Journal of Theoretical and Applied Information Technology
20th March 2015. Vol.73 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUDIO DATA AS AN ENCODING FOR COLOR QUICK RESPONSE CODE (CQR) 3D BARCODES

[1]JUDE J.L. MARTINEZ
[1]Binus International – Binus University, Faculty of Computing and Media, Computer Science Program
E-mail: [1]jmartinez@binus.edu

ABSTRACT

The goal of this paper is to discover and implement a method of encoding an audio data from a MIDI audio file format into a special type of Quick Response (QR) code 2D barcode called Color Quick Response (CQR) code which is a 2D barcode that has been extended into a 3D barcode system. The benefit of this research paper is; to show and explore the potential of using CQR code as a medium to encode audio files, which open the possibility to encode audio messages into a 2D barcode. By proving that it is possible to encode an audio data into a 2D barcode, it shows that there are unexplored potential uses for 2D barcodes other than just for typical Audio ID Date Capture (AIDC) uses or the more current trend mobile tagging, this research can also be a stepping stone for other future mass media applications wherein people can embed audio messages into 2D barcodes for mass media purposes. Utilizing 2D barcodes to encode audio data may also have the potential to counter audio piracy.

Keywords: *2D barcode, Quick Response (QR) code, Auto ID Data Capture (AIDC), Audio Encoding, Compression*

1. INTRODUCTION

For years bar codes have been promoted as a machine readable license plate. Each label contained a unique serial number coded in black and white bars that was a key into a database containing detailed information. That was the expert wisdom. Yet many end users wanted to code more information. They wanted the barcode to be a portable database rather than just a database key [1]. With this issue introduced to the users, 2D (2 dimensional) barcodes comes into play.

According to a research conducted, the 2D barcodes main application field is mobile multimedia content consumption with particular reference to advertising; according a typical use case, a user: 1. finds a barcode printed on a magazine, for instance in an advertising; 2. launches the application on the mobile phone and scans the barcodes; 3. retrieves a URL form a WAP pages and watches a video clip streamed over 3G network.[2].

2D barcode or two-dimensional barcode contains more information than conventional one dimensional barcodes. As more data are encoded in a one dimensional barcode, the length of the barcode increases. 2D barcodes have data encoded in both the horizontal and vertical dimensions. With 2D barcode, the size of the code can be increased both horizontally and vertically to increase the number of data encoded, therefore it is more efficient in terms of space usage in relation to the amount of data that can be stored within the code.
(Marketwire - October 5, 2009) – "A trend is growing throughout North America as marketers and media providers are using 2D (two-dimensional) barcode technology to make their traditional printed media more interactive." [3] In 2009, 2D barcodes have been used for marketing. In Japan, 2D barcodes have been commonly used for mobile tagging.

Color Quick Response Code or CQR Code for short, which is a 2D barcode that has been extended into a 3D barcode system. By implementing color patterns (instead of just black and white), utilizing mobile phone cameras, and data compression, more data into the 2D barcode is added.CQR Code is an extended version of the popular QR Code. 3D barcodes contain data not only in its x and y axis, but also in depth. By stacking the QR Codes we can create a new barcode system that can store more data than the existing QR barcode systems. Through the CQR barcode system, we can embed data that normally could not be done using existing barcode systems [4].

Journal of Theoretical and Applied Information Technology
20th March 2015. Vol.73 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The author would like to discover the possibilities of and implementation of a method for encoding an audio data from a MIDI audio file format into a CQR code 3D barcode, thereafter, creating a prototype application that is able to recognize and decode the content of a CQR Code 3D barcode. In further discussions, we will see that 2D barcode comes in many variations and standards, which later we discover that some are proprietary. The focus of this paper will revolve on a special type of 3D barcode called "CQR code" also known as "Color Quick Response code" [4].

2. PROBLEM ANALYSIS

The discussion will include two (2) standards of 2D barcodes, Data Matrix and QR based on researches. For the sake of the research focus, only these two (2) widely used standards are chosen for discussion from each barcode category. In addition, the research conducted by Kosala and Nurwono [4] regarding Color Quick Response Code for Mobile Content Distribution will be discussed since the authors used the technology to encode the audio data from a wav audio file format into a special type of Quick Response (QR) code 2D barcode called Color Quick Response (CQR) code 3D barcode.

