International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.9 (2014), pp.269-280
http://dx.doi.org/10.14257/ijmue.2014.9.9.29

Feature Points Extraction for Camera Tracking in Augmented Reality System

Boo-Gyum Kim[1], Jong-Soo Choi[2], and Jin-Tae Kim[3,*]

[1]*UNIQ Incorporated, Korea*
[2]*Department of Image Engineering, Chung-Ang University, Korea*
[3]*Department of Aerospace Software Engineering, Hanseo University, Korea*
kbg2111@naver.com, jschoi@cau.ac.kr, jtkim@hanseo.ac.kr

Abstract

In this paper, the natural features extracted from images do not use the specific markers or reference points proposed augmented reality system. First, for tracking objects in video footage after the creation of feature extraction and descriptors in the same way about the creation of feature extraction and descriptors will. Descriptor information and features $I_1I_2I_3$ color around the edge of space, size and rotation is created by using the information. This generated using the descriptors for each feature determine the similarity and differences between the values obtained using the distance to check for matching. This process is generated through the corresponding points for two-dimensional plane to obtain the homograph information to get the camera matrix is a 3-D object matching. Finally, the proposed system to put the game on algorithm-based augmented reality, natural feature points of the game will be implemented.

Keywords: *marker, augmented reality, tracking, feature extraction, descriptor*

1. Introduction

Recently, much attention has been paid to technologies combining a new type of imagery with daily living using virtual reality and mixed reality. Among the related technologies, augmented reality (AR) is a technology that implements a new type of computing environment by fusing real images acquired from cameras with additional information in real time [1,2].

The core of AR technology is to acquire camera images in real time and to estimate the features of acquired images accurately, thereby registering virtual objects stably into the images. Using the features acquired from the camera images, planar information is obtained as well as camera location information. The general method to implement augmented reality is to assume that actual and virtual camera location information is the same and to register virtual objects obtained from the information. A typical method for the implementation of augmented reality uses a reference point that has a specialized format or artificial markers. When markers or reference points are used, implementation of AR systems becomes convenient because a tracking problem can be solved simply and thus ensures stable performance.

However, marker-based systems have several drawbacks, since extraction of marker information is difficult due to obstruction of objects by markers or lighting condition changes and the difficulty of system operation without specific markers. To overcome

[*] Corresponding Author

ISSN: 1975-0080 IJMUE

such problems, a study on AR systems without using specific markers or reference points has been conducted [3].

To implement such a system, two methods have been used: one is a registration method that extracts feature points of objects from images or stored in a database and the other is a registration method that extracts 3D feature points from images. Although these methods have a complex implementation process because of the extraction of feature points, descriptor creations about feature points, and computation of comparison matching between descriptors, their advantages include not using additional markers. This has been highlighted as a great strength and related studies are now in progress.

In this paper, a descriptor has been created using color information, which is robust to lighting, and features, which are robust to rotations and sizes. An augmented reality system using real-time natural feature points is proposed by employing a fast feature point extraction algorithm and matching method and its performance is compared and evaluated.

2. Proposed Algorithm

This paper proposes an augmented reality system that can extract and track feature points in real time and register virtual objects using natural feature points without the use of specific markers. In pre-processing an RGB color image is converted into a gray image while the Gaussian pyramid is applied to extract features that are not changeable by image size. Coordinates of feature points that are robust to corners are acquired from images using the fast corner detector algorithm, while descriptors are generated using $I_1I_2I_3$ color space information about surrounding areas of feature points, edge size and direction information of feature points and surrounding areas of feature points. Since $I_1I_2I_3$ displays an unchangeable characteristic despite changes in lighting, viewing direction, geometry and strong light, it is appropriately used for feature information for descriptors. Features that are robust to rotation can be extracted by collecting direction information about the areas surrounding feature points.

