

## 오디오 바이너리 파일을 컬러 QR코드로 표현하는 방법과 그 응용

이충호

한밭대학교 정보통신공학과

### A Method to Express Audio Binary Files by Color QR Codes and Its Application

Choong Ho Lee\*

1 Department of Information and Communication Eng., Hanbat National University

**요약** 본 논문은 MP3 오디오 바이너리 파일을 일련의 컬러 QR 코드로 생성하여 종이에 인쇄할 수 있는 방법을 제안한다. 또한 이 방법이 상당한 압축효과를 가져올 수 있음을 기술한다. 이 방법은 먼저, 한 개의 MP3 파일을 QR코드가 바이너리로 표현할 수 있는 최대용량으로 나눈다. 그런 다음 각각의 분할된 파일들을 흑백 QR코드들로 변환한다. 최종적으로, 분할된 파일을 3개씩 중첩하여 1개의 컬러 QR코드를 만든다. 중첩 시에 3개의 흑백 QR 코드는 각각 적색, 녹색, 청색으로 간주된다. 이 방법에서 한 개의 컬러 QR코드는 2개의 흑백 QR코드 영역이 겹쳐지는 부분은 시안(Cyan), 마젠타(Magenta), 노란색(Yellow)로 표현되며, 3개의 흑백 QR코드가 겹쳐지는 부분은 흑색, 전혀 겹쳐지지 않는 부분은 백색으로 표현한다. 실험결과 약8.5Mb의 MP3파일은 A4 용지 9페이지에 인쇄될 수 있다. 부수적인 효과로서 인쇄하지 않은 컬러 QR코드의 크기는 원래의 MP3파일보다 약 15.7배의 압축효과를 가질 수 있음을 보였다. 제안된 방법은 인터넷 액세스가 불가능한 환경에서 사용될 수 있는 장점이 있다.

• 주제어 : QR코드, 패턴인식, 데이터 압축, 오디오파일

**Abstract** This paper proposes a method to express an MP3 audio file by a series of color QR codes which can be printed on the paper. Moreover, the method can compress the data considerably. Firstly, an MP3 file is divided into many small files which have maximum capacity of binary file of a QR code. Secondly, the multiple files are converted to multiple black-and-white QR codes. Lastly, every three QR codes are combined into color QR codes. When combining, each of three black-and-white QR codes are regarded as red, green, blue components respectively. In this method, the areas of a color QR code where two QR codes are overlapped are expressed by the colors Cyan, Magenta and Yellow. And the areas where three components are overlapped are expressed by white color. Contrarily, the areas that no components are overlapped are expressed by white color. Experimentation result shows that an MP3 file with 8.5MB the original MP3 files are compressed with the compression rate around 15.7. This method has the advantage that can be used in the environments that the internet access is impossible.

• Key Words : QR code, Pattern recognition, Data compression, Audio file

Received 11 June 2018, Revised 19 June 2018, Accepted 25 June 2018

\* **Corresponding Author** Choong Ho Lee, Department of Information and Computer Engineering, Hanbat National University, 125, Dongseodae-ro, Yuseong-gu, Daejeon, Korea. E-mail: chlee@hanbat.ac.kr

