

## User Authentication System using Visual Communication

Byung-Hun Oh<sup>1</sup>, Kyung-Won Chang<sup>2</sup> and Kwang-Seok Hong<sup>3</sup>

<sup>1,2,3</sup> School of Electronic Electrical Engineering, Sungkyunkwan University, Suwon-Si, 440-746, Korea

<sup>1</sup> [sincelife@skku.edu](mailto:sincelife@skku.edu), <sup>2</sup> [kwmnusa@naver.com](mailto:kwmnusa@naver.com) and <sup>3</sup> [kshong@skku.ac.kr](mailto:kshong@skku.ac.kr)

### Abstract

*We propose a novel user authentication system using visual communication. The proposed system consists of two parts: a sender system, and a receiver system. The sender system converts a digital password into a color-coded image using a color displaying screen attached to a sender device. The receiver system detects these color-coded images using a camera, extracts the color values, and decodes them into the original digital password. We measured the data transfer accuracy of the visual communication system, and found that the average accuracy at the bit level was about 99.2% at the suggested experiment environment, to show that this system can work in real situations.*

**Keywords:** Visual Communication, Color-coded Image, User Authentication

### 1. Introduction

Security of digitally stored personal information has become an important issue in modern society, and various types of security systems are being developed [1-2]. Accordingly, many research activities have been performed on the development of image-based authentication methods to replace existing research method. These later developments include direct password input systems, wireless remote input systems, and the iris or fingerprint [3]. Image-based data communication methods are usually described in patents rather than being published as papers, and the patent numbers are given in the description below. Many methods have been proposed regarding the encoding and decoding of color codes or gray markers. However, the existing methods have several problems in illumination robustness and image recognition.

In patent [4], a method is proposed that can express data by converting the gray (QR) codes to color QR codes of red, green, and blue. It splits the data and assigns them to different color QR codes, and then transforms the QR codes into one colored QR code by overlapping them. The intention of this patent is to extend the range of data expression by changing the gray code to color codes; however, because it does not improve on the existing QR code, it has the same problems as those of the QR code. In addition, there is a limit to the amount of transferable data because the QR code has its own fixed size of code printing area. Even if it uses colors, the colors used are only red, green, and blue, so the amount of transferable data is only increased four times. Additionally, the QR code can be more sensitive to the surrounding environment because it uses colors. In some cases, colors may not be recognized correctly because of the illumination. Also, when printed on paper, differences can occur in the printed colors because of the differences in the printer quality and printing methods.

In patent [5], the proposed method concerns a two-dimensional code. The code is separated into  $m \times n$  cells, and the data are encoded by assigning different colors to each cell. This method uses a (CYM) color model, and arranges eight different colors in code cells according to certain rules. The CMY color model is commonly used in paper printing processes. Even though this method uses eight colors and the colors are arranged in a two-dimensional code, this can also have problems because the code is printed on the

