ISSN: 1226-7244 (Print)
ISSN: 2288-243X (Online)
논문번호 14-02-16

j.inst.Korean.electr.electron.eng.Vol.18,No.2,291~297,June 2014
http://dx.doi.org/10.7471/ikeee.2014.18.2.291

Hand Language Translation Using Kinect

Junghwan Pyo*, Namhyuk Kang**, Jiwon Bang**, Yongjin Jeong**★

Abstract

Since hand gesture recognition was realized thanks to improved image processing algorithms, sign language translation has been a critical issue for the hearing-impaired. In this paper, we extract human hand figures from a real time image stream and detect gestures in order to figure out which kind of hand language it means. We used depth-color calibrated image from the Kinect to extract human hands and made a decision tree in order to recognize the hand gesture. The decision tree contains information such as number of fingers, contours, and the hand's position inside a uniform sized image. We succeeded in recognizing 'Hangul', the Korean alphabet, with a recognizing rate of 98.16%. The average execution time per letter of the system was about 76.5msec, a reasonable speed considering hand language translation is based on almost still images. We expect that this research will help communication between the hearing-impaired and other people who don't know hand language.

Key words: Kinect, sign language, hand language, hand detection, depth color calibration, decision tree

Ⅰ. Introduction

Due to the rapid technology leap in the field of image processing, many attempts were made for tracking individual figures inside an image. Hand and face regions, especially, have been the main concern in human body part detecting. Since the most widely used method in Human-Computer-Interaction(HCI) is hand gesture, we targeted our goal to tracking the hand region and use the data to translate sign language gestures, mainly hand language gestures which only use hand figures that match one-on-one with a letter. Since hand language is used as letter gestures while using sign language, our system will be convenient in writing letters between sign language motions, certainly much more easier than halting motions during sign language communication in order to write down letters with a keyboard or other input devices. The translating is realized through 2 stages; the hand detecting stage and the gesture recognizing stage.

In the hand detecting stage, we used 2 methods to track the human hand in a real time image: depth thresholding and color thresholding. Thanks to the depth sensor embedded inside the Kinect[1], we can extract depth data of the hand by using the hand depth coordinate which is provided in Kinect skeletal data. But, we may also receive the data in the wrist and arm region since the depth threshold we apply could include this area. By using the RGB image of the Kinect, we can extract skin toned objects by color thresholding. We converted the color model from RGB to YCbCr because RGB data isn't suitable to apply thresholds[2]. However, color thresholding may also detect other skin toned regions such as the face region. By merging the two thresholded images, we

* Dept. of Electronics and Communications Engineering, Kwangwoon University, e-mail : higre_pyo@hanmail.net Tel : +82-10-2019-6069
** Dept. of Electronics and Communications Engineering, Kwangwoon University
★ Corresponding author, e-mail : yjjeong@kw.ac.kr Tel : +82-10-5571-5551

※ The work reported in this paper was conducted during the sabbatical year of Kwangwoon University in 2012, and supported in part by MSIP(ministry of science, ICT and future planning) Korea under the human rssource development project for SoC (NIPA-2014-H0601-14-1001) supervised by the NIPA
Manuscript received Apr. 30, 2014; revised Jun. 10, 2014 : revised Jun. 16, 2014 ; accepted Jun. 18. 2014.

can discover the hand region. Some pixel noise occurs during this merging, so we use blob labelling and rule out the small pixel noises.

The gesture recognizing stage contains a decision tree that contains all the methods we used to find out what the real time image gesture means. In the tree, we used several ways to recognize the characteristics of the image such as number of fingers, number of contours, where the position of the hand is, whether the front of the image is the palm region or not, and whether the palm region contains fingers inside it. In the first layer of the tree, we set the agenda to the size of the image. If the image is horizontal, we resize it into a 125*70 image, and if the image is vertical, we resize it into a 80*135 image. After that, we use the differences of the factors written above.

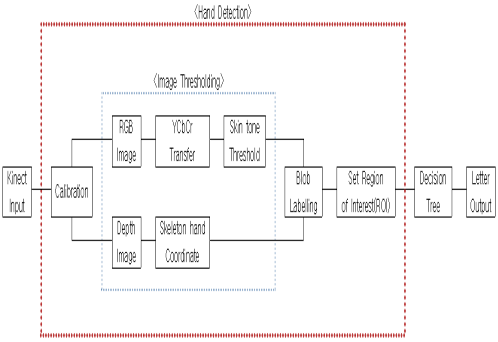
There have been many research attempts trying to detect and recognize hand language with computer devices. For example, in other countries, there have been attempts that used the Kinect device to recognize and translate the hand language of their own country[3][4].