2.1 Analysis of Data Matrix 2D Barcode Standard

Data Matrix is a very efficient, two-dimensional (2D) barcode representation that uses a small area of square modules with a unique perimeter pattern, which helps the barcode scanner determine cell locations and decode the symbol [19]. Data Matrix code allows the encoding of characters, numbers, text and actual bytes, including Unicode characters and photos. Data Matrix can encode data up to 800 characters confidently; this means that most camera based imagers and hand-held scanners will be able to read the code smoothly. The amount of data encoded within Data Matrix varies depending on the type of data. Data Matrix barcodes are made up of black and white rectangular cells arranged in rectangular or square patterns. Data bits that are encoded and mapped to a region of black and white cells are called a data region. The data region is surrounded by a finder pattern; the bottom and left hand side only contain black cells, while the top and right hand side contains only white cells.

All current implementations have been standardized on the ECC200 error correction method, which is approved by ANSI/AIM BC11 and the ISO/IEC 16022 specification. ID Automation 2D Data Matrix barcode products all support ECC200 by default and are based on the ANSI/AIM BC11 and the ISO/IEC 16022 specifications. The Reed-Solomon error correction algorithms of ECC200 allow the recognition of barcodes that are up to 60% damaged [3]. Therefore the security of Data Matrix is still quite reliable even if it partly damaged.

Data Matrix barcode is considered as one of the smallest and dependable barcode standard. Data Matrix code is about 30 times smaller than a Code 39 barcode when representing the same amount of data. In a study at The Center for Automatic Identification at Ohio University, the statistical probability of a misread error with Data Matrix is 1 in 10.5 million scans, compared to a misread error probability of 1 in 1.7 million with the Code 39 barcode [5]. Data Matrix become a standards in the US and Europe.

2.2 Analysis of QR (Quick Response) 2D Barcode Standard

QR Code is a 2D barcode created by a Japanese corporation Denso-Wave in 1994. QR code also known as quick response code. QR codes are common in Japan and are currently the most popular type of 2D barcode.

QR Code has been accepted internationally and the standardization can be seen in below.

Table 1: QR Code Standardization [6]

QR Code Standardization	
October, 1997	Approved as AIM International (Automatic Identification Manufacturers International) standard (ISS – QR Code)
March, 1998	Approved as JEIDA (Japanese Electronic Industry Development Association) standard (JEIDA-55)
January, 1999	Approved as JIS (Japanese Industrial Standards) standard (JIS X 0510)
June, 2000	Approved as ISO international standard (ISO/IEC18004)
November, 2004	Micro QR Code is Approved as JIS (Japanese Industrial Standards) standard (JIS X 0510)

Initially, QR codes are used for tracking parts in vehicle manufacturing. As time progress, the use of QR code increases widely. QR codes are

Journal of Theoretical and Applied Information Technology
20th March 2015. Vol.73 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

used in commercial tracking applications. In Indonesia, one of the mass media newspapers called "Kompas" have often used QR codes on some articles for mobile tagging. Blackberry devices also uses the QR code as a media to embed user contact details and allow contact details exchange between Blackberry users by scanning the QR code that the device generate using the camera. QR codes are also one of the smallest and dependable 2D barcode. The size of a QR code is approximately 30 times smaller than a Code 39 barcode when encoding the same amount of data [6].

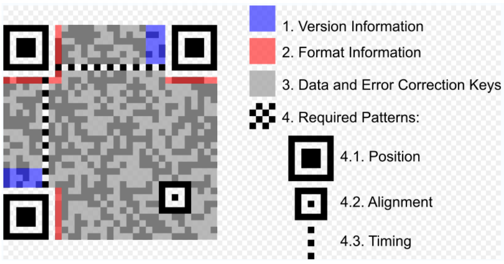

Figure 1: Structure of a QR Code [6]

QR codes are square shaped. On three corner of the square is a square bull-eye pattern that determines the position of a QR code. This allows the QR code to be scanned in any orientation but still giving a correct result. A more thorough discussion of the structure will be provided in the later chapter.

Table 2: QR Code Measurement Specifications [5]

Symbol size	21 × 21 - 177 × 177 modules (size grows by 4 modules/side)	
Type & Amount of Data (Mixed use is possible.)	Numeric	Max. 7,089 characters
	Alphanumeric	Max. 4,296 characters
	8-bit bytes (binary)	Max. 2,953 characters
	Kanji	Max. 1,817 characters
Error correction (data restoration)	Level L	Approx. 7% of codewords can be restored.
	Level M	Approx. 15% of codewords can be restored.
	Level Q	Approx. 25% of codewords can be restored.
	Level H	Approx. 30% of codewords can be restored.
Structured append	Max. 16 symbols (printing in a narrow area etc.)	

