Updating Information in Colored QR Code for Knowledge Sharing

Azizi Abas, Yuhanis Yusof and Farzana Kabir
University Utara Malaysia, Malaysia, yuhanis@uum.edu.my

ABSTRACT

Knowledge sharing is the process through which explicit or tacit knowledge is communicated to other individuals. Recently, the Quick Response (QR) Code which is a small size printed shape of black and white squares has been used in many domain as a storage medium. To date, it is a popular storage medium in the marketing domain. The code can easily be used to disseminate information about a product as it only requires a small printed area. The QR Code may contain various information such as URL, contact information, and application launching. However, the information stored in the QR Codes may require some modification or updating. Hence, this paper demonstrates a technique to update information stored in colored QR Code and regenerate the code using integrated techniques of compression, multiplexing and multilayered techniques (CoMM). The processes to regenerate an updated QR Code includes the decode and re-encode of the QR Code.

Keywords: QR Code; Colored QR Code, Data Storage.

I INTRODUCTION

The QR (Quick Response) Code is widely used in the application of marketing, retailing, advertising, production, tracking and others relating to product description (Kato & Tan, 2005; Kieseberg et al., 2010). It was categorized as two dimensional barcode because it can store more information compared with one dimensional barcode. It was developed by a company called Denso Corporation Japan in 1994 and officially recognized as an ISO international standard (ISO/IEC18004) (Falas & Kashani, 1994). The features of the QR Code covers the capability to encode high capacity data, printed out in a small size, robustness and the ability to be read in any direction of 360 degrees (Boob, Shinde, Rathod, & Gaikwad, 2014). Figure 1 below illustrates a sample of the QR Code.

Figure 1. The QR Code sample

Each of the QR Code module is represented as a single bit where a black square stores the value 1 and white square stores the value 0 (Nikolaos & Kiyoshi, 2010) (Grillo, Lentini, Querini, & Italiano, 2010). The type of QR Code is in black and white shape that able to encode and decode the characters with certain maximum capacity based on the type of characters. The type of characters used may as numeric (0 – 9) , alphanumeric (0-9, A-Z, a-z, space, $, %, *, +, -, ., /, :), byte/binary (8-bit bytes) and kanji (Japanase symbols) to store data (Jagan, Akila, & Nasrin, 2015; Kieseberg et al., 2010). Table 1 shows the storage capacity of a QR Code orderd by type of characters. This maximum capacity is based on the QR Cod version 40 with error correction level L.

Table 1. The storage capacity in a QR Code.

Characters	Maximum Characters
Numeric	7089
Alphanumeric	4296
Binary	2953
Kanji/Kana	1817

The QR Code is in the form of a square shape with three corners of small square pattern used to determined the position of the QR Code during reading process at any angle position without giving incorrect output result (Jude J.L., 2015). To date, there are 40 versions of QR Code type; the smallest version is 1 which is contain 21 x 21 matrixes and the largest version is 40 with 177 x 177 matrixes. Each matrix will increase four modules from the previous version to next version (Sutheebanjard & Wichian, 2010).

The data capacity of a QR Code is determined by its error correction level. This error correcting level was developed as a recovery purpose. The recovery may needed when the QR Code contains miss reading, damage or others related. There are 4 type of recovery levels, which includes level L: (7% or less error), level M (15% or less error), level Q : (25% or less error) and level H : (30% or less error).

Since colored QR Code (Abas, Yusof, & Kabir, 2015) was introduced, there are some issues that requires attention. This includes the re-generation of QR Code when the stored information is obsolete or requires some modification. Base on the research above, if the colored QR Code needs to be manipulated, the first process involve is decode. It need to generate 24 black and white QR Codes. After the data was updated, it needs to re encode all the 24 black and white QR Codes to generate a colored QR Code. Such a process consumes large processing time as it needs to decode and re encode the colored QR Code. This paper demonstrates the process of updating a colored QR Code and re-generate the code using the combination of multilayers (Pandya & Galiyawala, 2014; Pillai & Naresh, 2014), multiplexing (Vongpradhip, 2013) and compression (Victor, 2012) techniques. The proposed work involved text based with error correction level L.

According to Paulin and Suneson (2008) , knowledge is the source and method that were obtained and at least it has been discussed. Meanwhile, knowledge sharing (Schwartz, 2006) is exchanging of knowledge between or among individuals, teams, unit and organizational. The propose works is to share among expertise on how to reduce time consume when the QR Code needs to decode and re encode again.