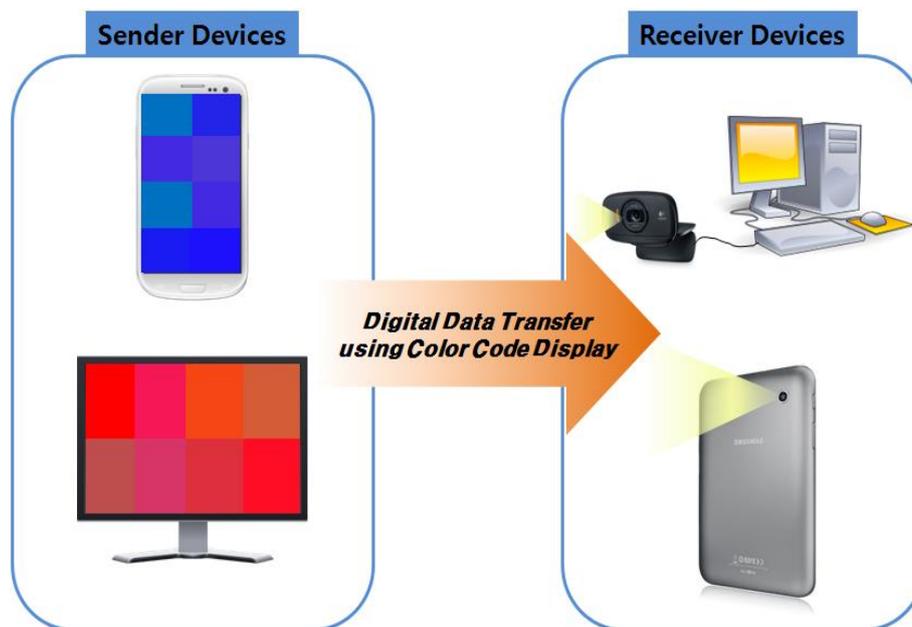
paper. The surrounding illumination or the way the code is printed can affect data transfer accuracy. Also, this method needs some special programs that show guidelines in taking code images, because the detection method for these CMY color codes is not suggested.

In patent [6], an internet address access method is suggested using code image recognition. This also uses codes arranged as  $m \times n$  cells, but gray codes (white, gray, black) are used instead of colors. Because this method makes codes using gray colors in a ternary shape, it is more robust to illuminations than the codes using colors. However, this can store three types of data per cell, and needs more area to store a large amount of data.

In order to solve the above problems, we propose a novel user authentication method that uses a red, green and blue (RGB) color code display on a subdivided mobile device screen area. While a number of existing works such as the above patents propose a similar method, our system differs in two ways: firstly, it uses 'reference colors', which can ensure the color code images are robust to illumination and secondly, it uses periodically-changing color codes to enable it to send larger password data.

## 2. System Architecture

The proposed system needs a color displaying screen attached to a sender device, and a receiver device with a camera. Devices that can receive digital data directly from the user and that have a color displayable screen are appropriate for the sender device. Personal computers and smart mobile devices can be used. The receiver device needs a camera that can continuously take images of the sender device screen, and should be able to show the received data to the user. Figure 1 shows the overall architecture of the proposed system.



**Figure 1. Overall Architecture of the Proposed System**

The sender device needs to convert digital data to RGB color codes and display color codes on the screen. When the user inputs digital data such as text or numbers, the sender device converts these data to color codes using a previously prepared coding algorithm. The sender device can transfer the converted color codes to the receiver device by periodically displaying them on the screen for short intervals.

To increase the speed of the data transfer, we used the method of dividing the device screen into several rectangular areas. If the screen is divided into an appropriate number of rectangular areas that fit exactly within the device screen size, we can transfer more data immediately by assigning different data to each rectangle area and displaying them

individually. For this process, the color to be displayed needs to be selected among red, green, and blue, considering which color would be easy to detect by the receiver device. This can help the receiver device in continuously detecting the sender device screen, even if the location of the sender device changes in the image taken. Also, if the data are divided into the number of rectangular areas and the displaying speed of the divided data can be matched to the image-capturing speed of the receiver device camera, any type of digital data can be transferred, even in very large amounts.

