International Journal on Robotics, Automation and Sciences

Multi-Color Code with High Data Capacity

Zheng You Lim, Kok Swee Sim*

***Abstract*— Quick Response (QR) code is commonly utilized in this era to serve multiple purposes. In order to uplift the data size of QR codes, higher versions of QR codes are introduced for greater data size. Nonetheless, the current QR code has met the bottleneck of data size. The latest version of QR code (version 40) now can only encode up to 4 kilobytes of data. Besides, QR code version 40 is unable to be decoded easily due to the high layout density. Hence, this research's objective is to develop a Multi-color Code that involves multiple colors to uplift the data size. Similar to the QR Code, the Multi-color Code (McC) also consists of the features of auto rotation and error correction. The outcome of the research is the developed McC can achieve up to 10Kb. Besides, McC is capable to encode an image file for offline identification. The results show that the auto rotation feature, Reed-Solomon error correction algorithm, developed Lanczos-8 interpolation method and adaptive piecewise transformed normalization contribute to yielding 100% decoding accuracy.**

Keywords: QR Code, Image Processing, Color Code, Encoding, Decoding, Error Correction

I. INTRODUCTION

Barcode is a one-dimensional (1D) code that was developed collaboratively by Bernard Silver and Norman Joseph Woodland in 1945 [1]. It is an optical image with black rectangular blocks of different widths and the gap between them. The barcode is usually decoded by utilizing an optical scanner which involves the hardware system of laser technology. With the advancement of the mobile phone nowadays, the mobile phone can be easily used to decode the barcode by capturing the barcode using the mobile phone camera [2].

In this era of technology, people keep on demanding higher data capacity. The barcode consists of a data capacity of 48 numeric characters. Hence, it is no longer sufficient to fulfill the demand for higher capacity [3]. In this way, two dimensional (2D) barcode is developed to succeed the single-dimensional (1D) barcode. There are several types of 2D barcodes that have been developed. However, the Quick Response (QR) code is the most popular 2D code that is used in the market nowadays.

In the year 1994, Denso Wave Incorporated introduced the first 2D barcode which is known as Quick Response (QR) code [4]. Masahiro Hara from Denso Wave Incorporated has led a team in developing a standard for the QR code. The design of the QR Code is made up of only black and white colors. The black and white colors represent data '1' and '0' in the form of a rectangular module. The model of a QR code is composed of two major parts. These two parts are the function patterns and encoding region. There are three components that made up the function patterns. They are known as the timing pattern, alignment pattern, and finder pattern. They are the reference points used to determine the location and orientation of the QR code during the decoding process. The encoding region is comprised of information about the QR code such as the format and version of the QR code. Most importantly, the encoding region is the region where actual and error correction data are encoded.

Several attributes of the QR code have contributed to the QR code being distinguished compared to the

*Corresponding Author email: kssim@mmu.edu.my, ORCID: 0000-0003-2976-8825
Zheng You Lim and Kok Swee Sim are with the Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia; (e-mail: limzhengyou@gmail.com, kssim@mmu.edu.my).

International Journal on Robotics, Automation and Sciences (2022) 4:35-45
https://doi.org/10.33093/ijoras.2022.4.6
Manuscript received: 22 Mar 2022 | Revised: 28 Apr 2022 | Accepted: 31 May 2022| Published: 8 Jul 2022
Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

conventional barcode and other 2D barcodes. Firstly, the data storage capability of QR codes is massive whereby it can achieve up to 4 kB of encoded data. In addition, QR code is designed to encrypt several data types. Aside from numerical data where barcode does, it is able to encode binary codewords, alphanumerical codewords, and kanji words. Besides, QR code consists of an alignment pattern. This pattern can enable the QR code to be successfully decoded regardless of any captured direction and orientation. Lastly, the QR code utilizes the error correction algorithm which is known as the Reed-Solomon error correction. This unique algorithm allows the QR code to correct the error up to 30% [5].

The major purpose of introducing the QR code is to serve inventory tracking. However, nowadays the QR code has extended its usage to various fields in our daily life. QR codes can be found on online movie tickets, bus tickets, and cashless payment applications [6]. Thus, QR code has brought convenience and played an important role in our daily life.