II PROPOSED WORK

The colored QR Code was developed based on the abstract model as shown in Figure 2. The process to update information is divided into two levels, which is level 1 and level 2. Level 1 involves 24 black and white QR Codes while Level 2 includes the red, green and blue QR Codes. Each level contains decode and re-encode processes.

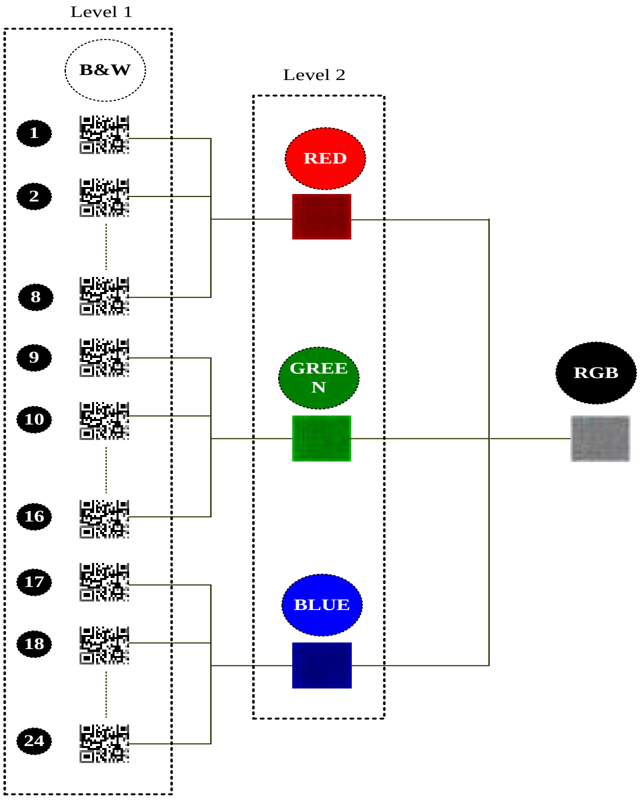

Figure 2. The colored QR Code development.

The proposed level 1 decode abstract model which is used to update the information in black and white QR Code is illustrated in Figure 3. This model will update the black and white QR Code information at location index 0. Meanwhile, for the re-encode abstract model is shown in Figure 4. This process will start updating text file information at location index 0.

