International Journal of Computer Applications Technology and Research
Volume 4– Issue 3, 179 - 183, 2015, ISSN:- 2319–8656

Implementation of 2D Optimal Barcode (QR Code) for Images

Awadhesh Kumar
AIET Jaipur,
India

Ajeet Kumar Nigam
Dr.KNMEC,
Modinagar, India

Abstract: Quick Response (QR) Code is very useful for encoding the data in an efficient manner. Here data capacity in 2D barcode is limited according to the various types of data formats used for encoding. The data in image format uses more space. The data capacity can be increased by compressing the data using any of the data compression techniques before encoding. In this paper, we suggest a technique for data compression which in turn helps to increase the data capacity of QR Codes generated for image. Finally, results are compared with the normal QR Codes to find the efficiency of the new technique of encoding followed by compression for generating optimal QR code.

Keywords: 2D barcodes, Data Capacity, Data Compression, Lossless Compression, QR Code

1. INTRODUCTION

Bar codes have become widely popular because of their reading speed, accuracy, and superior functionality characteristics. Barcodes can be divided as 1D, 2D and 3D. 1D barcodes can express information in horizontal direction only. Also, the data capacity is limited. 2D barcodes can hold data both in horizontal and vertical direction. As a result, the data capacity is 100 times more than the 1D barcode [1]. 3D barcode is usually engraved on a product or applied on a product so that the barcode has depth and thickness.

As bar codes became popular and their convenience universally recognized, the market began to call for codes capable of storing more information, more character types, and that could be printed in a smaller space. However, these improvements also caused problems such as enlarging the bar code area, complicating reading operations, and increasing printing cost. 2D Code emerged in response to these needs and problems [2].

QR Code is a kind of 2-D (two-dimensional) symbology developed by Denso Wave and released in 1994 with the primary aim of being a symbol that is interpreted by scanning equipment [3]. 2D bar codes can act like identifier (like in 1D) but takes less space. 2-D barcode minimizes the use of database; alternatively, it functions as database itself.

QR Code holds a considerably greater volume of information than a 1D bar code. These can be numeric, alphanumeric or binary data – of which up to 2953 bytes can be stored. Only a part of each QR bar code contains actual data, including error correction information. A large area of the QR code is used for defining the data format and version as well as for positioning, alignment and timing purposes. The smallest square dot or pixel element of a QR code is called a module. QR Codes have an empty area around the graphic. This quiet area is ideally 4 modules wide. Examination certificates can also use the QR Encoding techniques [4].

This paper proposes a method in which data capacity can be increased by first compressing the data and then encoding it. Actual requirement for compression arises when we need to encode image data into QR code. A lossless compression technique is proposed to increase the data capacity. For decoding the data, two steps will be followed: (i) de-compressing the data using the techniques which are just the reverse of compression technique used here and (ii) decoding the decompressed data. For this, the reverse technique used for encoding the data can be used.

2. LITERATURE SURVEY

QR Codes have already overtaken the conventional 1-D bar codes because of the capacity of data that can be stored by a 2-D barcode(QR Code) is much greater than that of conventional 1-D bar code. QR Code contains data both in horizontal and vertical directions. This stems in many cases from the fact that a typical 1-D barcode can only hold a maximum of 20 characters, whereas as QR Code can hold up to 7,089 characters [3]. QR Codes are capable of encoding the same amount of data in approximately one tenth the space of a traditional 1-D bar code. A great feature of QR Codes is that they do not need to be scanned from one particular angle, as QR Codes can be read regardless of their positioning. The data can be read successfully even if QR code is tampered while 1-D

barcode can't. QR Codes can be easily decoded with a smart phone with appropriate barcode reader software (for example:, Kaywa Reader, QRafter and I-Nigma etc.) [5]. Secure communication can also be established using QR Encoding techniques [6].

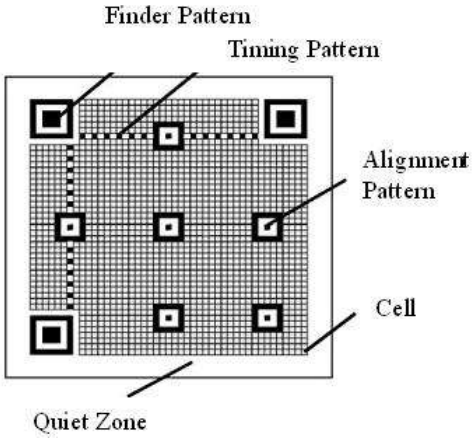

Fig.1: Structure of QR Code

2.1 Structure of QR Codes

QR Codes are actually black modules in square patterns on white background but many researchers have been working for colored QR code. It consists of the following areas having specific significance.