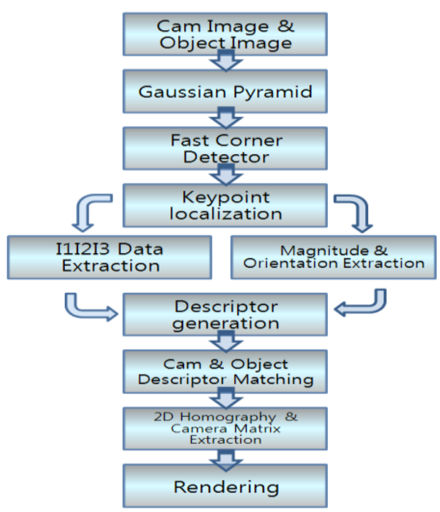

Figure 1. Flow Chart of the Proposed Algorithm

The second step is to measure similarity between the descriptors of camera feature points obtained by the above method and feature points of objects to be tracked. Instead of using Euclidean distance measurement, which is the most widely used similarity measurement method, this paper used the faster Manhattan distance measurement method to measure similarity with regard to each feature point. By comparing the measured similarity values, two points that have the least difference of similarity with each other are set as corresponding points.

The third step is to acquire a 2D planar homography matrix between generated corresponding points so as to extract camera coordinate information using the determinant. Using the acquired camera coordinate information and 2D planar information of images, virtual 3D objects are registered. Figure 1 shows a flow chart of the proposed algorithm.

2.1. Corner Location Information Extraction (Fast Corner Detector)

In the feature point extraction process of input images, features are extracted from differential images [4], which are edge images, among the previously-made difference of Gaussian (DoG) images. Here, a large amount of computation is required to perform pixel-unit processing since the number of images increases as the Gaussian pyramid process progresses. The corner tracking algorithm is used as the first step of the vision operation such as image search and recognition [5]. In this paper, the fast corner detector was used to reduce computational requirements.

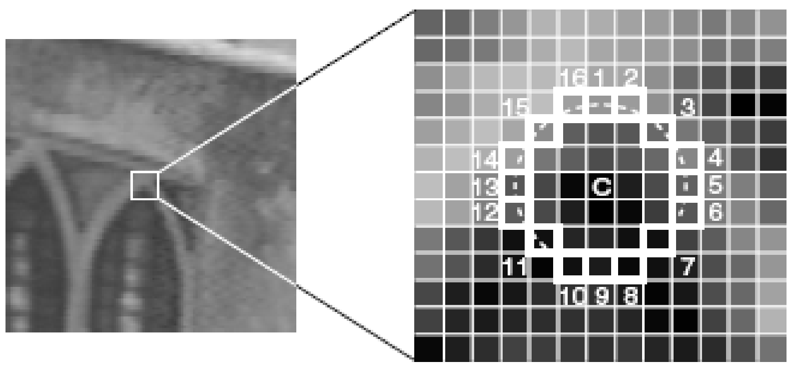

Figure 2. Detection Mask of the Fast Corner Detector

As shown in Figure 2, a corner detection mask consists of 16 pixels surrounding a given pixel *C*. A mask consisting of 16 pixels is classified according to predetermined criteria with regard to *C* that uses a threshold. Assuming that a set of 16 pixels of circular shape is $x \in \{1...16\}$ and a relative pixel of the position is p, $p \rightarrow x$ has one of the following three states shown in Equation (1).

$$S_{p\rightarrow x} = \begin{cases} d, & I_{p\rightarrow x} \le I_p - t \quad (darker) \\ s, & I_p - t < I_{p\rightarrow x} < I_p + t \quad (similar) \\ b, & I_p + t \le I_{p\rightarrow x} \quad (brighter) \end{cases} \tag{1}$$

where P sets a set with regard to all pixels p. Inside P, three subsets, P_d, P_s, and P_b are found and these are represented by Equation (2).

$$P_d = \{p \in P : S_{p\rightarrow x} = d\}$$

$$P_s = \{p \in P : S_{p \to x} = s\}$$
$$P_b = \{p \in P : S_{p \to x} = b\} \tag{2}$$

The corner detector that uses the above method was applied to the proposed system to extract the speed and data. Figure 3 shows the corner point extraction time with regard to the whole image DoG. The X axis in Figure 3 refers to the number of frames that were performed while the Y axis refers to the performance speed with a microsecond (ms) unit. The result showed that the speed was 4.6ms per frame on average.