## I. 서론

QR(Quick Response)는 일본 덴소사에 의하여 개발 [1]된 이후, 다양한 응용에 널리 사용되고 있다[2-4]. 그러나 그 중에서도 가장 흔히 사용되는 용도는 웹사이트 주소를 심어서 QR코드를 스캔하면 해당 사이트의 홈페이지로 인도하는 것이다. 나아가서 QR코드 스캔 시에 해당 인터넷 사이트에서 이미지를 보여 주거나 음악을 들려주거나 비디오를 보여 주는 것인데 이들 모두 네트워크를 통하여만 서비스가 가능한 것이 특징이다. 하지만, 인터넷 액세스가 불가능한 환경에서는 이런 용도로 사용할 수 없다. 이런 경우에 바이너리 파일 형식으로 된 오디오 파일이나 이미지를 QR코드 자체에 분할하여 심는 방법을 생각할 수 있다. QR코드에 정보를 심을 수 있는 용량은 숫자의 경우 7,089 바이트, 알파벳과 숫자가 혼합된 경우에는 4,296 바이트, 바이너리파일의 경우에는 2,953 바이트, 일본한자의 경우에는 1,817 바이트를 수용할 수 있다[1]. 본 논문에서 QR코드 자체에 심으려는 오디오파일인 MP3파일의 경우에는 바이너리 파일이므로 2,953바이트를 심을 수 있다. 바이너리 파일을 QR코드 자체로 변환하면 인쇄된 형태로 표현할 수 있으므로 인터넷이 없는 곳에서도 QR코드 인식기만 있으면 원래의 바이너리 파일로 변환할 수 있는 장점이 있다. 현재 QR코드를 인식하는 인식기와 생성기 등[5-12]은 많이 보고되어 있으나, 오디오파일을 QR코드 자체에 심는 실제적인 방법에 대한 연구에 대하여는 필자가 아는 한 잘 알려져 있지 않다. 본 논문은 오디오파일 형식 중 하나인 MP3(MPEG Audio Layer-3) 파일을 흑백 QR코드 자체로 표현하고 이를 [13-15]에 간략하게 제시된 방법을 프로그램으로 실제로 구현하고 이를 분석하였다. 오디오 파일은 결과적으로 컬러 QR코드로 표현되며 이를 인식하여 원래의 흑백 QR 코드로 분리하여 디코딩 프로그램으로 이를 인식할 수 있도록 구현하였다. 또한 원래의 오디오 파일을 컬러 QR코드로 표현하였을 때 수반되는 데이터 압축효과에 대하여도 기술하였다.

본 논문의 구조는 다음과 같다. 제 II장에서 MP3 파일을 분할하여 다수의 컬러 QR코드로 표현하는 구체적인 방법을 기술한다. 제 III장에서는 잘 알려진 3개의 샘플 MP3 파일을 가지고 실험 결과를 기술하고 분석한다. 마지막으로 제 IV장에서 본 논문의 내용을 요약하고 결론을 맺는다.

## II. 관련연구

### 2.1 QR코드 인코딩

QR,코드 인코딩 방법은 기본적으로 [1]에 설명되어 있으며 이를 구현하기 위한 실제 연구가 [5-11]에 다수 발표되어 있다. 이외에 텍스트를 주고 QR코드로 인코딩 하는 많은 상용프로그램들이 나와 있으며 대부분 http로 시작되는 인터넷 사이트를 인코딩하여 디코딩 후에 바로 그 사이트로 이동할 수 있는 응용프로그램들이 많이 휴대폰의 응용프로그램(app)으로 많이 나와 있다. QR코드를 컬러코드로 구현한 대부분은 장식용 위한 것이며 컬러에 정보를 포함하고 있지 않다. 컬러 코드에 정보를 포함하고 있는 것은 [13-15]에서 기술된 바 있다.

### 2.2 QR코드 디코딩

QR 코드를 디코딩하는 방법 또한 [1]에 기술되어 있으며 이를 구현하는 많은 상용프로그램이 개발되어 있으며 오픈 소스 또한 많이 공개되어 있다[5-8, 10,11]. 하지만 상용프로그램들은 대부분 문자를 디코딩하는 것으로 바이너리코드는 디코딩하지 못한다. 명령라인 형식으로 디코딩하는 것은 필자가 아는 한 발견되지 않는다.

