Journal of the Korea Academia-Industrial cooperation Society
Vol. 22, No. 1 pp. 24-30, 2021

https://doi.org/10.5762/KAIS.2021.22.1.24
ISSN 1975-4701 / eISSN 2288-4688

Detecting the screw-assembly state of a valve-body using the AR method

Moon-Ho Kang
Department of Information and Communication Engineering, Sunmoon University

AR 방식을 이용한 밸브바디의 나사 조립 상태 검지

강문호
선문대학교 정보통신학과

Abstract In this study, an augmented reality (AR) app that detects the screw-assembly state of a car valve-body and assists the assembly work is developed and the effectiveness of the app is shown through testing. The app creates the contents indicating the screw-assembly position and order, and the screw-assembly state. Then, the contents are registrated onto the valve-body image on a smart-phone screen to be shown to the worker during assembly. To this end, the features are extracted from the 2D image of the valve-body and the location of the valve-body is tracked. By extracting the areas where the screws are to be assembled, and periodically determining the luminance of these areas, it is checked whether the screws are assembled in order at the predetermined position of the valve-body. When an error is detected during assembly, a warning sound is notified to the worker, and the worker can check the assembly state on the smart-phone screen and handle the error, immediately. Study results found that it takes about 65 ms to detect the assembly state of the five screws, and the assembly state is detected without error for 1 hour.

요 약 본 연구에서는 자동차 밸브 바디 (valve body) 나사의 조립 상태를 검지하고 조립 작업을 보조하기 위한 증강현실 (AR) 앱을 개발하고 테스트를 통해 유효성을 보인다. 조립할 나사의 조립 위치와 순서 및 조립 상황들을 표시하는 컨텐츠들을 생성한 후 스마트 폰 화면의 밸브 바디 영상 위에 정합하여, 조립 도중에 작업자에게 보여준다. 이를 위해, 밸브 바디의 2차원 이미지로부터 특징 데이터를 추출하고 밸브 바디의 포즈를 알아낸다. 밸브 바디 위의 각 나사들이 조립될 영역의 영상을 추출하고 주기적으로 해당 영역의 휘도를 판별하며, 밸브 바디 위의 정해진 위치에 순서에 따라 각 나사들이 조립되고 있는 지를 확인한다. 미조립 또는 조립 순서 오류 등 조립 에러가 검지되면 경고 음이 발생되어, 작업자가 스마트 폰 화면에서 즉시 조립 상태를 확인하고 에러를 처리할 수 있다. 테스트를 통해 5개 나사의 조립 상태를 검지하는데 약 65ms가 걸리고 1시간 동안 오류 없이 조립 상태가 검지되는 것을 확인했다.

Keywords : Android app, Augmented reality, Detect assembly state, Valve-body, Pose tracking

1. Introduction

Due to the development of multimedia technology, augmented-reality (AR) technology [1, 2] is being applied to the manufacturing field [3-8]. By integrating virtual AR contents into real working environment, it is possible to realize ubiquitous and sensory job supports.

*Corresponding Author : Moon-Ho Kang(Sunmoon Univ./Automation & Energy Tech. Institute)
email: mhkang@sunmoon.ac.kr
Received September 16, 2020 Revised November 9, 2020
Accepted January 8, 2021 Published January 31, 2021

In this study, an AR-based android app is developed to assist screw assembly of a car valve-body. Assembly of automobile parts is a simple repetitive operation and errors are infrequent, but if any non-assembly or assembly sequence error occurs, a serious problem can occur. Therefore, accurately detecting the part-assembly state is very important in the car assembly process. Meanwhile, most AR apps used in manufacturing works are in a form in which the worker works while viewing the app, and it can be a hindrance to the worker. But the app in this study monitors the user's work, and when an error occurs like non-assembly or assembly-order mismatch, it immediately notifies the worker through an alarm or voice message to handle the error. Therefore, there is no interference to work caused by having to view the app screen.

The app creates AR contents that show five screws, assembly position and order, and assembling state, and then adds the contents on the valve-body image of the smart-phone screen. In order to add the contents exactly on the valve-body image, the feature points of the valve-body are extracted from a 2D image [9] and the pose of the valve-body is tracked through the features. And, in order to confirm whether the screws are assembled in order at correct positions on the valve-body, the assembly state of the screws is detected periodically. The detection is carried out under a simple assumption, that is, when a screw is assembled, the assembled area (a hole and its vicinity) becomes brighter by the screw than when the hole is vacant. So, by measuring only the luminance of the areas during the assembly operation, the screw assembly state is obtained easily and accurately.

