Journal of Multimedia Information System VOL. 5, NO. 2, June 2018 (pp. 131-138): ISSN 2383-7632(Online)
http://dx.doi.org/10.9717/JMIS.2018.5.2.131

Binary Image Based Fast DoG Filter Using Zero-Dimensional Convolution and State Machine LUTs

Seung-Jun Lee[1*], Kye-Shin Lee[1], and Byung-Gyu Kim[2]

Abstract

This work describes a binary image based fast Difference of Gaussian (DoG) filter using zero-dimensional (0-d) convolution and state machine look up tables (LUTs) for image and video stitching hardware platforms. The proposed approach for using binary images to obtain DoG filtering can significantly reduce the data size compared to conventional gray scale based DoG filters, yet binary images still preserve the key features of the image such as contours, edges, and corners. Furthermore, the binary image based DoG filtering can be realized with zero-dimensional convolution and state machine LUTs which eliminates the major portion of the adder and multiplier blocks that are generally used in conventional DoG filter hardware engines. This enables fast computation time along with the data size reduction which can lead to compact and low power image and video stitching hardware blocks. The proposed DoG filter using binary images has been implemented with a FPGA (Altera DE2-115), and the results have been verified.

Key Words: Binary Image, DoG Filter, Look-up Tables (LUTs), Zero Dimensional Convolution.

I. INTRODUCTION

Due to the advance in image processing systems and computer vision technology, there is an increasing demand for efficiently combining still images or video streams that are captured from different cameras in order to generate panoramic images. This procedure for combining multiple images into one single image is called image stitching [1]. The applications for image stitching range from driver assistance for smart cars, 3D mapping, defense/weapon systems, medical imaging, object or scene recognition, motion tracking, forensics and law enforcement to gaming and virtual reality – any vision system with multiple cameras that requires image processing and analysis [2] – [4].

Generally, image stitching algorithms can be classified into two categories – the pixel based [6] and the feature based [7] approach. The pixel based approach, also called the direct method shifts or warps the images relative to each other one pixel at a time, and finds the best alignment between the two images. The feature based approach consists of feature extraction, feature matching, and image blending steps, where distinctive features are first extracted from each image, and the common feature set that are included in both images will be identified. In addition, once a set of feature correspondences has been extracted, a feature set that will produce a highly accurate image alignment will be computed. The most widely used and efficient feature based image stitching algorithm is the scale invariant feature transform (SIFT) where feature extraction and matching can be realized with invariant to scale, rotation, and translation as well as partially invariant to affine distortion and illumination changes [8].

So far, image stitching has been mainly realized using software algorithms. However, although implementing image stitching algorithms with PCs are easy, the software based approach is slow, which critically limits the real time image data processing, i.e a general software image stitching algorithms can take more than 30-minutes for stitching a 1-minute video stream. Furthermore, as the image size increases and the image quality improves, software algorithms require humongous computing power, which makes the software based image stitching extremely inefficient and slow. Alternatively, hardware based image stitching schemes realized with field programmable arrays (FPGAs) or application specific integrated circuits (ASICs) can be faster than the software based approach, thus it is feasible to process real time image data. In addition, owing to improved VLSI technologies, the cost of the hardware based approach can be significantly lower than the software based image stitching schemes.

Manuscript received March 13, 2018; Revised April 27, 2018 ; Accepted May 10, 2018. (ID No. KMMS-2018-0025)
Corresponding Author (*): Seung-Jun Lee, ASEC260, Akron OH 44304, 01-330-696-2788, sl50@zips.uakron.edu
[1]Dept. of Electrical and Computer Engineering, The University of Akron, Akron, Ohio, USA, klee3@uakron.edu
[2]Dept. of IT Engineering, Sookmyung Women's University, Seoul, Korea, bg.kim@sm.ac.kr

Especially, FPGAs are ideal for realizing different types of imaging hardware due to their programmability, compact size, and low power consumption [5]. However, even though hardware based image stitching is promising, simply using the existing image stitching algorithms to implement the hardware will not work, since this will end up with huge circuit blocks. As a result, developing a compact, fast, and low power hardware based algorithms and circuit blocks for FPGAs still remains a challenge that requires intense research efforts.

In this work, a FPGA based DoG filter using binary images that can be applied for realizing fast image stitching hardware is proposed. The proposed DoG filter is based on the zero-dimensional convolution using state machine LUTs, which can significantly improve the computation time compared to existing image stitching hardware.

