# Increasing data storage of coloured QR code using compress, multiplexing and multilayered technique

Azizi Abas, Yuhanis Yusof, Roshidi Din, Fazli Azali, Baharudin Osman

School of Computing, College Arts and Sciences, Universiti Utara Malaysia, Malaysia

#### **Article Info**

## Article history:

Received Dec 18, 2019 Revised Mar 3, 2020 Accepted Mar 31, 2020

#### **Keywords:**

Compression Data storage Multilayer Multiplexing QR code

## **ABSTRACT**

Quick response (QR) code is a printed code of black and white squares that is able to store data without the use of any of the electronic devices. There are many existing researches on coloured QR code to increase the storage capacity but from time to time the storage capacity still need to be improved. This paper proposes the use of compress, multiplexing and multilayered techniques, as an integrated technique known as CoMM, to increase the storage of the existing QR code. The American Standard Code for Information Interchange (ASCII) text characters are used as an input and performance is measured by the number of characters that can be stored in a single black and white QR code version 40. The experiment metrics also include percentage of missing characters, number of produced QR code, and elapsed time to create the QR code. Simulation results indicate that the proposed algorithm stores 24 times more characters than the black and white QR code and 9 times more than other coloured QR code. Hence, this shows that the coloured QR code has the potential of becoming useful mini-data storage as it does not rely on internet connection.

This is an open access article under the CC BY-SA license.



# Corresponding Author:

Azizi Abas, School of Computing, UUM College Arts and Sciences, Universiti Utara Malaysia, 06010, Sintok, Kedah, Malaysia. Email: azizia@uum.edu.my

#### 1. INTRODUCTION

There are 40 versions of QR Code; the smallest version is 1 with  $21\times21$  matrixes and the largest version is 40 with  $177\times177$  matrixes. Each matrix will increase four modules from the previous version to the next version [7]. The data capacity of a QR code is determined by its error correction level. This error-correcting level was developed as a recovery mode when it was in the event of a miss reading. There are 4 recovery levels, which include level L: (7%), level M (15%), level Q: (25%) and level H: (30%).

While a QR Code can store more data compared to the two-dimensional barcode [8], it still requires improvement to improve its data capacity [9]. To enlarge the printing size of the barcode will create complex

Journal homepage: http://beei.org

2556 □ ISSN: 2302-9285

operation and increase the printing cost and space [8]. In addition, if the amount of data is large, it requires several QR Code to be printed [10, 11]. Hence, the insufficient storage capacity of a QR Code becomes an issue as some information cannot be embedded or stored inside it. This includes information such as long Unified Resource Locator [12] and security information in the cryptography [13-15].

Many techniques have been proposed on how to encode and decode a QR Code that can store the larger size of data and this includes multilayers [16, 17] multiplexing [18], compression [19] and base64 encoding/decoding [20]. This study proposes a technique that improves the data capacity of a QR Code by merging for the mentioned techniques and tested in QR Code version 40. The reason for choosing QR Code version 40 as the storage medium is because it stores the largest amount of data among various QR Codes version.

# 2. PROPOSED METHOD

The proposed processing method sequence is illustrated in Figure 1. The data (text) and image are separated as different processes in the development stage. The data are categorized in the shape of the compression technique and the image of the QR Code is divided into multiplexing and multilayer techniques. The priorities of the processes were divided due to capacity capabilities. It is found that compression technique should go first and follow by multiplexing and multilayer processes. All the technique and priority sequences use can be referred in Figure 1. All the sequence methods will be explained in next section below.



Figure 1. The abstract model and method priority sequence of data and image combination

#### 2.1. Compression

Compression is a technique to eliminate the redundancy in the data. This will help to reduce the size and it takes less time to transmit the data. This compression approach is divided into two phases which are using the best compression tools [21] and binary to text encoder. In the first phase, the lossless compression technique was chosen because it saves bandwidth [22] and storage space [23]. Also, the lossless compression technique is suitable for text data as every single bit of data that was originally in the file remains after the file is uncompressed. Lossless compression is also used in the Zip and GZip tools. The GZip compression is used in this study as it brings more percentage of compression compared to Zip, LZW, Huffman Coding and mixed Huffman coding and Gzip [21]. In the second phase, the base91 encoder, known as binary to text encoder, is used to convert the binary compressed file into the text file. As a result, two methods were implemented in the data compression, which is GZip compression and base91 encoder.