Through the test, it can be seen that it takes about 65 ms to detect the assembly state of the five screws, and the assembly state is detected without error for 1 hour.

2. Preliminary works

2.1 Feature-points extraction of valve-body

Fig. 1 shows the valve-body and the five-screws assembly points (part1~part5). The app developed in this study checks the assembly state of the automobile valve-body screws in real time.

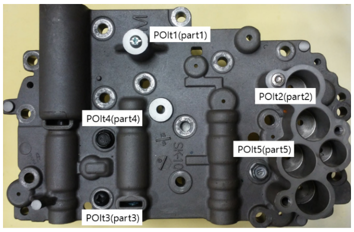


Fig. 1. Valve-body and five assembly positions.

Since the valve-body may move in the assembling line, the position of the valve-body should be traceable. For this, the valve-body feature points to be used for the tracking are extracted from a 2D valve-body image and saved as a file by using Vuforia's Target Manager [10]. This process is performed off-line only once before the app runs. Fig. 2 shows this process.

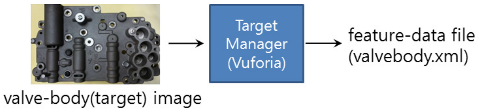


Fig. 2. Feature-points extraction of a valve-body.

2.2 Setting screws-assembly positions

Before the app runs, the assembly positions on the valve-body image in the smart-phone screen are set. Fig. 3 shows this process. When the valve-body (target) tracking succeeds, the model-view matrix and the perspective projection matrix are obtained [9]. Then, when the user touches the position where the screws should be assembled in the target image on the smart-phone screen, the app extracts the coordinates of the touched point (POIc) on the screen, and calculates the target coordinates (POIt) corresponding to the POIc with the inverse matrices of the model-view and perspective projection matrix. The obtained POIt is displayed on the screen and inserted into the app program as the assembly positions of the screws. This process is performed only once in the initialization routine.

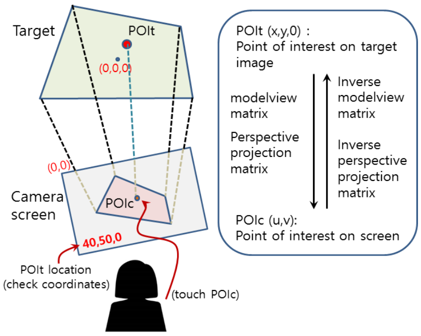


Fig. 3. Extraction of a part(screw)-assembly position (POIt) on the valve-body.

2.3 Setting AR contents

The AR contents are created using JME [11] library. The contents consist of 3D boxes (z-axis position is set to 0) to display assembly position and order of the screws, texts to show assembly-position coordinates, a screw image, and an assembly-state indicator (Fig.8(c)).

3. Detecting screws on the valve-body

To check the assembly states of the five screws on the valve-body, screw-assembly areas are extracted first, and simple image processing is performed to detect the five screws on the areas.

3.1 Extraction of screw assembly area

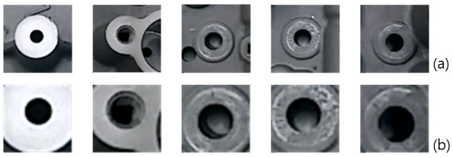


Fig. 4. Images of screw-assembly areas ((a) 80x80 pixels (b) 40x40 pixels).

Firstly, the areas where the five screws are to be assembled are extracted with no screws. Fig.4 shows two resultant images with two sizes.

Next, to obtain more precise assembly areas, a circle locus is searched in the assembly areas of Fig.4 and the center point of the circle is calculated. And finally, the area around the center point is extracted [12]. Fig.5 shows this process that finds the center point in the 40x40-pixels images shown in the Fig.4(b). Fig.5(a) shows the first-step result of searching circles. A circle is found in the screw-1, 3, 4, and 5 assembly areas, but no circle is found in the screw-2 assembly area. To solve this, the 40x40 pixels image is sharpened before searching the circle, and a circle is found in all assembly areas, but some circles slightly deviate from the center of the assembly area (Fig.5(b)). As final step, noise reduction is appended before the sharpening, and the result is shown in Fig.5(c) where the centers of the assembly areas are more accurately found.