As technology advances, people keep demanding more data capacity. Thus, the QR code had evolved from Version 1 to Version 40. At present, the QR code version 40 is able to achieve the highest data capacity of approximately 3 kilobytes of binary data [7]. Conversely, the high layout density of QR code Version 40 results in a high error rate. The camera resolution of the common mobile phone nowadays is still unable to capture the code clearly. In addition, some people may request the capability of a QR code to encode an image, a short article, or even a short video within the code without internet access [8]. In this way, people can directly reveal the image, article, or video anywhere and anytime. The data bottleneck of the QR code has restricted it to develop these features.

Hence, our research project utilizes multiple colors in encoding the data in order to increase the data capacity. It is hypothesized that the data capacity can be tripled by using 8 colors to encode data. Besides, specific color image enhancement and processing techniques are developed to serve the decoding purpose. It is hypothesized that the image processing techniques and the error correction algorithm will allow the decoder to decode the McC with higher accuracy.

II. Related Works

A. Color Quick Response Code

Researchers from the University of Brasilia propose the Colour Quick Response (CQR) code by employing 5 different colors with the intention of achieving double the data capacity of the QR code [9]. CQR code-5 is the first developed prototype and CQR code-9 is the second prototype. Both prototypes utilize the QR code Version 8 layout which consists of 49 x 49 modules, where the module is known as the tiniest square element in QR code that is used to store 1 bit of data. Figure 1(a) shows the example of CQR code-5 and Figure 1(b) shows the example of CQR code-9.

FIGURE 1. Example of (a) CQR code-5 (b) CQR code-9.

According to the encoding algorithm, the maximum capacity CQR code-9 can achieve is 576 bytes. However, the research did not state the performance of the decoder for the CQR code. This is due to the CQR layout did not consist of any reference color. In this way, it will lead to high recognition errors while performing the decoding process.

B. High Capacity Colour Barcode

In the year 2003, Microsoft Corporation introduced and patented the High Capacity Colour Barcode (HCCB) technology [10]. HCCB utilizes a grid of colored triangles to store information. Thus, the layout of the HCCB is comprised of colored triangles which are different as compared to the other two-dimensional codes that are comprised of square modules. The colored triangles are arranged sequentially in rows. Each row is separated by a white rectangular region. According to different requirements such as data size or data density (number of data per inch), the HCCB is customizable in terms of the number of colors involved. Thus, there are three versions that involve a different number of colors. The first version involves only two colors which are black and white. Then, the second version is introduced. This version involves four colors namely black, magenta, yellow, and cyan. Next, the latest version implemented is the third version. It involves eight colors which are white, black, red, magenta, green, yellow, blue, and cyan. Figure 2(a) shows the layout of 8-color HCCB and Figure 2(b) illustrates the layout of 4-color HCCB.

FIGURE 2. Example of (a) 8-Color HCCB (b) 4-Color HCCB.

As result, the 2-color HCCB is able to store 1 byte (8 bits) of data by using 8 symbols, where one symbol is represented by one colored triangle. The 4-color HCCB is capable to encode 1 byte of data by using 4 symbols. Besides, the 8-color HCCB can even achieve by using an average of 2.66 symbols to encode 1 byte of data. Microsoft has conducted its laboratory test to achieve the highest data capacity. The results show that the 8-color HCCB is able to store 2,000 binary bytes per square. The test is conducted by using a standard off-the-shelf printer to print out the HCCB on white paper and scan using a 600 dpi business card scanner [11]. However, the developed code can only be decoded using a scanner device but was not readied to be decoded by using a mobile phone camera.

C. High Capacity Colored Two Dimensional (HCC2D) Code

In the year 2010, researchers from the University of Tor Vergata, Rome, Italy published their developed High Capacity Colored Two Dimensional (HCC2D) Code [12]. The main objective of developing the HCC2D code is to uplift the data size while preserving the unique features of the QR code. Thus, for the HCC2D code layout, it follows exactly the same as the QR code to maintain full compatibility with the QR code. HCC2D extends the data capacity by embedding the data module with different colors from the color palette according to the data value. There are three kinds of color palettes that can be selected which are known as 4 colors palette, 8 colors palette, and 16 colors palette [12]. Figure 3(a) illustrates the example of HCC2D of 4 colors and Figure 3(b) illustrates the example of 16 colors.

FIGURE 3. Example of HCC2D (a) 4 colors (b) 16 colors.

