Indonesian Journal of Electrical Engineering and Computer Science
Vol. 19, No. 2, August 2020, pp. 1021~1027
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v19.i2.pp1021-1027

Depth image correction for intel realsense depth camera

Hyun Jun Park[1], Kwang Baek Kim[2]
[1]Division of Software Convergence, Cheongju University, Korea
[2]Department of Computer Engineering, Silla University, Korea

Article Info

Article history:

Received Jan 4, 2020
Revised Mar 2, 2020
Accepted Mar 16, 2020

Keywords:

Computer vision
Depth camera
Depth image correction
Image processing
Intel realsense

ABSTRACT

Intel RealSense depth camera provides depth image using infrared projector and infrared camera. Using infrared radiation makes it possible to measure the depth with high accuracy, but the shadow of infrared radiation makes depth unmeasured regions. Intel RealSense SDK provides a postprocessing algorithm to correct it. However, this algorithm is not enough to be used and needs to be improved. Therefore, we propose a method to correct the depth image using image processing techniques. The proposed method corrects the depth using the adjacent depth information. Experimental results showed that the proposed method corrects the depth image more accurately than the Intel RealSense SDK.

Corresponding Author:

Kwang Baek Kim,
Division of Computer and Information Engineering,
Silla University,
Busan 46958, Korea.
Email: gbkim@silla.ac.kr

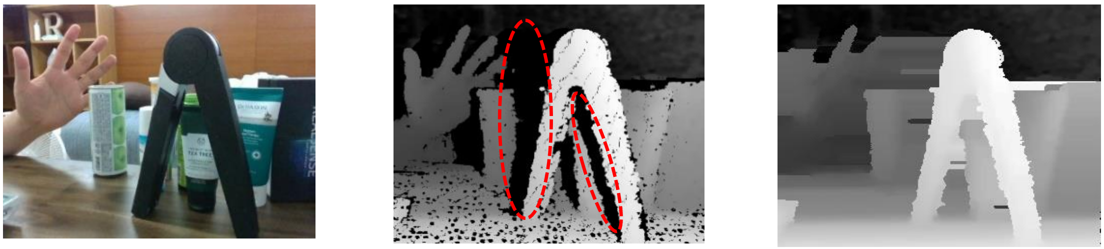
1. INTRODUCTION

The field of computer vision has grown rapidly with the development of deep learning techniques. It has reached the level that can recognize various objects from the image with high accuracy [1-6]. However, it is still hard to measuring the distance between the camera and objects. The conventional method to solve this problem is to use equipment such as stereo vision or laser sensors. Stereo vision has a problem with accuracy because it recognizes each object from images acquired by two cameras and measures the distance between the objects and the camera by mathematical calculations [7]. On the other hand, laser sensors have high accuracy, but these are expensive and can measure the distance only for a preset narrow region [8].

In recent years, the distance is measured using a depth camera (or 3D camera) [9-11]. Starting with Kinect [12], depth cameras became popular. Kinect, a kind of depth camera, is a motion detection sensor for X-box game consoles released by Microsoft. Currently, a variety of depth cameras have been developed and are on the market, and Intel RealSense depth cameras are used in many research fields including AR [13], VR [14], robotics [15], and drones [16] because of the advantage of small size and lightweight. Also, it can work with OpenCV which is the most popular computer vision open library and is available for a reasonable price.

Intel RealSense depth camera provides not only RGB color image but also depth information by recognizing infrared dots using infrared projector and infrared camera [17-19]. Using infrared radiation (IR) makes it possible to measure the distance with high accuracy but it has a disadvantage that the shadow region of IR appears as black holes in a depth image. Because when the shadow of IR occurs, cameras can not measure the depth of the shadow region. Intel RealSense SDK (Software Development Kit) [20] provides a postprocessing algorithm that fills the black holes (shadow regions), but this algorithm is so simple that it needs to be improved.