3.1.1 Generation of reference images

Fig.5(d) and (e) show 20×20 and 10×10-pixels images extracted from Fig.5(c), with centers found from the circles in Fig.5(c). Each image in Fig.5(d) and (e) is used as a reference image later in the determining screw-assembly state. The reference images are generated only once during the initialization process.

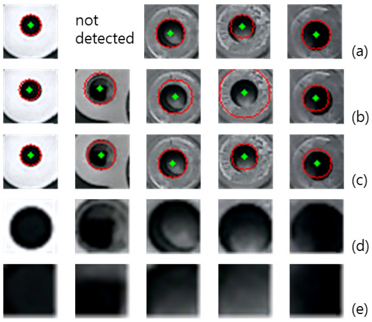


Fig. 5. Generation of reference images through circle searching (circle searching: (a) no compensation (b) with sharpening (c) with blurring and sharpening, reference images: (d) 20x20 pixels (e) 10x10 pixels).

3.1.2 Generation of target images

Fig.6(a) shows assembly areas when all screws are assembled. Fig.6(b) shows the result of blurring and sharpening, and then extracting circles and center points. Fig.6(c) shows 20x20-pixels images (target images) extracted from Fig.6(a) with the center points. During assembly work, the target images are generated periodically and compared with the reference images (Fig.5(d)), and the differences are analyzed to determine whether the screws are assembled or not.

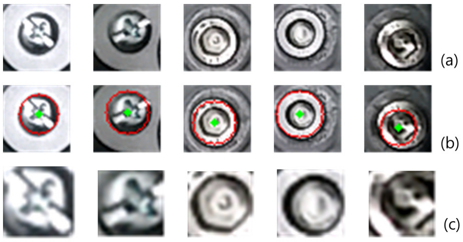


Fig. 6. Generation of target images (a) screw-assembly image (40x40 pixels) (b) circle detection with blurring and sharpening (40x40 pixels) (c) target image (20x20 pixels).

3.2 Determination of parts-assembly state

To determine whether the screws are assembled or not, luminance value (Y) is extracted for both reference and target images, and then the differences between the luminance of both images are calculated as follows.

$$Y_{d,n}[\%] = \frac{100}{2 \times 255 \times w \times h} \sum_{i=0}^{w \times h-1} \left| Y_{ref_i,n} - Y_{tar_i,n} \right| \quad (1)$$

n:number of screws, $Y_{d,n}$:luminance difference between the reference and target images for screw-n, $Y_{ref,n}$, $Y_{tar,n}$:luminances of the reference and target images for screw-n, w, h:width and height of the reference or target image, i:index of pixel positions

When $Y_{d,n}$ is calculated for all assembly areas, an assembly state is determined by comparing $Y_{d,n}$ and a threshold, $Y_{d,th}$ as shown in below condition. After selecting the maximum (minimum) value among the $Y_{d,n}$ obtained from a number of tests with no screws assembled (with all parts assembled), $Y_{d,th}$ is set to the median value between the maximum and minimum values. Threshold can be set only once during initial setup.

If $Y_{d,n} > Y_{d,th}$ screw-n assembled
Else screw-n not-assembled

$$Y_{d,th} = median(Y_{d,nas},\ Y_{d,as}) \quad (2)$$

$Y_{d,th}$: threshold value of $Y_{d,n}$
$Y_{d,nas}$: max $Y_{d,n}$ when no screws assembled
$Y_{d,as}$: min $Y_{d,n}$ when all screws assembled

4. Test and analysis

The smart-phone used for the test has display resolution of 1920 × 1080. The video frame rate is 30 frames per second and the video resolution is 1280×720. Fig.7 shows the test arrangement of the valve-body and smart-phone.

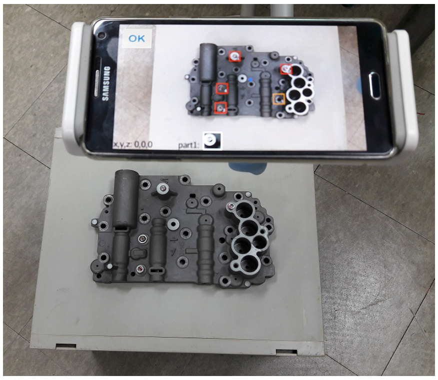


Fig. 7. Arrangement of valve-body and smart phone.

