be willingly submitted in the delay time of several frames when we try to access to the particular frame through the usage of the prediction from the previous frame. Therefore, a video codec to satisfy both compression efficiency and random access functionality is necessary to provide a panoramic video service. However, it has proven extremely difficult to find a commercial codec to satisfy these two functions. This indicates the need for the development of an optimal codec with the right balance between compression efficiency and random access functionality.

The random access plays an important role in the case of panoramic video with high compression efficiency. The random access function enables users to directly access a desired frame or region in an image sequence [11]. Fig 2 shows the simplified diagram for the concept of random access in the panoramic video. As shown in Fig. 2-(a), first, the panoramic video replays at number 1. If the user sends the signal to change the viewpoint, the service provider receives the viewpoint change request information, jumps to number 2 and replays the frames from that point.

Table 1 [12][13] shows the general comparison results depending on the type of codec, the compression efficiency and random access functionality and transmission cost. From the compression efficiency point of view, MPEG-4 AVC (Advanced Video Coding) inter prediction demonstrates the best efficiency [14]. In terms of random access, JPEG or Motion-JPEG shows the best performance. The complexity of the random access function describes the time for accessing a particular frame. While JPEG or Motion-JPEG does not need to compute the operation between the neighbor frames, Motion-JPEG2000, MPEG-4, and MPEG-4 AVC do need the operation to access the requested frame. The MPEG-4 AVC inter prediction mode, which has the highest compression capacity, has the low cost for transmitting.



(Figure 2) Random access and conversion of viewpoint for panoramic video

3. Streaming of High Capacity Panoramic Video

The improvement of network transmission capacity and advances in multimedia processing technology will make panoramic video streaming possible in the near future. However, various research projects are in progress in order to find the optimal method to stream high volume panoramic video [15－18]. In this paper, we present an effective scheme for streaming of high capacity panoramic video. When we take our experimental environment into consideration with 10Mbps for the average bandwidth of the used network, it is hard to stream high volume panoramic video for want of network bandwidth. In this section, we suggest a couple of plausible streaming scenarios, and give an account of the PAV (Panoramic Video) system and the transmission package format, which was suggested after examining the advantages and disadvantages.

3.1 Streaming scenario 1

The first scenario uses the patch scheme, dividing a number of sub patches for a panoramic video from the server side as shown in Fig. 3 [19]. The individual patch is then encoded with a JPEG to efficiently realize the random access function. This scheme shows efficiency for encoding a panoramic video but struggles with streaming in environments with less than 10Mbps of network bandwidth. This scheme has several drawbacks.

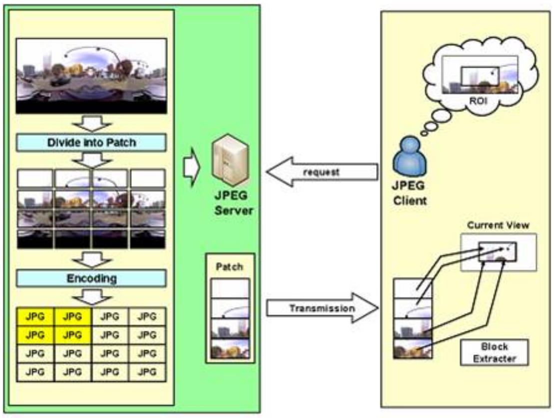
First, the implementation of the random access function is easy, but the compression ratio went down when we encoded the panoramic video with a JPEG codec. Fig. 4 shows the comparison results between the JPEG based codec and the H.264 based codec encoding. The H.264 inter prediction shows 2Mbyte of compression while JPEG based encoding results in 31Mbytes for the 192Mbytes of the tested video. In the case of the H.264 inter prediction, the necessary capacity for the one particular view point (the yellow mark) is about 500Kbytes using our scheme.

Second, when the user requests the region of interest out of the panoramic video, the server may transfer the neighbor patch frame to the client. At this moment, the transmitted data may include unnecessary information due to the transmission of the neighboring parts as well as the region of interest.

