A Real-time Rectification using an Adaptive Binary Encoding for High-resolution Video

Jong-hak Kim*, Jung-kyun Oh**, Seong-muk Kang***, Jun-dong Cho***

*Department of IT convergence, Sungkyunkwan University, Korea

**Department of Human ICT convergence, Sungkyunkwan University, Korea

***Department of Electrical and Computer Engineering, Sungkyunkwan University, Korea

jhakkim@vada.skku.ac.kr, jkoh@vada.skku.ac.kr, smkang@vada.skku.ac.kr, jdcho@skku.edu

***Abstract*—Previously, various rectification methods using compressed lookup table have been studied for real-time hardware stereo vision system. These loss compression methods may occur distortion that could corrupt disparity estimation process. Differentially encoded lookup table method which is lossless compression has no distortion with reasonable compression ratio. However, the method is limited to low-resolution and low warping movement. In this paper, we propose an adaptive binary encoding method. Our proposed algorithm has approximately 10~26% compression ratio which is comparable to the previous method and tolerates high-resolution video.**

***Keyword*—real-time processing, high-resolution rectification, lookup-table, adaptive binary encoding, lossless compression**

I. INTRODUCTION

A STEREO vision system has four steps. The first step is rectification as a pre-processing. A role of rectification is to align epipolar lines. And the second step is disparity estimation to obtain disparity-map. In the third step, filtering techniques are utilized to enhance disparity-map. Finally, disparity-map is used for various applications such as visual fatigue decrease, object segmentation and tracking, motion control and etc. Since rectification not only aligns epipolar lines but also decreases search-range for disparity estimation process, it is essential process to achieve real-time processing. Figure 1 shows reduced search-range.

Rectification could categorize into real-time calculation and pre-calculation using a LUT (Lookup table). Since real-time calculation consist a large amount of computation, many research tried to simplify calculation of epipolar geometry matrix as in [2]–[4]. And a hardware architecture using FPGA (Field-programmable gate array) is also used to achieve real-time processing as in [5]–[7]. However, simplified calculation methods are not suitable for real-time processing. And FPGA methods could process in real-time, but they consist of complex hardware architecture.

On the other hand, pre-calculation using a LUT has fast

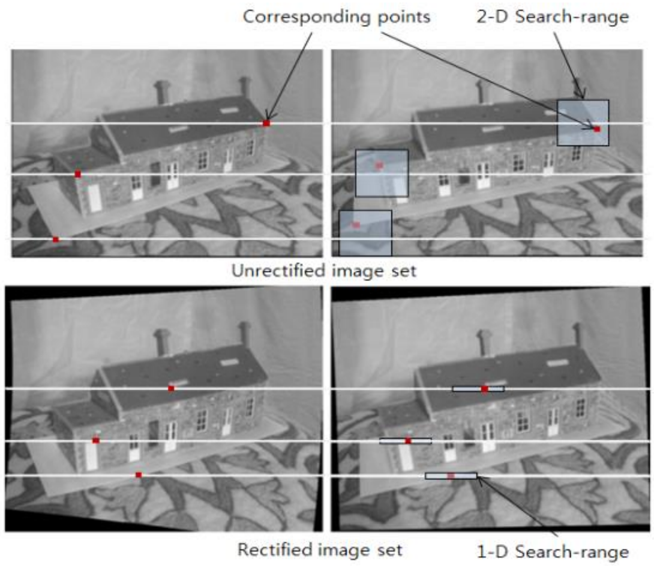

Fig. 1. Example of reduced search-range[1].

processing-time with compact hardware architecture as in [8]. But disadvantage of this method is high memory consumption. To overcome this issue, loss and lossless compression techniques are applied. One of the loss compression methods subsamples rectification mapping data and reconstructs using interpolation as in [9], [10]. Although the loss compression method has much higher compression ratio than the lossless compression method, it also has distortion, and could be critical for the performance of disparity estimation process. To avoid distortion, reference [11] uses differential encoding and decoding using only subtraction and accumulation that consumes only 1.3Mbytes for a 1280 by 720 image with 27% compression ratio. Another recent research using the run-length encoding technique with different order as in [12] shows high performance. However, these methods limit data magnitude bits, so data magnitude have to be less than 3 for differential coding and 1 for run-length coding which are not suitable for high-resolution rectification.