## 2.2. Multiplexing

Multiplexing is a method to combine or merge multiple streams into a single medium [23]. This technique can be used to increase the storage capacity of a QR Code. In the beginning, eight black and white OR Codes were generated based on the error correction level and its maximum amount of character storage capability. Theoretically, the multiplexing technique allows multiple blacks and white QR Code to be combined into a single mono-colour QR Code. The pixel index location will be identified in two dimensions (x, y) from each black and white QR Codes before they are combined. The combination is based on the same pixel index location from each black and white QR Code. Since the standard black and white QR Code uses two colours, the black is assigned as 1 and the white is assigned as 0 [24]. Then they will be merged into eight characters, which is equivalent to eight black and white QR Code in each pixel index location of it. The binary numbers are converted to a decimal number in the range between 0-255. This value complies with the range number of RGB colour system of red, green and blue. The decimal number will be put in either red, green or blue colour value system. For example, the 101011112 will be converted to 17510 and put into Colour (175, 0, 0) in Java program language as red colour. This example applies for the position (x, y) of the pixel index location in eight bits of binary number for black and white QR Code and the QR Code mono-colour will be generated in the red colour mode. This technique allows the green and blue QR Code in the RGB colour system.

#### 2.3. Multilaver

After the multiplexing processes were generated and completed, the multilayered process will continue the process to combine the red, green and blue QR Code into coloured QR Code. The last result of the multilayered process is to produce the coloured QR Code known as RGB which it's contained some amount of maximum characters inside The multi-layered QR Code has 93987 processes which can be calculated as (177 x-axis \* 177 y-axis) location index \* 3 red, green and blue QR Codes. The ways of collecting information are the same with multiplexing but the plotting decimal value in the RGB colour scheme is based on a combination of red, green and blue colour values. This process will read the whole colour code from each location index starting from (0, 0) until (176, 176) of the images from red, green and blue QR Codes. The same location index of red, green and blue QR Codes will be blended in RGB colour systems as colour (x, y, z) which indicates as x is the colour code of the red QR Code, y is for the green QR Code and z is representing the blue QR Code.

ISSN: 2302-9285

#### 2.4. Decode

The decoding process is to bring back the actual data which were encoded before. In the beginning, the process will start with extracting the coloured QR Code into the red, green and blue QR Code. After these processes are completed, each of the red, green and blue QR Code will be extracted into eight black and white QR Codes from each of them. For the next step, the twenty-four black and white QR Code from the result of previous processes will be decoded from each of it into base91 text files and combined it together to produce a complete merging base91 text file. The base91 text file will be converted to a binary GZip file by using a base91 decoder. Lastly, the binary GZip file can be decompressed and produce the original text file.

#### 3. RESEARCH METHOD

This experiment is performed based on three criteria which are computer processing time, data storage capacity and error correction level. The input data is a compilation of text based on short stories that are stored in twenty-four black and white QR Codes. The amounts of characters depend on the error correction level. The error correction level L can contain more characters compared to other levels. Figure 2 shows a part of the employed input text which includes various types of characters such as the numeric, alphabet and several symbols.

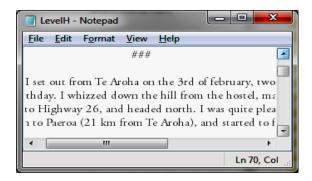


Figure 2. A part of the input data text

All the experiments were performed using Java programming language and executed on a computer with processor specification of Intel(R) Core (TM) i7-2670QM CPU @2.20GHz, RAM 8 Gigabytes, and 64-bit Windows 7 Ultimate operating systems. The utilized benchmark method is the QR Code of version 40. In this experiment, the text files are used as input. Meanwhile, the number of characters to each black and white QR Code is divided and set based on the error correction level. A maximum number of characters stored in the QR Code and size of the text file is shown in Table 1.

Table 1. Maximum number of characters and text file size based on error correction level

Error correction	Total characters	Text file size	Error correction	Total characters	Text file size
level	each QR code		level	each QR code	
L	2952	126 KB (129,512 bytes)	Q	1662	68.8 KB (70,545 bytes)
M	2330	94.2 KB (96,477 bytes)	H	1270	51.7 KB (52,996 bytes)

### 4. RESULTS AND DISCUSSION

Results can be presented in figures, tables and others that make the reader understand easily [2, 5]. From the data depicted in Table 2, it can be studied that the number of characters will be increased if the error correction level is lower. This is due to the feature of the recovery function that was embedded in the QR Code. The minimum amount of characters can be stored in the new coloured QR Code is 51240 characters excluding new line and carriage return in error correction level H. Meanwhile, the maximum amount of characters can be stored is 125114 characters in error correction level L. There are no missing characters during encoding and decode processes from this experiment conducted. All the characters can be recovered back in all error correction levels.