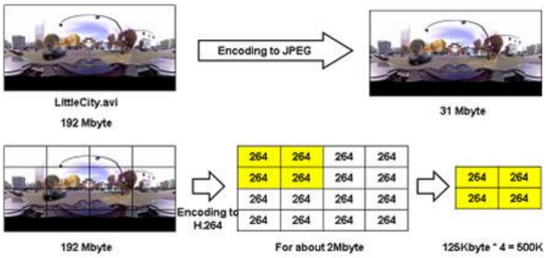
Third, the user needs to decode the transmitted patches to display the region of interest as shown in Fig. 5. Once the decoding process is done, it is possible to extract the ROI through the block extractor. This process results in an inefficient method of decoding unnecessary parts. Furthermore, this process involves a long processing time if large patches are transmitted. In order to solve this problem, our scheme introduces the concept of a multithread process for reducing decoding time through parallel processing of the transmitted patches.

(Table 1) The comparisons results on type of codec, compression efficiency and random access transmission cost

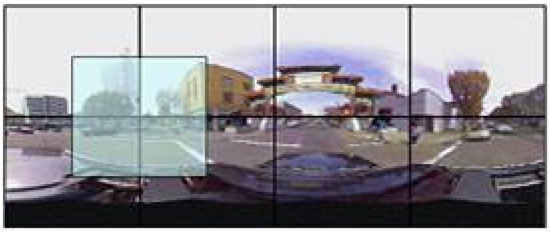
Codec	Compression Efficiency	Random Access	Transmission Cost
JPEG	Not better than a movie codec	Best efficiency	high
Motion-JPEG	Not better than a movie codec	Best efficiency	↓
JPEG2000	Better than JPEG	Full-frame coding and disadvantage for partial coding	
MPEG-4	Better than JPEG or JPEG2000	Slower than JPEG	
MPEG-4 AVC Intra prediction	Better than JPEG or JPEG2000	Slower than MPEG-4	
MPEG-4 AVC Inter prediction	Best efficiency in compression	Bad Disadvantage for random access	low



(Figure 3) Encoding/decoding for Streaming scenario 1



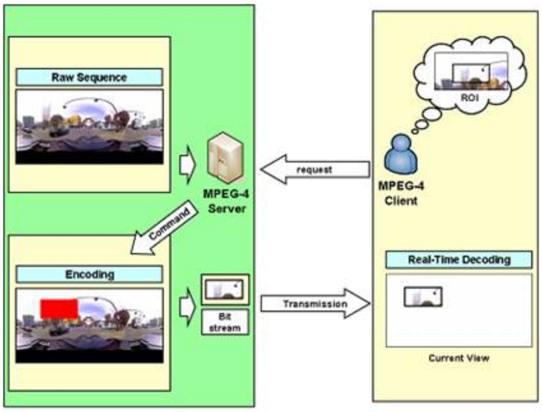
(Figure 4) Compression rate comparison between JPEG and MPEG4-AVC inter mode



(Figure 5) A weakness of Tiling
(□ = Region of interest)

3.2 Streaming scenario 2

The scenario shown in Fig. 6 is that the user designates a region of interest and requests the ROI from the MPEG-4 server which holds the panoramic video. The server encodes the ROI only, sends the bitstreams back to the user and displays the ROI immediately. In order to stream the bitstreams in this scenario, encoding/decoding of the MPEG-4 visual bitstreams should be executed in real time. However, it is not possible to stream due to the complexity delay. Even though the delay caused by the encoding/decoding process can be overcome, live streaming can only be implemented but the VOD (Video on Demand) streaming cannot be implemented. Since the panoramic video consists of a large amount of data, and it has been not encoded in advance, a heavy burden has been imposed on the server for the storage of the raw data. Broadcasting in this scenario can only be provided with live streaming. In order to prevent this problem, we suggest dividing the whole screen in to cells less than 256 x 256 in size, encode it, and then finally, transmit it using only the cells that are included in the requested ROI.