The size of a QR code depends on the size of its modules. Modules are smaller square, either black or white, that made up the data region of a QR code. The amount of data can be encoded depends on the QR code version. QR code version 1 means that the QR code size is of 21x21 modules and there are total of 40 versions. Each increase in version, it means an increase of 4 modules per side.

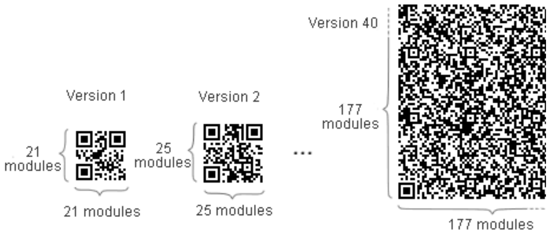

Figure 2: QR Codes Version Increment

2.3 Analysis of CQR (Color Quick Response) 3D Barcode

Color Quick Response Code or CQR Code for short, which is a 2D barcode that has been extended into a 3D barcode system. By implementing color patterns (instead of just black and white), 3D barcodes contain data not only in its x and y axis, but also in depth. By stacking the QR Codes we can create a new barcode system that can store more data than the existing QR barcode systems. Through the CQR barcode system, we can embed data that normally could not be done using existing barcode systems. The concept, which is shown in Figure 3, shows that by stacking QR Codes, CQR Code can store more data than other existing code systems. The target is to create a printable code system that could embed mobile contents into it. [4]

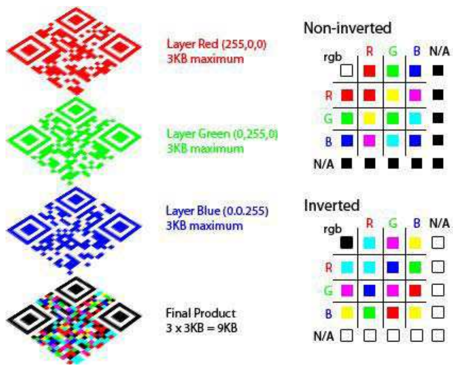

Figure 3: Color QR Code (3D barcode) concept

Journal of Theoretical and Applied Information Technology
20th March 2015. Vol.73 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. SOLUTION DESIGN

3.1 General Design and Architecture of Prototype Application

Overall architecture of the application can be divided into 3 main components. The first component is an application prototype that will take an audio file as an input. The audio file input will then be converted into another file format and the content of this new file format will be encoded into a QR code. The second component is the QR code decoder application prototype, this application prototype is essential as this will allow the author to decode an input image file, which should be a recognizable QR code. The third component is an application prototype that will try to reconstruct a file identical from the output of the first application with the input from the decoded message. Figure 4 summarizes the overall process of the prototype's applications.

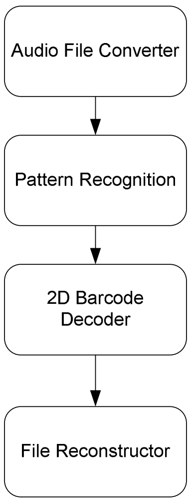

Figure 4: General Application Process Flow

The first step of the process will be an application that will take an audio file as an input. The audio file will then be converted into a comma separated value (CSV) file format, containing the string representation of the audio binary data. MIDI files are stored in a compact binary form intended to optimize speed (and hence reduce latency) on limited speed MIDI channels, while not overly taxing the processing power of the humble microcontrollers employed by instruments in the epoch when MIDI was developed. While well-suited to its intended purposes, MIDI files are somewhat difficult to read and write without a special-purpose library, The midicsv and csvmidi programs [8] allowed the author to use text processing tools to transform MIDI files or create them from scratch. midicsv converts a MIDI file into a Comma-Separated Value (CSV) text file, preserving all the information in the original MIDI file. The format of the CSV file is deliberately designed to facilitate processing by text manipulation tools. The csvmidi program reads CSV files in this format and creates an equivalent MIDI file. Processing a MIDI file through midicsv and then csvmidi will result in a completely equivalent MIDI file.

The second and third step of the process is combined in one CQR code decoder application. First the application will request the user to open a file to be decoded. The file should be an audio file (MIDI). Then the file will be checked for pattern recognition, this step will ensure that the input file is a workable CQR code. After the pattern recognition, the input file will then be decoded. The result of the decoded CQR code will be displayed after decoding process completes. The final step of the process is an application that will take the decoded message from the CQR code and from these output, the application will try to reconstruct them into a file that must be identical to the output file of the first application. Both files from the output of the first application and the file output from the third application must have identical data content in order to satisfy the concept proposed.