Table 2. Amount of characters encoded based on the sequence of compression, multiplexing and multilayered

Characters	Level L		Level M		Level Q		Level H	
	Before	After	Before	After	Before	After	Before	After
Total characters	125114	125114	93295	93295	68201	68201	51240	51240
Total characters include new line and carriage	129512	129512	96477	96477	70545	70545	52996	52996
return								

The base91 encoder is used to convert the binary file to the text file. An experiment was conducted to check the availability of characters during the encoding and decoding processes. Table 3 shows the result of total characters during base91 encode (before) and decode (after) processes. It seems that base91 encodings can be used to reduce the number of characters and this result can be compared as in Table 1 above. From Table 3 also, there are no missing characters starting encode to decode.

Table 3. Total characters of encoding and decoding using base91 encoder

Characters	Level L		Level M		Level Q		Level H	
	Before	After	Before	After	Before	After	Before	After
Total characters	70845	70848	53275	53280	39887	39888	30479	30480
Total characters include new line and carriage	70845	70848	53275	53280	39887	39888	30479	30480
return								

Table 4 shows the elapsed time of the encoding process. The encoding process starts from compression (GZip and Base91 encode), multiplexing until multilayer. The error correction level H has the capability to complete the encoding process fastest compared with the other error correction level, which consumed 6 seconds and 254 milliseconds to complete the processes. But, at the same time, the total bytes of the image (Coloured QR Code) with error correction level L has a minimum size which has only consumed 105 kilobytes (108,339 bytes). From this perspective, it can be concluded that time consume to complete the process is not contributing to the file size of the QR Code for this research.

Table 4. The elapsed time of the encoding process

Error level correction	Compression	Multiplex	Multilayer	Multiplexing and multilayer	Total elapse time	Total bytes colour QR code image
L	0s	13s	0s	13s	16s	105 KB
	64ms	238ms	169ms	406ms	683ms	(108,339 bytes)
M	Os	6s	0s	6s	7s	105 KB
	29ms	531ms	97ms	629ms	551ms	(107,997 bytes)
Q	0s	6s	0s	6s	6s	107 KB
	24ms	276ms	92ms	638ms	494ms	(110,241 bytes)
H	0s	6s	0s	6s	6s	108 KB
	21ms	254ms	98ms	353ms	964ms	(110,679 bytes)

Based on Table 5, the decode elapsed time process starts from demultilayer, demultiplex and decompression. The last result is to go back to the actual text file that was decoded before. From the tabular data, the error correction level Q has a minimum total decode elapsed time, which is 5 seconds 858 milliseconds. Meanwhile, the error correction level H has consumed the maximum decode elapsed time with 6 seconds 11 milliseconds. From the result, the time consumed to decode the coloured QR Code not a big issue because the range time between it, is not too long. It took approximately 6 seconds to complete the process.

ISSN: 2302-9285

Table 5.	The ela	nsed	time	of the	decoding	process

Error level correction	Demultilayer	RGB demulitlex	White & black demultiplex	Total multiplex	Decompress	Total
L	0s	0s	0s	1s	0s	5s
	925ms	402ms	470ms	656ms	12ms	959ms
M	0s	0s	0s	1s	0s	5s
	899ms	437ms	465ms	650ms	11ms	929ms
Q	0s	0s	0s	1s	0s	5s
	902ms	411ms	450ms	614ms	9ms	858ms
Н	0s	0s	0s	1s	0s	6s
	900ms	421ms	473ms	655ms	8ms	11ms

The traditional QR Code version 40 has been employed and used to compare with the new proposed colour QR Code in the form of data capacity. Table 6 shows the different text capacity between QR Code version 40 and proposed coloured QR Code. From this table, the error correction level L has a large difference between two QR Codes. It consists of 29.12337 times of extended characters compared with traditional QR Code version 40. Certainly, the new proposed QR Code has a large capacity to extend the characters with a maximum of 125114 characters.