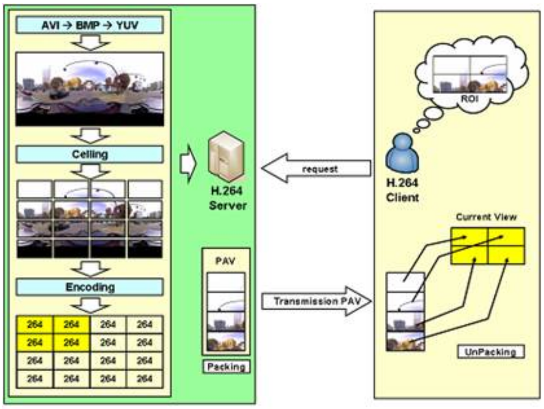


(Figure 6) Encoding/decoding for Streaming scenario 2

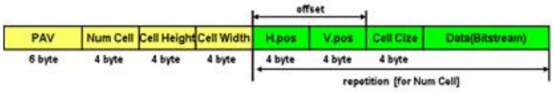
3.3 Proposed scheme

The proposed scheme for streaming of panoramic video investigates the pros and cons of the two scenarios. Fig. 7 describes the proposed scheme. The server divides the panoramic video into tens or hundreds of cells prior to streaming. In the case of VOD, these cells are encoded in advance. In the case of live streaming, raw data that is coming in real time is encoded by the cell unit. When the user requests the ROI from the H.264 server, the server can send the PAV (Panoramic Video) package containing the corresponding cells as described in Fig. 8. The user decodes the received package in parallel.

Fig. 8 describes the structure of the PAV package. The first "PAV" indicates a file type of 6 bytes. The second part "Num Cell" means the number of bitstreams in the PAV file. The third and fourth are the cell height and width when the image is celled after the YUV conversion. Finally, the image data is repeated by a number of cells with the following information: horizontal position, vertical position, cell size and bitstream data for each cell. If the cell size is big as shown in Fig. 9-(a), it is not easy to operate the fine view point transformation on the current view. In this case, the alternative is controlling of the cell size as shown in Fig. 9-(b).



(Figure 7) The proposed PAV system

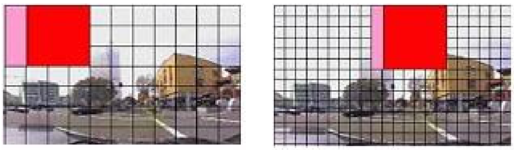


(Figure 8) The structure of PAV package format

4. Experimental Results

For large screen panoramic video streaming, compression capacity and random access functionality are extremely important. Random access ability is particularly important for the user. In the client part, it is embodied along with the decoding ability. In this research, we experimented after only embodied the client part of the PAV system. The cells needed for the experiment divided the YUV file and each were encoded through the JM 10.2. The client was changed to use JM 10.2 source as a shed and was also embodied to be used as library mode in a premade UI.

In our experiment, we reviewed and verified the performance of the PAV system based on the decoding time, PSNR (Power signal-to-noise ratio) and the compression ratio. We used the H.264 inter prediction coding scheme. In the case of decoding time, we utilized the single/multiple cell method to measure the video performance using the decoding time as frame per second. For the image quality, we used the PSNR. The test sequence was LittleCity.avi as described in Fig. 1 and Table 2. Fig. 10 displayed the proposed scheme and the screen shot of the execution. It was possible to replay the ROI depending on any arbitrary change of the viewpoint.