Figure 1 shows an example of the postprocessing of the Intel RealSense SDK. Figure 1 (a) is an RGB color image, and Figure 1 (b) is an original depth image. Black pixels in Figure 1 (b) mean where depth information is not measured by IR shadows region. Figure 1 (c) is a corrected depth image generated by Intel RealSense SDK postprocessing. The postprocessing provides inaccurate correction because it simply copies the left depth value of the IR shadows region to depth unmeasured region, so the shape of hand is almost disappeared on the left side of the depth image.



(a) RGB image (b) Depth image (c) Postprocessing result

Figure 1. Postprocessing example of Intel RealSense SDK

Therefore, this paper proposes a new depth correction method to improve the postprocessing. The proposed method finds the same object of the depth unmeasured region and estimates the new depth using the the depth information of the same object. It corrects depth values of the IR shadow region more accurately.

2. DEPTH IMAGE CORRECTION METHOD

The postprocessing of Intel RealSense SDK uses only adjacent pixels in the depth image for estimating the depths of the IR shadow region. However, since it is already lost the depth information at the IR shadow region in the depth image, the accuracy of the depth estimation cannot be high. Therefore, in this paper, the proposed method uses not only depth image but also RGB image to estimate the depth of IR shadow region and makes the following assumptions.

Assumption 1. Pixels having similar colors within a given small region correspond to the same object.
Assumption 2. The depth of a pixel is similar to the depth of an adjacent same object.

According to the Assumption 1, it can determine that which object is containing a DUP (Depth Unmeasured Pixel), which is the pixel without a depth value in a depth image, by using its position and color. In Assumption 2, since the depth of a specific position is assumed to be similar to the depth of an adjacent same object, that is, a region having a similar color, the depth of the object can be estimated. Therefore, in this paper, the proposed method generates a criterion using RGB image for determining which object containing a DUP, and the depth of the DUP is estimated using depth information of adjacent pixels of the object including the DUP. Figure 2 shows the entire process of the proposed method.

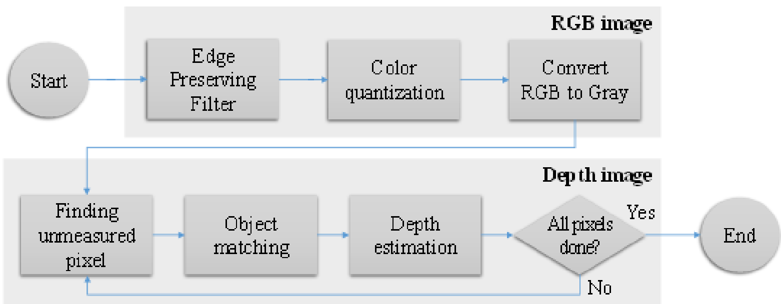


Figure 2. The process of the proposed method

2.1. RGB image segmentation by colors

This step is the process of generating a criterion to find the object containing a DUP. In Assumption 1, pixels with similar colors in a given small region are the same object, this process aims to represent regions with similar colors as a single color (object) in an RGB image.

First, the proposed method removes the noise and simplifies the RGB image. It can be done effectively by removing the texture. The meanshift filter [21], a well-known method for removing textures, is very slow and unsuitable for real-time processing. Therefore, in this paper, edge preservation smoothing that can preserve the boundary between objects while simplifying the texture of the image is used. Edge preserving smoothing include median filter [22], bilateral filter [23], edge preserving filter [24], and so on. The proposed method uses edge preserving filter which is fast and has good performance. Figure 3 shows an example of edge preserving filtering. The noise and texture in Figure 3(a) are effectively removed.

(a) Original image

(b) Edge perserving filtering

Figure 3. An example of edge preserving filtering

Then, the proposed method applies color quantization to the edge preserving filtered image. As the color quantization is a method to reduce colors in an image, it can represent many similar colors in one color. It means similar color regions can be represented as same color, so it is equivalent to the purpose of RGB image segmentation step. Among the various color quantization methods [25-28], the proposed method uses color quantization using color importance-based self-organizing map (SOM) [25]. Since this color quantization method learns all pixels in the image using SOM, it reduces colors effectively with a small error.