II. FEATURE EXTRACTION USING BINARY IMAGES

DoG is a widely used filtering for image stitching, however it is not suitable to directly implement this with hardware due to complicated processing steps and humongous computations. In this work, to overcome this limitation, binary images that are composed of only two colors – black and white will be used. Although binary images are simple representation of the original images, they still preserve the global features of the original image such as contours, corners, edges, and lines which will make the image stitching algorithms to properly work.

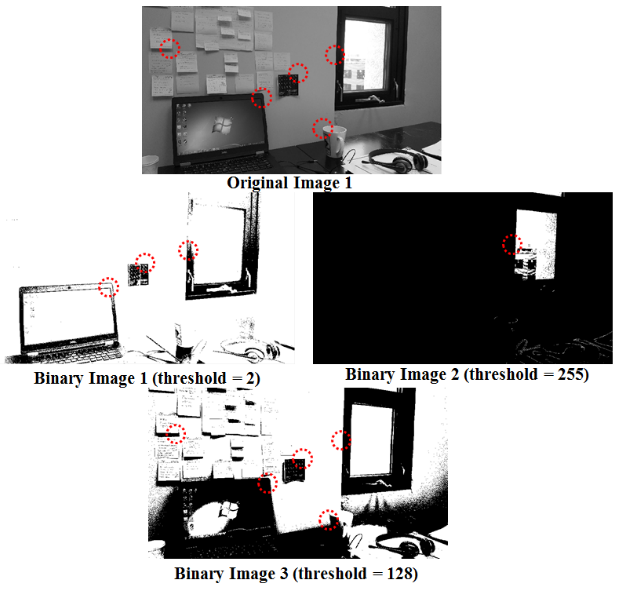

Original Image 1
Binary Image 1 (threshold = 2)
Binary Image 2 (threshold = 255)
Binary Image 3 (threshold = 128)

Fig 1. Binary Images with Different Threshold Levels

Figure 1 shows the gray scale (8 bit) input image is converted into binary images. The binary image can be obtained by comparing each pixel with a threshold level – usually the mid-level of the minimum and maximum pixel value, and representing the pixel value with either 255 (pixel intensity greater than threshold) or 0 (pixel intensity less than threshold), where 255 and 0 stands for the white and black level, respectively. As shown in Figure 1, if the threshold is too low (th=2), it eliminates the blob like objects (notepads on the wall). If the threshold is too high (th=255), a large number of essential feature points are eliminated. In this case, the threshold, 128 is suitable for our application since it still preserves useful edges and corner information. The dotted red circles indicate feature points of the image, where threshold levels can change the feature point information. For the threshold level of 2 and 255, only three and one feature points are preserved from the original image, respectively. For the threshold level of 128, all the feature points in the original image are preserved. In addition, by using binary images, the amount of data processed in the DoG filtering will be significantly reduced, which will enable a very fast computation for image stitching.

III. DOG HARDWARE IMPLEMENTATION

The proposed hardware implementation consists of three main modules: 1) binary image conversion, 2) zero-dimensional convolution with state machine, 3) LUTs.

3.1. Gaussian Filter

The basic concept of extracting features, such as corner and edge detection are based on the identifying the rapid changes in pixel intensity. It can be computed by taking the derivative of the image: the difference between the previous pixel's intensity and current pixel's intensity

$$\frac{df}{dx} = lim_{\Delta x->0}\frac{f(x)-f(x-\Delta x)}{\Delta x} \qquad \textbf{(1)}$$

where Δx is always 1 since it is discrete pixel. As shown in Figure 2, the edge (3 by 3 window on the right) can be found where the intensity changes only along horizontally and the corner (3 by 3 window on the left) can be found where the intensity changes both along horizontally and vertically. In both cases, the derivative will result in 1 or -1 and such results are detected as feature points.

However, the problem with the derivatives is that it also amplifies the noises and accentuates high frequencies, thus the Gaussian Filtering (convolution with Gaussian Kernel: k by k matrix) has to be applied, which removes the noise by smoothing the image, similar to a low pass filter that convolves with its impulse response. However, for the SIFT algorithm, the computation time of the Gaussian Filter convolution module, consumes more than 70% of the entire computation time due to the heavy multiplications and additions [9].