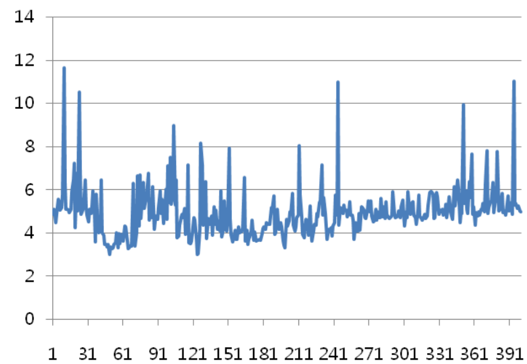

Figure 3. Processing Speed per Frame in the Corner Detector

Figure 4 shows a total sum of the corner points extracted as per frame. For every frame, the sum of the corner points extracted on average was about 1,400 on the basis of 500 frames. Since features that corresponded to the coordinates of the corner points using the above method are extracted, this method reduces execution time more than a method that extracts features by searching the whole image.

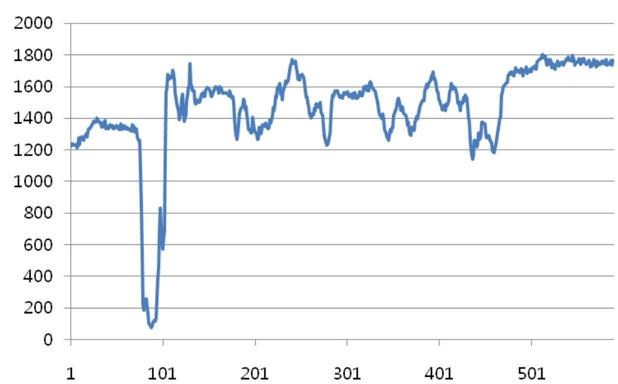

Figure 4. Sum of Corner Points Extracted as per Frame

Table 1 shows the sum of the average corner points of each DoG image for an execution time of 1,000 frames. Figure 5 shows an image that represents the corner coordinates extracted from the DoG image using the corner detector. In the figure, as the cross mark increases, it shows extracted coordinates from the reduced image. It is also verified that a coordinate where the cross mark is larger was detected less often than a coordinate where the cross mark is smaller.

Table 1. Sum of the Extracted Corner Points of the Edge Images in the Gaussian Pyramid

	octav_0	octav_1	octav_2	octav_3
interval_0	565	252	103	35
interval_1	92	49	25	6
interval_2	26	12	8	3
interval_3	8	1	1	1

Figure 5. Extracted Corner Coordinate Image

2.2. Feature Point Localization

The average sum of the generated corner points is more than 1,000. If descriptors are generated with many such corner points, required computation time increases and speed of matching decreases. To resolve this problem, localization is performed on the corner points. First, pixel values of the extracted corner points are removed by applying a certain threshold. In this paper, a threshold value 0.03 was applied, thereby removing corner points which have inappropriate contrast values. Next, points that are considered strong edges in one direction for each corner point are removed. The removal method uses a Hessian matrix [9].

The elements in a Hessian matrix are used to calculate a relationship of vertical, horizontal, and orthogonal directions as well as removing inappropriate points using a threshold value. The Hessian matrix was applied to about 2,000 points and the number of corner points can be reduced to about 100. Figure 6 shows the result of corner point removal.