- Finder Pattern
- Alignment Pattern
- Timing Pattern
- Quiet Zone
- Data Area

Fig.1 shows the structure of QR Code. The significance of each area is as described as follows:
Each QR Code symbol consists of mainly two regions: an encoding region and function patterns. Function patterns consist of finder, timing and alignment patterns which does not encode any data. The symbol is surrounded on all the four sides by a quiet zone border [7]. A QR Code can be read even if it is tilted or distorted. The size of a QR Code can vary from 21 x 21 cells to 177 x 177 cells by four cell increments in both horizontal and vertical direction.

2.1.1 Finder Pattern

This pattern can be used for detecting the position, size and angle of the QR Code. These can be determined with the help of the three position detection patterns (Finder Patterns) which are arranged at the upper left, upper right and lower left corners of the symbol as shown in Fig. 1.

2.1.2 Alignment Pattern

The alignment pattern consists of dark 5x5 modules, light 3x3 modules and a single central dark module. This pattern is actually used for correcting the distortion of the symbol [8]. The central coordinate of the alignment pattern will be identified to correct the distortion of the symbol.

2.1.3 Timing Pattern

The timing patterns are arranged both in horizontal and vertical directions. These are actually having size similar to one module of the QR Code symbol. This pattern is actually used for identifying the central co-ordinate of each cell with black and white patterns arranged alternately.

2.1.4 Quiet Zone

This region is actually free of all the markings. The margin space is necessary for reading the bar code accurately. This zone is mainly meant for keeping the QR Code symbol separated from the external area [9]. This area is usually 4 modules wide.

2.1.5 Data Area

It consists of both data and error correction code words. According to the encoding rule, the data will be converted into 0's and 1's. Then these binary numbers will be converted into black and white cells and will be arranged accordingly. Reed-Solomon error correction is also used here [10].

2.2 Data Capacity

The data storage capacity of QR Code is very large as compared to 1-D barcode. The number of characters that can be encoded as QR Code varies according to the type of information that is to be encoded. The various information types and the volume that the QR Code can hold are explained in Table 1.

Table 1. Information Types and Volume of Data

Information Type	Volume of Data
Alphabets and Symbols	**4296**
Numeric Characters	**7089**
Binary Data (8 bit)	**2953**
Kanji Characters	**1817**

2.3 Data Compression

In the history of computer science, data compression, source coding [1] or bit-rate reduction includes encoding information using fewer bits than the original representation. There are two kinds of data compression: lossy and lossless. Lossy compression reduces bits by identifying marginally important information and removing it. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression.

Data Compression is very useful due to reducing the consumption of resources such as data space or transmission capacity. Because compressed data must be decompressed to be used, this extra processing imposes computational or other costs through decompression. The design of data compression schemes involve trade-offs among various factors, including the degree of compression, the amount of distortion introduced and the

computational resources required to compress and uncompress the data [11].

Lossless data compression algorithms usually exploit statistical redundancy to represent data more concisely without losing information. Lossless compression is possible because most real-world data has statistical redundancy. The Lempel–Ziv (LZ) compression methods are among the most popular algorithms for lossless compression. DEFLATE is a variation on LZ which is optimized for decompression speed and compression ratio, but compression can be slow.

3. PROPOSED SCHEME

The efficiency of QR Codes is increased by applying compression before encoding. This paper focuses on the high capacity QR Codes for encoding image within barcode symbol. Our approach consists of mainly four steps for encoding:

1) Convert data in image format to base64 character format,
2) Compresses the data obtained in step 1.
3) Encodes the compressed data into a QR Code.

The whole process of converting image into QR Code is represented by flowchart in Fig 2.