In this paper, we propose an adaptive differential encoding method. We analysed probability of appearance of

Manuscript received September 4, 2014. Followings are results of a study on the "Leaders in Industry-university Cooperation" Project, supported by the Ministry of Education, Science & Technology (MEST). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2013R1A1A2058942)

Jong-hak Kim is with the department of IT convergence, Sungkyunkwan University, Suwon, Gyeonggi-do 440-746, Korea (phone : 031-290-7200, e-mail: jhakkim@vada.skku.ac.kr).

Jung-kyun Oh is with the department of Human ICT convergence, Sungkyunkwan University, Suwon, Gyeonggi-do 440-746, Korea (e-mail: jkoh@vada.skku.ac.kr).

Seong-muk Kang and Jun-dong Cho are with the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Gyeonggi-do 440-746, Korea (e-mail: smkang@vada.skku.ac.kr; jdcho@skku.edu).

differential values and built two codebooks for high performance and resolution including two and four modes. The first mode in the both codebooks is comprised of minimum bit size to decrease compression ratio, and the fourth mode in the codebook with four modes could extent bit size for magnitude to tolerate high-resolution video.

The rest of paper is organized as follows. We will introduce basic concept of rectification using LUT in Section II. Our proposed algorithm will be presented in Section III. Proposed hardware architecture will be presented in Section IV. Experimental result will be given in Section V, and finally, we will draw the conclusion in Section VI.

II. BASIC CONCEPT OF RECTIFICATION USING LUT

Rectification is used for various applications such as ortho-photo or ortho-image as in [13], [14], multi-view camera system as in [6], [15], intelligent vehicle using stereo vision system as in [16], [17] and etc. Difference between stereo vision system and others is that position of each camera is fixed, which needs only one set of rectification parameters if there is no individual movement of each camera and uses it consistently. Therefore, we are allowed to pre-calculate rectification mapping data and rectify images by loading mapping data from LUT memory.

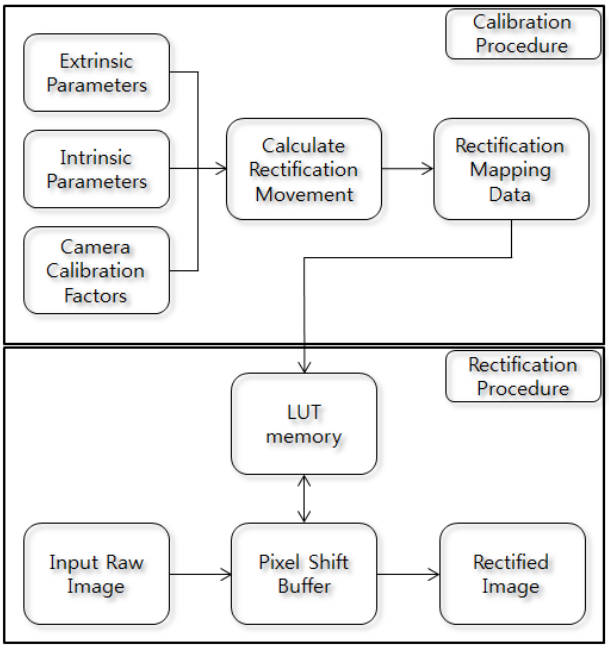

Fig. 2. Overall flow of rectification using LUT

Rectification using LUT could divide into calibration and rectification procedures. The calibration procedure is to calculate rectification movement using intrinsic parameters, extrinsic parameters and camera calibration factors and obtain rectification mapping data. The mapping data includes coordinate information that moves pixels on the original image to the rectified image. This procedure processes in software environment. And the rectification procedure compensates input images using mapping data from the LUT memory. Since complex computation is mostly calculated in the calibration procedure, the rectification procedure could be designed compact hardware architecture and be suitable for real-time processing. However, there are four different mapping data set which are X and Y coordinate information from left and right images. And it affects high memory consumption. Therefore, each mapping data set ought to be compressed. Compressed LUT methods are simply added encoding and decoding parts before and after LUT memory, respectively. And complexity increase depends on which compression method is utilized. Compressed LUT methods classify loss and lossless compression. Figure 2 shows overall flow of rectification using LUT

A. Loss compression

The loss compressed LUT method usually applies down-sampling to reduce a size of LUT in the encoding process and interpolation to fill void pixels in the decoding process as in [9], [10]. Given that interpolation change calculated values based on relationship with adjacent values, a rectified image is distorted. Although rectified image using a loss compressed LUT shows reasonable PSNR over 30 dB, it could occur critical errors in the disparity estimation process. To avoid potential errors, the lossless compression is more suitable.