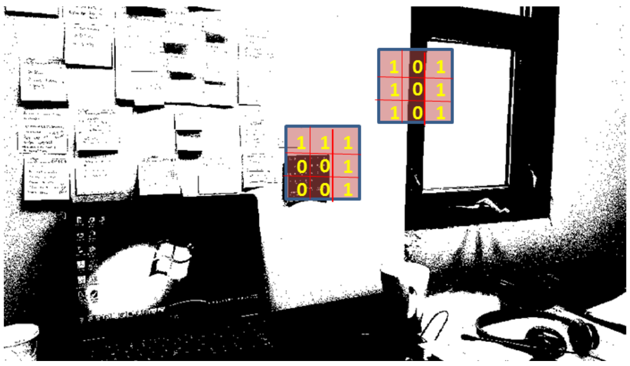


Fig 2. Binary Image Intensity Matrix

If the Gaussian Filter is realized with a FPGA, the computation is proportional to W^2×K^2 where W is the image of width and K is the Gaussian kernel matrix dimension. However, since the Gaussian kernel is linear and symmetric, it can be separated into two 1-d Gaussian Kernel (vertical and horizontal):

$$\begin{bmatrix} \multicolumn{3}{c}{\text{Covoution}} \\ \multicolumn{3}{c}{\text{Kernel}} \\ G1 & G2 & G3 \\ G4 & G5 & G6. \\ G7 & G8 & G9 \end{bmatrix} = \begin{bmatrix} \text{Vertical} \\ \text{Kernel} \\ G1 \\ G4 \\ G7 \end{bmatrix} \cdot \begin{bmatrix} \multicolumn{3}{c}{\text{Horizontal}} \\ \multicolumn{3}{c}{\textit{Kernel}} \\ G1 & G2 & G3 \end{bmatrix}. \quad \textbf{(2)}$$

This reduces computation time to $2 \times W \times K^2$.

After the Gaussian Filter (low-pass filter) is applied for blurring the noises, DoG filter should be used for enhancing the feature point of the blurred images. DoG is an approximation of LoG (Laplace of Gaussian) [1]. The DoG is given by:

$$\begin{aligned} DoG &= (h_{\sigma 1} - h_{\sigma 2}) * f(x,y) \\ &= G_{\sigma 1}(x,y,k\sigma) - G_{\sigma 2}(x,y,\sigma) \end{aligned} \quad \textbf{(3)}$$

where σ is the standard deviation of Gaussian distribution, k is the constant multiplicative factor and $f(x,y)$ is binary input images. The Gaussian Kernel h_σ is the discretized K by K matrix from continuous Gaussian function [12]:

$$h(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \textbf{(4)}$$

The input image is repeatedly convolved with a multiple scale (k=1, 2, 3, n) Gaussian Kernel, and the difference produces the DoG; a subtraction of blurred image from a more blurred image.

The conventional convolution of Gaussian Kernel can be computed by sliding the two dimensional Kernel matrix through the input images as shown in Figure 3. In this method, the horizontal and vertical convolution can be computed in parallel in a FPGA by utilizing the multiple shift registers so that multiplication and addition are computed concurrently. The tradeoff is that it requires more adders and multipliers.

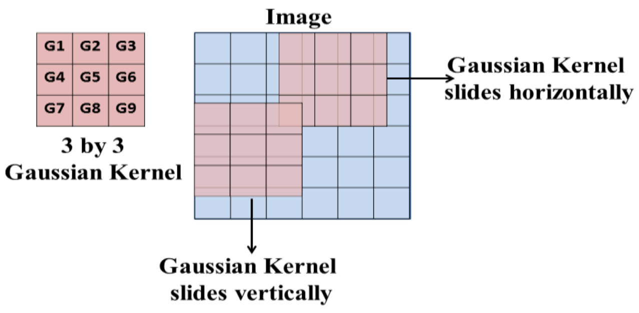
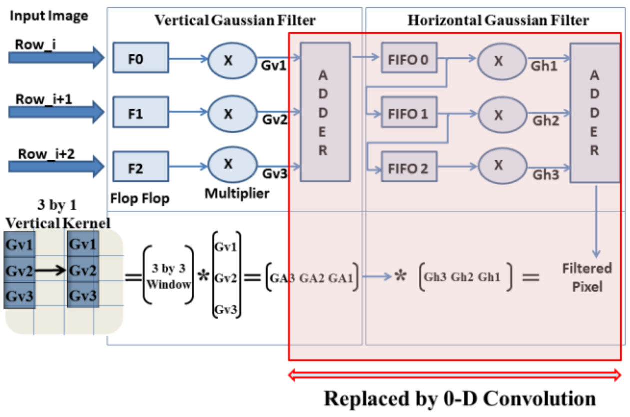