In Korea, there were early attempts using special devices such as data gloves to recognize hand language[5][6]. After the gloves, there were some attempts using VGA cameras to detect the finger region and figure out hand gestures[7][8]. However, we have to make specific electronic glove devices in order to realize hand language translation when we use data gloves, and we receive inaccurate hand region information when we only use RGB image data to detect the hand. In this paper, we propose a translating system using Kinect to increase the detection and recognition rate of Korean hand language. Previously, there were related studies that proposed methods in hand detecting[9] and using a decision tree algorithm to translate hand language with Kinect[10]. In this paper, we propose a flexible depth threshold value by using the right hand skeleton coordinate and a decision tree that can detect fingers inside a palm region by using Sobel edge detecting.

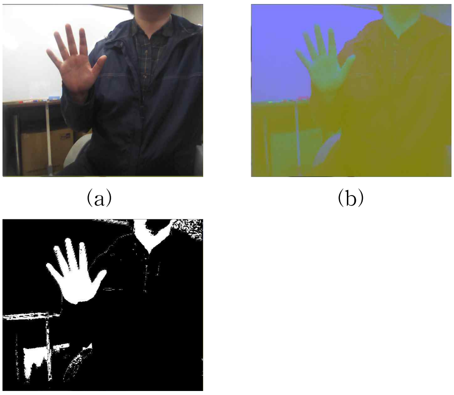
We assume the user is a Asian person with bare hands inside a building(where sunlight is almost blocked) due to the color threshold applied in the hand detecting stage and the effects of sunlight to the Kinect device. The system should be more reliable comparing to the recognition rates provided by the VGA camera-based translating method, and more convenient in usage comparing to data glove-based translating methods. By realizing hand language translation, it will provide the hearing-impaired population a convenient method of communication between each other and also with those who don't know hand language, and sign language in long term.

II. Translating Flow

The translation is performed in 2 stages, the hand detecting stage and the gesture recognizing stage. The hand detecting stage is then divided into 3 phases, color thresholding, depth thresholding, and color-depth merged image labelling. <Fig. 1> shows the flow of the translation.



<Fig. 1> Translation flow



(a)

(b)

(c)

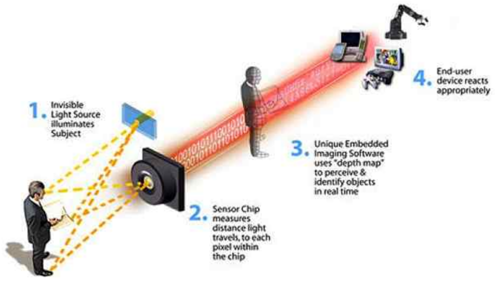
<Fig. 2> Color Image. (a) RGB image. (b) YCbCr image. (c) Binarized image.

2.1 Color Thresholding

The Kinect provides the user a RGB image stream from an embedded VGA camera. We perform 3 adjustments to this input RGB image. Since the YCbCr color model is convenient in tracking skin color, we convert the input RGB image into a YCbCr image. With the YCbCr image, we perform binarization by applying color thresholds in order to separate the skin color toned image from the YCbCr image. The Asian skin tone in the YCbCr scale is 133 < Cr < 173, 77 < Cb < 127, and 69 < Y < 255[2]. <Fig. 2> shows the stages up to the binarization result image: (a) is the original RGB image, (b) is the YCbCr converted image, and (c) is the image result after binarization.

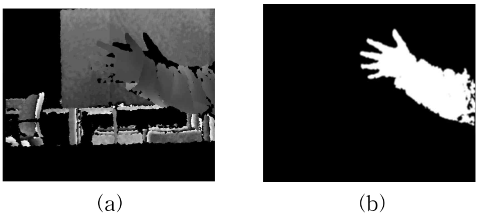
2.2 Depth Thresholding

The Kinect also provides depth information of an image stream by using light source measurement. <Fig. 3> shows how the Kinect device generates depth information[11].