B. Lossless compression

The lossless compressed LUT method utilizes lossless compression technique such as Huffman, differential, run-length and etc. Among them, a differential encoding method is used with high compression ratio and compact hardware architecture as in [11]. It shows good trade-off between complexity and compression ratio. However, since it limits differential magnitude of sequential pixel values on 3 to minimize memory consumption, an error is occurred on the current coordinate if magnitude of differential value is more than 4. Moreover, when one error is occurred, it is propagated on the current line by accumulation of erroneous difference. Therefore, the previous differential encoding method operates properly only with low warping movement.

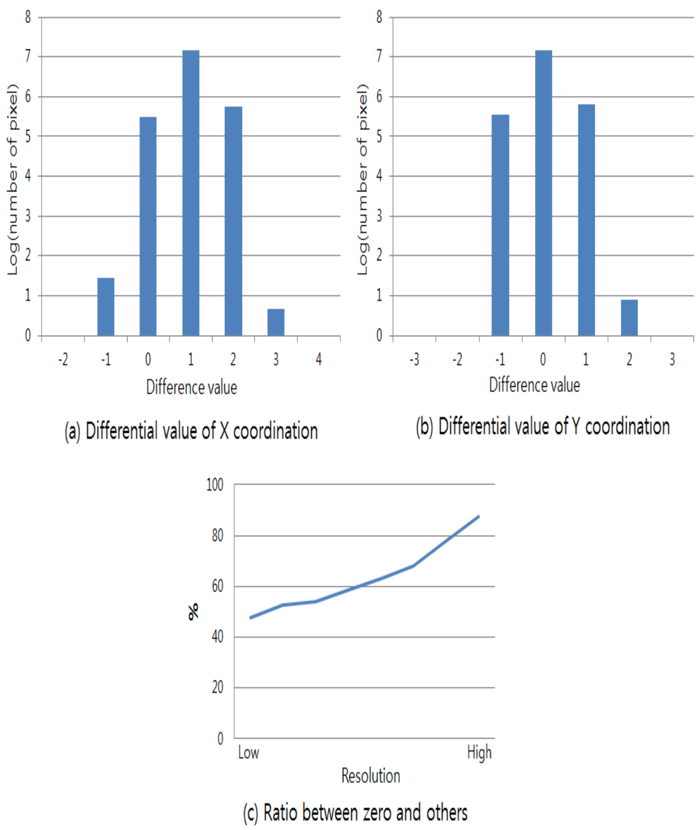

Fig. 3. Distribution of difference values (a) differential values of X coordinate (b) differential values of Y coordinate (c) ratio between the highest probability case and others.

III. Proposed Algorithm

Adaptive compression method basically uses characteristic of input. Reference [18] adjusted sampling frequency depend on frequency of ECG (electrocardiogram) signal. Especially, ECG signal with low frequency sampled with low sampling frequency, which decrease amount of data. In the same manner, our proposed method adjusted size of memory bits to store coordinate data depend on magnitude of differential values. Size of memory bits is minimized to decrease compression ratio for the highest probability case and also maximized to have enough bit size to store maximum magnitude for the opposite case. Therefore, compression ratio is decreased with increasing the highest probability case.

As shown in figure 3(a) and 3(b), variance of differential coordinate values from 50 stereo image sets is distributed around 1, 0 for X and Y coordinate, respectively. Moreover, as shown in figure 3(c), since the ratio between the highest probability case and others increases with higher resolution, our proposed method is more suitable for higher resolution. To satisfy above-mentioned conditions, we composed two codebooks as shown in table 1.

TABLE I
Two Codebooks for Adaptive Binary Encoding

Codebook I		
Range of magnitude	1 for x axis 0 for y axis	-1, 0 ,2, 3 for x axis -2, -1, 1, 2 for y axis
Mode bit	0	1
Sign bit	-	1 bit
Magnitude bit	-	1 bit
Total bit	1 bit	3 bits

Codebook II				
Range of magnitude	0	±1~ ±2	±3~ ±6	±7~
Mode bit	00	01	10	11
Sign bit	-	1 bit	1 bit	1 bit
Magnitude bit	-	1bit	2 bits	3 bits~
Total bit	2 bits	4bits	5 bits	6 bits~

The codebook I is used for small magnitude, which consists of two modes. The first mode is for the highest probability case; 1 for x axis, 0 for y axis. And the second mode encodes rest of cases. The codebook II is used for large magnitude, which consists of four modes. The difference value that has the highest probability case sets as mode '00' to decrease compression ratio. And to cover larger range of magnitude, the codebook could rearrange by expansion of magnitude bit in mode '11'.