Figure 4(a) shows an example of applying color quantization to Figure 3(b). Regions with similar colors are represented in the same color, so it shows contour-like shapes. In this paper, the number of the colors in the RGB image is reduced to 16 colors. Now, to determine which object contains a pixel of a certain coordinate, it only needs to find adjacent pixels of the same color. Figure 4(b) shows the result of converting a quantized image to a grayscale image. It can determine whether the same color or not by using RGB values, but the proposed method uses grayscale image because it is faster to compare one channel.

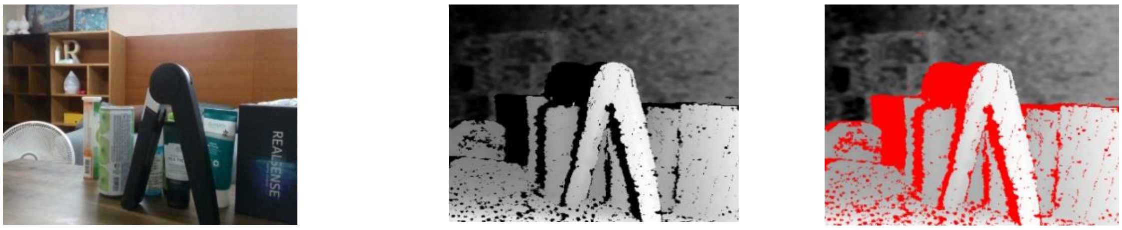
(a) Color quantized image

(b) Grayscale image

Figure 4. An example of image segmentation

2.2. Depth correction

Depth correction step is a process of estimating depth of DUP using the depth information of adjacent same objects. Figure 5 shows an example of DUPs in the depth image. Figure 5 (a) shows the RGB image, (b) shows the raw depth image, and (c) shows the DUPs in red. Depth correction step estimates the depth for every pixel marked in red.



(a) RGB image (b) Raw depth image (c) Depth unmeasured pixels

Figure 5. An example of depth unmeasured pixels

First, this step finds the DUP in the depth image. To do this, the proposed method uses the GrassFire algorithm [29], which is used as a labeling algorithm, to find black pixels. The Grassfire algorithm is a method of sequentially visiting adjacent pixels like fire spreading on dry grass starting from seed, so it finds DUPs sequentially. Then, the proposed method corrects the depth of DUPs using (1) and (2). The Equation (1) defines a method of finding a depth of a specific coordinate in a depth image.

$$depth(x, y, color) = \begin{cases} depth(x, y), & if\ color = color(x, y) \\ 0, & otherwise \end{cases} \tag{1}$$

where x and y are the coordinates of the pixel, $depth(x, y)$ is the depth of the x and y coordinates, and $color(x, y)$ is the color of the x and y coordinates. Only the depth of the same object is needed to correct the depth. Therefore, if the color is the different then it gives 0. The Equation (2) defines a method for depth correction.

$$new\,depth(w, h) = \frac{\sum_{x=w-1}^{w+1} \sum_{y=h-1}^{h+1} depth(x, y, color(w, h))}{count} \tag{2}$$

where w and h are coordinates and *count* is the number of depth values. This is the average of the depths of the pixels corresponding to the same object among the 8-connected adjacent pixels of (w, h) pixels divided by the number. The Equation (2) means that the sum of the depths corresponding to the same object among 8-connected adjacent pixels of (w, h) coordinate is divided by *count*, which is an average depth value. The proposed method performs this step until the depth correction for all DUPs is complete.