1. Invisible Light Source illuminates Subject
2. Sensor Chip measures distance light travels, to each pixel within the chip
3. Unique Embedded Imaging Software uses "depth map" to perceive & identify objects in real time
4. End-user device reacts appropriately

<Fig. 3> Kinect depth data receiving procedure.

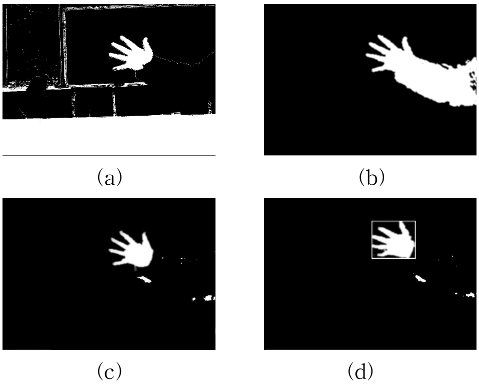
Thanks to the depth sensor of the Kinect, we can also use the depth value of the hand to separate it from the image. Since the front object of the Kinect will be the hand when sign language translation procedures are made, we can apply a threshold that originates from the hand depth received. We can receive the hand depth from the skeleton data of the hand which is provided by the Kinect sensors. There are some environmental restrictions when we use skeletal data from the Kinect. The Kinect device only provides skeletal data when it can recognize a whole body(or upper body when sitting mode is applied). So, the user must expose his/her entire(upper) body to provide the skeletal data of the right hand region. Once the skeleton's right hand depth information is provided, we binarize the depth image by whitening all the depth data near hand depth. <Fig. 4> shows the depth image and the binarized image.



<Fig. 4> Depth image. (a) Original depth image. (b) Binarized depth image.

2.3 Color-Depth Image Merging and Labelling

Both of the methods of hand extraction shown previously have weaknesses. In the color thresholding method, we can see that the binarized image also contains not only the hand, but the neck region and additional color noises of the background. In the depth region, we can notice that the whole arm appears in the binarized image. So, we merged the 2 binarized images and extracted the hand region by whitening the pixels which were white in both of the binarized images. Before we merged the two images, we used a calibration API provided in the Kinect Software Development Kit(SDK) to show the RGB image and the Depth image in the same coordinate system. When the merging process is complete, we can extract the hand with some noises that still remain in very small pixel areas. We solved this problem by blob labelling[12] only the hand region and



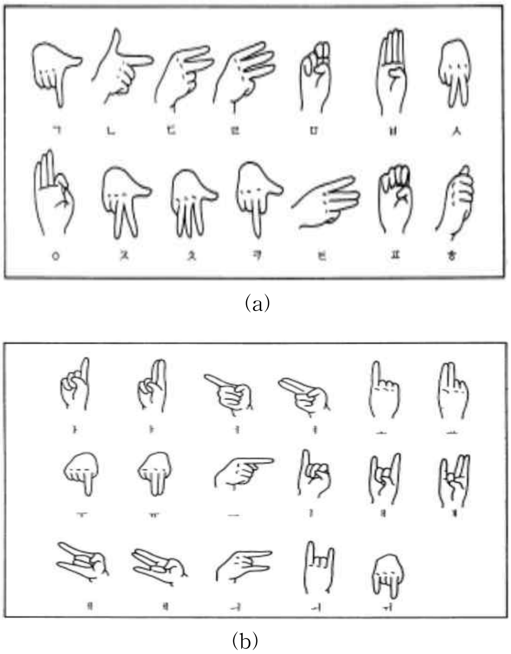
<Fig. 5> Calibration image. (a) Color-binarized image. (b) Depth-binarized image (c) Merged image. (d) Labelled image.

eliminated all the noises by applying size thresholds for them. <Fig. 5> shows the color-binarized image, the depth binarized image, the merged image.

In the final labelled image, we can notice that only the hand region is detected from the Kinect original RGB input image. For accurate hand detection, we use eroding and dilating to smoothen the hand region's edge area. Once the largest label, the hand region, is detected, we set a Region-of-Interest(ROI) to this label in order to neglect all the other parts of the given input image. Since the Kinect device provides a 640*480 image and the hand size is likely to be bigger than 50 pixels, we applied 50 as the pixel size threshold which eliminates the noise of the binarized image.