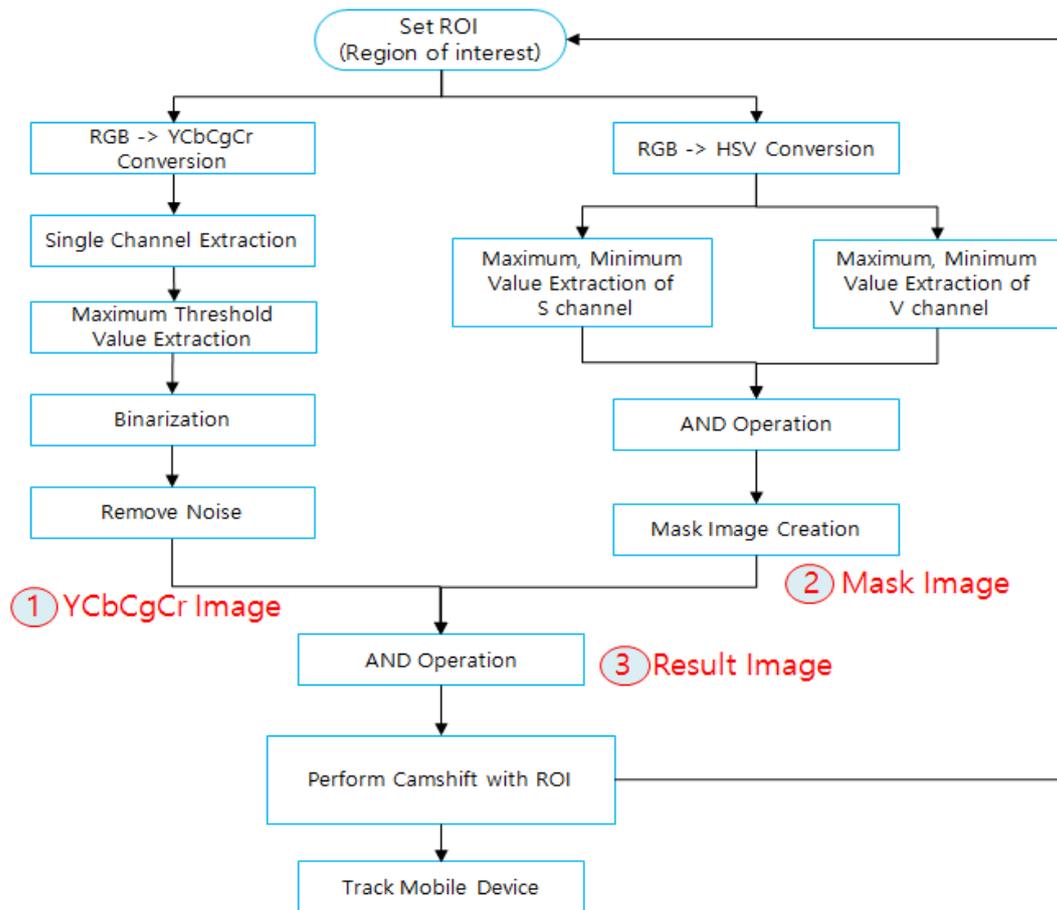
The first task the receiver device needs to handle is detecting the sender device screen in real time from images taken by the camera. Using image-based object detecting algorithms searching for the displayed colors, the sender device screen can be easily detected. When detected, the RGB color values should be extracted from all the divided rectangular areas of the screen. The extraction speed can be determined considering the displaying speed of the sender device and image processing time. The receiver device needs to extract RGB values periodically, maintaining the chosen extraction speed. The extracted color values can then be converted into digital data. When this data is reported to the user, the wireless data transfer of the proposed system is used.

### **3. User Authentication System**

#### **3.1. Mobile Device Detection Algorithm**

To detect the mobile device, we suggest using displayed colors. We can display one of the three primary colors (red, green, or blue) on the screen. We can then search for the displayed color in the camera image using object detection or tracking algorithms. For example, we can use the Camshift algorithm [7-10], which is appropriate for real-time tracking of color features. However, a process first needs to be established for setting the start location and size of the target object to use Camshift. This task cannot be performed manually each time, so we need an additional algorithm that can automatically find the location of the mobile device in the image. Thus, we assessed the displayed color values on the mobile device screen using the database, and researched the automatic detection of the mobile device start location in the image.