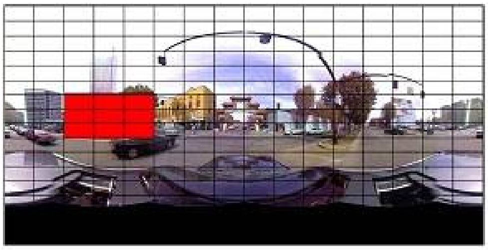


(a) (b)

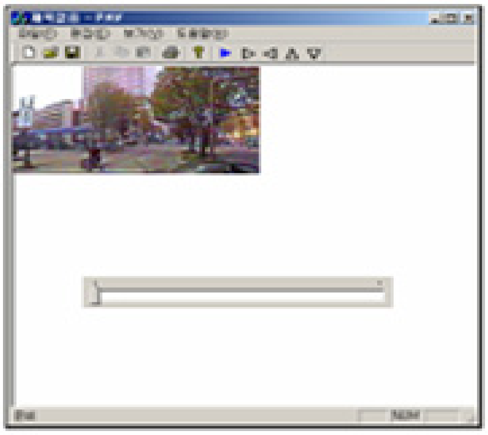
(Figure 9) The differencecell size, (a) big size cell, and (b) smallsize cell

(Table 2) The specification of test sequence

Test Sequence	LittleCity.avi
Source	Immersive Media
Resolution	2048 x 1024 (100frames)
Frame rate	15 Hz



(a)



(b)

(Figure 10) The implementation result of the proposed algorithm, (a) cell using in display, and (b) screen shot

The comparison results of the decoding time are summarized in Table 3. The cell size indicates the divided cell size individually and the current view means the number of cell displaying on the current screen. The decoding time was measured between starting and ending time of decoding. The empty spaces in Table 3 indicate the measurement is not possible due to high values on bit/picture.

If we look at the frame item, the measurement is low and even though the present view is 1, the measurement if lower than 10. This is related to the limitations of the JM source which is the standard reference S/W of H.264/AVC. Actually, in a normal PC (CPU:3.6G, RAM:2GB) environment, there is no HD screen that has 1fps decoding ability. On the other hand, an FFMPEG has a 24fps decoding ability [20]. Henceforth, if we use an optimized codec such as the FFMPEG as a source, we should be able to improve its ability.

Fig. 11 shows the performance of the image quality with the average PSNR. The measurement values depend on the cell size and the number of quantization. The PSNR is calculated as follow.

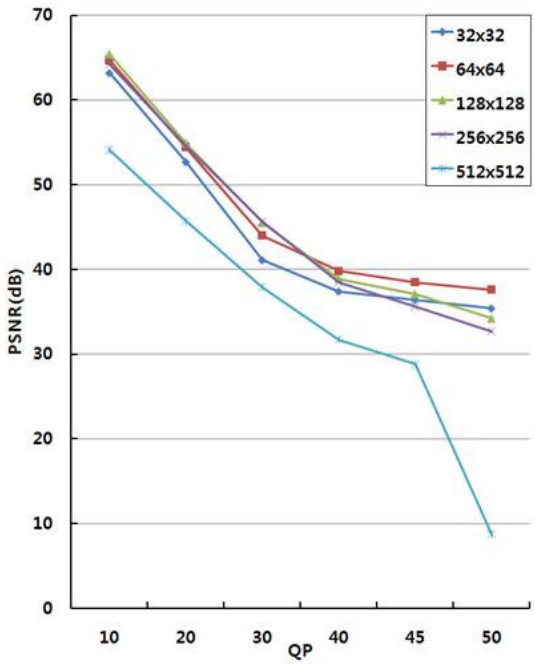
$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (1)$$

where, MAXi is the maximum possible pixel value of the image and MSE is the mean square error.

According to Fig. 11, we can determine that PSNR has been decreased by increasing the number of quantization parameter. This means the image quality is decreased as an increment of the number of quantization. The image quality dropped abruptly when the cell size became more than 256 x 256 image. In general, in the case of PSNR of over 40dB, human beings cannot detect the drop in image quality [21]. Therefore, it is possible to prevent image quality deterioration when we use cell sizes less than 256 x 256 image and quantization below 40dB. Table 4 summarizes the compression ratio with the capacity before and after codec. Embodiment based on the H.264/AVC codec shows high efficiency of compression capacity.