## III. 오디오 바이너리 파일을 컬러 QR코드로 변환하는 방법

### 3.1 바이너리 파일의 분할

바이너리 파일을 QR코드로 변환하기 위하여 먼저 MP3 파일을 QR코드가 수용할 수 있는 최대 용량인 2,953 바이트씩 분할한다. 그러면 다수의 분할된 바이너리 파일을 얻을 수 있는데 그 수는 파일전체의 크기를 2,953 바이트로 나눈 몫이 되며, 나머지가 있는 경우 그 크기보다 1만큼 크다. 이 경우에 분할되어 생성된 파일들은 원래의 파일의 크기가 정확히 2,953바이트로 나누어떨어지는 경우에는 분할되어 생성된 파일 모두가 2,953 바이트의 크기를 가지나, 그 외의 경우에는 맨 마지막 파일은 2,953 바이트보다 작은 크기를 가진다. 구체적인 방법은 그림 1과 같이 C언어로 중요한 부분을 나타내었다. 여기서 noOfMps 은 분할된 파일의 개수를 나타낸다.

```
//입력: MP3 파일
//출력: QR코드로 만들기 위한 2,953바이트의 파일
noOfMps = filesize/2953;//분할된 바이너리 파일 개수
if(filesize%2953)
    noOfMps++; //여분 파일 1개 추가
fseek(fpi,0,SEEK_SET); //파일의 맨 앞으로 이동
//연속적 번호를 가진 파일로 저장한다.
for(i=0;i<noOfMps;i++)
{
    fread(Buf, 2953, 1, fpi);
    sprintf(ofilename,"mp%04d",i);
    fpo=fopen(ofilename, "wb");
    fwrite(Buf,2953,1,fpo);
    fclose(fpo);
}
```

Fig. 1. Divide an MP3-file into multiple files each of which has the size of 2953 bytes

### 3.2 컬러 QR 코드로 변환

MP3 파일을 분할하여 생성된 조각난 파일들을 QR코드로 변환하면 그 개수가 상당히 많다. 그래서 이 파일들을 3개씩 묶어서 한 개의 컬러 QR코드로 표현할 필요가 있다. 이 방법은 3개의 흑백 QR코드를 각각 적색, 녹색, 청색 성분으로 간주하여 1개의 컬러 QR코드를 만드는 것이다. 그러면 2개의 QR코드가 겹치는 경우에는 Cyan, Magenta, Yellow로 나타나며 3개가 겹치는 곳은 백색, 전혀 겹치지 않는 경우에는 흑색이 된다. 하지만 [9], [10]과 같이 빛의 삼원색의 원리와는 달리 흑백 QR코드 3개가 겹치는 영역은 흑색, 전혀 겹치지 않는 영역은 백색으로 표현한다. 구체적인 알고리즘을 C언어로 구현한 부분을 그림 2에 나타내었다. 여기서 배열 r[], g[], b[]는 흑백 QR코드를 적색, 녹색, 청색 성분으로 간주하는 경우의 화소 값을 나타내며, 배열 ppm[]은 컬러 QR 코드의 화소 값을 나타낸다.

```
//입력: 흑백 QR 코드 3개
//출력: 컬러 QR 코드 1개
for(i=0;i<rows;i++)
    for(j=0;j<cols;j++){
        //3개 QR 코드가 백색인지 체크
        if (r[i][j] && g[i][j] && b[i][j]) {
            ppm[i][j*3]=255; //적색 코드
            ppm[i][j*3+1]=255; //녹색 코드
```

```
ppm[i][j*3+2]=255; //청색 코드
        }
        //3개 QR 코드가 흑색인 경우
        else if((r[i][j]==0)&&(g[i][j]==0) && (b[i][j]==0)) {
            ppm[i][j*3]=0;
            ppm[i][j*3+1]=0;
            ppm[i][j*3+2]=0;
        }
        else { //3개 코드 중 서로 겹치는 부분
            ppm[i][j*3] = r[i][j] == 255 ? 0: 255;
            ppm[i][j*3+1] = g[i][j] == 255 ? 0: 255;
            ppm[i][j*3+2] = b[i][j] == 255 ? 0: 255;
        }
    }
}
```

Fig. 2. Make a color QR code using three black-and-white QR codes

Fig. 2에서 3개의 QR코드가 모두 백색인 부분은 백색으로, 모두 흑색인 부분은 흑색으로 한 다음, 나머지 부분은 백색은 흑색으로 흑색은 백색으로 바꾸어 PPM 파일에 저장하였다.