3. EXPERIMENTAL RESULTS

The experimental environment for evaluating the proposed method is as follows. We used laptop computer with Intel i7-7500U 2.70 GHz CPU and 8.0 GB RAM, and the proposed method is implemented in Visual Studio 2017 with MFC-based C ++ language. For depth camera, Intel RealSense D435 and Intel RealSense SDK 2.0 were used, and OpenCV 3.4.2 was used as an image processing library.

Figure 6 shows the comparison between the proposed method and the postprocessing results of the Intel RealSense SDK. In Figure 6 (a), the proposed method represented the fan, bookcase, and the height of the objects more accurately. The right region of the bass guitar in Figure 6 (b), the dark passage region in Figure 6 (c), the shape of the rabbit doll in Figure 6 (d), the shape of the rabbit doll and background in Figure 6 (e), and the shape of the hand and the monitor in Figure 6 (f) were represented by the proposed method more accurately.

Experimental results show that the proposed method corrects the shape of objects better than the Intel RealSense SDK when DUP is occurred by IR shadow. However, if there are other objects or backgrounds with similar colors around the object, the proposed method may not accurately estimate the depth. It is caused by the Assumption 1, so it can be improved by making object segmentation more accurate.

RGB image | Intel RealSense SDK | The proposed method

(a)

(b)

(c)

(d)

(e)

(f)