(Table 3) The comparison results for decoding time

Cell size	Current View	Decoding time (sec)	frames / sec
32x32	1	9.87	10.13
	4 (2 : 2)	19.72	5.07
	9 (3 : 3)	39.64	2.52
64x64	1	9.92	10.08
	4 (2 : 2)	19.75	5.06
	9 (3 : 3)	-	-
128x128	1	10.00	10.00
	4 (2 : 2)	20.48	4.88
	9 (3 : 3)	-	-
256x256	1	14.29	6.99
	4 (2 : 2)	-	-
	9 (3 : 3)	-	-
512x512	1	27.98	3.57
	4 (2 : 2)	-	-
	9 (3 : 3)	-	-



(Figure 11) PSNR results

(Table 4) Comparison in compression efficiency

Cell size	Before (KB)	After (KB)	Compression rate (%)
32x32	333414	2457.6	99.26
64x64	330291	716.8	99.78
128x128	327155	320.0	99.90
256x256	324009	281.6	99.91
512x512	320864	536.0	99.83

5. Conclusion

A cell based panoramic video codec has been designed and developed for high volume video data. It was proven that compression ratio and random access functionality play an important role in the efficient process of the panoramic video used in high volume multi-viewpoint. Furthermore, the proposed scheme reduced the waste of resources by displaying the current view without unnecessary background information. This allows for streaming service with low resource use of the system when the user replays the requested cells. It is also possible to utilize the low bandwidth by transmitting the necessary cells and not the whole image. The proposed scheme introduced the thread based parallel processing method to process multiple cells at the same time and showed that system performance and multiple cells are proportional. In our experiment, we used the MPEG-4 AVC codec and inter prediction coding. Even though a satisfying performance was not achieved due to the long encoding/decoding time characteristic of the MPEG-4 AVC codec, compression capacity and random access functionality showed the likelihood of success. Henceforth, streaming server part must be embodied and there is a need of research in the efficency of the capacity of streaming service.

6. Acknowledgement

This research was supported by the Daegu University Research Grant, 2007.

References

[1] S.E.Chen, "QuickTime VR - An Image-Based Approach to Virtual Environment Navigation", ACM SIGGRAPH, pp. 29-38, August 1995.

[2] Foote, J, Kimber, D., "FlyCam: practical panoramic video and automatic camera control", Multimedia and Expo, Vol. 3, pp.1419-1422, 2000.

[3] Nagase, Y., Yamamoto, T., Kawamura, T., Sugahara, K., "Hardware realization of panoramic camera with speaker-oriented face extraction for teleconferencing", IEEE International Symposium on Circuits and Systems, Vol. 6, pp. 6256-6259, May 2005.

[4] Kar-Han Tan, Hong Hua, Ahuja, N., "Multiview panoramic cameras using mirror pyramids", Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 26, Issue 7, pp. 941-946, July 2004.

[5] Smolic, A,. "Robust generation of 360 degree panoramic views from consumer video sequences", Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom, pp 431-435, June 2002.

[6] Tzavidas, S., Katsaggelos, A.K., "A multicamera setup for generating stereo panoramic video", Multimedia, IEEE Transactions on, Vol. 7, Issue 5, pp. 880-890, Oct. 2005

[7] Li-Shen Juhn,, Li-Ming Tseng., "Harmonic broadcasting for video-on-demand service", Broadcasting, IEEE Transactions on, Vol. 43, Issue 3, pp. 268-271, Sept. 1997.

[8] Grunheit C, Smolic, A, Wiegand T, "Efficient representation and interactive streaming of high-resolution panoramic views", Image Processing. 2002. Proceedings. 2002 International Conference on, Vol. 3, pp. 209-212, June 2002.