3.2 General Overview of Solution Implementation

To better understand the solution implementation proposed by the authors, figure 5 explains the general overview of the solution design. The whole solution involves 4 stages of different processes and also involves some third party applications. The first and fourth stage is done using the application prototypes that the author developed. The second stage of the solution involves using a third party CQR code generator created by Kosala and Nurwono [4], which is mainly used to generate CQR barcodes. The third stage of the application is an application developed by the latter; this application is a CQR 3D barcode decoder application. Altogether, the four different stages will simulate the solution implementation proposed by the author.

Journal of Theoretical and Applied Information Technology
20th March 2015. Vol.73 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

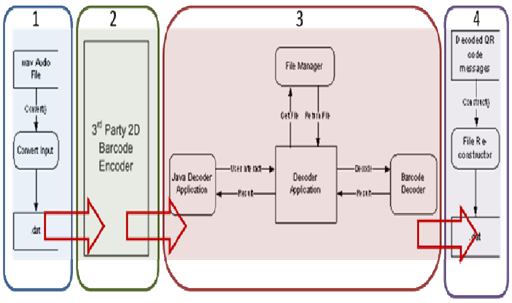

Figure 5: General Overview of Solution Implementation

Figure 5 shows the general overview of the solution proposed by the author. The solution is divided into 4 main stages. The first stage, highlighted in blue, is called the file conversion stage; it is where an audio input file is converted into a Comma-Separated Value (CSV) file, which will be used for the second stage of the solution. The second stage of the solution, highlighted in green, is called CQR code encoding stage. The output file from the first stage is processed and the data content of the output file will be encoded into CQR codes with the help of a third party application. Once the CQR codes are generated, these CQR images will be used in the third stage. The third stage is called the CQR decode process, this stage will take all CQR images generated in the second stage, and the application will decode the underlying data. The decoded data will be stored in temporary CSV files, which will be used in the fourth stage of the solution. The fourth stage is called file reconstruction stage; this stage take all temporary files from stage three as inputs and combines all of the contents the files into a single CSV file which must be identical to the output file from the first stage.

3.3 Testing

In this stage, the implementation of the three different application prototypes will be tested. The criteria for each of the different application prototype are displayed in the below tables.

Table 3: Audio File Converter Testing Result

No	Input File Size	Output File Size	Number of Characters
1	124 Kb	416 Kb	302,058 Characters
2	292 Kb	880 Kb	607,321 Characters
3	298 Kb	968 Kb	690,926 Characters

Table 3 describes three audio file samples of different configurations and the result after the file conversion. The converted file becomes larger due to some string manipulation in order to ease the retrieval of strings for barcode encoding. The decoder prototype application runs properly according to the proposed design. The decoder program is able to open an image file from a local disk, if the input file is a proper CQR code image, the application will successfully decode the underlying data. If the image file is not recognized as a CQR code, user will be notified. The third application's role is to reconstruct the decoded message, the output file from this application is as desired. It creates an output file of .csv format.

3.4 Method for Proof of Concept

The aim of this research is to prove that there is a method to encode audio data into a CQR code. Therefore a proving method is required as a part of the testing and implementation phase. The proving method is to take output file from the pre-process application and the output file from the post-process application and compare both output files. The content of both files will be compared and if both files contain identical data, this will become the proof of concept. An audio file of .MIDI format contains huge amounts of data when converted into its binary form. Only a section of the converted file will be taken as a sample for testing. The author decided to take 9,216 x 5 characters from the output file of the pre-process application as a sample. From the 9,216 x 5 characters, this sample data will be converted into 5 CQR codes. The author decided to take 9,216 characters because a CQR code can only encode maximum of 9,216 numbers of alphanumeric characters.

The 5 CQR codes will be generated using a third party program; it will then be decoded using the decoder application prototype. The decoded data from the CQR codes will be taken and stored into a CSV file, where the 5 different chunks of data will be appended into a single CSV file. Theoretically speaking, the data decoded from each of the CQR codes must have identical values to the output file from the pre-process application. Therefore, the data chunks that had been appended into a CSV file will be re-constructed on the third application. After the re-construction of the decoded data, then the output file from the conversion application and the output file from the re-construction application will be compared. If both files contain identical data, the proof of concept is successful.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. DISCUSSION

4.1 Major Observation

The goal of the proposed solution is to prove the concept of encoding audio data into a 3D barcode, thus exploring new type of application that had not been done before. In this subchapter, a general observation of what had been accomplished and issues that had been handled and need to be handled in the future will be done.