Image
G1 G2 G3
G4 G5 G6
G7 G8 G9
3 by 3 Gaussian Kernel
Gaussian Kernel slides horizontally
Gaussian Kernel slides vertically
Fig. 3. 2-d Gaussian Filter



(a)

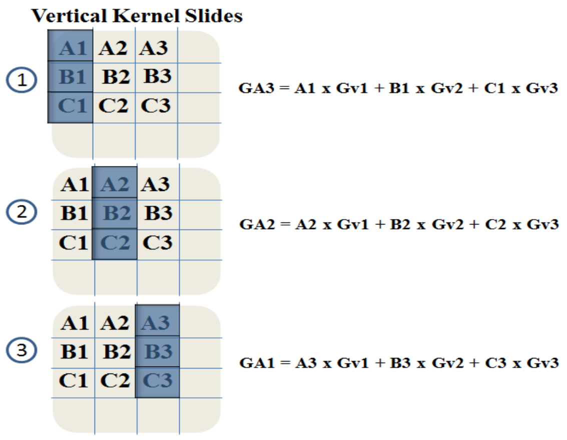

Vertical Kernel Slides
1
A1 A2 A3
B1 B2 B3
C1 C2 C3
GA3 = A1 x Gv1 + B1 x Gv2 + C1 x Gv3
2
GA2 = A2 x Gv1 + B2 x Gv2 + C2 x Gv3
3
GA1 = A3 x Gv1 + B3 x Gv2 + C3 x Gv3
(b)

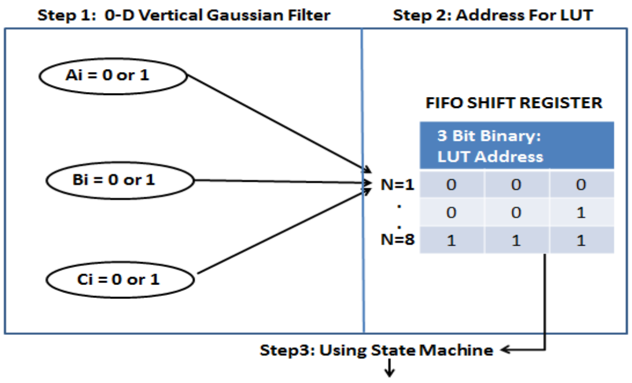
Fig. 4. (a) Two 1-d Gaussian Filter, (b) Vertical Gaussian Filter Operation

Another conventional method is using two 1-d convolutions as shown in Figure 4. The vertical Gaussian Filter can be obtained by sliding the 3 by 1 vertical kernel. The incoming pixel of input image from memory is stored in the flip flop first, then each pixel is convolved with the vertical Gaussian kernel [Gv1 Gv2 Gv3] concurrently in the FPGA by the multiplier and the adder. The output pixel from the adder, GA3 is computed when the vertical kernel slide is on the first position, where the three vertical pixels are multiplied by the vertical Gaussian kernel [Gv1 Gv2 Gv3]. The GA2 and GA1 are obtained from the same procedure, but with the different kernel position. Once [GA3 GA2 GA1] are stored in the FIFO in sequence, they will be convolved with the horizontal Gaussian kernel [Gh1 Gh2 Gh3], which produces the filtered output. The problem with this method is that the column Gaussian filter has to wait until [GA3 GA2 GA1] arrive in the FIFO after convolving with the vertical Gaussian Filter. This scheme still requires multipliers and adders. The drawbacks of this architecture can be eliminated by the proposed 0-d Gaussian filter with state machine and LUTs.

