Overview

The use of 2D barcode is more common in our daily lives. Also the prevalence of cameras in hand-held devices and electronic display creates a new opportunity for wireless communication. For these reasons, our project aims to implement an invisible 2D barcode system by using complementary colors flickering at 60Hz as signal which is unobtrusive to human eyes but can be captured by a camera with fast capturing mode. In this manner, the display can carry meaningful visual information such as text and artwork to humans while simultaneously transmit hidden message to camera.

Methodology

Algorithm to create invisible modules in 2D barcode

With reference to VRCodes[1], the color space on the right shows how hue can be broken down. By drawing a line on the CIE chart, complementary colors (color A and color B) can be found and alternating them beyond a certain frequency may produce a perceived color (center point of line). To segment the color of an image into complementary colors, RGB value of each pixel will be modified directly. For example, when a pixel is turning to green, we will reduce red and blue channel and increase green channel.

Design and Implementation

The coding system can be divided into two parts: encoder and decoder. The encoder is to generate a video barcode from an image and message. The decoder is to detect the barcode in a practical situation and retrieve the hidden message by comparing the color change of several consecutive frames. To increase the reliability, two columns of reference bits and Reed-Solomon error-correction code are applied. The encoder is implemented as a MATALB program. The decoder is implemented as an Android application.

Workflows of Invisible Barcode System

Encoding system

Character String → Mapping → Bits → ECC Encoder → Message Embedding → Video Barcode

Decoding system

Video Barcode shown on 60Hz display → Phone Camera → Image Resizing → Color Change Detection by Frame Differentiating → Barcode Detection → Perspective Correction → Color Adjustment → ECC Decoder → Bits → Message Extraction → Corrected Barcode

Results

Specification

- Display: 60Hz refresh rate
- Camera: 640 x 480 pixels with sports mode
- Maximum length of message: 48 bytes
- Software platform
  - Encoder: PC with MATLAB
  - Decoder: Android 4.1.2

Performance among different devices and conditions

Default setting: angle 0° and distance 40 cm

Fujitsu AH552 display with HTC m8

Message can also be retrieved with different settings:

- With angle ~20°
- With long distance ~70 cm

iPad as display

Sony Xperia J as camera

The screenshot of corresponding result

Message can also be retrieved with different settings:

- With angle ~20°
- With long distance ~70 cm

iPad as display

Sony Xperia J as camera

Conclusion

Our application achieves 90% successful barcode detection with default setting. The average detection time of HTC m8 is about 2.35s. The result is satisfactory and the data capacity is enough for general uses such as coupon code and short URLs. For real world application, it is useful for advertising especially for exhibition using electronic display.