For experimental setup, the researchers have realized a common Print&Scan scenario using an off-the-shelf 3-in-1 inkjet printer with print and scan capabilities. As result, the HCC2D code is able to achieve 1.881 KB/inch data density while the QR code is able to achieve 0.627 KB/inch data density and HCCB is able to achieve up to 2 KB/inch data density [12].

The HCC2D code's latest prototype utilizes 16 colors whereby it can compress four binary bits from the QR code to form one color module within the HCC2D code. Thus, theoretically, the HCC2D can achieve up to four times the data capacity of a QR code. However, there is no relevant result that can show the performance of the decoder. Recognizing the colors within the HCC2D with 16 colors may not be achievable due to the HCC2D code layout does not consist with any reference color to assist in color recognition.

III. MULTI-COLOR CODE ENCODING ALGORITHM

A. Multi-color Code Layout

Figure 4 illustrates the design layout of the developed Multi-color Code. In the designed layout, the Multi-color Code is comprised of the five major elements: data encoding region, quiet zone, reference alignment pattern, reference pattern, and timing pattern. The data encoding region is made up of color modules that carry the error correction information and actual information. The color module is the smallest square box that represents the encoded data. Each module is squared in size where the color of the module depends on the color multiplexing technique during the encoding procedure. The reference boundary of the code is comprised of four colors. They are known as cyan, yellow, magenta, and black colors in a rectangular form that enclosed the quiet zone and data encoding region. Each rectangle that makes up the boundary consists of a width of at least doubles the width of one color module. In addition, the length of the boundary should be at least 54 times the width of one color module. This reference boundary is the element used to allocate the position of the code. Besides, the colors involved in the reference boundary act as the reference colors for decoding purposes. The reference alignment pattern is made up of three square blocks. They are positioned on the top left, top right, and bottom-right corners of the reference boundary. The three square blocks are comprised of three primary colors which are red, green, and blue colors. It is employed to determine the orientation of the Multi-color Code and aid in performing auto rotation when the Multi-color Code is misaligned. In addition, the colors embedded in the reference alignment pattern also acts as the reference colors for decoding purpose. The quiet zone is the white margin that distinguishes the data encoding region from the reference boundary. The intention of designing this zone is to ease the process of extracting the data encoding region for further decoding purposes. Thus, it should be white and clean without the existence of any marking. The minimum size of the gap should double the width of one color module. The timing pattern is located in the quiet zone in the middle of the reference boundary and data encoding region. The existence of the timing pattern also has divided the data encoding region into four whereby the timing pattern. The timing pattern is designed to aid in detecting the location of each module within the data encoding region.

B. Conversion of Image into ASCII Characters

In order to implement the feature of storing an image within the McC, it is proposed to convert the image into ASCII characters before encoding it into the McC. Vice versa, the ASCII characters are utilized to convert back to the image file while decoding the McC back into an image file. Figure 5 shows the flowchart on how the conversion between image and ASCII characters works.

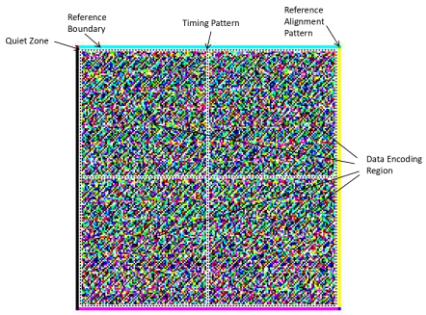

FIGURE 4. Multi-color code layout.

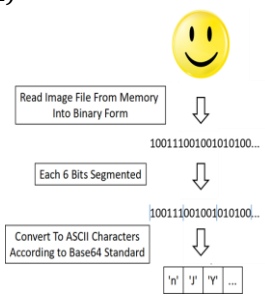

FIGURE 5. Conversion from image to ASCII characters.

The process of conversion from the image into ASCII characters starts with reading the selected image file in binary form as to how the image is stored in the memory. Then, every six bits of the binary data are segmented and every six bits are converted into ASCII characters according to the Base64 standard [13].

C. Encoding Process for Multi-color Code

The encoding process which receives the string as input and provides the Multi-color Code as output is shown in Figure 6.

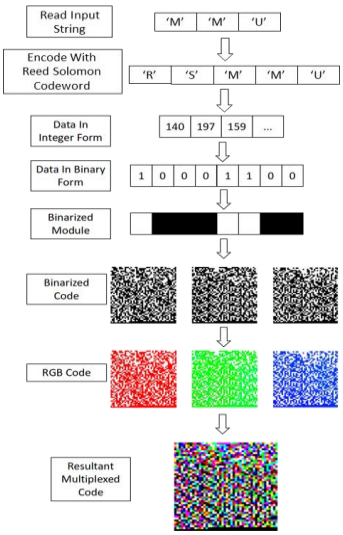

FIGURE 6. Encoding process for multi-color code.