3.2 Proposed: 0-d Convolution with State Machine and LUT

There are three steps in the proposed 0-d Gaussian Filter as shown in Figure 5. The first step detects the binary pixel data Ai, Bi, Ci, which is 0 or 1. The second step stores the three binary pixel data in the FIFO shift register and the stored three bit data becomes the address of LUT. The third step uses the State Machine to retrieve the pre-calculated Gaussian Filtered pixel output. The Gaussian Kernel values used in LUT for the pre-calculation are $[Gv1\ Gv2\ Gv3] = \frac{1}{16} \times [1\ 2\ 1]$ and $[Gh1\ Gh2\ Gh3] = \frac{1}{16} \times [1\ 2\ 1]$. This generates the same results as the conventional Gaussian Kernel. However, it can achieve faster computation using binary image and LUT with 0-d convolution. In addition, the power consumption can be reduced by eliminating huge number of multipliers and adders. Since our binary input image has only one bit, either 0 or 1, the filtered pixel value after the vertical Gaussian Filter is same as the value of the Gaussian Kernel (if pixel data = 1) or 0 (if pixel data = 0). Thus, there is no multiplier required for the vertical convolution. Once we obtained the 1 by 3 matrix, such as $[GA3\quad GA2\quad GA1]$ as shown in Figure 4, then it can be convolved with the 3 by 1 horizontal Gaussian Kernel $[Gh1\quad Gh2\quad Gh3]$.

However, since the horizontal Gaussian kernel values are always constant and all the possible combination of multiplication for the vertical and the horizontal Gaussian kernels are known, the horizontal convolution is not required during the process. Fortunately, the possible combination of 1 by 3 matrix from the vertical Gaussian Kernel is N=8, and since the horizontal kernel values are always constant, the possible combination of final output will be still 8.



N	Address	Gi
1	000	0
2	001	Gv3xGhi
3	010	Gv2xGhi
4	011	Gv2xGhi+Gv3xGhi
5	100	Gv1xGhi
6	101	Gv1xGhi+Gv3xGhi
7	110	Gv1xGhi+Gv2xGhi
8	111	Gv1xGhi+Gv2xGhi+Gv3xGhi

LUT: Pre-Calculated Horizontal Gaussian Filtered Output

Fig. 5. 0-d Gaussian Filter

The possible combination can be known in FIFO when three binary pixel data (array of 0 or 1) from flip flop arrive in sequence. At this point, the final Gaussian filtered output can be known before the convolution with horizontal kernel. Thus, the conventional horizontal Gaussian filter can be removed from Figure 4(a). After 3 vertical Gaussian convolution sum data [GA3 GA2 GA1] are stored in the FIFO, the state machine uses these values as an address of the LUT (ie. 101). The multiplication of vertical Gaussian Kernel and horizontal Kernel values are stored in the LUT for each N, and the summation of each row in N is the pre-computed convolution of the data. Once the pre-computed convolution value Gi is retrieved from the LUT, the final Gaussian filtered output of the pixel is generated by summing the three consecutive Gi (for i=1,2,3).

IV. RESULTS

This work optimizes the hardware architecture of DoG algorithm by reducing the data size (using binary image input) and computability (using state machine and LUTs). The computability is reduced based on the big O analysis. The computation of matrix multiplication is O(n^3) and linear search for LUT is O(n). The proposed DoG hardware is realized using FPGA Altera DE2-115 board by storing incoming pixel in the SRAM, M9K blocks (embedded memory block in FPGA: 9k+ bits per block and support for shift registers, FIFO buffer and LUTs), DMA (Direct Memory Access) for write and read from the SRAM memory directly, 8-bit FIFO shift register for convolution, state machine, built-in peripheral IP module for SRAM Controller and VGA Interface, and NIOSII CPU for controlling the process and interface between our implemented DoG algorithm and memory.