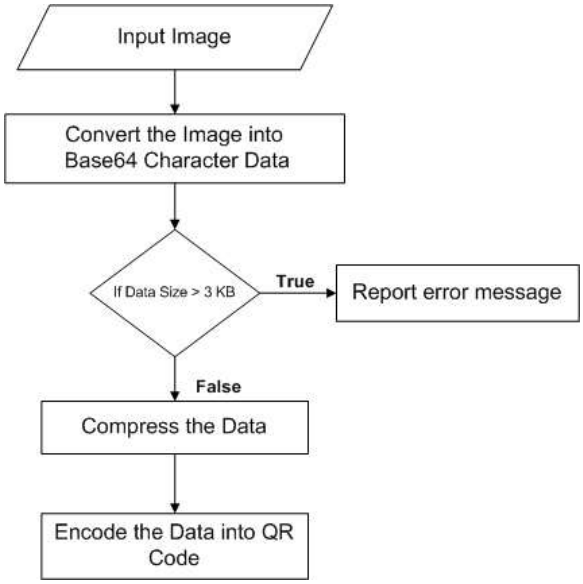

Fig. 2 Flowchart for generation of QR Code for Image

3.1 Compression Technique

This is a lossless approach to short the string such that after expansion we get the same string without any loss of character. In QR Code, we are using alphabets (A-Z), numbers(0-9) and some special characters ($,space,%,_,+,□,_,:,=). Thus, a total of 45 characters we needed for QR Code. Normally, each character in the string is represented by 8-bits. So, if there are 100 characters in a string then we need 800 bits to represent that string. Our approach gives each 45 characters a particular code. The code is a fixed length code. We need 6 bits fixed length for each character to distinguish from one another, since $45<2^6$.

3.2 QR Encoding

The normal encoding of data is done through various steps such as [1]:

1) Analyse the data to be encoded. Convert the data to symbol characters. Find out the error correction and detection level.
2) Encode the data.
3) Error Correction Coding
4) Add reminder bits and data masking patterns.
5) Generate the format information and version information [12].

The entire process can be made clear with the help of the simple flowchart given below.

3.3 QR Decoding

Normally, QR decoding is done with the help of camera equipped mobile phones. Decoding process is just the reverse of the encoding procedure applied. We need to identify the quiet zone in order to decode the correct data. Alignment patterns help the decoding procedure by correcting the distortion of the symbol.

Kaywa reader is the most commonly used software to decode the original text. Image processing with J2ME is found to be more powerful. J2ME is designed to work on low-end devices in terms of processing power and memory capabilities. The various pre processing steps include gray scaling of the captured colour image, histogram stretching of the image, local adaptive thresholding, noise filtering, cropping, rotation correction and tilt correction. After pre processing, the finder patterns are detected and then the original data is decoded [5]. Another technique for decoding involves "edge to similar edge" estimation method is employed to check whether the detected bar-space pattern is correct [12].

3.4 Compression and Decompression of String

Algorithm 1: Compression

//String1 is the input and output.
1. for each character in String1. do
   Extract a character from the String1.
   Convert this character into respective 6-bit length code.
   Append this code to String2.//Initially String2 is empty.
   end for
2. Remainder = (Length of the String2) mod 8.
3. Convert (Remainder+1) into 8 bits and add in the starting of String2.
4. Add 0's equal to the Remainder at the end of String2.
5. for each 8 characters in String2. do
   Extract 8 character from the String2.
   Convert this value into equivalent ASCII character.
   Store this character in String1.
   end for
6. Return String1

Algorithm 2: Decompression

//String1 is the input and output.

1. for each character in String1. do
   Extract a character from the String1.
   Convert this character into respective 8-bit length ASCII code.
   Append this code to String2.//Initially String2 is empty.
   end for
2. Extract starting 8-bits from String2.
3. Convert this into number.
4. Remove _rst 8-bits from String2.
5. Remove 0's equal to (number-1) from the end of String2.
6. for each 6 characters in String2. do
   Extract 6 character from the String2.
   Convert it into equivalent code.
   Store this character in String1.
   end for
7. Return String1.

4. RESULTS

Using the approach discussed above, we are compressing the data before generating QR Code, and hence efficiency can be improved. This technique suggests simple ways to accomplish this task. The whole concept is implemented by designing a small application using C#.Net on Visual Studio 2008. Fig-3 shows the conversion of image in png format to Base64 character format.