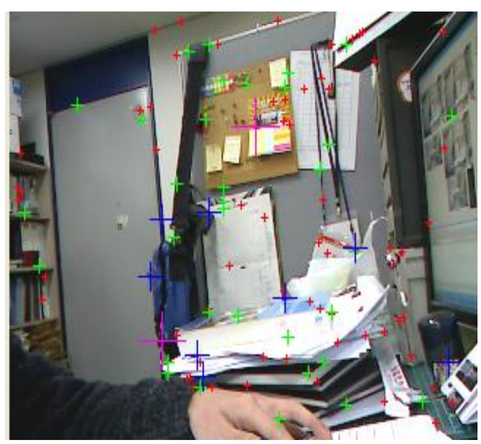

Figure 6. Unnecessary corner point removal

2.3. $I_1I_2I_3$ color Information Extraction

Since the $I_1I_2I_3$ color space shows robust characteristics against changes in lighting, viewing direction or geometry, it is a suitable feature vector as a feature of descriptors and one of the color spaces used widely in the object recognition field. In this paper, the $I_1I_2I_3$ color space feature was added to the descriptor in contrast with the existing scale-invariant feature transform (SIFT) or speed up robust feature (SURF). Each element of $I_1I_2I_3$ is extracted by calculating it using Equation (3).

$$\begin{cases} I_1 = \frac{1}{3}(R+G+B) \\ I_2 = \frac{1}{2}(R-B) \\ I_3 = \frac{1}{4}(2G-R-B) \end{cases} \tag{3}$$

The $I_1I_2I_3$ color information is provided with each average of I_1, I_2, and I_3 in all points within the detailed area of the extracted corner points, which are calculated by Equation (4) to generate each feature vector C1, C2, and C3. It is provided with the sum of the features in the surrounding pixels.

$$V(C_1,C_2,C_3) = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N} P_{ij}(I_1,I_2,I_3) \tag{4}$$

2.4. Extraction of the Size Direction of the Gradient

As another feature in the generation of descriptors, direction and size of the surrounding area of the extracted corner points can be found. At the 16×16 area around the extracted points, a 4×4 mask is formed, setting up 16 space areas. Once the size and direction of the gradient with regard to each pixel within the 16 spaces are extracted, the direction of each pixel inside the space is formed with an 8-direction vector. The size of each direction can be represented by the sum of the gradient size of each pixel, which subsequently creates 8 vectors for each area in the 16 spaces. This 128-dimension vector formed above is in the descriptor. Figure 7 shows one such example. The size and direction of the gradient are calculated using the gradient vector in Equation (5). Equations (6) and (7) are formulae to extract the size and direction of the gradient. Figure 8 shows the size and direction of the gradient extracted using the above equations.

$$Gradient\ vector = \begin{bmatrix} L(x+1,y) - L(x-1,y) \\ L(x,y+1) - L(x,y-1) \end{bmatrix} \tag{5}$$

$$m(x,y) = |L(x+1,y) - L(x-1,y)| + |L(x,y+1) - L(x,y-1)| \tag{6}$$

$$\theta(x,y) = \tan^{-1}\frac{L(x+1,y) - L(x-1,y)}{L(x,y+1) - L(x,y-1)} \tag{7}$$

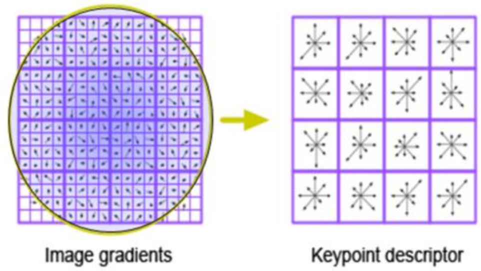

Figure 7. Extraction of the Size and Direction of the Gradient and Generation of the Descriptor

