색상 및 채도 값에 의한 이미지 코드의 칼라 인식

Recognition of Colors of Image Code Using Hue and Saturation Values

김태우*, 유현중**, 박흥국***

한양사이버대학교 정보통신학과*, 상명대학교 정보통신공학과**, 상명대학교 미디어학부***

Tae-Woo Kim(twkim22@hanmail.net)*, Hyeon-Joong Yoo(yoohj@smu.ac.kr)**, Hung-Kook Park(parkh@smu.ac.kr)***

요약

유비쿼터스 컴퓨팅에 대한 관심이 증가함에 따라, 이미지 코드도 다양한 영역에서 관심을 끌고 있다. 유비쿼터스 컴퓨팅에서 이미지 코드가 중요한 이유는 비용면과 함께 많은 영역에서 RFID(radio frequency identification)를 보완하거나 대체할 수 있기 때문이다. 그렇지만, 칼라의 왜곡이 심하여 정확한 칼라를 읽는데 어려움이 있기 때문에, 그 응용은 아직까지는 매우 제한적이다. 이 논문에서는, 칼라의 색상 및 채도 값을 이용하여 자동으로 이미지 코드를 찾아내는 것을 포함하여, 이미지 코드 인식에 관한 효율적인 방법을 제시한다. 이 논문의 실험에서는 현재 상용되고 있는 것들 중 가장 실용적이라고 판단되는 디자인을 사용하였다. 이 이미지 코드에는 여섯 개의 안전 칼라, 즉, R, G, B, C, M, Y가 사용되었다. 실험 영상들로는 크기가 2464 x 1632인 72개의 트루 칼라 필드 영상들을 사용하였다. 히스토그램에 의해 칼라를 보정한 경우, 코드 검출 정확도는 96%, 검출된 코드에 대한 칼라 분류 정확도는 91.28%이었다. 이미지 코드를 검출 및 인식하는데 2 GHz P4 PC에서 약 5초가 소요되었다.

■ 중심어 :| 이미지 코드 | 칼라코드 | 코드 검출 | 칼라 분류 | 코드 인식 |

Abstract

With the increase of interest in ubiquitous computing, image code is attracting attention in various areas. Image code is important in ubiquitous computing in that it can complement or replace RFID (radio frequency identification) in quite a few areas as well as it is more economical. However, because of the difficulty in reading precise colors due to the severe distortion of colors, its application is quite restricted by far. In this paper, we present an efficient method of image code recognition including automatically locating the image code using the hue and saturation values. In our experiments, we use an image code whose design seems most practical among currently commercialized ones. This image code uses six safe colors, i.e., R, G, B, C, M, and Y. We tested for 72 true-color field images with the size of 2464×1632 pixels. With the color calibration based on the histogram, the localization accuracy was about 96%, and the accuracy of color classification for localized codes was about 91.28%. It took approximately 5 seconds to locate and recognize the image code on a PC with 2 GHz P4 CPU.

■ Keyword :| Image Code | Color Code | Code Localization | Color Classification | Code Recognition |

* 본 연구는 상명대학교 공학기술연구소 연구과제로 수행되었습니다.

접수번호 : #050107-001

접수일자 : 2005년 01월 07일

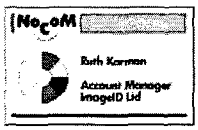
심사완료일 : 2005년 04월 27일

교신저자 : 유현중, e-mail : yoohj@smu.ac.kr

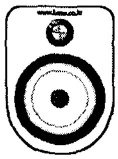
I. Introduction

Although bar code [1] and other binary codes [2] are being widely used in managing commercial products, it is confronted with restriction in further expanding its applications particularly because of cosmetic reason. On the contrary, the image code using colors is more advantageous not only in the appearance but also in the number of combinations. And compared to the RFID(radio frequency identification) [3], which is receiving drastic attention especially in the areas of ubiquitous computing [4], they share some application areas and the image code still finds some niche applications against the RFID for the cases where RFID can not be applied. And the image code is more economical than the RFID and needs no other special device to capture it -- it can be captured by usual camera-equipped cell phones or PC cameras. Most of all, the RFID is inherently susceptible to fraudulent uses because RF is exposed to anyone, and, for example, sophisticated EW (Electronic Warfare) techniques [5] might be used to interfere with RFID readers.