The information inputted by the user is read in text form, also known as string form. Next, the string is used to construct the Reed Solomon codeword using the Reed Solomon encoding algorithm [14]. The algorithm follows by converting the Reed Solomon codeword into binary numbers. These binary numbers are further utilized to convert into color values in hexadecimal form using the Color Multiplexing Method [15]. The concept of this algorithm is to convert the data into three layers of binarised code with identical sizes and the total number of modules. Then, each layer of binarised code is signified by one primary color channel. In this way, the red channel signifies the first binarised code, the green channel signifies the second binarised code and the blue channel signifies the third binarised code. At last, all the three layers are multiplexed together using the color addition method which will eventually produce the resultant code with colored modules. Lastly, the resultant code will be further embedded into the data encoding region of Multi-color Code.

IV. Multi-Color Code Decoding Algorithm

A. Decoding Process for Multi-color Code

The decoding process for McC is illustrated in Figure 7.

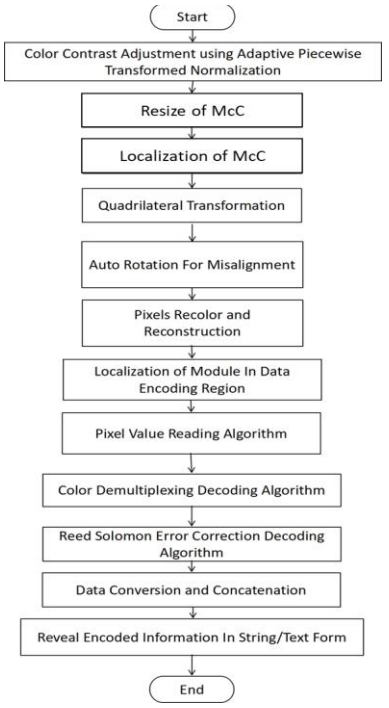

FIGURE 7. Flowchart of decoding process for multi-color code.

The decoding procedure starts with color contrast adjustment using the developed adaptive piecewise transformed normalization. Then, the McC is resized by using the developed Lanczos-8 interpolation method. Next, localization of the McC is performed in order to allocate the position of the McC on the captured image. The procedure is followed by the auto rotation to correct the misalignment of the captured McC. Next, the technique of quadrilateral transformation [16] is applied to ensure that the McC is resized to the desired size in perfect square form. The process continues with the pixels recolor and reconstruction technique to repaint the color of every pixel within the data encoding region. This process is employed to uplift the accuracy of pixel value reading. Eventually, the color values retrieved using the pixel reading algorithm are decoded into ASCII characters using the color demultiplexing decoding algorithm. Then, all the integer values are passed through the Reed Solomon Error Correction Decoder [17] to perform error detection and auto error correction. After that, the actual message in integer form is extracted from the Reed Solomon block. The actual message in integer form is then converted into character form according to the ASCII table, and concatenated together

back into string/text form. At last, the concatenated string which is in the readable form will be revealed to the user.

B. Adaptive Piecewise Transformed Normalization

This technique is used to perform the automatic contrast and brightness adjustment to ease the process of color recognition. This technique is specially developed in order to enhance the colored images instead of grayscale images. First, the histogram of each red, green, and blue (RGB) is constructed. The discrete functions are shown in (1), (2), and (3).

$$H_{red}(I_k) = n_k \tag{1}$$

$$H_{green}(I_k) = n_k \tag{2}$$

$$H_{blue}(I_k) = n_k \tag{3}$$

Where H_{red} is the histogram for red channel, H_{green} is the histogram for green channel, H_{blue} is the histogram for blue channel, I_k is the kth intensity level and n_k is the number of pixels with the intensity of I_k .Then, the histogram is normalised using (4), (5) and (6).

$$\overline{H}_{Red}(I_k) = \frac{H_{Red}}{h * w} \tag{4}$$

$$\overline{H}_{Green}(I_k) = \frac{H_{Green}}{h * w} \tag{5}$$

$$\overline{H}_{Blue}(I_k) = \frac{H_{Blue}}{h * w} \tag{6}$$

Where $\overline{H}_{Red}$ is the normalised histogram for red channel, $\overline{H}_{Green}$ is the normalized histogram for the green channel, $\overline{H}_{Blue}$ is the normalized histogram for the blue channel, h is the height of the image and w is the width of the image.