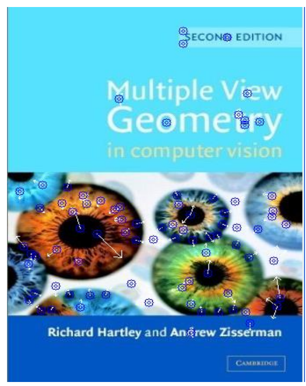

Figure 8. Size and Direction of the Extracted Gradient

2.5. Limited Area Matching Method

In this paper, a limited area matching method was used to provide faster matching than existing methods for matching corresponding points in real time. It was verified that existing SIFT and SURF methods experienced considerable reductions in matching speed due to the comparison of feature points over the whole area for each frame. To resolve this problem, a limited area matching method was used, performing matching operations only within the interest area by setting the matching area to the surrounding area of the object. Figure 9 shows the matching method by setting the interest area. Overall, the computational resources required were significantly decreased due to the reduction of the preprocessing image in the next processing procedure. This was thanks to the setup of the interest area after matching as shown in Figure 9.

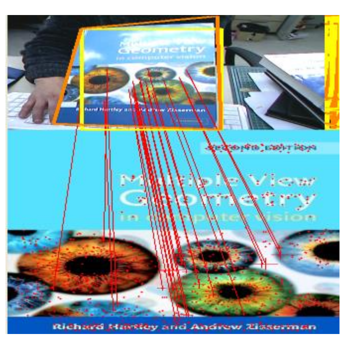

Figure 9. Limited Area Matching Image

3. Experiment Result and Verification

3.1. Algorithm Speed and Performance Result

The algorithm speed check is divided into two subparts: preprocessing and descriptor generation, and similarity matching between two descriptors. The first subpart showed good performance of 28 frames on average while the matching subpart showed a decrease in the number of frames. Figure 10 shows the speed change of the system. A speed of 15 to 20 frames was used so that the possibility of implementation within augmented reality in real time was proved. As features about the $I_1I_2I_3$ color space were employed, a robust response to lighting was revealed. Figures 11 and 12 show the result of matching according to changes in lighting as lighting changes in the same location.

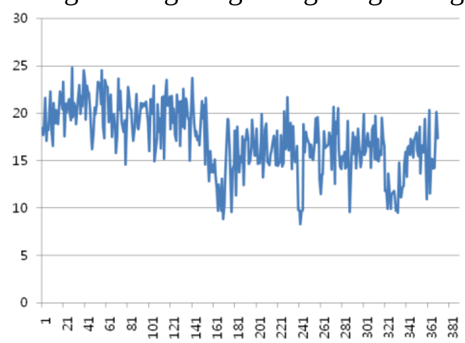

Figure 10. Speed change in the proposed system

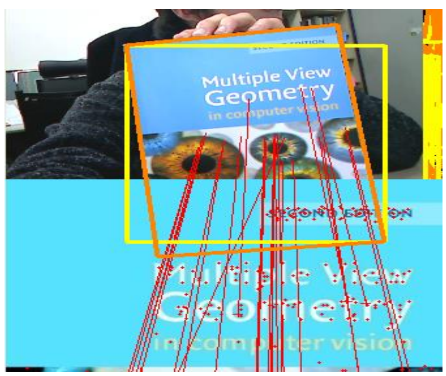

Figure 11. Matching Result in Bright Lighting

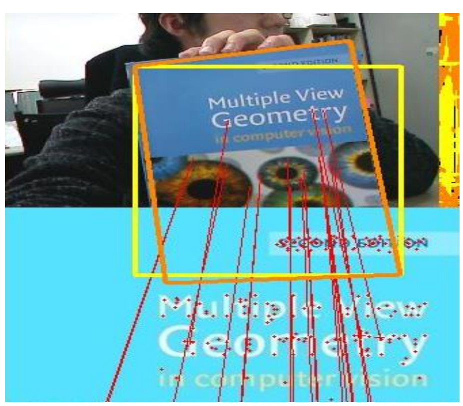

Figure 12. Matching Result in Dark Lighting

3.2. Comparison with the Existing Algorithms

This paper compared and analyzed the proposed algorithm with SIFT and SURF, two representative existing algorithms using natural feature points [8, 11, 14]. Figure 13 shows a speed comparison of our algorithm with SIFT and SURF with regard to preprocessing and descriptor generation subparts.