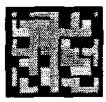
The fast and wide spread of inexpensive high-resolution digital cameras is also making the application of image codes rapidly emerge. There are some companies like Vividot [6], Imedia [7], and ColorzipMedia [8] which allegedly have successfully commercialized image codes: [Figs. 1] (a)~(c) show Vividot's image code tag, Imedia's image code tag, and ColorzipMedia's image code, respectively. [Figs. 2] (a) and (b) show their applications. In [Fig. 2] (a), after photographers have uploaded pictures from an event to the web site of host, visitors to the event visit the site and find their pictures to have them printed [6, 7]. [Fig. 2] (b) shows the image code in [Fig. 1] (c) which can be marked on news papers, magazines, or business cards, and it can be taken by using usual PC cameras to be directly linked to a related DB or services [8]. This kind of service is attractive in that it links off-line to on-line. The function that recognizes image codes on off-line products and then links customers to the site that provides related information is expected to encounter increasing demands with the development of various services using image codes.



(a) Vividot's image code tag

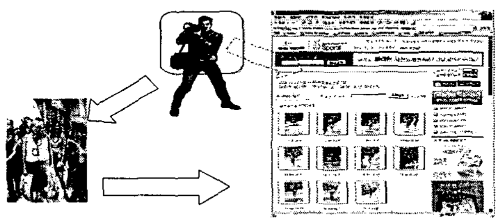


(b) Imedia's image code tag

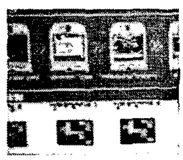


(c) Colorzip's image code

Fig. 1. Image codes.



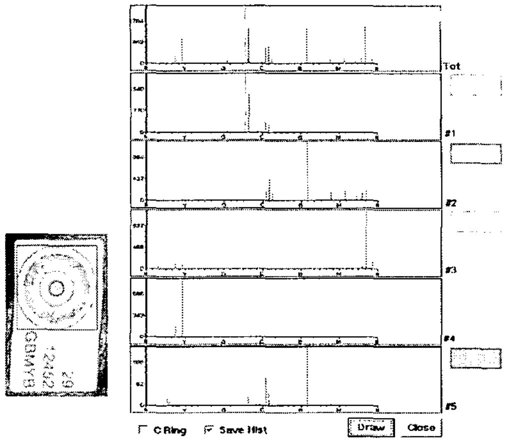
(a) Application 1



(b) Application 2

Fig. 2. Image codes and online services.

However, there are some difficulties in achieving the practical use of image codes mainly because of the severe distortion of colors. That might be one of the reasons why not many companies are launching business using image codes and why we could not find any research reports on the subject of image code recognition for which the analysis of precise color information is needed, while there are quite a few papers using rough information of colors to find specific objects like faces [9, 10, 11, 12, 13]. [Figs. 3](a) and 3(b) show an image code in a tag with a string indicating its colors and the histograms of it, respectively. The histogram clearly shows the severity of the degree of color distortion: In [Fig. 3](b), the upper-most graph shows the histogram of the hue values of the entire pixels in the circular code region, and the other graphs show the histograms of the hue values of the pixels in the individual color rings in the order of G ring, B ring, M ring, Y ring, and B ring. The boxes on the right column show the colors obtained by computing the average hue value from each histogram. Notice that the original color code was Green-Blue-Magenta-Yellow-Blue as shown in [Fig. 3] (a). [Fig. 4] shows rather a mildly obscured image codes cropped from a field image, which show examples of quite usual distortion. Though those who are using the designs of [Fig. 1] (a) and (c) have registered some patents about error checking techniques like parity check, [Fig. 4] might tell us that those designs and techniques are hardly practical and useless without having the code information appear repeatedly on the tag. In that sense, the design in [Fig. 1] (b) is the most practical among them, and, therefore, we choose it for our experiments in this paper. In next chapters, we also explain how to efficiently localize the image code and classify the colors in the code.



(a) (b)

Fig. 3. Histograms of image code regions: (a) An image code tag. (The red rectangle is merely a marker in the windows program.) (b) The histograms of the whole image code region (upper-most) and each color region. The boxes in the right column show the colors computed from the histograms.

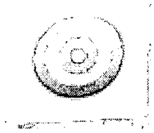
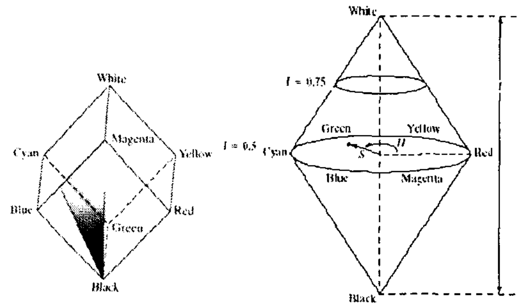


Fig. 4. Distortion of image codes in field images.