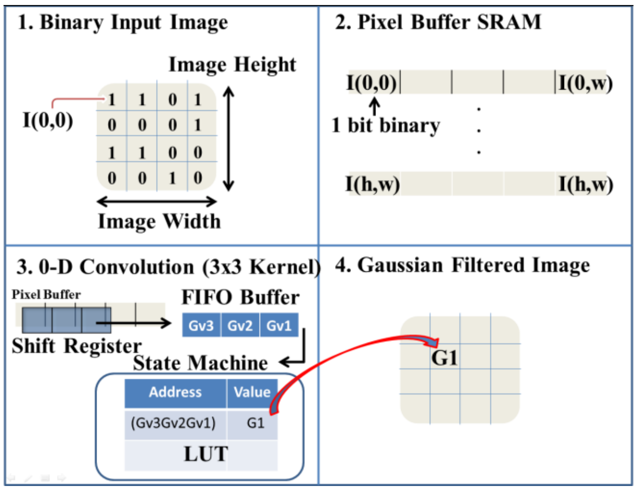


Fig. 6. The Hardware Implementation Overview

Figure 6 shows the hardware implementation logic overview. In the first stage, the input image is converted to a one bit binary format (0 or 1) either from RGB or greyscale. The greyscale image (generally 8-bits) is converted to binary image with pixels either with 255 (white) or 0 (black). The threshold should be between 0 and 255. Later, 255 is replaced with a 1 for easy processing. The threshold value of 128 was selected for binary image in our hardware implementation. In the second stage, each one bit pixel of the binary image is stored in the FPGA SRAM. The size of SRAM is determining by the height and width of the image. In the third stage, the convolution is computed with 3 by 3 Gaussian Kernel. Our proposed 0-d convolution method uses the concept of two 1-d convolution, but the vertical and the horizontal Gaussian Kernel are replaced by 0-d convolution. Taking advantage of the binary format, the computation steps are reduced compared to the conventional 2-d convolution. For every cycle, three adjacent input pixels (0 or 1) are stored in the FIFO, and the state machine will use the three input binary pixel (0 or 1) as the LUT address to retrieve the pre-calculated convolution value. In the last stage, the output from stage 3 is Gaussian filtered and stored in the pixel buffer. The difference between the Gaussian modules (with different Gaussian Kernel) is the DoG. Finally, the output images are displayed via the VGA interface in FPGA.

Figure 7 shows two 0-d filtered binary image with σ =1.4, where the right image is more blurred image with multiplying constant factor k, 2. Figure 8 shows the DoG filter results, which is the subtraction of two 0-d Gaussian Filter results. As shown, most of the features contained in the original images are reserved.

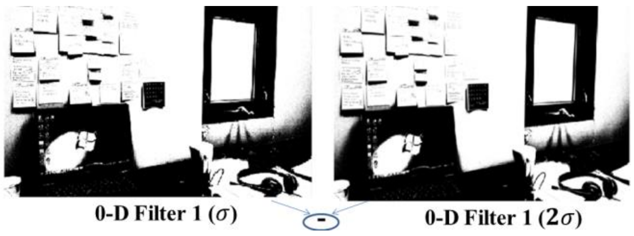


Fig. 7. Gaussian Filter Results

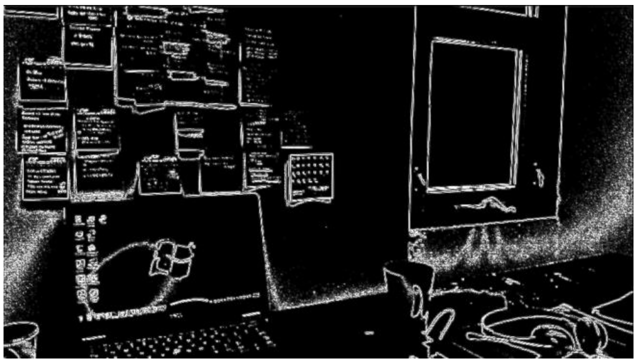


Fig. 8. DoG Filter Results

V. CONCLUSION

A binary image based fast Difference of Gaussian (DoG) filter using zero-dimensional (0-d) convolution and state machine look up tables (LUTs) has been implemented with a FPGA (Altera DE2-115), and the results have been verified. With the proposed approach for using binary images, the data size is significantly reduced compared to conventional gray scale based DoG filters, yet binary images still preserved the key features of the image such as contours, edges, and corners. Also, the proposed design eliminated the major portion of the adder and multiplier blocks that are generally used in conventional DoG filter hardware engines. Furthermore, this enabled fast computation time along with the data size reduction which led to compact and low power image and video stitching hardware blocks.

REFERENCES

[1] R. Szeliski, “Image alignment and stitching: a tutorial,”

Found. Trends. Comput. Graph. Vis., vol. 2, pp. 1-104, 2006.

[2] S. Chen, "QuickTime VR – An image based approach to virtual environmental navigation," SIGGRAPH 95, vol. 29, pp. 29-38, 1995.

[3] R. Szeliski and H. Shum, "Creating full view panoramic image mosaics and environmental maps," SIGGRAPH 97, vol. 31, pp. 251-258, 1999.

[4] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," Int. J. Computer Vision, vol. 74, pp. 59-73, 2007

[5] M. Alawad and M. Lin, "Stochastic-based deep convolutional networks with reconfigurable logic fabric," IEEE Trans. Multi-Scale Comp. Syst., vol. 2, pp. 242-256, 2016.

[6] J. Kim, "Visual correspondence using energy minimization and mutual information," Int. conf. on Comput. Vision (ICCV 2003), Oct. 2003, pp. 1033-1040.