By using the three normalized histograms, every position of the highest amplitude components is determined. The positions of the highest amplitude component are denoted as $P_{Red}^{Amplitude}$, $P_{Green}^{Amplitude}$ and $P_{Blue}^{Amplitude}$. For each red, green and blue component, if the $P^{Amplitude}$ lies within the left segment of the non-zero histogram components but not in the first component of the non-zero histogram components, the histogram is segregated into two secondary histograms. The first secondary histogram starts from 0 to the$P^{Amplitude} - 1$, and the second secondary histogram starts from $P^{Amplitude}$ to the maximum intensity level, known as$L - 1$. The first secondary histogram is applied with histogram equalization as shown in the equation as shown in (7).

$$T_{HE}(I_k) = \frac{L-1}{w * h} \sum_{j=0}^{k} n_j \tag{7}$$

Then, the second secondary histogram is applied by using Contrast Limited Adaptive Histogram Equalisation. This technique starts to calculate the cumulative distribution function by using the histogram as shown in (8).

$$C(k) = (L-1) \sum_{i=0}^{k} \overline{H}_i(k) \tag{8}$$

where $C(k)$ is the cumulative distribution function, $\overline{H}_i(k)$ is the normalized value of the k intensity.

By using the calculated cumulative distribution function, the distribution matrix D_k from 0 to L-1 is formed. Then, the least mean square algorithm is applied to determine the most suitable weight, W^* for the desired output, Y_k and the maximum allowed error, e_k. Thus, Y_k is the matrix of linear variation of contents from 0 to $L - 1$ in a uniform manner. D_k is the input matrix that is used to estimate to best fit W^*. In order to eradicate the nonlinear values of the cumulative distribution function by using the least mean square method, e_k is assumed constant. Then, the Weiner weight, W_o is calculated by using (9), (10), and (11) respectively.

$$W_o = R^{-1}P \tag{9}$$

$$R = D_k D_k^T \tag{10}$$

$$P = D_k Y_k \tag{11}$$

The algorithm proceeds with the steepest descent method, where the k+1 iteration's weight, W_{k+1} can be calculated by using (12).

$$W_{k+1} = W_k - \mu \nabla_k \tag{12}$$

Where W_k is iteration k's weight and μ is the gain constant with value of 0.01. ∇_k is the gradient estimate of the image that can be calculated by using (13).

$$\nabla_k = 2\mu e_k D_k \tag{13}$$

Thus, it yields the equation as (14).

$$W_{k+1} = W_k - (0.01) * 2\mu e_k D_k \tag{14}$$

Then, operator E is used to represent the expected value of the variable as shown in (15), (16), and (17) to find out W^* in (19).

$$E[W_{k+1}] = E[W_k] + 2\mu E[e_k D_k] \tag{15}$$

$$E[W_{k+1}] = E[W_k] + 2\mu (E[Y_k D_k] - E[D_k D_k^T W_k]) \tag{16}$$

$$E[W_{k+1}] = E[W_k] + 2\mu (P - RE[W_k]) \tag{17}$$

$$E[W_{k+1}] = (1 - 2\mu R) E[W_k] + 2\mu R W^* \tag{18}$$

$$W^* = 0.5\mu^{-1} R^{-1} (E[W_{k+1}] - (1 - 2\mu R) E[W_k] \tag{19}$$

Whereby $E[W_{k+1}]$ is the expected value of weight at iteration $k + 1$, $E[W_k]$ is the expected value of weight at iteration k, and W^* is the desired output. Thus, the new cumulative distribution function is reformed according to the best fit line which is calculated as $D_k W^*$. According

to the new cumulative distribution function, the new probability of occurrence of each pixel intensity $\overline{\mathrm{H_{new}}}(\mathrm{k})$ at $\mathrm{k^{th}}$ intensity value is calculated by using (20), (21) and (22).

$$\overline{H_{new}}(k) = \overline{H}(k) - \overline{H_{new}}(k-1)\ , 1 < k < L-1 \quad (20)$$

$$\overline{H_{new}}(k) = \overline{H}(0) \quad (21)$$

$$T_{CLAHE}(I_k) = \overline{H_{new}}(k) \quad (22)$$

Whilst, if $\mathrm{P^{Amplitude}}$ lies within the right segment of the non-zero histogram components, the Contrast Limited Adaptive Histogram Equalisation is directly applied to the histogram. Thus, the piecewise transformed function is shown as (23).

$$T(I_k) = T_{HE}(I_k) + T_{CLAHE}(I_k) \quad (23)$$

Where $\mathrm{T(I_k)}$ is the piecewise transformed function. The piecewise transformed functions for each R, G, and B component are then applied to the original image to produce an enhanced contrast image.