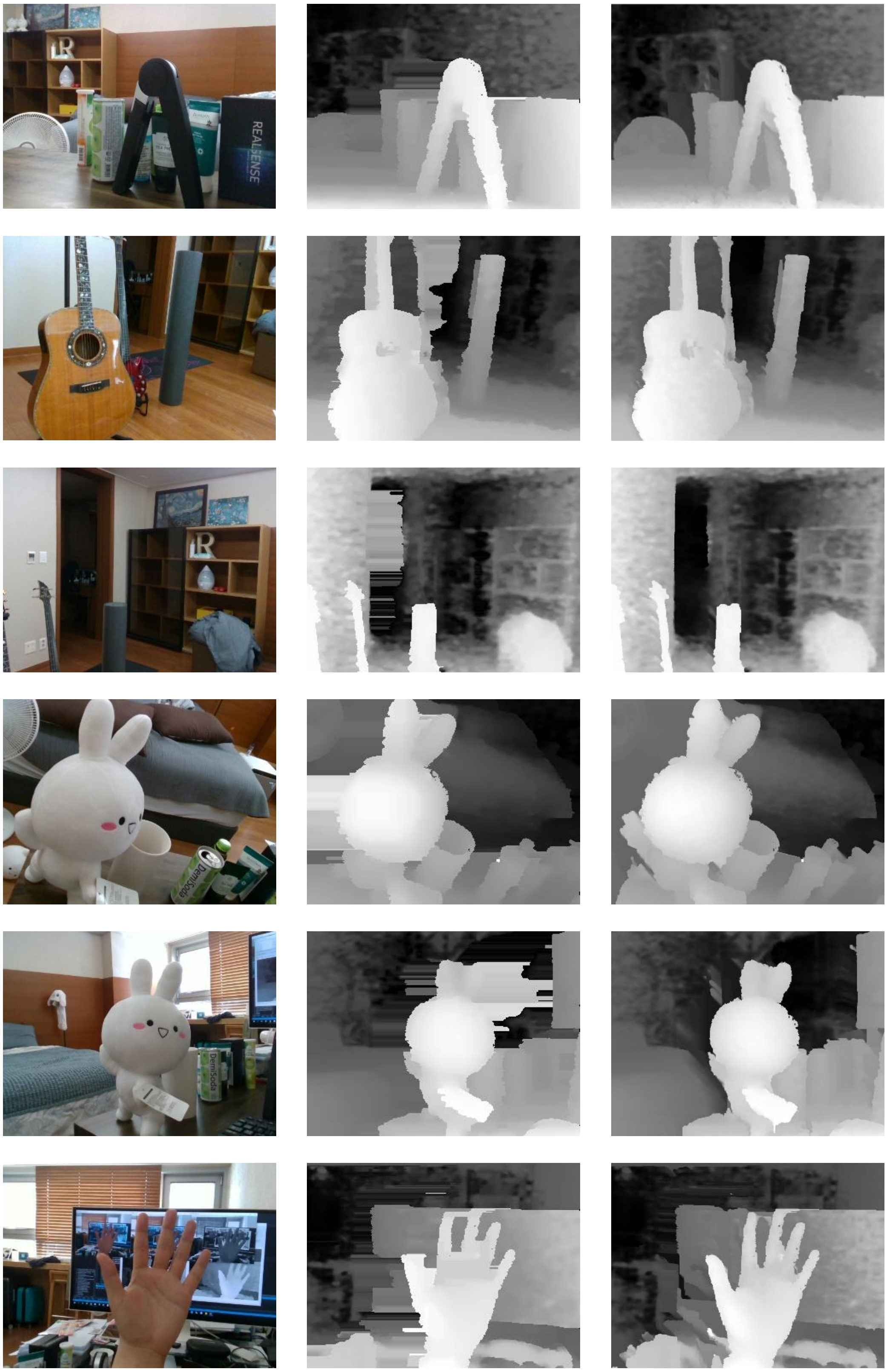


Figure 6. The results of the Intel RealSense SDK and the proposed method

4. CONCLUSION

In this paper, we proposed a depth image correction method to improve the postprocessing of Intel RealSense depth camera. The Intel RealSense SDK uses only a depth image for depth correction. However, since the depth information is already missing, depth correction results cannot be accurate. First, in this paper, we made two assumptions that the adjacent same object has similar colors and the depth can be estimated based on the depth of the adjacent same object. The proposed method consists of RGB image segmentation and depth correction. In the RGB image segmentation step, edge preserving filter and color quantization is applied to simplify the RGB image. This is used as a basis for determining which pixels among adjacent pixels are included in the same object. In the depth correction step, the depth is estimated based on the adjacent depth information of the same object. Experimental results showed that the proposed method corrects the depth of objects more accurately than the Intel RealSense SDK. However, when different objects with similar colors are adjacent to each other, accurate correction may not be possible. In future work, we will improve the estimation method in depth correction step to obtain more accurate depth correction results.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016.

[2] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3367-3375, 2015.

[3] Z. Fu, T. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2635-2644, 2015.

[4] M. T. M. Khairi, S. Ibrahim, M. A. M. Yunus, and A. R. Wahap, "Detection of foreign objects in milk using an ultrasonic system," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 15, no. 3, pp. 1241-1249, 2019.

[5] T. Bathini and B. Gadgay, "Hybrid enhanced ICA & KSVM based brain tumor image segmentation," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 14, no. 1, pp. 478-489, 2019.

[6] G. Ramkumar and E. Logashanmugam, "Multimodal verge for scale and pose variant real time face tracking and recognition," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 13, no. 2, pp. 665-670, 2019.

[7] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *Journal of Real-Time Image Processing*, vol. 11, no. 1, pp. 5-25, 2016.

[8] K. Konolige, J. Augenbraun, N. Donaldson, C. Fiebig, and P. Shah, "A low-cost laser distance sensor," *In 2008 IEEE International Conference on Robotics and Automation*, IEEE, pp. 3002-3008, May 2008.

[9] K. Khoshelham, "Accuracy analysis of kinect depth data," *In ISPRS workshop laser scanning*, vol. 38, no. 1, pp. 133-138, Aug. 2011.

[10] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," *In 2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012.

[11] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, "Temporal filtering for depth maps generated by kinect depth camera," *In 2011 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, IEEE, pp. 1-4, May 2011.

[12] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4-10, 2012.

[13] J. Gimeno, P. Morillo, J. M. Orduña, and M. Fernández, "A new AR authoring tool using depth maps for industrial procedures," *Computers in Industry*, vol. 64, no. 9, pp. 1263-1271, 2013.