Sometimes, the performance of Camshift diminishes because, when detecting the rectangle while tracking, it becomes distorted or stays still in another area. The Camshift algorithm needs to be improved to eliminate these issues and increase tracking accuracy. We analyzed the reasons for these problems and tracking failure, and propose a method that can improve the performance of Camshift by combining the YCbCr color model [11] and the HSV. YCbCr is a color model that separates brightness (Y) and color differences (Cb, Cg, Cr) from the RGB color model. Figure 2 shows a flow chart of the Camshift operation, which is offered by OpenCV. If the region of interest is selected, Camshift changes the RGB color model image to an HSV color model image, and separates the image into H, S, and V channels. It then makes a histogram of the H channel image, and recreates the backproject image using a maximum frequency feature. It sets minimum and maximum threshold values in the S and V channel images, and computes an 'AND' operation with the two images, making a mask image. Finally, it tracks the target object in the result image, created by computing the AND operation with the mask image and the backproject image.



**Figure 2. Flow Chart of Revised Mobile Device Tracking Algorithm**

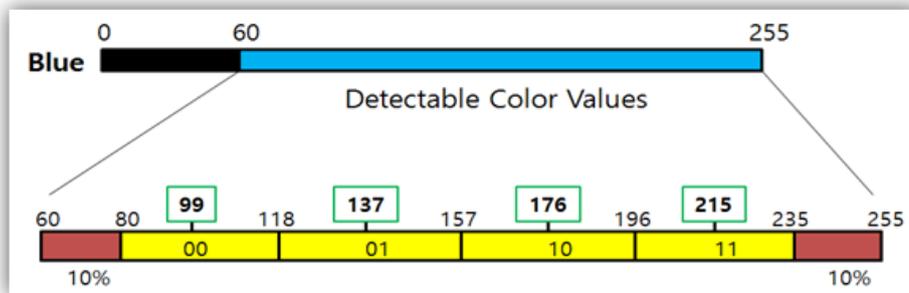
However, many tracking failures occurred when there was significant noise in the backproject image or when the colors are similar to the target object. This shows that the H channel is sensitive to noise and similar colors. Thus, we propose a method of applying the YCbCr color model, which is relatively robust, to noise and similar colors. Figure 2 shows the flow chart of the revised object tracking algorithm. Instead of using the H channel image, this algorithm changes the RGB color model image to a YCbCr image. This is performed using the RGB to YCbCr transformation equation and the RGB to YCrCb transformation equation. The algorithm then separates the image into Cb, Cr, and Y channels, and extracts the maximum threshold values. It then binarizes the image and removes the noise using a morphology operation. When this YCbCr image performs the AND operation with the mask image, the result image can make the Camshift more robust to noise and similar colors.

### 3.2. Data Encoding Algorithm

The data transfer accuracy of the proposed system is very sensitive to the way in which the encoding and decoding algorithms are organized. To design high-performance algorithms, experiments must first be performed several times with the initially designed algorithms. Several reasons for data transfer errors may become apparent in such experiments, so solutions to the problems can be found. After revising the algorithm, several iterations can be made to optimize the algorithms. In this paper, the proposed algorithms are considered to show the best performance in data transfer by repeating the experiments several times.

To detect the sender device, we used an object-detecting algorithm, which is used in the mobile device tracking interface, as mentioned above. Because the blue color showed the best detection rate in the experiments, we explain the algorithm using the color blue in this paper, but red and green can also be used in the same way.

If we set the red value to 0, the green value to 0, and the blue value to a number between 60 and 255, the object-detecting algorithm can detect the color continuously in a general office environment. Then, using the blue color, we can organize the encoding algorithm as follows. First, we divide the range of 60 to 255 into four smaller sections. Using the fact that all the digital data comprises bits (0 and 1), we assign 00, 01, 10, and 11 bit values to each section. This means that if the sender device wants to send one of those four values, it needs simply to display the representative color of the section. We determined the median colors to be representative colors of each section. The receiver device can then distinguish which bit value among the four is being sent by the sender, using a decoding algorithm.