II. Background Theory

1. HSI Color Model

The color model we are using in this paper is the HSI (hue, saturation, intensity) color model. It decouples the intensity component from the color-carrying information (hue and saturation) in a color image [14]. The HSI color model is derived from the RGB cube, and [Figs. 5] (a) and 5(b) shows the relationship between the RGB and HSI color models.



(a) (b)

Fig. 5. Relationship between the RGB and HSI color models.

We see that the HSI space consists of a vertical intensity axis and the loci of color points that lie on a plane perpendicular to this axis. In this plane, the primary colors (R, G, and B) are separated by 120°, and the secondary colors (C, M, and Y) are 60° from the primaries, which means that the angle between secondary colors also is 120°. [Fig. 5] (b) shows that the hue of a color point is determined by an angle from the red axis. The hue increases counterclockwise from there. The saturation (distance from the vertical axis) is the length of the vector from the origin to the point. Note that the origin is defined by the intersection of the color plane with the vertical intensity axis. Given an image in RGB color format, the H and S components of each RGB pixel is obtained using Eqs. (1) and (2).

$$H=\begin{cases}\theta & \text{if } B \le G \\ 360-\theta & \text{if } B > G\end{cases} \quad (1a)$$

with

$$\theta = cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \quad (1b)$$

$$S = 1-\frac{3}{R+G+B}[min(R,G,B)] \quad (2)$$

It is assumed that the RGB values have been normalized to the range [0, 1].

2. k-Means Clustering

k-means clustering algorithm is an elementary but very popular approximate technique that can be used to simplify the computation and accelerate convergence [15]. k stands for the number of cluster centers, and its goal is to find the k mean vectors $\mu 1$, $\mu 2$, ..., μk. Denoting the known number of patterns as n, and the desired number of clusters k, the algorithm can be represented as shown in [Fig. 6].

```
begin initialize n, k, μ1, μ2, ..., μk
  do classify n samples according to nearest μi
     recompute μi
  until no change in μi
  return μ1, μ2, ..., μk
end
```

Fig. 6. Pseudocode of the k-means clustering algorithm.

III. Experiments

We use the image code which consists of five concentric rings as shown in [Fig. 1] (b). The color of each ring is randomly selected from six RGB colors, i.e., R, G, B, C, M, and Y. Each of these colors has the same and maximum distance from its adjacent color in the hue domain, and these colors are called the safe web colors. We used 72 field images of three different size categories, i.e., waist shot, knee shot, and full shot. In this section, we first explain how to localize the image code tag, and then how to classify the colors in it for code recognition.

1. The Image Code Localization

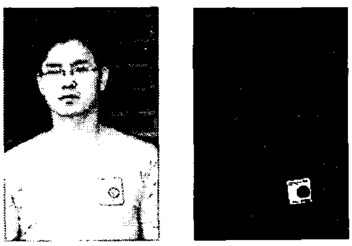
Image code localization is to detect the image code tag in each of the 72 true-colored field images of the size of 2464 x 1632 pixels. To automatically find the location of the tag and determine the size of the image code, we go through three steps: i) Threshold the image to find candidate areas of the tag using Eq. (3) [16], assuming that the image code lies on the white tag; ii) detect the image code region in the tag; iii) and classify colors.

$$g(x,y)=\begin{cases}1, & m(x,y)\geq T\\ 0, & m(x,y)<T\end{cases} \tag{3}$$

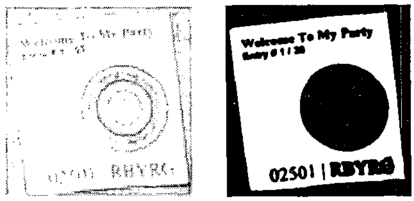
where

$$m=(f_R(x,y)+f_G(x,y)+f_B(x,y))/3$$

In the equation above, f_R, f_G, and f_B are the component images of input color image f ; g is the output image which is binary; x and y are the coordinates of pixels; and R, G, and B stand for red, green, and blue components, respectively. [Fig. 7] (a) shows a subimage cropped from a waist-shot field image and [Fig. 7] (b) shows the thresholded result of the image in [Fig. 7] (a) by using Eq. (1). The value of T was chosen based on the average intensity level. The region of image code tag in these figures is enlarged in [Figs. 7] (c) and 7 (d), respectively. The region of image code appears as a black disc in white background in the thresholded result.