2.4 Decision Tree

After the hand detecting stage is complete, we enter the gesture recognizing stage. This stage is composed of a decision tree. Before we look inside the tree, we have to know how the Hangul hand gestures look like. <Fig. 6> shows the characteristics of the Hangul hand language.[13]



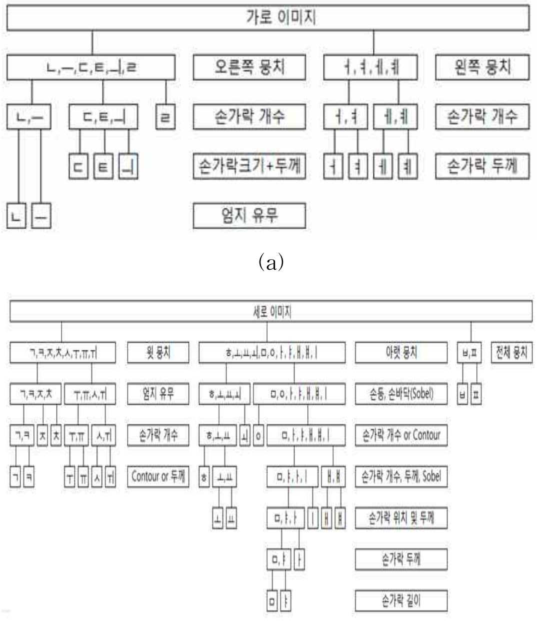
(a)

(b)

<Fig. 6> Hangul hand language, (a) Consonants, (b) Vowels

The Hangul hand language contains 31 letters, 14 consonants and 17 vowels. There are no letters that look exactly the same. Each letter contains noticeable characteristics such as number of fingers, position of the hand, direction of the hand, etc. So, we didn't divide our recognizing method into consonants and vowels. We made a decision tree by using the other characteristics. The first layer of the tree is noticing the size of the ROI image. If the ROI image's width is bigger than the height, we resize the image into a 125*70 image, and if the image's height is bigger than the width, we resize the image into an 80*135 image. This separates the letters into two big groups.

After resizing the input image into a certain size, we decided what the gesture means by using 8 characteristics, the position of the hand region, number of finger blobs, existence of the thumb region, finger blob thickness, finger blob pixel size, height of the entire hand blob, number of contours, and whether or not fingers are inside the palm area. The tree's depth is 8 layers and each layer contains one of the characteristic information above. For instance, we can detect 'ㄱ' by counting the number of finger blobs, looking for contours in the forefinger area, and finding out whether or



(a)

(b)

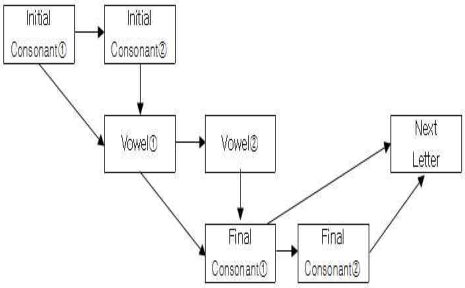
<Fig. 7> Decision Tree, (a) Horizontal images, (b) Vertical images

not the thumb region is detected. We minimized the usage of detecting fingers inside the palm region because the Sobel edge detecting algorithm takes a lot of time comparing to the other methods of gesture recognizing. <Fig. 7> shows the details of the decision tree algorithm we propose.

Since the whole decision tree is based on exact images, we can't expect the system to translate tilted images or severely contaminated gestures.

2.5 Letter Output

Hangul syllables are consisted of 3 different parts, an initial consonant, a vowel, and a final consonant. Each part of the syllable can contain either single or complex letter values. If another consonant comes after a consonant, the two consonants are combined into a complex consonant. The vowel can also go through this process and form a complex vowel. For instance, '닭', which means chicken in Hangul, can be decomposed to 'ㄷ' + 'ㅏ' + 'ㄺ'. in this case, 'ㄷ' and 'ㅏ' are single letter values and 'ㄺ' is a complex letter value. Considering this structure, we designed a code that realizes this combination and returns syllable level outputs. The output values are Unicode-based values. <Fig. 8> shows how we composed a Hangul syllable.