**Figure 3. The Digital Data Encoding Algorithm**

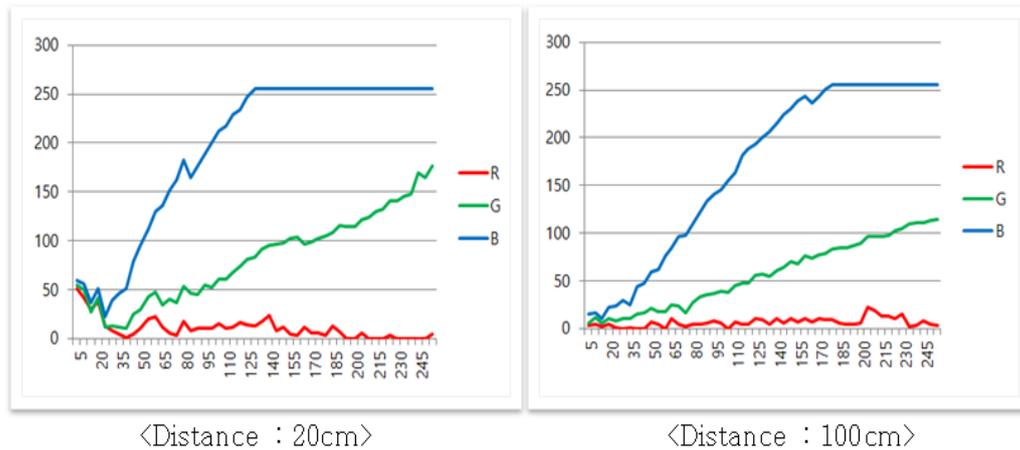
However, the received color values can be changed each time, even if the displayed colors are the same. This happens because images taken using a digital camera are very sensitive to the surrounding environment, especially illumination conditions. To address this and to make this system robust to illumination, we suggest a method using reference colors. The detectable value of the color blue is 60 to 255. We can fix two rectangular areas to display 60 and 255 only, and make the receiver device extract colors from these two areas in each frame. Using this method, we can determine degree to which the colors are being affected by the illumination. First, the receiver device should analyze how the received reference color value differs from the original reference color value. If this difference is calculated, we can then use it to check the degree to which the value of the received color is affected, if there are no illumination effects. Also, because we used two edge values in the detectable range, we can determine the range of colors extracted by the receiver device. Thus, if we divide this received color range into four sections, we can check the section to which the received colors belong. We can then convert the received color values to digital data. Figure 3 shows the structure of the designed digital data encoding algorithm.

The upper bar in Figure 3 indicates the range of the blue color value, and the brightly colored part is the range of the detectable color value. This part is expanded in the lower bar, which consists of four small sections (yellow part) that have assigned bit values. Reference colors need to be distinguished from other colors, so additional sections are provided for reference colors. The red-colored parts, which comprise 10% at each side of the lower bar, indicate the sections of the reference colors. The remaining 80% of the area is divided into four smaller sections having bit values assigned. Thus, if we select the median values of each section, then 99, 137, 176, and 215 will be the representative blue values, which will be displayed on the sender device screen when transferring digital data.

### 3.3. Data Decoding Algorithm

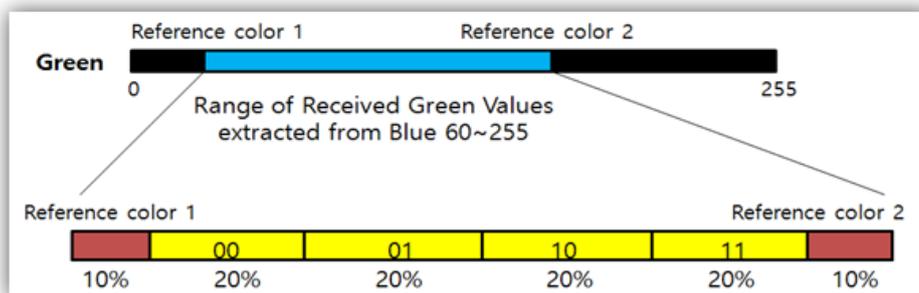
The receiver device needs a decoding algorithm that can convert extracted color values to digital data. This can be designed by taking images of displayed colors using the camera, and analyzing the RGB values of the received colors. Figure 4 shows graphs of the received RGB color variation when the displayed blue color is changed from 0 to 255.