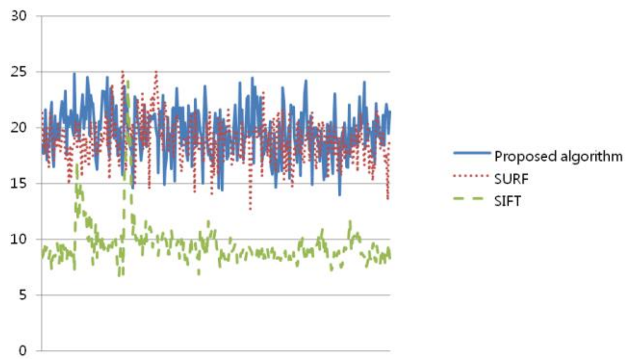

Figure 13. Time Taken for Preprocessing and Descriptor Generation for Each Algorithm

As shown in Figure 13, the SIFT algorithm, which examined all pixels of each of the Gaussian pyramid images during preprocessing, showed a considerable speed difference compared to the proposed algorithm, which computed only the corner point coordinates. The SURF algorithm showed performance comparable to the proposed algorithm because it reduced the dimension of descriptors, thereby speeding up performance. However, the experiment revealed that our algorithm was a little faster than SURF.

Figure 14 shows a speed comparison with regard to the matching subpart. In contrast with SURF or SIFT, both of which examined the whole area, the proposed limited area matching algorithm had better performance in matching speed.

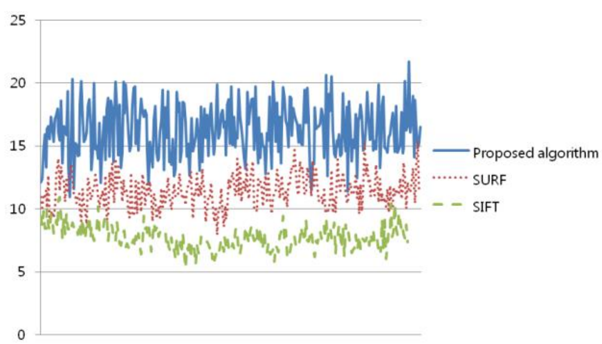

Figure 14. Speed Comparison of the Matching Subpart

4. Conclusion

In this paper, a natural augmented reality implementation system was proposed where natural feature points were extracted from images acquired from cameras followed by matching. First, preprocessing of images acquired from cameras was done followed by using a fast corner detector to extract corner points within the Gaussian pyramid image in real time. From the extracted corner points, $I_1I_2I_3$ color space feature information and strength and direction of the gradient were extracted, generating the descriptors. Using the image frames acquired from cameras and descriptors of target images to be tracked, matching was conducted using the similarity measurement while the limited area matching method was used to increase matching speed. Once planar homography and camera matrixes were obtained using the matched corresponding points, virtual objects were registered using the extracted planar information.

In the experiment, a speed test was conducted by dividing into two subparts: a preprocessing and descriptor generation subpart and a corresponding point matching subpart; data was extracted to compare and analyze our algorithm against the existing algorithms, SIFT and SURF. To apply the proposed system, registration of virtual objects and implementation of a game system are required. In future research, our proposed algorithm will be applied to not only game systems, but also various application areas of augmented reality.

Acknowledgements

This work was supported by 2013 Hanseo University research grants.