4.1 Screw detection with luminance threshold

In this test, not only the luminance (Y) of the screw-assembly area but also the color values (croma-red/blue: C_r/C_b) were used to determine whether the screw was assembled or not, but the color values were not effective because there was little color difference depending on the assembly state, and finally, the screw-assembly state was determined using only the luminance. To determine the luminance thresholds, $Y_{d,th}$ according to the Eq.2, some tests have been performed and the results are analyzed. Table 1 shows the analysis results for three test cases.

In Table 1, case I shows the test result performed with 40×40-pixels reference and target images. If no screws are assembled, the maximum $Y_{d,n}$ is 11.2, whereas when all screws are assembled, the minimum $Y_{d,n}$ is 15.4. So, $Y_{d,th}$ is set to 13, which is the median value between 11.2 and 15.4. By the same way, in case II with 20x20-pixel images, $Y_{d,th}$ is set to 24. On the other hand, in case III, with 10x10-pixel images, $Y_{d,th}$ is also set to 24. But in case III, since the distance between the minimum and maximum $Y_{d,n}$ is the largest, $Y_{d,th}$ reliability of the case III is highest.

As shown in the sixth column of Table 1, a detection cycle detecting the five screws is completed within 300 ms in all cases. Last column in the Table shows detecting error rates of 7200 cycles (1 hour), with screw-1,2,3 assembled and screw-4,5 not-assembled. When 40x40-pixels reference/target images were used without the circle detection (case I), 20 detection errors occurred. However, when images of 20x20 pixels or 10x10 pixels were used with the circle detection (case II, III), no detection error occurred, but, case III is more desirable because the detection time is about 50 % shorter than case II.

Table 1. Test results of screw detection.

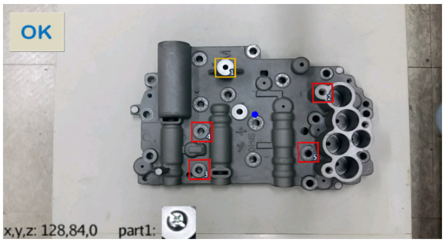
case	screw (n)	not-assembled[1] $Y_{d,n}$ (Eq.1) (min/max/avr)	not-assembled[1] $C_{rbd,n}$ [6] (min/max/avr)	all-assembled[2] $Y_{d,n}$ (Eq.1) (min/max/avr)	all-assembled[2] $C_{rbd,n}$ [6] (min/max/avr)	$Y_{d,th}$ [3] (Eq.2)	detection time [ms][4] (min/max/avr)	errors/ detection-cycles[5]
(I) 40x40 pixels	1	4.1/10.4/6.1	0.5/0.6/0.56	21.0/21.9/21.5	0.7/0.7/ 0.7	13	215/254/223	20/7200
	2	4.3/9.8/6.6	0.5/0.5/0.5	15.4/16.9/15.8	0.6/0.7/ 0.65			
	3	6.1/11.2/8.1	0.5/0.6/0.52	19.5/22.9/21.5	1.1/1.2/1.12			
	4	7.0/11.2/8.4	0.5/0.6/0.51	24.4/26.8/25.8	0.7/0.8/0.704			
	5	1.6/9.7/4.4	0.5/0.5/0.5	22.1/22.7/22.2	0.6/0.6/0.6			
(II) 20x20 pixels + circle detect	1	2.2/10.9/8.6	0.5/0.8/0.7	35.8/53.8/45.7	1.0/1.2/1.1	24	101/159/129	0/7200
	2	6.5/21.4/8.4	0.4/0.5/0.42	36.5/49.3/45.8	0.8/0.9/0.84			
	3	3.6/20.5/7.7	0.5/0.7/0.57	34.3/41.5/39.0	2.0/2.6/2.1			
	4	3.8/19.0/7.4	0.5/0.7/0.6	26.4/37.9/32.7	1.4/2.6/2.1			
	5	1.7/7.1/3.2	0.3/0.5/0.4	38.5/45.1/40.2	0.7/1.1/1.0			
(III) 10x10 pixels + circle detect	1	1.0/3.1/1.7	0.7/0.9/0.74	70.0/75.7/72.1	0.8/1.4/1.2	24	50/113/65	0/7200
	2	1.6/6.9/3.8	0.3/0.6/0.4	35.7/61.8/57.4	0.6/1.4/1.1			
	3	10.8/14.5/12.7	0.6/0.9/0.7	33.4/46.8/38.1	1.4/2.6/1.9			
	4	8.5/12.4/10.0	0.6/0.9/0.72	36.6/60.4/48.3	1.1/2.1/1.7			
	5	0.7/2.6/1.6	0.3/0.6/0.4	44.7/64.4/58.2	0.8/1.7/1.4			