C. Lanczos-8 Interpolation Technique

This interpolation method is developed to resize the captured McC to desired window size before extracting it out for decoding. The interpolation technique employed to estimate the value for the interpolated pixel refers to the Lanczos resampling technique with the mask size of 8 x 8, thus namely the Lanczos-8 interpolation method. Lanczos resampling technique is usually employed in increasing the sampling rate of discrete signals in one dimensional. The Lanczos resampling technique is developed based on the Sinc function whereby the formula is shown in (24).

$$f(x) = \begin{cases} \sin c(x) \cdot \sin c(\frac{x}{s}) & if \quad -s < x < s \\ 0 & otherwise \end{cases} \quad (24)$$

Where $f(x)$ is the Lanczos is function and s is the kernel size which is a positive integer. Figure 8 shows the Lanczos window with different kernel size (s), where red indicates $s = 1$, green indicates $s = 2$ and blue indicates $s = 3$.

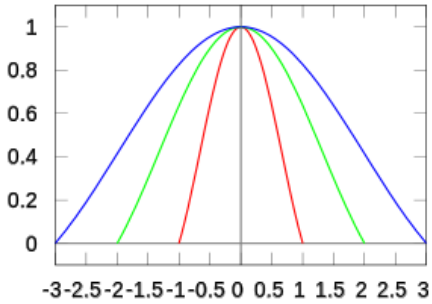

FIGURE 8. Lanczos window with different kernel size.

Figure 9 shows the discrete signal with Lanczos interpolation of kernel size, $s = 1$.

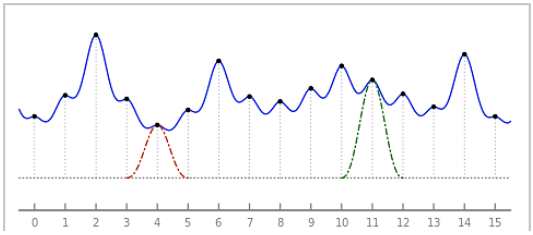

FIGURE 9. Discrete signal with interpolation value on 4th and 11th samples.

Thus, the Lanczos resampling technique is required to be adapted into two-dimensional (2D) planes in order to achieve image interpolation. The interpolation algorithm starts with calculating the scale ratio for the horizontal plane and vertical plane. The scale ratio in the horizontal plane is given by (25) and the scale ratio in the vertical plane is given by (26).

$$W_k = \frac{S_x}{8} \quad (25)$$

$$H_k = \frac{S_y}{8} \quad (26)$$

Where W_k is the width of the kernel and H_k is the height of the kernel. Then, horizontal interpolation is performed on all three color channels. The Lanczos function for horizontal interpolation is given by (27).

$$f_H(x) = \begin{cases} \sin c(x) \cdot \sin c(\frac{x}{W_k}) & if \quad -W_k < x < W_k \\ 0 & otherwise \end{cases} \quad (27)$$

The horizontal interpolation for each interpolated pixel on the red, green and blue channel is given by (28), (29) and (30).

$$R_{\mathrm{int}}(x) = \sum_{i=\lfloor x \rfloor - W_k + 1}^{\lfloor x \rfloor + W_k} f_H(x-i) \cdot R(i) \quad (28)$$

$$G_{\mathrm{int}}(x) = \sum_{i=\lfloor x \rfloor - W_k + 1}^{\lfloor x \rfloor + W_k} f_H(x-i) \cdot G(i) \quad (29)$$

$$B_{\mathrm{int}}(x) = \sum_{i=\lfloor x \rfloor - W_k + 1}^{\lfloor x \rfloor + W_k} f_H(x-i) \cdot B(i) \quad (30)$$