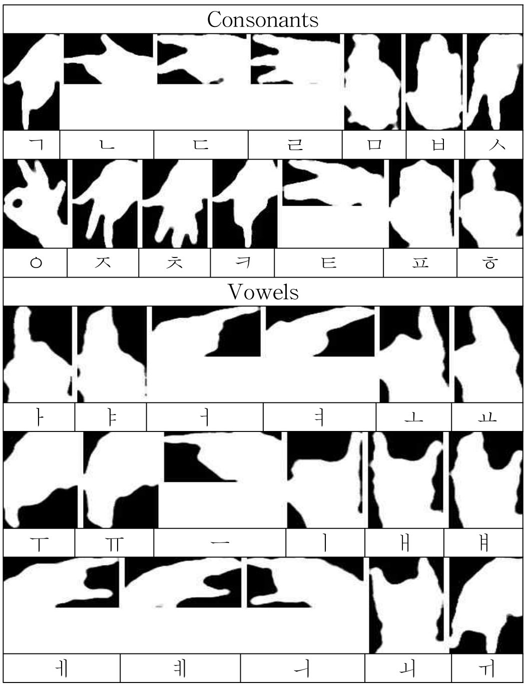


<Fig. 8> Hangul syllable forming flow chart

III. Results and Discussion

Before checking the recognition rate and speed of the system, we first checked how the hand image looked like. Table 1 shows the output image for each Hangul letter. By comparing the letter output images with the original gesture shown in <Fig. 6>, we can confirm that the system's output image matches each letter's characteristics. Note that images are rotated horizontally and height-based, weight-based images aren't the same size.

Table 1. Output image of each letter


Consonants
ㄱ
ㄴ
ㄷ
ㄹ
ㅁ
ㅂ
ㅅ
ㅇ
ㅈ
ㅊ
ㅋ
ㅌ
ㅍ
ㅎ
Vowels
ㅏ
ㅑ
ㅓ
ㅕ
ㅗ
ㅛ
ㅜ
ㅠ
ㅡ
ㅣ
ㅐ
ㅒ
ㅔ
ㅖ
ㅢ
ㅚ
ㅟ

In order to check the recognition rate of the system, we tested the program with 10 people. Each person performed a letter 20 times. The testing was performed with an Intel i-5 processor(clock frequency 2.5GHz). There were some differences in the execution time of each letters because of the Sobel edge detecting algorithm. The letters which used the algorithm took about 50msec more time since the Sobel algorithm contains diffrential matrix operations. The average execution time of the letters which used the Sobel algorithm(ㅁ, ㅇ, ㅎ, ㅏ, ㅑ, ㅗ, ㅛ, ㅣ, ㅐ, ㅒ, ㅚ) was about 105msec. Table 2 shows the recognition rate and the execution time of each letter.

Table 2. Recognition rate and execution speed of the proposed system

Hangul	Number of Samples	Recognition rate(%)	Execution time(msec)
ㄱ	199	99.50	59
ㄴ	188	98.40	60
ㄷ	197	92.90	60
ㄹ	199	98.50	59
ㅁ	199	100	117
ㅂ	200	100	59
ㅅ	188	96.28	59
ㅇ	195	100	119
ㅈ	193	96.37	58
ㅊ	196	97.96	89
ㅋ	198	96.46	60
ㅌ	199	99.50	60
ㅍ	198	98.99	59
ㅎ	196	100	149
ㅏ	196	100	89
ㅑ	194	100	89
ㅓ	193	98.96	58
ㅕ	197	100	57
ㅗ	148	99.32	118
ㅛ	179	98.32	119
ㅜ	194	91.75	60
ㅠ	184	95.65	60
ㅡ	188	95.74	57
ㅣ	178	97.75	90
ㅐ	192	97.92	92
ㅒ	191	98.43	90
ㅔ	190	98.95	60
ㅖ	194	99.48	59
ㅢ	191	97.38	60
ㅚ	196	99.49	88
ㅟ	196	98.98	58
Average	174.35	98.16	76.49

We sampled 20 images from 10 different people for random result checking. The samples were taken inside a room without people in the background of the image in order to show only one skeleton. As a result, we could receive about 17.5 images per person. This means that we can expect the translating program to recognize 87.5% of all of the input hand images given. This rate is a result of non-detected images such as weird-shaped hand gestures or multi-detected images which occur when the left hand is also shown. We can see that the average execution time is about 76.49msec. Considering that the target input images don't need high speed rates due to its low mobility, the results can be used in an actual translating program.

The recognizing rate of the program is quite promising. The average recognition rate is 98.16% and the lowest recognition rate was 91.75% for 'ㅜ'. In the case of 'ㅜ', the detection rate was lower compared to the other letters because people with thick index fingers can send out 'ㅠ' outputs to the translating system. Table 3 shows the recognition rate compared to data glove-based systems[5][6] and VGA camera-based systems[7][8].