For all tests, detection period/cycle is 500[msec]
1. No screws are assembled during 100 detection cycles.
2. All screws are assembled during 100 detection cycles
3. For 100 detection cycles, screw-detection threshold
4. For 100 detection cycles, detection-time duration for one cycle
5. For 7200 detection cycles, with screw-1,2,3 assembled and screw-4,5 not-assembled
6. $C_{rbd,n}[\%] = \frac{100}{2\times 255 \times w \times h}\sum_{i=0}^{w\times h-1}(|C_{rref_i,n} - C_{rtar_i,n}| + |C_{bref_i,n} - C_{btar_i,n}|)$

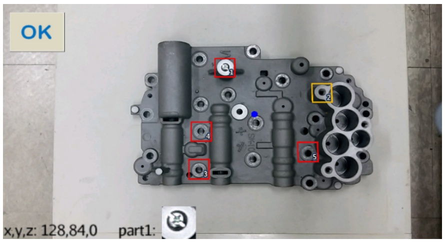
$C_{rbd,n}$: chroma-red and chroma-blue difference between the reference and target images for screw-n
C_{rref}, C_{rtar}: reference and target chroma-reds, C_{bref}, C_{btar}: reference and target chroma-blues

4.2 Screw-assembly test

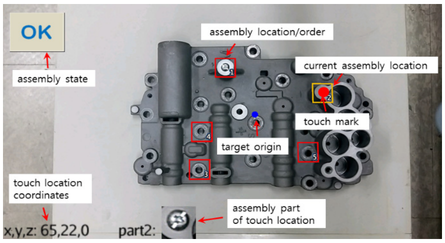
This test has been carried out under the same conditions as the case III in Table 1. Fig. 8 shows an assembling procedure of the five screws in order at the predetermined position of the valve-body. The number, shown in the 3D box displayed at the screw-assembly position, indicates the assembling order. The position where a screw should be assembled now is indicated by flashing the box color in red and yellow. When present assembling is correct, the assembly status displays OK, and when an assembling error occurs, it displays ERR. Then, a warning sound is immediately generated, allowing the worker to check present error state through the smartphone screen and correct it. When an assembly cycle is completed correctly, END is displayed. When an assembly position is touched on the smart-phone screen, the image of the screw to be assembled at the touched position is displayed at the bottom of the screen so that the worker can check it (Fig.8 (c)). Fig.8(d) shows a case that screw-3 should be assembled but screw-4 is assembled, so ERR is displayed. After screw-4 is removed and screw-3 is assembled, assembly situation changes back to OK (Fig.8 (e)).



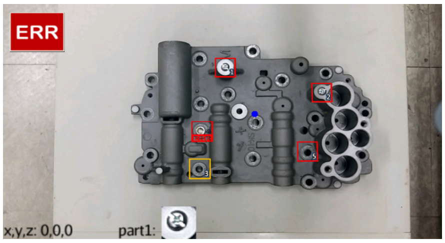
(a) assembled:no next:screw-1 (OK)



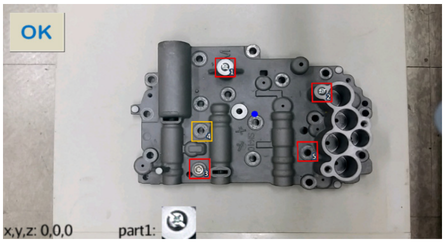
(b) assembled:screw-1 next:screw-2 (OK)



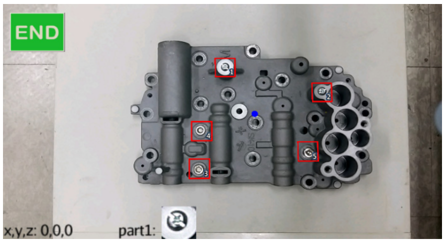
(c) check screw-2



(d) assembled:screw-1, 2, 4 (ERR: error)



(e) assembled:screw-1, 2, 3 next:screw-4 (OK)



(f) assembled:screw-1, 2, 3, 4, 5 next:no (END)

Fig. 8. Screw-assembly test.