The left graph shows the case when the sender is positioned 20 cm from the receiver, and the right graph shows the case at 100 cm distance. The horizontal axis indicates the blue color displayed from the sender device screen.



**Figure 4. The Received RGB Color Variation According to Change of the Displayed Blue Color**

When the displayed blue color value increases, the red value is fixed at under 25, and the blue value increases rapidly to 255, finally being held at 255. The green value shows a tendency to increase linearly compared to the other colors. Thus, in this paper, we judged the green color to be more useful than the blue color when designing the decoding algorithm; however, this can vary depending on the surrounding illumination and the type of receiver device. Figure 5 shows the structure of the designed decoding algorithm.

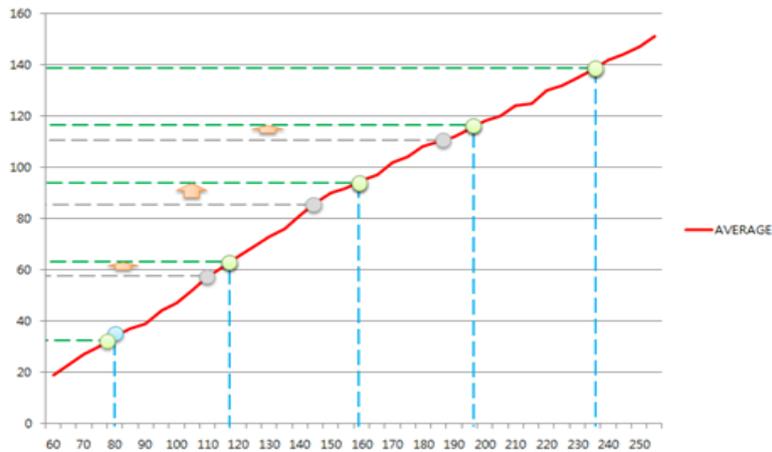


**Figure 5. The Color Code Decoding Algorithm**

While the structure of the decoding algorithm appears to be similar to that of the encoding algorithm, it does differ significantly, whereby the range of the detectable color value is not fixed for the decoding algorithm. The receiver device needs to set the range of valid green values by using the received reference colors in each frame. The upper bar in Figure 5 indicates the range of green values, and the bright part indicates the range of

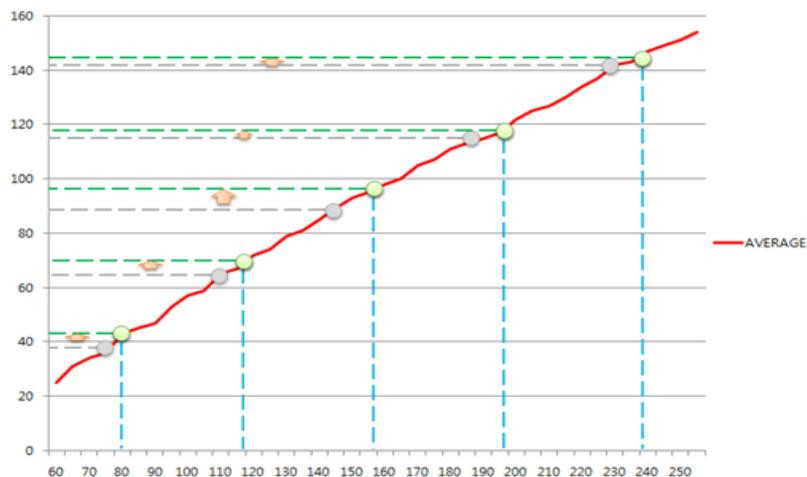
valid green values set by two reference colors. The bright part is expanded in the lower bar. The red-colored parts indicate the sections of reference colors, and the remaining 80% of the area is divided into four smaller sections of 20% area, having bit values assigned.