## IV. 실험

본 논문의 실험에 사용된 오디오파일은 Windows 7 운영체제 환경에서 음악 폴더 아래에 위치하는 MP3 파일들로서 다음과 같은 파일 3개이다. 표 1의 3개의 파일을 편의성을 위해 이하에서는 각각 ‘Kalimba’, ‘Maid’, ‘Sleep’ 라고 부르기로 한다.

Table 1. MP3 audio files used in the experimentation

파일이름	바이트 수
Kalimba.mp3	8,414,449
Maid with the Flaxen Hair.mp3	4,113,874
Sleep Away.mp3	4,842,585

컬러 QR 코드는 제 II장에서 기술된 바와 같이 같은 방법으로 구현할 수 있다. 실험 단계를 간략하게 요약하면 다음과 같다.

단계 1) MP3 파일을 2953바이트 크기를 가진 복수의 파일로 분할한다.

단계 2) 이 파일들을 참고문헌 [8]에서 제공되는 인코딩 명령어를 사용하여 일괄적으로 PNG 이미지형식의 흑백QR코드 이미지를 생성한다. 이 때 명령어와 옵션은 “QREncode -o mp0000 -8” 을 사용하였다. 옵션 “-o” 는 생성되는 QR코드 파일이름을 주기 위한 것이며 옵션 “-8” 은 바이너리파일임을 의미한다.

단계 3) 프로그램을 용이하게 작성하기 위하여 생성된 PNG형식의 QR코드를 일괄적으로 PGM파일형식으로 바꾼다. 이미지 에디터를 사용하였다.

단계 4) PGM형식의 흑백 QR코드를 상기 설명한 방법으로 3개씩 묶어서 컬러 QR코드로 만든다. 단, 흑백 QR코드의 개수가 3의 배수가 아닐 경우에는 3개씩 변환하고 남은 파일들만 겹쳐서 PPM 이미지형식의 컬러 QR코드를 만든다.

단계 5) PPM형식의 컬러 QR코드 파일들을 다시 일괄적으로 PNG형식의 파일로 변환한다.

단계 1에서 분할된 파일의 개수는 ‘Kalimba’, ‘Maid’, ‘Sleep’ 에 대하여 각각 2,850개, 1,394개, 1,640개이다.

단계 2에서는 분할된 파일의 수만큼 인코드 명령어를 사용하여 반복적으로 PNG파일로 생성하였다. Fig. 2에 이 단계를 수행하는 프로그램을 나타내었다. 여기서 argv[1]과 argv[2]는 각각 시작 파일의 번호와 마지막 파일의 번호이다. 예를 들어 ‘Kalimba’ 의 경우에는 시작파일의 번호와 마지막 파일의 번호는 각각 0과 2,849이다.

```
//입력: 흑백 QR코드 이름의 시작번호와 끝번호
//출력: 3개씩 결합하여 생성된 컬러 QR 코드
//명령라인을 이용하여 바이너리 MP3파일을 QR 코드로 인코딩
from = atoi(argv[1]), to = atoi(argv[2]); //
for (i = from; i <= to; i++) { //명령어 반복 실행
printf(str,"QREncode -o mp%04d.png -8 mp%04d\n", i, i);
system(str);
}
```

Fig. 3. Method to encode the divided files repeatedly

생성된 흑백 QR코드 이미지들의 실제크기는 표 2와 같이 가로와 세로가 63x63으로 일정하였고 바이트 수는 100바이트 전후의 약간의 차이가 있다. 실제 가로와 세로 크기가 일정하기 때문에 3개씩 중첩하여 컬러

QR코드를 생성할 때에 크기를 확대하거나 축소하여 맞추는 필요가 없어 용이하다. 표 2는 ‘Kalimba’ 가 분할된 파일로 생성한 PNG 이미지형식의 흑백 QR코드 12개를 나타낸 것이다.