(a) (b)



(c) (d)

Fig. 7. Thresholding to detect image code tag: (a) A true color image with the image code. (b) Thresholded image using Eq. (1). (c) and (d) Enlarged portions of (a) and (b), respectively.

After the thresholding, there can be multiple candidate tag areas. Determining the image code region inside each candidate tag can be divided into two stages. First, we determine the candidate center pixels of image code region using Eqs. (4)~(6)

$$g(x,y)=\begin{cases}1, & \sigma_L<T_C\\ 0, & \sigma_L\geq T_C\end{cases} \tag{4}$$

where

$$\sigma_L=\sqrt{\frac{1}{N_D}\sum_{i=1}^{N_D}(L_i-\mu_L)^2} \tag{5}$$

where

$$\mu_L=\frac{1}{N_D}\sum_{i=1}^{N_D}L_i \tag{6}$$

In the equations above, i stands for the index of N_D (= 8) search directions; L_i is the distance from a pixel inside the disc to the boundary of the disc. [Fig. 8] explains these parameters. D_i's in this figure stand for the search directions. T_C in Eq. (4) is a threshold which is inversely proportional to the number of candidate tags. Eq. (6) provides the average of distances L_i. Eq. (5) determines the standard deviation of L_i's. [Fig. 9] (a) illustrates an example of the candidate centers obtained using the above equations.

Then, we perform clustering of candidate centers to determine the center of the image code. The clustering is based on the maximum distance algorithm [16], which can be used to determine the number of clusters, and k-means clustering algorithm [15] to estimate the center of clusters based on the number of clusters. [Fig. 9] (b) shows the center of clusters obtained by the algorithm.

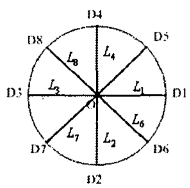
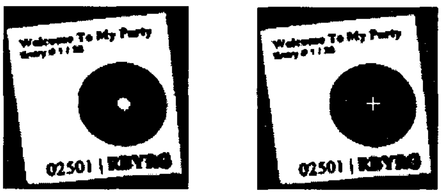


Fig. 8 Parameters in Eqs. (2)~(4).



(a) (b)

Fig. 9. Computing the center pixel: (a) Candidate centers. (b) The center of the disc (marked by the plus sign).

2. The Image Code Recognition

The image code is recognized by reading color component values of some pixels regularly selected at the distances determined by Eq. (7).

$$L_{ij}=\frac{L_i}{N_R}(j-0.5) \qquad (7)$$

Here, N_R stands for the number of rings in the image code, and i and j = 1, ..., N_R stand for the indices of search directions and rings, respectively. Hence, L_{ij} is the distance from the center pixel determined above to a point in the middle of the ring j in the i^{th} search direction D_i. A sample is taken at each L_{ij} for all i and j.

Color classification is based on the hue value in the HSI color space [14]. The hue value is calculated by Eq. (1).

The image code recognition uses NN (nearest neighbor) algorithm [16] for hue values of the image code. [tab. 1] shows the hue values of the six colors used for the rings and the measured hue values.

Tab. 1. Hue values of six safe colors used and the averages of their measured hue values

Colors	R	Y	G	C	B	M
Hue values used	0	60	120	180	240	300
Hue values calibrated	6	50	143	188	227	339

Ⅳ. Experimental Results and Analyses

We used 72 true-colored field images containing image codes of the design in [Fig. 1] (b) because it seemed most practical among the designs of currently commercialized image codes.

The image code recognition was performed in the sequence of localizing the image code, regularly sampling pixels in the code, and classifying their colors. The success rate in localizing the image code was about 96%. [Fig. 10] shows some typical cases of motion blur which led to failure in the localization.

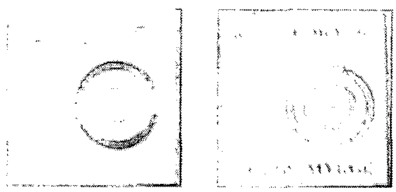
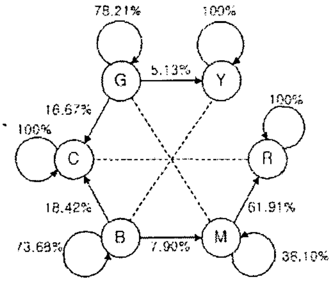
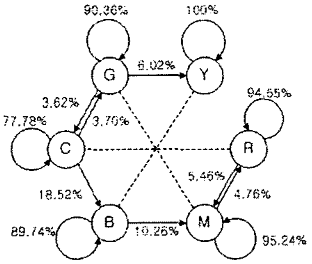


Fig. 10. Typical cases which led to failure in the image code localization due to motion blur.