Let's assume that we have differential value 0 and -5 in y axis. First, 0 is able to encode with two codebooks. In the codebook I, 0 in the y axis is in the mode '0'. And mode '0' has no sign bit and magnitude bit. Therefore, 0 is encoded as '0'. On the other hands, in the codebook II, 0 is included in mode '00'. And mode '00' is the same as mode '0' in the codebook I. Therefore, 0 is encoded as '00'. Second, -5 cannot be encoded using the codebook I due to large magnitude. However, the codebook II can encode -5 that is included in mode '10'. Sign is minus and magnitude is 5. We can assign mode bit as '10', sign bit as '1', magnitude bit as '10'. Therefore, -5 is encoded as '10110'. Our proposed method chooses a proper codebook based on magnitude. Selected codebook is stored to the codebook table register in the decoder. The encoding flow is shown in figure 4.

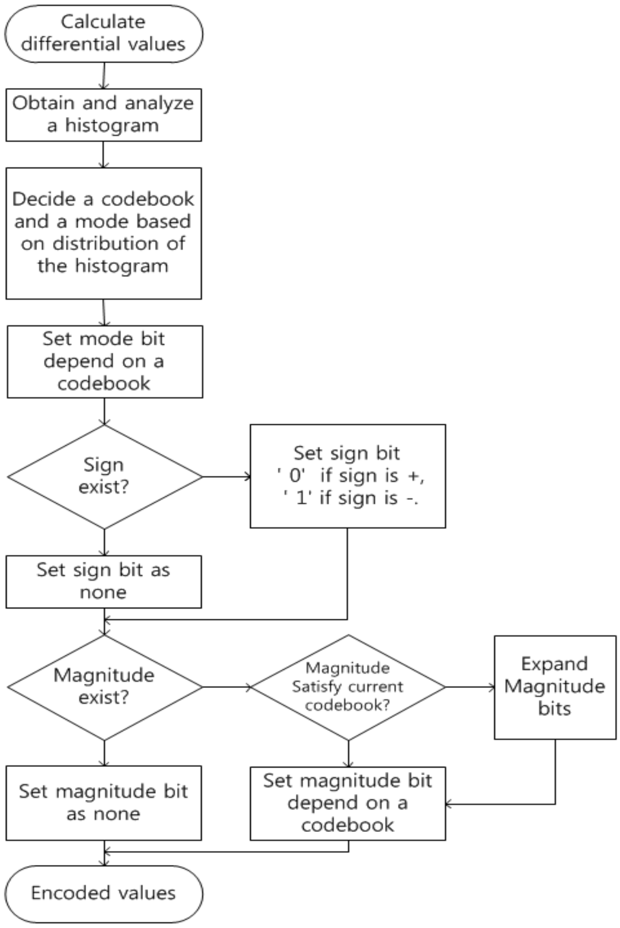

Fig. 4. Encoding flow

IV. Proposed Hardware Architecture

Our proposed hardware architecture consists of LUT memory, a decoder and a pixel address generator. First, the decoder reads differentially encoded data from LUT memory and decode into X and Y coordinate values. Then the pixel address generator reconstruct X and Y coordinate values into pixel address on the original image. The decoder consists of data controller and codebook table registers and the pixel address generator consists of one accumulator for X coordinate, one accumulators and one multiplier for Y coordinate and one adder to combine X and Y coordinate values. Overall hardware architecture is compact with low latency and simple computation, which is suitable for real-time processing.

Decoding flow is processed as followings. First, each data controller obtains the information about which codebook is used and reads encoded data from LUT memory. Second, data is decoded from the binary codes to the differential values using the stored codebook. Lastly, we accumulate differential values to decode into coordinate values, and generate pixel addresses. The proposed hardware architect is shown in fig. 5.