Table 2. Black-and-white QR codes generated from binary MP3 files.

파일명	바이트수	QR코드 크기 (화소수)
mp0000.png	793	63x63
mp0001.png	722	63x63
mp0002.png	756	63x63
mp0003.png	752	63x63
mp0004.png	759	63x63
mp0005.png	785	63x63
mp0006.png	682	63x63
mp0007.png	767	63x63
mp0008.png	766	63x63
mp0009.png	770	63x63
mp0010.png	769	63x63
mp0011.png	778	63x63

그 다음으로 실제 프로그램을 용이하게 하기 위하여 PNG 이미지 형식의 파일을 이미지 변환툴을 이용하여 PGM파일의 형식으로 일괄 변환하였다. 그 다음 PGM 파일을 3개 읽어 들여 한 개의 PPM파일의 각각 적색, 녹색, 청색 성분으로 이용한다.

Table 3에서 ‘Kalimba’ 는 8,414,449 bytes의 크기를 가지며 이것을 2,953 bytes씩 분할하면 2,850개로 분할된다. 이렇게 분할된 파일들을 PNG 이미지형식의 QR코드 파일로 변환할 경우 한 개의 크기는 대략 800bytes이하로 표현되며 전체 크기를 합산하면 2,194,375bytes이다. 이 PNG형식의 이미지파일을 PGM 파일로 변환하여 3개씩 묶어서 컬러 QR코드로 만들었다. 이 경우에 PGM 총 파일용량은 3,782,900 bytes이다. 결과적으로 생성된 컬러 QR코드의 총용량은 11,324,000 bytes이다. 이것을 다시 PNG형식으로 변환하면 총파일의 크기는 533,835 bytes이다. 결과적으로 얻게 된 압축률은 원본 MP3 파일의 바이트 수를 최종적으로 생성된 컬러 QR코드의 바이트 수로 나눈 것이다. 예를 들면 ‘Kalimba.mp3’ 파일의 경우  $8,414,449/533835=15.8$ 이다. 3개의 샘플 오디오파일에 대한 상세한 결과는 표 3과 같다. PNG파일에서 PGM

파일로의 일괄변환과 생성된 PPM파일에서 PNG로의 일괄 역변환은 상용 이미지 변환 툴을 사용하였다.

Table 3. Black-and-white QR codes generated from binary MP3 files

MP3 파일이름	바이트 수 (MP3)	흑백 QR코드 바이트 수 (png)	컬러 QR코드 바이트 수 (png)	압축률
Kalimba	8,414,449	2,194,375	533,835	15.8
Maid	4,113,874	1,072,307	261,887	15.7
Sleep	4,842,585	1,262,696	307,500	15.7

Fig. 3은 ‘Kalimba’의 첫 3개의 흑백 QR코드 (a), (b), (c) 3개를 중첩하여 한 개의 컬러 QR코드 (d)를 생성한 것을 나타낸다.