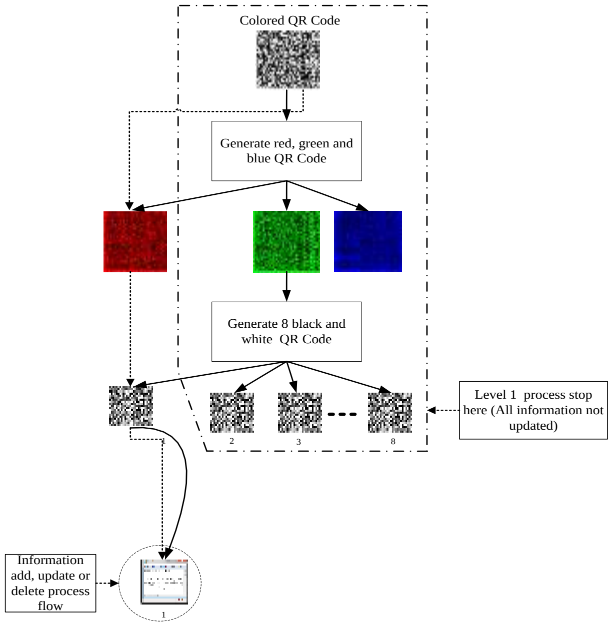

Figure 3. Level 1 decode abstract model

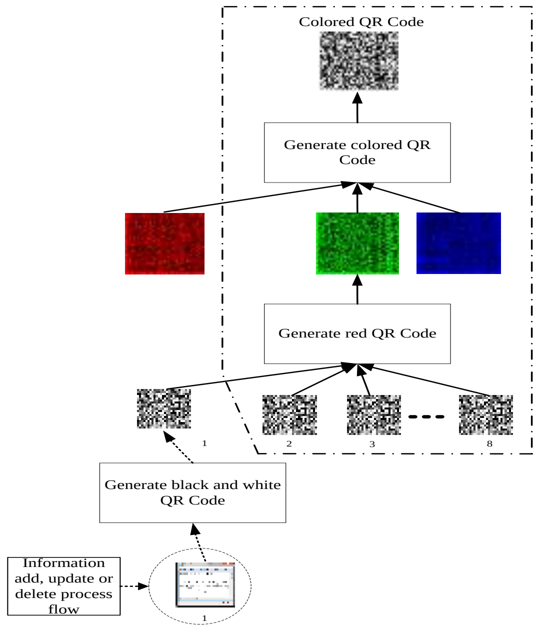

Figure 4. Level 1 re-encode abstract model

In addition, level 2 abstract model consists the operation to update the red, green and blue QR Code. This process will generate eight text files and the provider can manipulate a larger amount of information. Figure 5 and 6 show the processes of decode and re-encode red color QR Code.

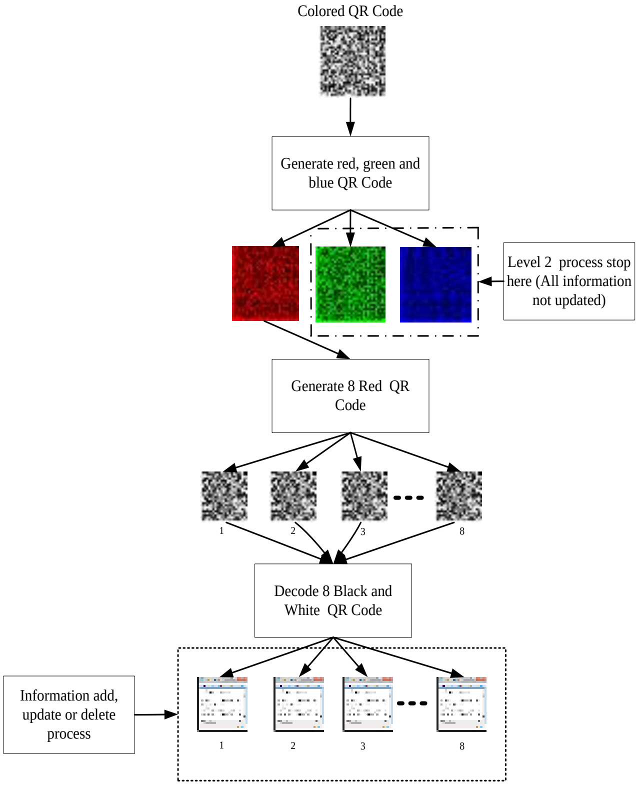

Figure 5. Level 2 decode abstract model

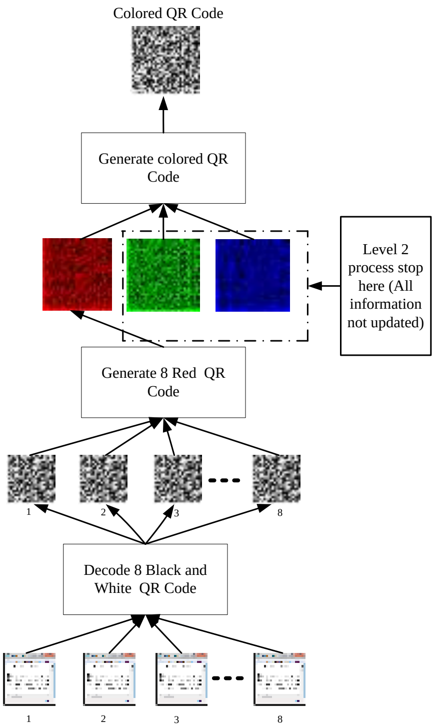

Figure 6. Level 2 re-encode abstract model

The input data is a compilation of text based on short stories which are stored in twenty four black and white QR Codes. The amount of the characters depends on the error correction level. The error correction level L will be used in this experiment. Figure 7 shows a part of the employed input text which includes various types of characters such as numeric, alphabet and several symbols.

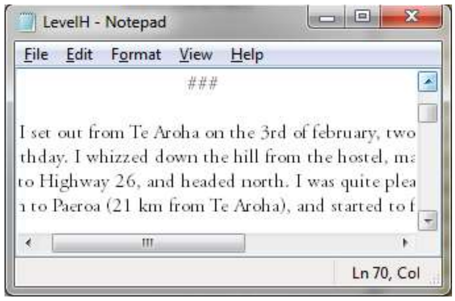

Figure 7. A part of input data text.