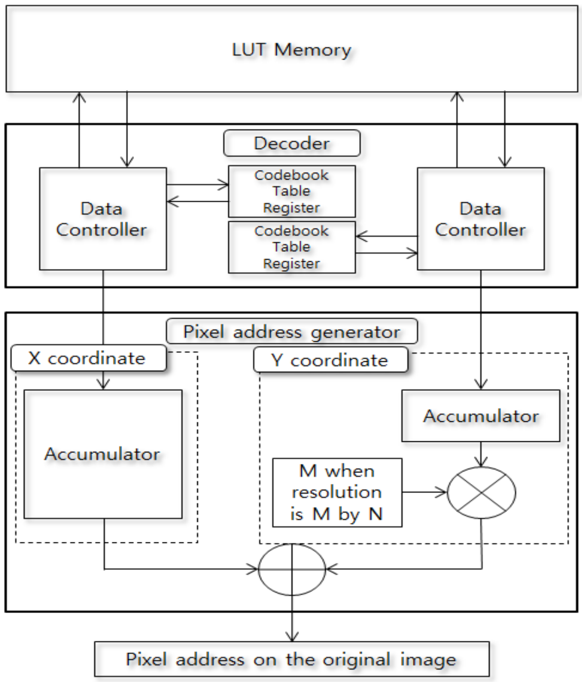

Fig. 5. Proposed hardware architecture

V. Experimental Result

Our proposed method is designed using OpenCV for the encoding process and Verilog for the decoding process. For the evaluation, we applied two methods. The first one is a binary image of difference to show distortion. The equation for binarization is as:

$$B(x,y) = \begin{cases} 0, & Rect_{img}(x,y) = Ori_{img}(x,y), \\ 1, & Rect_{img}(x,y) = Ori_{img}(x,y). \end{cases} \quad (1)$$

Since distortion is the same as difference, distortion regions represent with white pixels in the result images. As shown in figure 6, the result images of the loss compression method have white regions, whereas our proposed method does not.

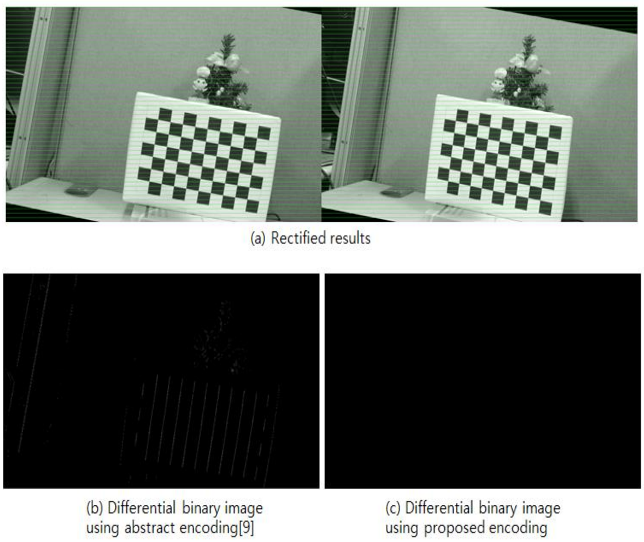

Fig. 6. Differential binary image

The second method is compression ratio. Compression ratio (CR) is defined as:

$$CR = \frac{Compressed\ data}{Uncompressed\ data} \quad (2)$$

We compared to previous differential encoding method as in [11] and two codebooks with various resolution conditions and 50 stereo image sets. As shown in figure 7, compression ratio of our proposed method has minimum 18.9% and maximum 26.1% lower than the previous one. And difference of compression ratio in the same resolution becomes larger with higher resolution.

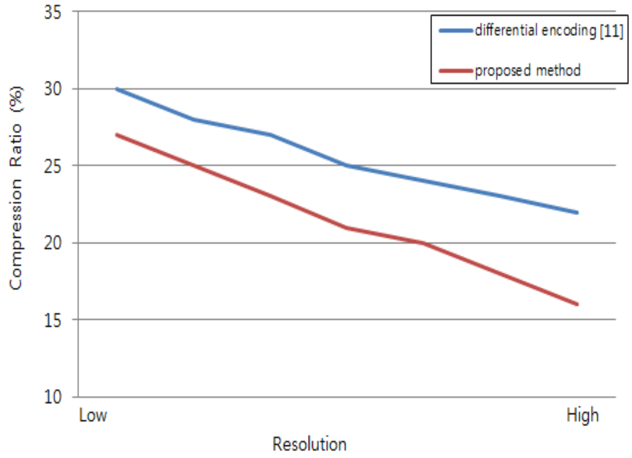

Fig. 7. Compression ratio results with various resolutions

As shown in figure 8, differential encoding shows only 31.69% compression ratio regardless of stereo environment due to fixed encoding bits. On the other hands, our proposed methods show different results as per stereo environment, since it has different encoding bits depend on modes. And for performance, our proposed methods have approximately two-third and one-third lower compression ratio with the codebook II and the codebook I, respectively.