Where R_{int} is the red value for the interpolated pixel, G_{int} is the green value for the interpolated pixel, B_{int} is the blue value for the interpolated pixel, $\lfloor x \rfloor$ is the floor value of x, R is the red value of a pixel with X-coordinate of i, G is the green value of a pixel with X-coordinate of i, B is the blue value of a pixel with X-coordinate of i. The interpolation algorithm follows with vertical interpolation whereby the Lanczos function for vertical interpolation is given by (31).

$$f_V(x) = \begin{cases} \sin c(x) \cdot \sin c(\frac{x}{H_k}) & if \quad -H_k < x < H_k \\ 0 & otherwise \end{cases} \quad (31)$$

The vertical interpolation for each interpolated pixel on the red, green and blue channel is given by (32), (33) and (34).

$$R_{\text{int}}(x) = \sum_{i=\lfloor x \rfloor - H_k + 1}^{\lfloor x \rfloor + H_k} f_V(x-i) \cdot R(i) \quad (32)$$

$$G_{\text{int}}(x) = \sum_{i=\lfloor x \rfloor - H_k + 1}^{\lfloor x \rfloor + H_k} f_V(x-i) \cdot G(i) \quad (33)$$

$$B_{\text{int}}(x) = \sum_{i=\lfloor x \rfloor - H_k + 1}^{\lfloor x \rfloor + H_k} f_V(x-i) \cdot B(i) \quad (34)$$

After all the values of the interpolated pixels in both horizontal and vertical planes are calculated using the Lanczos-8 interpolation method, all the interpolated pixels are combined together with the pixels of the original image. At last, this technique is able to produce the resultant image with the desired scale.

D. Localization of Module in Data Encoding Region

The process of the localization of the module is illustrated in Figure 10.

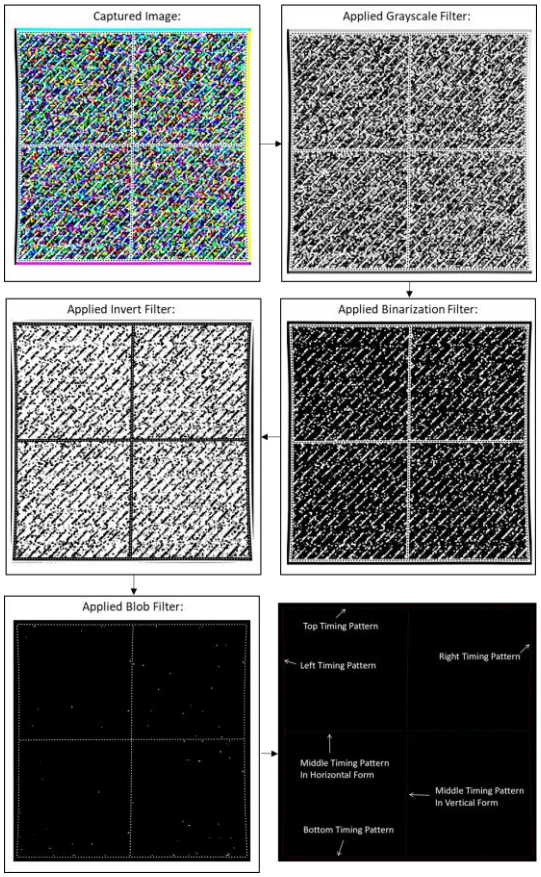

FIGURE 10. Process of localization of data module.

The process starts with applying a grayscale filter [19] to the captured image. Then, a binary filter [20] is applied to the image followed by an invert filter. Next, the timing patterns are extracted by applying a blob filter with a pre-defined size. At last, the blobs that are not within the region of the timing pattern are filtered out. In this way, blobs that made up the timing pattern can be obtained. The position of each blob is obtained by using the intersection point between two lines constructed by the timing pattern. Figure 11 presents all the coordinates of the module within the data encoding region that are localized on the interception between two red lines.