While the sections that have bit values comprise areas of 20%, as in the encoding algorithm, because the green value is not perfectly linear with respect to the displayed blue values, additional adjustments are needed for the ratio of section areas to increase the data transfer accuracy. Data transfer accuracy can be increased by shrinking the area where the slope of the blue-green graph is gradual, and expanding the area where the slope is steep.



**Figure 6. Graph of Average Value of Extracted Green Color in 20cm Distance**

Figures 6 and 7 show graphs of the average values of the extracted green colors compared to the displayed blue colors of the sender device screen at 20 cm and 100 cm distances. The horizontal axis indicates the displayed blue color value and the vertical axis indicates the average value of the extracted green color. The blue dotted line indicates the border between the bit value sections of the encoding algorithm and the gray dotted line indicates the border between the bit value sections of the decoding algorithm. In the two graphs, the blue and gray dotted lines are not matched at the average line and this may possibly increase the data transfer error rate. Thus, we adjusted the gray dotted line to match the point of the blue dotted line.



**Figure 7. Graph of Average Value of Extracted Green Color in 100cm Distance**

The orange arrows indicate the degree to which the line has moved, and the green dotted line indicates the adjusted borders of the bit value sections in the decoding algorithm. The new ratios in each section in the decoding algorithm are listed in Table 1 in the cases of 20 cm and 100 cm. Method 1 in the Table indicates the ratios before adjusting the borders, when the sections each comprised areas of 20% each.

**Table 1. Revised Ratio of Sections in Decoding Algorithm**

	Reference Color 1	00	01	10	11	Reference Color 2
<b>Method 1</b>	10%	20%	20%	20%	20%	10%
<b>20cm</b>	11.4%	22.0%	22.7%	17.4%	16.7%	9.8%
<b>100cm</b>	13.6%	20.5%	19.7%	16.7%	18.9%	10.6%

As a result, a maximum of 3.6% to a minimum of 0.2% is adjusted from Method 1. In the experiment, we used average values of the two cases, which were 12.5%, 21.25%, 21.2%, 17.05%, 17.8%, and 10.2%. The RGB color code transfer system, which is very robust to illumination conditions, can be designed using the encoding algorithm and the decoding algorithm proposed above.

## 4. Experiment and Implementation

### 4.1 Experimental Result

To measure the data transfer accuracy, we divided the screen of the mobile device into 10 rectangular areas as a  $2 \times 5$  window, and displayed two reference colors at the bottom row of the window. We extracted color values from the center of each divided area. The distance between the sender device and the receiver device was about 20 cm.

In the experiment, we converted four alphabetical characters to ASCII codes, and made each rectangular area display two bits in one frame. Excluding the reference colors, there was a total of eight rectangular areas, so we could send two alphabetical characters (16 bits) per frame. The speed of changing colors was set to 5 times per second, so 80 bits could be sent per second.

**Table 2. Data Transfer Accuracy of the Visual Communication System**

	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>	Average
<b>Letter</b>	99.2%	94.9%	99.7%	99.4%	99.2%	98.4%
<b>Word</b>	98.9%	87.5%	98.9%	97.9%	97.9%	96.2%
<b>Bit</b>	99.4%	97.5%	99.9%	99.7%	99.6%	99.2%

We transferred data for 1 minute in each experiment, and performed five experiments that measured the average transfer accuracy at the word, letter, and bit levels. The average data transfer accuracy at the bit level was about 99.2%. Table 2 shows the results of the experiment.

### 4.2 Implementation



**Figure 8. Implemented user Authentication System**

To test our system more intuitively, we implemented our user authentication system as shown in Figure 8, and programmed our own system using C++ and MFC. The implemented system can be divided into two parts: the user's password register and the password authentication using color-coded images.