5. Conclusion

In this study, an AR-based android app is developed to assist screw assembly of a car valve-body and the effectiveness of the app is shown through testing. The main features of the app are summarized as follows.

- Tracking target (valve-body) pose by using natural 2D feature points

- Guiding assembly work by using AR contents
- Detecting five screws through image processing
- Detecting assembly-error and notifying to worker

Test results show that it takes about 65 ms to detect the assembly state of the five screws and the assembly state is detected without error for 1 hour. And the screw-assembly test proves the app can be easily applied to assembly-work assistance. In the future, research will be conducted to verify the performance of the app in actual work sites and to develop various AR contents that can be used in a wider variety of work environments. In addition, a method of measuring and displaying the tightening state of the screw will be studied.

References

[1] Schmalstieg and T. Hollerer, Augmented Reality, Principles and Practice, Dieter, Addison-Wesley, pp. 78-84, 2016.

[2] S. Aukstakalnis, Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR, Addison-Wesley, pp.227-329, 2016.

[3] V. Paelke, "Augmented Reality in the Smart Factory Supporting Workers in an Industry 4.0. Environment", *Proc. of the IEEE Emerging Technology and Factory Automation*, pp. 1-4, 2014.
DOI: https://doi.org/10.1109/ETFA.2014.7005252

[4] A. Deshpande and I. Kim, "The Effects of Augmented Reality on Improving Spatial Problem Solving for Object Assembly", *Advanced Engineering Informatics*, vol. 38, pp. 760-775, 2018.
DOI: https://doi.org/10.1016/j.aei.2018.10.004

[5] O. Danielsson, A. Syberfeldta, M. Holma, and L. Wangb, "Operators Perspective on Augmented Reality as a Support Tool in Engine Assembly", *Procedia CIRP*, vol. 72, pp. 45-50, 2018.
DOI: https://doi.org/10.1016/j.procir.2018.03.153

[6] D. Aouam, S. Benbelkacem, N. Zenati, S. Zakaria, Z. Meftah, "Voice-based Augmented Reality Interactive System for Car's Components Assembly", *3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pp. 1-5, 2018.
DOI: https://doi.org/10.1109/PAIS.2018.8598516

[7] K. Qian, X. Zhang, H. Yu and X. Ma, "Industrial Maintenance and Assembly Guidance Using a Markerless AR System with Monocular Camera", *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1911-1016, 2018.
DOI: https://doi.org/10.1109/ICARCV.2018.8581087

[8] I. E. Nugraha, T. W. Sen, R. Wahyu, B. Sulistyo, et al., "Assembly instruction with augmented reality on Android application "assembly with AR"", *4th International Conference on New Media Studies*, pp. 32-37, 2017.
DOI: https://doi.org/10.1109/CONMEDIA.2017.8266027

[9] R. Laganière, OpenCV Computer Vision Application Programming Cookbook, 2nd Edition, Packt, pp. 281-313, 2014.

[10] Getting Started with Vuforia Engine for Android Development,
https://library.vuforia.com/articles/Solution/Getting-Started-with-Vuforia-for-Android-Development.html

[11] R. Kusterer, jMonkeyEngine 3.0: Beginner's Guide, June 2013.

[12] S. Kapur and N. Thakkar. Mastering OpenCV Android Application Programming, Packt Publishing Ltd, pp. 23-46, 2015.

Moon-Ho Kang [Regular member]

- Feb. 1990 : Korea Univ. Dept. of Elec. Eng. MS
- Aug. 1995 : Korea Univ. Dept. of Elec. Eng. Ph.D.
- Sep. 1995 ~ Aug. 1997 : KRRI, Researcher
- Sep. 1997 ~ current : Sunmoon Univ. Dept. of Com.&Info. Eng. Professor

〈Research Interests〉
Mobile and Imbedded System