[14] C. Camporesi, M. Kallmann, and J. J. Han, "VR solutions for improving physical therapy," *In 2013 IEEE Virtual Reality (VR)*, pp. 77-78. IEEE, Mar. 2013.

[15] A. Oliver, S. Kang, B. C. Wünsche, and B. MacDonald, "Using the Kinect as a navigation sensor for mobile robotics," *In Proceedings of the 27th conference on image and vision computing New Zealand*. ACM, pp. 509-514, Nov. 2012.

[16] K. Boudjit, C. Larbes, and M. Alouache, "Control of flight operation of a quad rotor ar. drone using depth map from microsoft kinect sensor," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 3, no. 3, pp. 15-19, 2008.

[17] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel (r) realsense (tm) stereoscopic depth cameras," *In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, pp. 1267-1276, Jul. 2017.

[18] M. Draelos, Q. Qiu, A. Bronstein, and G. Sapiro, "Intel realsense= real low cost gaze," *In 2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 2520-2524, Sep. 2015.

[19] Intel Realsense depth camera, Available: https://software.intel.com/en-us/realsense/d435, [Accessed: August 27, 2019]

[20] "Intel Realsense SDK 2.0," Intelrealsense, [Online] Available: https://software.intel.com/en-us/realsense/sdk, [Accessed: August 27, 2019]

[21] G. Wu, X. Zhao, S. Luo, and H. Shi, "Histological image segmentation using fast mean shift clustering method," *Biomedical engineering online*, vol. 14, no. 24, pp. 1-12, 2015.

[22] K. Verma, B. K. Singh, and A. S. Thoke, "An enhancement in adaptive median filter for edge preservation," *Procedia Computer Science*, vol. 48, pp. 29-36, 2015.
[23] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 839-846, Jan. 1998.
[24] E. S. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," In *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 69-79, Aug. 2011.
[25] H. J. Park, K. B. Kim, and E. Y. Cha, "An effective color quantization method using color importance-based self-organizing maps," *Neural Network World*, vol. 25, no. 2, pp. 121-137, 2015.
[26] H. J. Park, K. B. Kim, and E. Y. Cha, "An effective color quantization method using octree-based self-organizing maps," *Computational intelligence and neuroscience*, vol. 2016, pp. 1-11, 2016.
[27] M. E. Celebi, Q. Wen, and S. Hwang, "An effective real-time color quantization method based on divisive hierarchical clustering," *Journal of Real-Time Image Processing*, vol. 10, pp. 329-344, 2012.
[28] K. L. Chung, Y. H. Huang, J. P. Wang, and M. S. Cheng, "Speedup of color palette indexing in self-organization of kohonen feature map," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2427-2432, 2012.
[29] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd edition, Prentice Hall, Nueva Jersey, 2008

BIOGRAPHIES OF AUTHORS

Hyun Jun Park. He received his M.S. and Ph.D. degrees from the Department of Computer Engineering, Pusan National University, Busan, Korea, in 2009 and 2017, respectively. In 2017, he was a postdoctoral researcher at BK21PLUS, Creative Human Resource Development Program for IT Convergence, Pusan National University, Korea. From 2018 to the present, he is an associate professor at the Division of Software Convergence, Cheongju University, Korea. His research interests include computer vision, image processing, factory automation, neural network, and deep learning applications.

Kwang Baek Kim. Kwang Baek Kim received his M.S. and Ph.D. degrees from the Department of Computer Science, Pusan National University, Busan, Korea, in 1993 and 1999, respectively. From 1997 to the present, he is a professor at the Department of Computer Engineering, Silla University, Korea. He is currently an associate editor for Journal of Intelligence and Information Systems and The Open Computer Science Journal (USA). His research interests include deep neural network and applications, bioinformatics, computer vision and image processing.