[7] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. Pattern Analysis and Mac. Intell., vol. 27, pp. 1615-1630, 2005.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," vol. 60, pp. 91-110, 2004.

[9] S. A. Bakar, M. S. Hitam and W. N. Yussof, "Content-based image retrieval using SIFT for binary and grey scale images," IEEE ICSIPA 2013, pp. 83-88.

[10] C. Lyu, J. Peng, W. Zhou, S. Yang and Y. Liu, Fellow, "Design of a High Speed 360-degree Panoramic Video Acquisition System Based on FPGA and USB 3.0",IEEE Sensor 2011, pp 3.

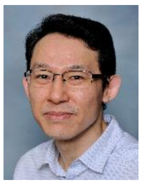
[11] Altera TriMatrix Embedded Memory Blocks in Stratix IV Devices https://www.altera.com/en_US/pdfs/literature/hb/stratix-iv/stx4_siv51003.pdf, 2011.

[12] R. Szeliski, Computer Vision: Algorithms and Applications, Draft 2010, pp. 450

[13] T.S. Fleming and B.D. Thomas, "Hardware Acceleration of Matrix Multiplication over Small Prime Finite Fields", Springer-Verlog Berlin Heidelberg 2011, pp 111.

Authors

Seung-Jun Lee received the B.S. degree from The University of Akron, Akron, OH, in electrical engineering in 2018. Since 2016 he has been an undergraduate research assistant in the Department of Electrical and Computer Engineering. His research interests include computer vision and image processing applications.



Kye-Shin Lee(S'02–M'06) received the B.S. degree from Korea University, Seoul, Korea, in 1992, the M.S. degree from Texas A&M University, College Station, TX, USA, in 2002, and the Ph.D. degree from the University of Texas at Dallas, Richardson, TX, USA, in 2005, all in electrical engineering. He was with Texas Instruments Inc., Dallas, TX, USA, from 2005 to 2008. In 2009, he was an Assistant Professor with the Department of Electronics, Sun Moon University, Asan-si, Chungnam, Korea. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, The University of Akron, Akron, OH, USA. His research interests include integrated circuits for sensors and image processing applications.

Byung-Gyu Kim has received his BS degree from Pusan National University, Korea, in 1996 and an MS degree from Korea Advanced Institute of Science and Technology (KAIST) in 1998. In 2004, he received a PhD degree in the Department of Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST). In March 2004, he joined in the real-time multimedia research team at the Electronics and Telecommunications Research Institute (ETRI), Korea where he was a senior researcher. In ETRI, he developed so many real-time video signal processing algorithms and patents and received the Best Paper Award in 2007. From February 2009 to February 2016, he was associate professor in the Division of Computer Science and Engineering at SunMoon University, Korea. In March 2016, he joined the Department of Information Technology (IT) Engineering at Sookmyung Women's University, Korea where he is currently an associate professor. In 2007, he served as an editorial board member of the International Journal of Soft Computing, Recent Patents on Signal Processing, Research Journal of Information Technology, Journal of Convergence Information Technology, and Journal of Engineering and Applied Sciences. Also, he is serving as an associate editor of Circuits, Systems and Signal Processing (Springer), The Journal of Supercomputing (Springer), The Journal of Real-Time Image Processing (Springer), and International Journal of Image Processing and Visual Communication (IJIPVC). From March 2018, he is serving as the Editor-in-Chief of The Journal of Multimedia Information System and associate editor of IEEE Access Journal. He also served or serves as

Organizing Committee of CSIP 2011, a Co-organizer of CICCAT2016/2017, and Program Committee Members of many international conferences. He has received the Special Merit Award for Outstanding Paper from the IEEE Consumer Electronics Society, at IEEE ICCE 2012, Certification Appreciation Award from the SPIE Optical Engineering in 2013, and the Best Academic Award from the CIS in 2014. He has been honored as an IEEE Senior member in 2015. He is serving as a professional reviewer in many academic journals including IEEE, ACM, Elsevier, Springer, Oxford, SPIE, IET, MDPI, and so on. He has published over 200 international journal and conference papers, patents in his field. His research interests include image and video signal processing for the content-based image coding, video coding techniques, 3D video signal processing, deep/reinforcement learning algorithm, embedded multimedia system, and intelligent information system for image signal processing. He is a senior member of IEEE and a professional member of ACM, and IEICE.