III EXPERIMENT

All of 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. The maximum number of characters stored in each QR Code version 40 is shown in Table 1 while the size of the employed text file is 126 KB (129,512 bytes). Evaluation of the undertaken experiments is based on computational time.

IV RESULT

Several experiments were conducted and some figures and tables were tabulated as the results of these experiments. The process flow results for benchmark method (ie. QR Code version 40) and the proposed technique are shown in Figure 8, 9 and 10.

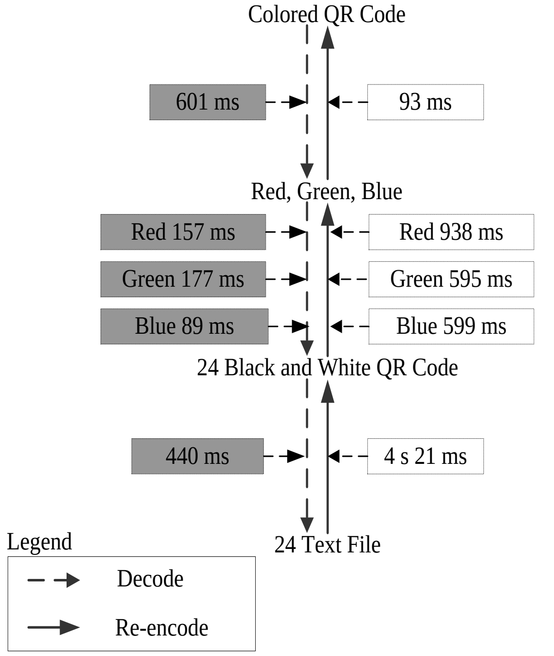

Figure 8. The process flow results for QR Code version 40

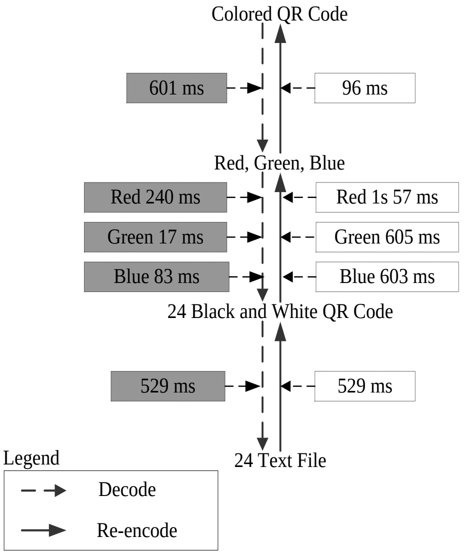

Figure 9. The process flow results for proposed technique level 1.

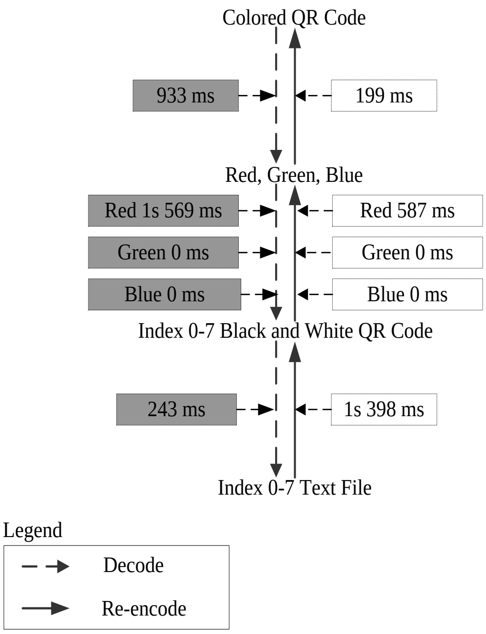

Figure 10. The process flow results for proposed technique level 2.

Based on Figures 8, 9 and 10, data were collected and tabulated in four tables ordered by level 1 and 2. The comparisons were made between each levels and the benchmark method. At the same time, each level is divided into decode and re-encode process. Table 2,3,4 and 5 shows the comparison between levels and resulted the different between them also separated within encode and decode processes.

Table 2. The comparison between benchmark and proposed technique in level 1 of decode process.

Level 1 (Decode)

From	To	Benchmark	Proposed Level 1	Difference
Colored QR Code	Red Green, Blue	601 ms	978 ms	**- 377 ms**
Red, Green, Blue QR Code	24 / Index 0 Black and White QR Code	423 ms	187 ms	236 ms
24 / Index 0 Black and White QR Code	24 / Index 0 File Text	440 ms	84 ms	356 ms
Total		1s 464 ms	1s 251 ms	231 ms

Table 3. The comparison between benchmark and proposed technique in level 1 of re-encode process.