In the beginning of the testing phase, the author decided to directly convert the MIDI file into its binary format. Then with the new file containing the byte data of the audio file, the author intended to use the data and encode them into the CQR code. But, during this trial, the author found out that it is impossible to directly encode the binary data directly into the CQR code. With the problem mentioned above, the author found a solution for this encoding problem. CQR code can store alphanumeric characters, but the binary data from the converted audio file doesn't have a proper alphanumeric data representation. So, the author then decided to convert this binary audio data into a string representation. The original audio .MIDI file is converted into string representation instead of binary data using a MIDI to CSV program which translates MIDI music files into a human- and computer-readable CSV (Comma-Separated Value) format, suitable for manipulation by spreadsheet, database, or text processing utilities, and re-encoding processed CSV files into MIDI. With the string data obtained from the conversion, the author now is able to encode the data into a CQR code.

4.2 Result

The result of this research is 3 different applications with specific functionalities. The first application is what the authors call "pre-process" application; this application take a MIDI audio file format and convert it into a CSV file format containing string representation of the binary data. The string data will then be used for a third party CQR code generator.

The second application is a CQR code generator created by Kosala and Nurwono [4], which is mainly used to generate CQR barcodes. The third applications main functionality is to decode the barcode. This application will prompt user to open an image file from a local disk, then the application will try to recognize. If the input file is a CQR code, the application will try to decode the underlying message.

The fourth application is what the authos call "post-process" application. This application takes the encoded string from the CQR codes, and will try to reconstruct the string back into a file that is identical to the output file from the pre-process application.

The main concept proposed in the introduction is to find possibilities to encode an audio data into a QR code. The resulting applications prototypes have been successfully proven this concept. The initial audio file in MIDI format is converted into string so that it can be embedded into a CQR code. The CQR codes that act as a carrier for these audio data in the form of string are then decoded by the second application. The decoded message is then stored temporarily using a text editor, which then serves as an input for the third application to reconstruct them into a file identical from to the output of the pre-process application. Although the method for proving the concept of encoding audio data into a CQR code is not the most efficient method, it has substantially served its purpose in proving the concept.

5. CONCLUSION

In this modern future, both 1D barcode and 2D barcode are still commonly used. Even up to today, 1D barcode are still widely used for AIDC purposes. Although 1D barcode still serves its purposes, especially in retail items, the capacity to encode data is minimal. 1D barcode do not have the ability to efficiently encode large amount of data. There is a need in the market to encode larger amount of data into a small spaces. Thus the role of 2D barcodes comes into play. The thesis main idea is based on this motivation and to explore further the potential uses of 2D barcode, this research tried to explore the possibility of encoding audio data into 2D barcode and also try to retrieve the encoded data and reconstruct it such that the embedded audio data can be recognized again by a third party program.

The result of this research thus far is still at a very early development stage. Applications that were successfully created were very basic in terms of functionality and were created only to satisfy the proof of concept delivered within the research. There are definitely a lot of rooms for further development including combining the applications into one deliverable application; and also

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

discovering the potential uses that can be implemented in real life applications. An application that is able to reconstruct the data retrieved from the encoded barcode can also be developed in continuation of this research.

REFRENCES:

[1] Specifications For Popular 2D Bar Codes. Adams Communications. [Online]. Available at:http://www.adams1.com/stack.html

[2] Lisa, S. ; Telecom Italia Lab., Turin ; Piersantelli, G. "Use of 2D Barcode to Access Multimedia Content and the Web from a Mobile Handset." Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE

[3] 2D Barcode Technology Taking Off in North America as Major Marketers & Media Providers Launch New Projects Powered by ScanLife. Scanbuy, Inc.. [Online].Available at: http://www.marketwire.com/press-release/2D-Barcode-Technology-Taking-Off-North-America-as-Major-Marketers-Media-Providers-Launch-1054672.htm

[4] K. Nurwono and R. Kosala, "Color Quick Response Code for Mobile Content Distribution," International Conference on Mobile Computing and Multimedia. Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia. 2009.

[5] Data Matrix Barcode ISO/IEC 16022 FAQ. *IDAutomation.com, Inc.* [Online]. Available at: http://www.idautomation.com/datamatrixfaq.html

[6] QR Code Standarization. *Denso Wave Inc.* [Online]. Available at: http://www.denso-wave.com/qrcode/qrstandard-e.html

[7] QR-Code Barcode FAQ & Tutorial. *IDAutomation.com, Inc*. [Online]. Available at: http://www.idautomation.com/barcode/qr-code.html

[8] J. Leben and R. Van Rooyen, "Percussion Detection in Polyphonic Music: Progress Report". University of Victoria, Victoria, 2014