Table 6. The different text capacity between QR Code ver. 40 and proposed coloured QR Code

Error correction level	Version 40 (maximum)	Proposed QR code (maximum)	Times difference expanded
L	4296	125114	29.12337
M	3391	93295	27.51253
Q	2420	68201	28.18223
Н	1852	51240	27.66739

Recently, there are many studies or researches on extending data capacity by using a QR Code. Based on Table 7 below, most of the researchers have successfully completed their studies by extending the QR Code storage up to 3 times. But only the research by Hiren J.Galiyawala, and Kinjal H. Pandya [16] was able to extend it up to 24 times which is the same with the proposed coloured QR Code in this paper. But there is a constraint in their research as shown in Table 8 that is the total processing time to encode and decode of 14 QR Code took about 53.153 and 1236.105 seconds which are too much time consuming to complete. This is clearly shown that [25] need to improve their processing time of decode and encode process.

Table 7. The accomplishment of the existing QR code generation

		8	
No	Researchers and year	Colours	Increments
1	Hiren J. Galiyawala and Kinjal H. Pandya [16]	2 <sup>n</sup> colours whereas n=total QR code	24 times
2	Prathibha. N. Pillai and K. Naresh [17]	Black, white, red, green, blue	3 times
3	Sartid Vongpradhip [18]	Black, white	3 times
4	M. Ramya, M. Jayasheel [26]	Cyan, magenta, yellow	3 times based on layers combine
5	Henryk Blasinski, Orhan B. Gaurav Sharma [27]	Cyan, magenta, yellow	3 times
6	Kris A. H. Nurwono and Raymondus Kosala [28]	Red, green, blue	3 times based on layers combine.

Table 8. The processing time of encoding and decode by Hiren J. Galiyawala and Kinjal H. Pandya [25]

Sr No.	Number of QR codes	Assigned distinct	Processing ti	me (seconds)
	multiplexed	colour (2 <sup>2</sup> )	During encoding	During decoding
1	2	4	2.922	2.892
2	3	8	3.194	2.988
3	4	16	3.364	2.982
4	5	32	3.131	3.084
5	6	64	3.297	3.381
6	7	128	3.489	3.624
7	8	256	3.911	4.359
8	9	512	4.811	6.128
9	10	1024	6.229	11.406
10	11	2048	9.453	28.398
11	12	4096	15.787	91.393
12	13	8192	27.94	323.198
13	14	16384	53.157	1236.105

#### 5. CONCLUSION

The proposed coloured QR Code that employs the CoMM technique has shown the increase in QR Code data storage compared with current QR Code and previous researches. The evaluation was realized in a text-based short story that contains not more than 125114 characters. As the undertaken experiment produces a positive result and meets the objective of the research. On another hand, it is belief that storing or embedding product description for advertisement purposes would also be successful. Nevertheless, in order to fully utilize the proposed work, there is a need to study the hardware issue such as the capability of the code reader to capture the actual colour produced in the coloured QR Code. Another limitation is the lighting in the environment while reading the code. This will lead to more studies in the area of capacity improvement, security and information hiding in the QR Code.

# **ACKNOWLEDGEMENTS**

This paper was funded under grant MyGrant Racer (Ref: RACER/1/2019/ICT03/UUM//2) (SO: 14441)