## 5. Conclusion

We propose a user authentication system that uses a color code display on a subdivided mobile device screen area. We also proposed an encoding algorithm and a decoding algorithm that can be used in our system, and estimated the data transfer accuracy at bit, letter, and word levels using experiments. Our approach can compensate the defects of existing security systems. It is more secure, because the password that is encrypted by a specialized algorithm passes through a mobile device, and is transferred using a visible light communication method. For future work, we plan to increase the data transfer accuracy and transfer speed, so that it can be used in real environments.

## ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014023704) and This research was supported by Basic Science Research Program through NRF of Korea, funded by MOE (NRF-2010-0020210)

## References

- [1] H. Zhang, M. Fathy and R. Chellappa, "Touch Gesture-Based Active User Authentication Using Dictionaries", 2015 IEEE Winter Conference on Applications of Computer Vision, (2015), pp. 207-214.
- [2] R. Guidorizzi, "Security: Active authentication", IEEE IT Professional Magazine, vol. 15, no. 4, (2013), pp. 4-7.
- [3] O. Byunghun and H. Kwangseok, "A User Authentication System Using Color-coded Images", Advanced Science and Technology Letters (Security, Reliability and Safety 2015), vol. 93, (2015), pp. 28-33.
- [4] B. Chae, M. Kim and I. Oh, "Code overlap output system code using color channel", KR Patent, No. 1012270550000, (2013).
- [5] H. Endo, "TWO-DIMENSIONAL CODE, TWO-DIMENSIONAL CODE READER, AND PROGRAM", KR Patent, No. 1020127008099, (2012).
- [6] K. Kim and J. Kim, "Image Recognition Code for Approaching Contents, and Method and Apparatus for Approaching Contents by Image Recognition Code in Broadband convergence Network", KR Patent, No. 1020080008674, (2008).

- [7] X. Liu, H. Chu and P. Li, "Research of the Improved Camshift Tracking Algorithm", International Conference on Mechatronics and Automation, China, (2007), pp. 968-972.
- [8] J. Peng and L. Gu, "The Hand Tracking for Huamnoid Robot Using Camshift Algorithm and Kalman Filter", Journal of Shanghai Jiaotong University, vol. 40, no. 7, (2006), pp. 1161-1165.
- [9] H. Chu and S. Ye, "Object Tracking Algorithm Based on Camshift Algorithm Combinating with Difference in Frame", Proceeding of the IEEE International Conference on Automation and Logistics, China, (2007), pp. 51-55.
- [10] O. Nouar and G. Ali, "Improved Object Tracking with Camshift Algorithm", IEEE International Conference on Acoustics, Speech and Signal Processing, France, (2006), pp. 657-660.
- [11] O. Byung-Hun, J. Kyung-Won and K.-S. Hong, "Multi-object Detection and Tracking Method based on Color Information", The 24<sup>th</sup> Joint Conference on Communications and Information, (2014).

## Authors

**Byung-Hun Oh**, received the B. S. degrees in electronic engineering from the Eulji University, in 2011 and his M.S. from the Department of Electrical and Computer Engineering at Sungkyunkwan University in 2013. He is presently a Ph.D. candidate at the College Information and Communication Engineering, Sungkyunwkwan University. His current research focuses on digital image processing and pattern recognition.

**Kyung-Won Chang**, received bachelor's degree from electronic engineering department of Sungkyunkwan University in 2013, and M. S. degree from electronic and computer engineering department of Sungkyunkwan University in 2015. His researches are about image recognition, focusing on finding proper machine learning and pattern recognition algorithms.

**Kwang-Seok Hong**, received his B.S., M.S., and Ph.D. in Electronic Engineering from Sungkyunkwan University, Seoul, Korea in 1985, 1988, and 1992, respectively. Since March 1995, he has been a professor at Sungkyunkwan University, Suwon, Korea. His current research focuses on human-computer interaction, five-sense recognition, interaction, and representation.