Level 1 (Re-encode)

From	To	Benchmark	Proposed Level 1	Difference
24 / Index 0 File Text	24 / Index 0 Black and White QR Code	4s 21 ms	231 ms	3s 790 ms
24 / Index 0 Black and White QR Code	Red, Green, Blue	2s 132 ms	621 ms	1s 511 ms
Red, Green, Blue QR Code	Colored QR Code	93 ms	195 ms	**- 102 ms**
Total		6s 246 ms	1s 47ms	5s 199 ms

Table 4. The comparison between benchmark and proposed technique in level 2 of decode process.

Level 2 (Decode)

From	To	Benchmark	Proposed Level 2	Difference
Colored QR Code	Red Green, Blue	601 ms	933 ms	**-332 ms**
Red, Green, Blue QR Code	24 / Index 0-7 Black and White QR Code	423 ms	1s 569 ms	**- 1s 146 ms**
24 / Index 0-7 Black and White QR Code	24 / Index 0-7 File Text	440 ms	243 ms	197 ms
Total		1s 464 ms	2s 745 ms	**-1s 281 ms**

Table 5. The comparison between benchmark and proposed technique in level 2 of re-encode process.

Level 2 (Re-encode)

From	To	Benchmark	Proposed Level 2	Difference
24 / Index 0-7 File Text	24 / Index 0-7 Black and White QR Code	4s 21 ms	1s 398 ms	2s 623 ms
24 / Index 0-7 Black and White QR Code	Red, Green, Blue	2s 132 ms	587 ms	1s 545 ms
Red, Green, Blue QR Code	Colored QR Code	93 ms	199 ms	**-106 ms**
Total		6s 246 ms	2s 184 ms	4s 62ms

From Table 3, the overall result in level 1 (decode) shown that proposed technique able to complete the task within 1 second 251 milliseconds compared with benchmark which is 1second 464 milliseconds. The different range between these techniques is 231 milliseconds. During the process to generate the red, green and blue QR Code, the time consume of the proposed technique process is slower than benchmark technique due to the task to collect information of pixels in the red QR Code is implemented. This process is not available in the benchmark technique. As a result, this will generate or consume the time processing.

When the decoding process in Table 3 is completed, the re-encode process will take over after update, delete or add information done. From Table 4, the total time consume has a large different range between benchmark and proposed technique which is 5 seconds 199 milliseconds. The benchmark technique has to generate 24 black and white QR Code, meanwhile, the proposed technique only needs to generate only one black and white QR Code. This is why the benchmark technique consumes more times compared with proposed technique. During the generation of colored QR Code, proposed technique consumes nearly twice of total time consume in benchmark technique. This is because the proposed technique has to collect pixel information in green and blue QR Code. From the data in Table 4 shows the time consume process has been conqueror by the proposed technique. The proposed technique took only 1 second 47 milliseconds to complete the process and the benchmark technique has completed in 6 seconds 246 milliseconds.

In level 2, the decode and re-encode process are executed in order to recover the red QR Code. Table 5 shows the same result of the decode process in level 1 at Table 3. The proposed technique took 933 milliseconds to complete the decode process from colored QR Code to red, green and blue QR Code but the benchmark technique only took 601 milliseconds. For the next process, the proposed technique consumes 1 second 569 milliseconds, but benchmark process only took 423 milliseconds. The range difference is 1second 146 milliseconds. The reason is the red QR Code needs to separate the image file into a group of black and white QR Code which is starting from index 0 until 7. This process needs extra time to process. Overall, the time range difference between benchmark and proposed techniques is 1 second 281 milliseconds.

During the re-encode in level 2 at Table 6, the proposed technique has lead the time process with time range difference is 4 seconds 62 milliseconds. But in generating the colored QR Code, the benchmark QR Code has led to 106 milliseconds in time range difference. This result is same with level 1 re-encode processes and the reason is same with level 1 re-encode processes mention before.

From Table 6 below, it shows the information about overall Level 1 has time range difference with 5 seconds 430 milliseconds and Level 2 has time range difference with 2 seconds 781 milliseconds. From this result, the proposed techniques have a good performance compared with benchmark techniques.

Table 6. Level 1 and Level 2 time range difference.