Fig-3: Image to Base64 Character Format

Fig-4 depicts the generation of QR code from image. The difference between uncompressed and compressed QR code can be seen in the Fig-4 below. The size of QR code without compression is greater (i.e. 26KB for sample signature image) than the QR code (i.e. 19KB for same sample image) of compressed data.

Fig-4: Generation of QR Code for Image

As our objective focuses on compression of data (after converting image to character format) to before generating QR code since QR code has limited data storage capacity. Fig-5(a) shows the error message when uncompressed image is browsed.

Fig-5(a): Verification of QR Code

Fig-5(b) shows the successful verification of signature image encoded in QR code.

Fig-5(b): Verification of QR Code

5. CONCLUSIONS

Normal QR Codes can compress only up to 4 KB of data. Using the techniques followed here, the data capacity can be increased drastically. As compared to the normal QR Codes, the data capacity of the QR Code after following technique was found to be more than 4 KB. Efficient data compression techniques can be used to store more than 4 KB of data inside a QR Code. A variety of data compression techniques can be used to obtain more data storage capacity. Comparing with the existing technologies used to generate bar codes, QR Codes were found to be of great advantage to the manufacturer because of its great data storage capacity, reading speed and accuracy. The data capacity was further improved by combining the most distinguishing features of compression and bar code generation. Using this novel technique of data compression followed by data encoding, the data storage capacity of QR Codes were increased drastically.

Currently only Smartphone's are technically equipped to do this. Many users that have mobile phones that have cameras are unable to get QR reading software for their phones. Future enhancements focus on QR Encoding of images which is more than 4 KB of size. Secure QR Coding can also be implemented using encryption techniques. Also, more advanced data compression techniques can be used to add more to the data capacity of the normal QR Codes.

6. REFERENCES

[1] Xiaofei Feng, Herong Zheng, "Design and Realization of 2D Color Barcode with High Compression Ratio" 2010 International Conference On Computer Design And Appliations (ICCDA 2010), 978-1-4244-7164-51, 2010 IEEE, 978-1-4244-7164-51, 2010 IEEE, Volume 1

[2] Nancy Victor, "Enhancing the Data Capacity of QR Codes by Compressing the Data before Generation", International Journal of Computer Applications (0975-8887), Volume 60 - No.2, December 2012.

[3] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, Edgar WeipplT. J., "QR Code Security"

[4] Chun-lei XIA, "Examination Certificate Based on Two-Dimensional Bar Code Technology", 2008 International Symposium on Computer Science and Computational Technology, 978-0-7695-3498-5/08/2008 IEEE DOI 10.1109/ISCSCT.2008.102

[5] Tasos Falas, Hossein Kashani, "Two-Dimensional Bar-code Decoding with Camera-Equipped Mobile Phones", Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops(PerComW'07) 0-7695-2788-4/07/2007

[6] William Claycomb, Dongwan Shin, "Using A Two Dimensional Colorized Barcode Solution for Authentication in Pervasive Computing", 1-4244-0237-9/06/2006 IEEE

[7] Guenther Starnberger, Lorenz Froihofer and Karl M. Goeschka, "QR-TAN: Secure Mobile Transaction Authentication", 2009 International Conference on Availability, Reliability and Security, 978-0-7695-3564-7/09 IEEE DOI 10.1109/ARES.2009.96

[8] ISO/IEC 18004:2000 Information Technology - Automatic Identification and Data Capture Techniques – Barcode Symbology- QR Code (MOD), June 2000.

[9] Sarah Lyons and Frank R. Kschischang, "Two-Dimensional Barcodes for Mobile Phones", 25th Biennial Symposium on Communications, 978-1-4244-5711-3/10/2010

[10] R. Bose and D. Ray-Chaudhuri. On a class of errorcorrecting binary group codes*. Information and control, 3(1):68{79, 1960.

[11] David L. Donoho, Martin Vetterli, Fellow, IEEE, R. A. DeVore, and Ingrid Daubechies, Senior Member, IEEE," Data Compression and Harmonic Analysis", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 6, OCTOBER 1998, 0018–9448/98$10.00 ã 1998 IEEE

[12] Hee I1 Hahn and Joung Koo Joung, "Implementation of Algorithm to Decode Two-Dimensional Barcode PDF-417", ICSP'O2 Proceedings, 0-7803-7488-6/02/$17.00 Q 2002 IEEE.