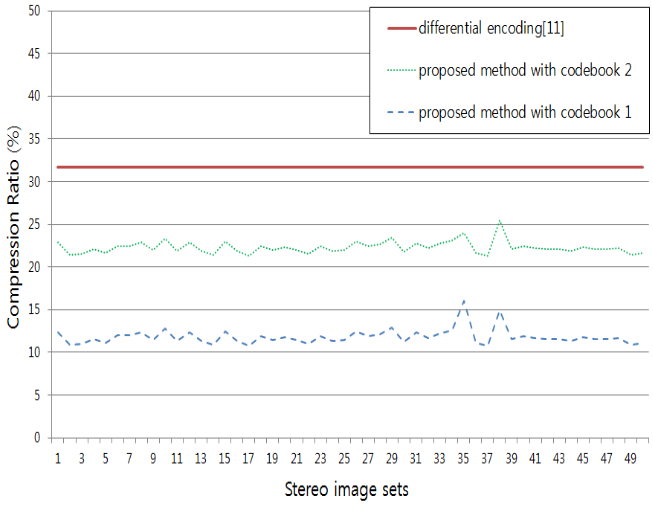

Fig. 8. Compression ratio results with 50 stereo image sets

VI. Conclusion

In this paper, we proposed an adaptive differential encoding method for high-resolution video. Our proposed method effectively encodes satisfying two conditions such as minimization to decrease compression ratio and maximization to tolerate large magnitude values. And decoding process also achieves compact hardware architecture with processing in real-time. Compression ratio is approximately from 10.8% to 26.1% depending on resolution with no distortion. And the higher resolution shows the lower compression ratio which is suitable for image processing with high-resolution.

REFERENCES

[1] Yunseok Lee, Kar-Ann Toh and Sangyoun Lee, "Stereo image rectification based on polar transformation," *Optical Engineering*, 47(8), 087205, 2008.

[2] Nicholas Ayache and Charles Hansen, "Rectification of Images for Binocular and Trinocular Stereovision," *in Proc. Pattern Recognition*, vol. 1, pp. 11-16, Nov, 1988.

[3] Marc Pollefeys, Reinhard Koch and Luc Van Gool, "A Simple and Efficient Rectification method for General Motion," *in Proc. Computer vision*, vol. 1, pp 496-501, Sep, 1999.

[4] Andrea Fusiello, Emanuele Trucco and Alessandro Verri, "A Compact Algorithm for Rectification of Stereo pairs," *Machine Vision and Applications*, Vol. 12, pp. 16-22, 2000.

[5] S. Jin, J. Cho, X. D. Pham, K. M. Lee, S. K. Park, M. Kim and J. W. Jeon, "FPGA Design and Implementation of a Real-time Stereo Vision System," *IEEE Trans. Circuits and Systems for Video technology*, Vol. 20, No. 1, pp. 15-26, Jan, 2010.

[6] Minsu Choi, Jinsang Kim, Won-Kyung Cho and Yunmo Chung, "Low Complexity Image Rectification for Multi-view Video Coding," *in Proc. ISCAS 2012*, pp. 381-384, May, 2012.

[7] Heiko Hubert, Benno Stabernack and Frederik Zilly, "Architecture of a Low Latency Image Rectification Engine for Stereoscopic 3-D HDTV Processing," *IEEE Trans. Circuits and Systems for Video technology*, Vol. 23, No. 5, pp. 813-822, May, 2013.

[8] Cristian Vancea and Sergiu Nedevschi, "LUT-based Image Rectification Module Implemented in FPGA," *in Proc. Intelligent Computer Communication and Processing*, pp. 147-154, Sep, 2007.

[9] Anders Kjaer-Nielsen, Lars Baunegaard With Jensen, Anders Stengaard Sorensen and Norbert Kruger, "A Real-time Embedded System for Stereo Vision Preprocessing using an.FPGA," *in Proc. Reconfigurable Computing and FPGAs*, pp. 37-42, Dec, 2008.