References

[1] [1] R. Azuma, “A Survey of Augmented Reality,” Teleoperators and Virtual Environments, vol. 6, no. 4, (**1997**), pp. 355-385.
[2] [2] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. Macintyre, “Recent Advances in Augmented Reality,” IEEE Computer Graphics and Applications, (**2001**), pp. 34-47.
[3] [3] S. -H. Lee, A Study on Real Time Camera Tracking and 3D Geometric Information Estimation for Augmented Reality System, Chung-Ang University, Ph. D. Dissertation, (**2008**).
[4] [4] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol. 60, no. 2, (**2004**), pp. 91-110.
[5] [5] E. Rosten, R. Porter, and T. Drummond, “Faster and Better: A Machine Learning Approach to Corner Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, (**2008**), pp. 110-115.
[6] [6] E. Rosten and T. Drummond, “Fusing Points and Lines for High Performance Tracking,” IEEE International Conference on Computer Vision, (**2005**), pp. 114-189.
[7] [7] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” Proc. European Conference on Computer Vision, (**2006**), pp. 1394-1399.
[8] [8] H. Bay, T. Tuytelaars, and L. V. Gool, ”Surf: Speeded up Robust Features,” Proc. Computer Vision and Image Understanding, (**2008**), pp. 346-359.
[9] [9] V. Lepetit, J. Pilet, and P. Fua, “Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation,” Computer Vision and Pattern Recognition, vol. 26, no. 2, (**2004**), pp. 244-250.
[10] [10] K. Mikolajczyk and C. Schmid, “An Affine Invariant Interest Point Detector,” Proc. European Conference on Computer Vision, (**2002**), pp. 128-142.
[11] [11] Z. Xiaomiao, Z. Xiaohu, J. Fangfang, and Y. Qifeng, “Known Landing Area Rough Locating from Far Distance Based on DDM-SIFT,” Proc. Congress on Image and Signal Processing, (**2008**), pp. 132-135.
[12] [12] F. Mokhtarian and R. Suomela, “Robust Image Corner Detection Through Curvature Scale Space,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, (**1998**), pp. 1399-1411.
[13] [13] T. T. H. Tran and E. marchand, “Real-time Key Point Matching: Application to Visual Servoing,” Proc. IEEE International Conference on Robotics and Automation, (**2007**), pp. 3787-3792.
[14] [14] Z. Wei, X. Weisheng, and Y. Youling, “Area Harmony Dominating Rectification Method for SIFT Image Matching,” Proc. International Conference on Electronic Measurement and Instruments, (**2007**). pp. 600-612.

Authors

Boo-Gyum Kim, he received his B.S. degree in Computer and Information Engineering from Hanseo University, Seosan, Chungnam, Korea, his M.S. degree in Image Engineering from Chung-Ang University, Seoul, Korea, in 2008 and 2010, respectively. He worked at Zenitum Inc. from 2010 to 2012. He worked at Solutionix Inc. from 2012 to 2013. He worked at Raphabio Inc. from 2013 to 2014. Now, he is working at UNIQ Incorporated. His research interests include augment reality, computer vision, and digital image processing.

Jong-Soo Choi, he received his B.S. degree from Inha University, Inchon, Korea, his M.S. degree from Seoul National University, Seoul, Korea, and his Ph.D. degree from Keio University, Yokohama, Japan, all in electrical engineering, in 1975, 1977, and 1981, respectively. He joined the faculty at Chung-Ang University in 1981, where he is now a professor of the Graduate School of Advanced Imaging Science, Multimedia, and Film. His current interests are in computer vision, image coding, and electro-optical systems.

Jin-Tae Kim, he received his B.S., M.S., and Ph.D. degree in Electronics Engineering from Chung-Ang University, Seoul, Korea, in 1987, 1989, and 1993, respectively. He worked at Institute of Industrial and Technology at Chung-Ang University from 1993 to 1995. Since 1995 he has been a faculty member of Department of Aerospace Software Engineering, Hanseo University. From January 2008 to January 2009, he was a visitor professor at the University of North Carolina at Charlotte. His research interests include image compression, augment reality, face recognition, and digital watermarking.