(a)



(b)

Fig. 11. The state diagrams of image code classification results: (a) Without calibration. (b) With calibration (Table 1). The nodes and arrows represent colors and classification or misclassification rates obtained from 72 test images, respectively.

For color classification, the sampling using Eq. (7) and the NN algorithm were applied to the localized image code with / without a reference image code or with / without color calibration. Histogram was used to determine the representative hue value for each color in color calibration. The accuracies of color classification with reference colors, and with or without calibration were 100%, 91.3%, and 81.7%, respectively. [Fig. 11] shows the results of the latter two experiments in the form of state diagram, where the nodes represent colors and the links (arrows) represent classification or misclassification rate. The self-directed links indicate correct classification, while transition links indicate misclassification. Without calibration. the correct classification rates for R. Y. and C colors were perfect, while the rate for M color was very low. With calibration, the classification rate for M color was noticeably improved, while those for R and C colors decreased somewhat. Color calibration improved the classification rate on the whole.

[Tab. 2] shows such color statistics as the mean, standard deviation, and range of hue values obtained from the 72 test image codes. It shows that variations of G and B colors were higher than the others, which may explain why recognition accuracy for colors G and B are poor in [Fig. 11] (a). In [Fig. 11] (a) Magenta color shows poor accuracy and that might be explained by the fact that its distribution is right-shifted and overlapped with that of Red color as shown in [Fig. 3](b).

Tab. 2. Statistics of colors in 72 image codes.

Color	Mean ± Std. dev.	Range (max.- min.)
R	353.67±6.29	20(359-339)
Y	47.88±2.28	11(53-42)
G	146.38±13.47	91(194-103)
C	190.47±1.41	5(193-188)
B	215.76±12.87	65(261-196)
M	334.00±8.36	27(346-319)
R	2.65±2.11	6(6-0)

IV. Discussion

The design in [Fig. 1] (b) provides quite restricted number of combinations though the design can be the most practical in the presence of distortion such as reflection. [Figs. 12] (a) and 12(b) show alternate versions of the design to increase the number of combinations. [Fig. 12] (a) can have repeated code in different quadrant by copying the same code in every other quadrant. In that sense, the design in [Fig. 12] (b) can provide more combinations. These image codes could be located and recognized with a slight modification of the method stated above.

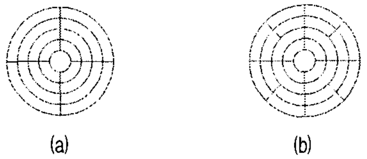


Fig. 12. Alternate versions of the design in Fig. 2 (b) to load the code redundantly.

Having taken photos in different size categories such as waist shot, knee shot, and full shot, we may derive the size or resolution restriction of image codes in images by exploring distortion caused by interference or artifacts like zipper effect between adjacent color regions. [17] We could observe that each color region should be at least 7 pixels wide in any direction to deal with such effects.

We may increase the accuracy of detecting the tag area by filtering out false candidates by inserting procedures such as performing edge detection and then searching for concentric circles inside the tag. If a candidate tag is the true one, it would contain concentric circles. Hough transform and least squares method may be employed in finding circles and determining its exact center.

It could be important to figure out the radius of code area. One of the convenient ways to help detect the outer-most circle may be to convert the image to the hue component image, transform it using such saw-tooth function as shown in [Fig. 13], and then threshold it to obtain a disc, which is white this time, corresponding to the code region. Combining the result with what we obtained by threholding a component image may help find more perfect code region [18].

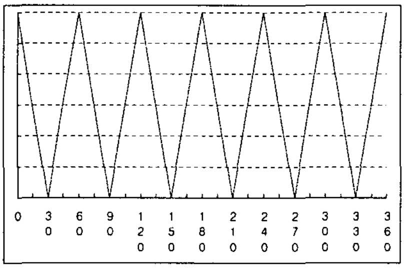


Fig. 13. The transformation function to assign the highest value to the six safe colors in the hue-component image.