[10] Khurram Jawed, John Morris, Tariq Khan and Georgy Gimel'farb, "Real time Rectification for Stereo Correspondence," *in Proc. Computational Science and Engineering*, pp. 277-284, Aug, 2009.

[11] Deuk Hyun Park, Hyoung Seok Ko, Jae Gon Kim and Jun Dong Cho, "Real Time Rectification using Differentially Encoded Lookup Table," *in Proc. ICUIMC 2011*, No. 47, Feb, 2011.

[12] Abdulkadir Akin, Ipek Baz, Luis Manuel Gaemperle, Alexandre Schmid and Yusuf Leblebici, "Compressed Look-Up-Table based Real-Time Rectification Hardware," *in Proc. VLSI-SoC 2013*, pp. 272-277, Oct, 2013.

[13] Ayman F. Habib, Eui-Myoung Kim and Chang-Jae Kim, "New Methodologies for True Orthophoto Generation," *Photogrammetric Engineering & Remote Sensing*, Vol. 73, No. 1, pp. 25-36, Jan, 2007.

[14] W. Karel, M. Doneus, G. Verhoeven, C. Briese, C. Ressl, N. Pfeifer, "Oriental – Automatic Geo-referencing and Ortho-rectification of Archaeological Aerial Photographs," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Science*, Vol II-5/W1, 2013, XXIV International CIPA Symposium, Sep, 2013.

[15] Yun-Suk Kang, Cheon Lee and Yo-Sung Ho, "An Efficient Rectification Algorithm for Multi-view Images in Parallel Camera Array," *in Proc. 3DTV conference: The True Vision- Capture, Transmission and Display of 3D Video*, pp. 61-64, May, 2008.

[16] Wannes van der Mark and Dariu M. Gavrila, "Real-time Dense Stereo for Intelligent Vehicles," *IEEE Trans. Intelligent transportation systems*, Vol. 7, No. 1, March, 2006.

[17] Anwar Hasni Abu Hasan, Rostam Affendi Hamzah and Mohd Haffiz Johar, "Range Estimation in Disparity Mapping for Navigation of Stereo Vision Autonomous Vehicle using Curve Fitting Tool," *IJVIPNS*, Vol. 9, No. 9, pp. 5~9, Oct, 2009.

[18] Zhongyun Yuan, Jong-hak Kim and Jun-dong Cho, "Adaptive Sampling for ECG Detection based on Compression Dictionary," *JSTS*, Vol. 13, No. 6, DEC, 2013.

Jong-hak Kim received the B.S. degree in radio communication engineering from the Kyunghee University, Suwon, Korea, in 2009, the M.S. degree from the Department of Electrical and Computer Engineering, Sungkyunkwan University, in 2012, and he is studying for a Ph. D degree at Sungkyunkwan University.
He is interested in the efficient low power hardware implementation for a real-time image processing system in mobile equipment. He currently studies a visual fatigue reduction scheme for a stereo vision system.

Jung-kyun Oh received the B.S degree in Information Communication Engineering from Myongji University, Yongin, Korea, in 2009. He is a M.S candidate in the Human ICT Convergence Department at Sungkyunkwan University, Suwon, Korea.
His research interests include SoC for 3D Image Processing Applications and 3D visual fatigue reduction.

Seong-muk Kang received the B.S degree from the Department information communication from Baekseok University, Korea, in 2013. And he is studying for a master's degree in Electrical Engineering at Sungkyunkwan University, Suwon, Korea.
His current research interests include image stabilization and stereo vision.

Jun-Dong Cho received the B.S. degree from the Department of Electronic Engineering, Sungkyunkwan University, Suwon, Korea, in 1980, the M.S. degree from the Department of Computer Science, Polytechnic University Brooklyn, New York, in 1989, and the Ph.D. degree from the Department of Computer Science, Northwestern University, Evanston, in 1993.
He was working for Samsung Electronics Company for 6 years. He joined the Department of Electrical and Computer Engineering, Sungkyunkwan University (SKKU), Suwon, Korea, since 1995, where he is currently chair of graduate school of Human ICT Convergence.
His research interests include Low Power Design, 3-D Image Processor, Embedded Multimedia, and Human ICT Application.
Prof. Cho is an IEEE Senior Member.