#### REFERENCES

- [1] M. Querini, A. Grillo, A. Lentini, and G. F. Italiano, "2d colour barcodes for mobile phones," *Int. J. Comput. Sci. Appl.*, vol. 8, no. 1, pp. 136-155, 2011.
- [2] P. Kieseberg, M. Leitner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, and E. Weippl., "QR code security," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia-MoMM '10*, pp. 430-435, 2010.
- [3] T. Falas and H. Kashani, "Two-dimensional bar-code decoding with camera-equipped mobile phones," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops* '07, pp. 2-5, 2017.
- [4] A. Boob, A. Shinde, D. Rathod, and A. Gaikwad, "QR code based mobile app and business process integration," Int. J. Multidiscip. Curr. Res., vol. 2, pp. 1014-1017, 2014.
- [5] T. Nikolaos and T. Kiyoshi, "QR-code calibration for mobile augmented reality applications: Linking a unique physical location to the digital world," in ACM SIGGRAPH 2010 Posters, p. 4503, 2010.
- [6] A. Grillo, A. Lentini, M. Querini, and G. F. Italiano, "High capacity coloured two dimensional codes," *Proc. 2010 Int. Multiconference on Comput. Sci. Inf. Technol. (IMCSIT)*, vol. 5, no. 1, pp. 709-716, 2010.
- [7] P. Sutheebanjard and W. Premchaiswadi, "QR code generator," in 2010 Eighth International Conference on ICT and Knowledge Engineering, pp. 89-92, 2010.
- [8] N. Harish and S. Gurav, "Embedding a large information in QR code using multiplexing technique," *Taraksh J. Commun.*, vol. 1, no. 6, pp. 6-9, 2014.
- [9] S. Abdullah, N. M. Din, S. J. Elias, A. W. Y. Khang, R. Din, and R. Bakar, "Design of a cell selection mechanism to mitigate interference for cell-edge macro users in femto-macro heterogeneous network," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 180-187, 2019.
- [10] M. Querini and G. F. Italiano, "Reliability and data density in high capacity colour barcodes," *Comput. Sci. Inf. Syst.*, vol. 11, no. 4, pp. 1595-1616, 2014.
- [11] A. Denso, "QR code essentials," Denso Wave 900, 2011.
- [12] M. Warasart and P. Kuacharoen, "Paper-based document authentication using digital signature and QR code," in 4th International Conference on Computer Engineering and Technology (ICCET 2012), 2012.
- [13] S. Majumdar, A. Maiti, B. Bhattacharyya, and A. Nath, "A new encrypted Data hiding algorithm inside a QR Code mplemented for an Android Smartphone system: S \_ QR algorithm Introduction," *Int. J. Innov. Res. Adv. Eng.*, vol. 2, no. 4, pp. 40-46, 2015.
- [14] N. A. N. Azman, S. Ali, R. A. Rashid, F. A. Saparudin, and M. A. Sarijari, "A hybrid predictive technique for lossless image compression," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1289-1296, 2019.
- [15] A. Al-Rawashdeh and Z. Al-Qadi, "Creating colour image features using local contrast method," *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 3, pp. 367-376, 2018.
- [16] K. H. Pandya and H. J. Galiyawala, "A survey on QR codes: in context of research and application," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 3, pp. 258-262, 2014.
- [17] P. N. Pillai and K. Naresh, "Improving the capacity of QR code by using colour technique," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 3, no. 7, pp. 10561-10568, 2014.
- [18] S. Vongpradhip, "Use multiplexing to increase information in QR code," in *The 8th International Conference on Computer Science & Education (ICCSE 2013)*, pp. 361-364, 2013.
- [19] N. Victor, "Enhancing the data capacity of QR codes by compressing the data before generation," *Int. J. Comput. Appl.* (0975–8887), vol. 60, no. 2, pp. 17-21, 2012.
- [20] M. Liu, Y. Zhang, J. Li, J. Shu, and D. Gu, "Security analysis of vendor customized code in firmware of embedded device," in *International Conference on Security and PRivacy in Communication Systems*, pp. 722-739, 2017.
- [21] A. Abas, Y. Yusof, and F. K. Ahmad, "Expanding the data capacity of QR codes using multiple compression algorithms and base64 encode/decode," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2-2, pp. 41-47, 2017.

[22] K. Kumar, P. Sharma, and A. K. Singh, "Configuring the system to share internet from single user to multi-user with single internet dongle," *Int. J. Soft Comput. Eng.*, no. 4, pp. 32-35, 2012.

ISSN: 2302-9285

- [23] A. Mayeli and M. Razani, "Multiplexing and demultiplexing frame pairs," *Commutative and Noncommutative Harmonic Analysis and Applications*, vol 603, pp. 189-195, 2013.
- [24] M. Antonijević, S. Sučić and H. Keserica, "Augmented reality for substation automation by utilizing IEC 61850 communication," 2016 39th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2016-Proc., pp. 316-320, 2016.
- [25] H. J. Galiyawala and K. H. Pandya, "To increase data capacity of QR code using multiplexing with colour coding: An example of embedding speech signal in QR code," in 11th IEEE India Conference: Emerging Trends and Innovation in Technology, INDICON 2014, pp. 2-7, 2014.
- [26] M. Ramya and M. Jayasheela, "Improved colour QR codes for real time applications with high embedding capacity," *Int. J. Comput. Appl.*, vol. 91, no. 8, pp. 8-12, 2014.
- [27] H. Blasinski, O. Bulan, and G. Sharma, "Per-colourant-channel colour barcodes for mobile applications: An interference cancellation," in *IEEE Transaction on Image Processing*, vol. 22, no. 4, pp. 1498-1511, 2013.
- [28] K. A. H. Nurwono and R. Kosala, "Color quick response code for mobile content distribution," in *MoMM2009-The* 7th International Conference on Advances in Mobile Computing and Multimedia, pp. 267-271, 2009.