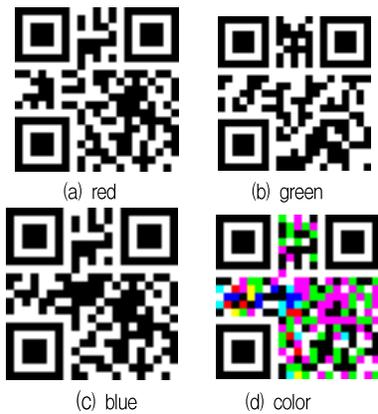


Fig. 4. Three black-and-white QR codes and an encoded color QR code

3개의 샘플파일을 이용하여 생성된 흑백 QR코드와 컬러 QR코드의 개수는 표 4와 같다. 생성된 한 개의 컬러 QR 코드는 가로세로 2cm x 2cm 크기이며 이것을 21cm x 29.7cm 크기의 A4용지 위에 인쇄하는 경우 가로 3cm, 세로 3.7cm의 여백을 둔다면, 117 (=9 x 13) 개를 1면에 인쇄할 수 있다. 그러면 ‘Kalimba’는 약 8.5 MB을 9 (=950/117) 면에, ‘Maid’는 4 (=465/117) 면에, ‘Sleep’는 5 (=7/117) 면에 인쇄 가능하다.

Table 4. Number of QR codes generated

파일이름	바이트 수	흑백 QR코드 개수	컬러 QR코드 개수
Kalimba	8,414,449	2850	950
Maid	4,113,874	1394	465
Sleep	4,842,585	1640	547

인쇄된 컬러 QR 코드를 휴대폰으로 촬영한 것을 3개의 분리된 흑백 QR코드로 분리한 것은 Fig. 5에 나타내었다. 실제 크기는 1900x1900 화소이며 이를 128x128화소로 축소하여 나타낸 것이다.



(a) color QR code (b) red QR code



(c) green QR code (d) blue QR code

Fig. 5 A color QR code captured by cell phone. Original size is 1900x1900 pixels

한 개의 컬러 QR코드를 3개의 QR코드로 분리하는 방법은 결합 시의 역과정으로 쉽게 구현된다. 구체적 인 C언어 코드는 Fig. 6에 나타내었다.

```
//입력: 한 개의 컬러 QR 코드
//출력: 3개의 흑백 QR 코드
for(i=0;i<rows;i++)//배경색 흰색으로 설정
    for(j=0;j<cols;j++)
        qr1[i][j]=qr2[i][j]=qr3[i][j]=255;
for(i=0;i<rows;i++)
    for(j=0;j<cols;j++)
    {
```

```

r=cqr[i][j*3]; //red 성분
g=cqr[i][j*3+1]; //green 성분
b=cqr[i][j*3+2]; //blue 성분
if (r>128 && g>128 && b>128)
//rgb성분 각각 백색 2치화
//qr1, qr2, qr3는 흑백 QR코드 지칭
qr1[i][j]=qr2[i][j]=qr3[i][j]=255;
else if (r<128 && g<128 && b<128)
//rgb 성분 흑색 2치화
qr1[i][j]=qr2[i][j]=qr3[i][j]=0;
else if (r>128 && g<128 && b<128)
qr1[i][j]=0; //red
else if (r<128 && g>128 && b<128)
qr2[i][j]=0; //green
else if (r<128 && g<128 && b>128)
qr3[i][j]=0; //blue
else if (r>128 && g>128 && b<128)
qr1[i][j]=qr2[i][j]=0; //yellow
else if (r>128 && g<128 && b>128)
qr1[i][j]=qr3[i][j]=0; //magenta
else if (r<128 && g>128 && b>128)
qr2[i][j]=qr3[i][j]=0; //cyan
}

```

Fig. 6. The procedure a color QR code divided into three QR codes

Fig. 6에서 컬러 QR코드의 영역에서 백색인 부분은 3개의 흑백 QR코드를 모두 백색으로 하고, 흑색인 부분이 있으면 3개의 QR코드도 모두 흑색으로 한다. 그러나 2개 이상이 겹쳐져서 컬러인 영역에서는 각각의 색상에 따라 해당되는 화소를 3개의 흑백 QR코드에서 흑색으로 표시하여 출력한다. 그렇게 함으로써 1개의 컬러 QR코드로부터 3개의 흑백 QR코드를 얻을 수 있다.