		Benchmark	Proposed	Total Time (Level)	Total Time
Level 1	Decode	1s 464 ms	1s 251 ms	231 ms	5s 430ms
	Re-encode	6s 246 ms	1s 47ms	5s 199ms	
Level 2	Decode	1s 464 ms	2s 745 ms	**-1s 281ms**	2s 781ms
	Re-encode	6s 246 ms	2s 184ms	4s 62ms	

V CONCLUSION

In today's world that offers explosive information, various medium haven been used to store the required knowledge, information and data. And, this includes the use of QR Code which only requires a small size of printed area to store certain amount of data. This paper has demonstrate the use of a combination technique that allows the updating of information stored in a colored QR Code. As the undertaken experiment produces positive result, it is shown that updating information in QR Code can be realized using less computation time as compared to the existing method (i.e QR Code version 40). With such achievement, knowledge sharing via QR Code can further be expanded. Information stored in the code can always be updated according to the providers needs.

In the future, there is a need to investigate its effectiveness on other error correction level such as M, Q and H. The reason is to see whether it is affecting the performance of the updated QR Code processes. Another possible direction is on concurrent processing. This will lead to studies in the area of performance capability, updated information and security in the QR Code.

ACKNOWLEDGMENT

The authors acknowledge the support given by Universiti Utara Malaysia.

REFERENCES

Abas, A., Yusof, Y., & Kabir, F. (2015). Expanding the data capacity of QR codes using multiple compression algorithms and base64 encode/decode. *Jurnal Teknologi*, *1*.

Boob, A., Shinde, A., Rathod, D., & Gaikwad, A. (2014). Qr Code Based Mobile App and Business Process Integration. *International Journal of Multidisciplinary and Current Research*, *2*(Sept and Oct 2014), 1014-1017.

Falas, T., & Kashani, H. (1994). Two-Dimensional Bar-code Decoding with Camera-Equipped Mobile Phones. *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference* (pp. 2-5).

Grillo, A., Lentini, A., Querini, M., & Italiano, G. F. (2010). High Capacity Colored Two Dimensional Codes. *Proceedings of the International Multiconference on Computer Science and Information Technology* (Vol. 5, pp. 709-716).

Jagan, A., Akila, P., & Nasrin, N. (2015). QR Code Based E-voting System Using an Android Smart Phones, *13*(2), 263-267.

Jude J.L., M. (2015). Audio Data As An Encoding for Color Quick Response Code (CQR) 3D Barcodes. *Journal of Theoretical and Applied Information Technology*, *73*(2), 183-189.

Kato, H., & Tan, K. T. (2005). 2D barcodes for mobile phones. *Proceeding Mobile Technology, Applications and Systems, 2005 2nd International Conference* (p. 8).

Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Sinha, M., & Weippl, E. (2010). QR Code Security. *The 9th International Conference on Advances in Mobile Computing & Multimedia* (pp. 430-435).

Nikolaos, T., & Kiyoshi, T. (2010). QR-Code Calibration for Mobile Augmented Reality Applications. *SIGGRAPH 2010* (p. 4503).

Pandya, K. H., & Galiyawala, H. J. (2014). A Survey on QR Codes : in context of Research and Application. *International Journal of Emerging Technology and Advanced Engineering*, *4*(3), 258-262.

Paulin, D., & Suneson, K. (2008). Knowledge Transfer , Knowledge Sharing and Knowledge Barriers – Three Blurry Terms in KM, *10*(1), 81-91.

Pillai, P. N., & Naresh, K. (2014). Improving the Capacity of QR Code by Using Color Technique. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *3*(7), 10561-10568. doi:10.15662/ijareeie.2014.0307025

Schwartz, D. G. (2006). Encyclopedia of Knowledge Management. IGI Global. Retrieved from http://library.books24x7.com/toc.asp?bookid=14700.

Sutheebanjard, P., & Wichian, P. (2010). QR-Code Generator. *2010 Eighth International Conference on ICT and Knowledge Engineering* (pp. 89-92).

Victor, N. (2012). Enhancing the Data Capacity of QR Codes by Compressing the Data before Generation. *International Journal of Computer Applications (0975 –8887)*, *60*(2), 17-21.

Vongpradhip, S. (2013). Use Multiplexing to Increase Information in QR Code. *The 8th International Conference on Computer Science & Education (ICCSE 2013)* (pp. 361-364).