V. Conclusion

In this paper, we presented a classification technique for a commercial image code of a design that would be the most practical considering the severe distortion of colors in field images. The structure of the code and its classification method have benefits in robustness for partially obscured cases and in processing time. In the experiments, we used 72 true-color field images of the size of 2464 x 1632 pixels, and it took approximately 5 seconds in localizing the tag and recognizing the image code. The localization was successful for about 96% cases. The accuracy of color classification for the localized codes with calibration of colors was about 91.3% while it was 81.7% before calibration.

For future study to reduce the number of candidate tag areas, we are planning to use additional component images, and use Hough transform or least squares method to detect features of code region.

참 고 문 헌

[1] T. Pavlidis, J. Swartz, and Y. Wang, "Fundamentals of bar code information theory," IEEE Computer, Vol.23, pp.74-86, 1990.

[2] http://www.hitl.washington.edu/artoolkit

[3] I. Robertson and I. Jaialy, "RF id tagging explained," Communications Engineer, Vol.1, pp.20-23, 2003.

[4] Ubiquitous market forecast report, ISBN: 89-89861-48-9 94500, Strategic Technology Management Institute (STEMI), Nov. 2004.

[5] S. Vakin, L. Shustov, and R. Dunwell, Fundamentals of electronic warfare, Artech House Publishers, June 2001.

[6] http://www.vividot.com

[7] http://www.colorzip.com

[8] http://print.imedia.co.kr

[9] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," IEEE Transactions on Multimedia, Vol.1, No.3, 1999.

[10] M. Jones and J. Rehg, "Statistical color models with application to skin detection," Cambridge Research Laboratory Technical Report Series, 1998.

[11] K. Sobottak and I. Pitas, "Extraction of facial regins and features using color and shape information," IEEE Int. Conf. on Pattern Recognition, Vol.3, Aug. 1996.

[12] D. Chai and K. Ngan, "Face segmentation using skin-color map in videophone applications," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 9, No.4, 1999.

[13] J. Park, J. Seo, D. An, and S. Chung, "Detection of human faces using skin color and eyes," Proceeding of IEEE Int. Conf. on Multimedia and Expo, Vol.1, pp.133-136, 2000.

[14] R. Gonzalez, R. Woods, and S. Eddins, Digital Image Processing Using MATLAB, Prentice-Hall, 2004.

[15] R. Duda, P. Hart, and D. Stork, Pattern Classification, 2nd ed., Wiley, 2001.

[16] S.-T. Bow, Pattern classification and Image Preprocessing, Marcel Dekker, Inc., 1992.

[17] R. Ramanath, W. Snyder, and G. Bilbro, "Demosaicking methods for Bayer color arrays," J. Electronic Imaging, Vol.11, pp.306-315, 2002.

[18] S. Shin and S. Choi, "Fast face detection in video using the HCr and adaptive thresholding method," Trans. of IEEK, Vol.41, No.SP-6, Nov. 2004.

저 자 소 개

김 태 우(Tae-Woo Kim) 정회원

• 1990년 2월 : 한양대학교 전자통신공학과(공학사)
• 1992년 2월 : 한양대학교 전자통신공학과(공학석사)
• 1995년 2월 : 한양대학교 전자통신공학과(공학박사)
• 1995년 3월~1997년 2월 : 한양대학교 산업과학연구소 선임연구원
• 1997년 3월~2003년 2월 : 삼성종합기술원 U-health팀 수석연구원
• 2003년 3월~현재 : 한양사이버대학교 정보통신학과 조교수

<관심분야> : 영상 처리, 생체 신호 처리, 영상 통신

유 현 중(Hyeon-Joong Yoo) 정회원

- 1982년 2월 : 서강대학교 전자공학과(공학사)
- 1991년 7월 : 미주리대학교 전기및컴퓨터공학과(공학석사)
- 1996년 5월 : 미주리대학교 전기및컴퓨터공학과(공학박사)
- 1996년 10월~현재 : 상명대학교 정보통신공학과 부교수

<관심분야> : 인공신경망, 패턴인식, 영상처리

박 흥 국(Hung-Kook Park) 정회원

- 1977년 2월 : 서울대학교 경영학과 (경영학사)
- 1985년 5월 : Claremont Graduate University(경영학 석사: MBA)
- 1990년 5월 : Claremont Graduate University (정보시스템 박사)
- 1992년 9월~현재 : 상명대학교 미디어학부 교수

<관심분야> : 멀티미디어, 임베디드 시스템