## V. 결론

오디오 파일을 컬러 QR코드로 표현하는 방법을 기술하였다. MP3 파일형식의 오디오 파일을 QR코드가 수용할 수 있는 바이너리파일 최대 용량인 2,953바이트로 분할하여 복수의 파일을 만들고 이것들을 모두 흑백 QR코드로 변환하였다. 그 다음 이들 파일을 3개씩 묶어서 컬러 QR코드로 표현하는 방법을 기술하였다. 또한 실제로 인쇄된 컬러 QR코드를 읽어 들여 각각 3개의 흑백 QR코드로 분리하고 이를 읽어 들이는 실험을 통하여 본 방법의 유효성을 보였다. 인코딩과

정에서 3개의 MP3 파일을 컬러 QR코드로 표현 시 이들의 각 단계에서의 용량을 계산하여 파일로 저장 시에는 약 15.7 압축 효과가 있음을 보였다. 또한 실험에 사용된 MP3 파일들은 8.4MB, 4.1MB, 4.8MB인데 A4용지 종이 위에 인쇄되는 경우에 각각 9면, 4면, 5면에 인쇄될 수 있음을 보였다. 실험을 통하여 제안된 방법이 인터넷을 접근할 수 없는 환경에서 오디오 파일을 컬러 QR코드 자체로 표현하고자 하는 응용에 활용 가능함을 보였다.

## ACKNOWLEDGMENTS

This research was supported by the research fund of Hanbat National University in 2014.

## REFERENCES

- [1] ISO/IEC 18004: 2000. Information technology-automatic identification and data capture techniques-Bar code Symbology QR Code. 2000.
- [2] C. Panyindee, C. Pintavirooj, "QR Codes Application for Reversible Watermarking Alogorithm in Biomedical Images," The 6<sup>th</sup> 2013 Biomedical Eng. Int' l Conf., pp. 1-4, 2013.
- [3] A QR-Code Localization System for Mobile Robots: Application to Smart Wheelchairs, 2017 European Conf. on Mobile Robots, pp. 1-6, 2017.
- [4] K. Zhao, C. Lee, "Short Research for Watermarking with QR Code in Space Domain," 2013 Spring Conf. of Multimedia Society, p. 205. 2013.
- [5] <http://www.qrcode.com/en/aboutqr.html>
- [6] <http://www.libpng.org/pub/png/libpng.html>
- [7] <http://zbar.sourceforge.net/index.html>
- [8] <http://code.google.com/p/zxing/>
- [9] P. Sutheebanjard, W. Premchaiswadi, "QR-Code Generator," 2010 8th Int' l Conf. on ICT and Knowledge Engineering, pp. 89-92, 2010.
- [10] <http://fukuchi.org/works/qrencode/index.en.html>
- [11] <https://github.com/josephholsten/libdecodeqr>
- [12] Y. Liu, M. Liu, "Recognition of QR Code with Mobile Phones," 2008 Chinese Control and Decision Conference, pp. 203-206, 2008.

- [13] C. Lee, "A Compound Color QR Code," The 9th International Conf. on Multimedia Info. Technology & Applications(MITA 2013), p. 141, 2013.
- [14] I. Oh, Code Overlap Output System Code Using Color Channel, Patent Reg. No. 10-1227055, Rep. of Korea, 2013.
- [15] C. Lee, "A Method to Express Audio Files by Color QR Codes," in Proc. of Intl' Conf. on Elec., Info. and Comm. (ICEIC) 2016, pp. 887-888, Danang, Vietnam, January 2016.

---

저자 소개

---

이 충 호 (Choong Ho Lee)



1985년 2월 : 연세대학교

전자공학과(공학사)

1987년 2월 : 연세대학교 대학원

전자공학과(공학석사)

1998년 3월 : 도호쿠대학 대학원

정보과학연구과(공학박사)

1987년 2월~2000년 2월 : KT

멀티미디어연구소 전임연구원

2000년 2월~현재 : 한밭대학교 정보통신공학과 교수

관심분야 : 디지털신호처리, 영상처리, 패턴인식,

인공지능, 응용소프트웨어