Table 3. Recogniton rate comparison

Recognition rate(%)	Data glove		VGA camera		Proposed system
	[5]	[6]	[7]	[8]	
	80	80.27	93.22	78	98.16

IV. Conclusion

In this paper, we developed a hand language translating system by using Kinect depth and RGB images. We changed the color model to enhance the accuracy of skin tone detection and combined the color threshold image with the depth image(with the same coordinate system) to extract the exact hand region. After detecting the hand region, we used a decision tree to decide the gesture's letter output value in a Unicode value. With the given output value, we formed a Hangul syllable which supports not only single consonants or vowels but also complex ones. The average detection speed of the program was about 76.49msec, enough to track and detect almost still hand gestures, and the average recognition rate for letters was 98.16%. It can be used in providing letter inputs during sign language communication systems. The system can be applied also in other systems that use hand gestures. For instance, the program's algorithm can be used in a gesture-based game controlling program, an air mouse, or as a remote control for a television.

For further development, sign language translation can be realized when we use the Kinect device and add time variant machine learning methods to the system. The system can also be available to various races when we change the color threshold value of our system. Once sign language translation is applied, the program can be used as a communicating device between the hearing-impaired and other people. With additional applications, it could be used as a telephone-like device for the hearing-impaired.

References

[1] http://www.microsoft.com/en-us/kinectforwindows/

[2] Douglas Chai, "Face Segmentation Using

Skin-Color Map in Videophone Applications", p. 551-564, IEEE Transactions on circuits and systems for video technology, Vol. 9, No. 4, June, 1999

[3] Pedro Trindade, "Hand gesture recognition using color and depth images enhanced with hand angular pose data", presented at the IEEE Conference of Multisensor Fusion and Integration for Intelligent Systems, p. 71-76, September, 2012

[4] Yi Li, "Multi-scenario Gesture Recognition using Kinect", presented at the 17th International Conference on Computer Games, p. 126-130, August, 2012

[5] Sanghyeok Oh, "Interactive Learning of Korean Finger Spelling using Data Glove", p.733-736, in Proc. Spring Ann. Conference of the Korea Multimedia Society, Korea, May, 2007.

[6] Seungki Min, "Optimize Data Glove-based System for Korean Finger Spelling Recognition", p. 237-241, Korea Information Science Society, June, 2007

[7] Min-Ji Kang, "The Study on Dynamic Images Processing for Finger Languages", p. 184-189, KIISE Journal of Korea Computer Congress, Vol. 34, No. 1, April, 2004

[8] Hee-Deok Yang, "Automatic Spotting of Sign and Fingerspelling for Continuous Sign Language Recognition", KIISE Journal of Software and Application, Vol. 8, No. 2, p. 102-107, Febuary, 2012

[9] Hanhoon Park, "A Study on Hand Region Detection for Kinect-Based Hand Shape Recognition", p. 393 - 400, Journal of The Korean Society of Broadcast Engineers, Vol. 18, No. 3, May, 2013

[10] http://www.jameco.com/Jameco/workshop/howitworks/xboxkinect.html

[11] http://en.wikipedia.org/wiki/Connected-component_labeling

[12] Guochan Chang, "A Decision Tree based Real-time Hand Gesture Recognition Method using KINECT", p. 1393 - 1402, Journal of The Korea Multimedia Society, Vol. 16, No. 12, December, 2013

[13] http://www.korean.go.kr(National Institute of the Korean Language)

BIOGRAPHY

Junghwan Pyo (Student Member)

2007 ~ current : Course of BS in Electronics and Communications Engineering, Kwangwoon University.

Namhyuk Kang (Student Member)

2008 ~ current : Course of BS in Electronics and Communications Engineering, Kwangwoon University.

Jiwon Bang (Student Member)

2009 ~ current : Course of BS in Electronics and Communications Engineering, Kwangwoon University.

Yongjin Jeong (Member)

1983 : BS degree in Control and Instrumentation Engineering, Seoul National University.

1995 : MS, PhD degree in Electronics and Computer Engineering, University of Massachusetts, Amherst

1983~1989 : Research Engineer, ETRI

1995~1999 : Principle Research Engineer, Samsung Electronics

1999~current : Professor, Dept. of Electronics and Communications Engineering, Kwangwoon Univ.