[9] King-To Ng, Shing-Chow Chan, Heung-Yeung Shum, "Data compression and transmission aspects of panoramic videos", Circuits and Systems for Video Technology, IEEE Transactions on, Vol. 15, Issue 1, pp. 82-95., Jan. 2005.

[10] Hsiu-Cheng Chang, Chien-Chang Lin, Jiun-In Guo. "A novel low-cost high-performance VLSI architecture for MPEG-4 AVC/H.264 CAVLC decoding", Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, Vol.6, pp. 6110-6113, May 2005.

[11] Immersive Midea, http://www.immersiveme-dia.com.

[12] Hannuksela, M., "Random access using isolated regions", ICIP 2003., Vol. 3, pp. 841-844, Sep. 2003.

[13] Ghina G., Sophia K., "Multimedia Codec Evaluation and Overview", Master program in Electrical Engineering, Blekinge Institute of Technology, Jan. 2007.

[14] Jentz B., "Low-Cost Solutions for Video Compression Systems", Altera Corporation, 2005.

[15] Raja, G, Mirza, M.J, "JVT H.264/AVC: Evaluation with Existing Standards for Low Bit Rate Video Coding", Microelectronics, 2005. ICM 2005, The 17th International Conference on, pp. 301-304, Dec. 2005.

[16] Wai-Kwan Tang, Tien-Tsin Wong, Heng, P., "A System for Real-Time Panorama Generation and

Display in Tele-Immersive Applications", Multimedia, IEEE Transactions on, Vol. 7, Issue 2, pp. 280-292, Apr. 2005.

[17] P. J. Shenoy, P. Goyal, and H. M. Vin, "Issues in multimedia server design", Comput. Surv., vol. 27, no. 4, pp. 636-639, 1995.

[18] Network Working Group, "Real time streaming protocol (RTSP)", Request for Comments 2326, Apr. 1998.

[19] Weerasinghe C., Wanqing Li, Ogunbona, P., "STEREOSCOPIC PANORAMIC VIDEO GENERATION USING CENTRO-CIRCULAR PROJECTION TECHNIQUE", Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on, Vol.3, pp. 473-476, April 2003.

[20] Yo-Sung Ho, Seung-Hwan Kim, H.264 Recommendation Advanced Video Coding, H.264/AVC standard for source code analysis (Korean version), Dooyangsa, p. 253, Seoul, 2006.

[21] De Vito, F, De Martin, J.C, "PSNR control for GOP-level constant quality in H.264 video coding", Signal Processing and Information Technology, pp 612-617, 2005.

◐ 저 자 소 개 ◑

김 영 백(Young Back Kim)
1996년 영남대학교 사회학과 졸업(학사)
2007년 경북대학교 컴퓨터공학과 졸업(석사)
2010년 경북대학교 전자전기컴퓨터학부 수료(박사)
2010 ~ 현재 대구가톨릭대학교 산학협력교수
관심분야 : 유비쿼터스, 임베디드, 모바일, 멀티미디어
E-mail : realtech@hanafos.com

김 태 호(Tae-Ho Kim)
2007년 대구대학교 정보통신공학부 졸업(학사)
2009년 경북대학교 의용생체공학과 졸업(석사)
E-mail : crewx1@nate.com

이 대 규(Dae-Gyu Lee)
2008년 대구대학교 정보통신공학부 졸업(학사)
2009년 ㈜아크로엠 연구원
2010년 ~ 필아이티㈜ 연구원
E-mail : gyu1211@nate.com

김 재 준(jaejoon Kim)
1990년 한양대학교 전자공학과 졸업(학사)
1995년 아이오아주립대학교 전기공학과 졸업(석사)
2000년 아이오아주립대학교 전기공학과 졸업(박사)
2002 ~ 현재 대구대학교 정보통신공학부 부교수
관심분야 : 의료영상처리, JPEG2000 코덱설계, MPEG-21, 패턴인식, 비파괴공학
E-mail : jjkimisu@daegu.ac.kr