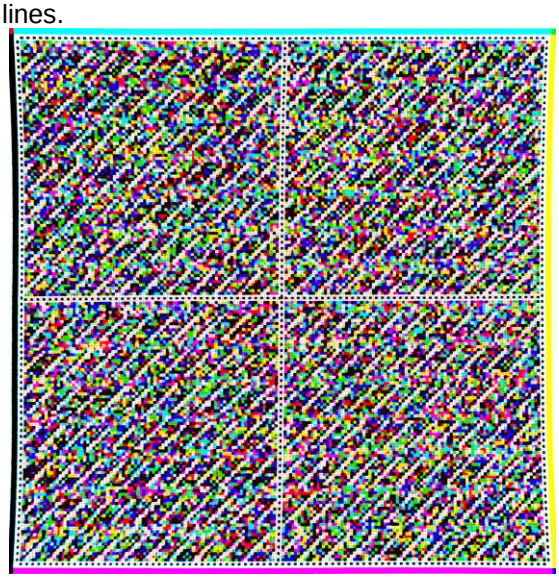

FIGURE 11. Localization of module using timing pattern.

E. Decoding Process for Multi-color Code

Figure 12 shows the color demultiplexing decoding algorithm to decode the McC back into string form.

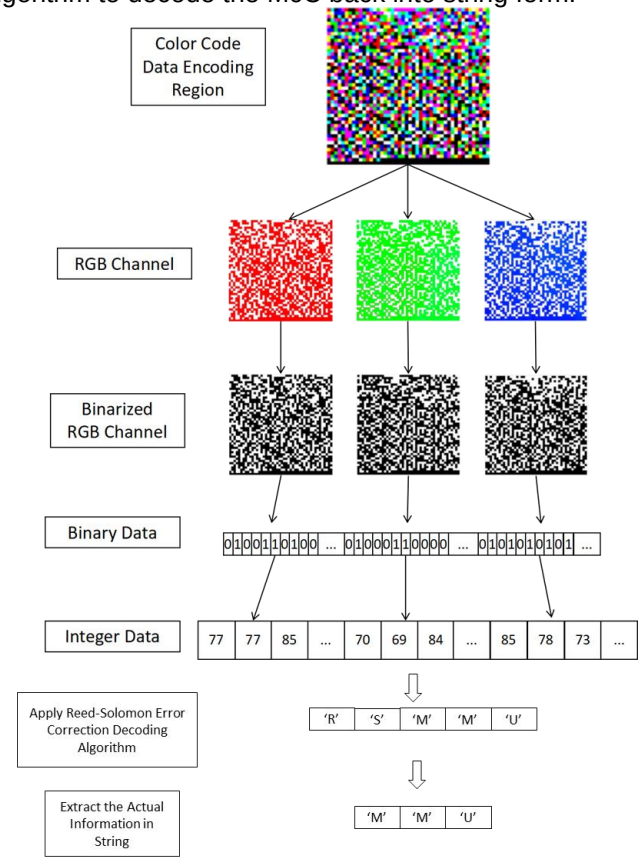

FIGURE 12. Decoding algorithm for multi-color code.

Once the data encoding region of the McC is extracted, three layers of the RGB channel are extricated from the colored data encoding region. Then,

each RGB channel is binarised and eventually forms three binarised data encoding regions. The algorithm follows by reading the color of every module from the three binarised data encoding region starting with the red channel, followed by the green channel, and lastly blue channel. During this process, every module is converted into binary data whereby the white module is converted into '1' and the black module is converted into '0'. Once all three layers of binarised data encoding region are converted into binary data, each 8 bit of the data is converted into data in decimal value. All the data in decimal form are then stored in the memory in array form. Then, the Reed-Solomon decoding algorithm [21] is applied to the decimal data and converted into ASCII characters. At last, the actual data is extracted from the Reed Solomon block and revealed to the user [22].

F. Conversion from ASCII Characters to Image

Since the developed McC is designed with the feature to encode an image, the conversion from ASCII characters to image form is mandatory [23]. The process is illustrated in Figure 13.

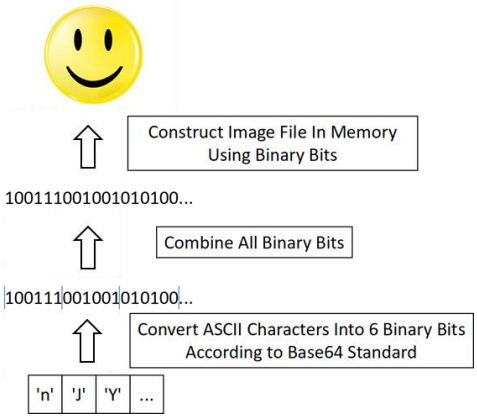

FIGURE 13. Conversion from ASCII characters to image.

During the decoding process, the decoded ASCII characters are used to convert back into data in 6-bit binary form according to the Base64. Then, all the 6-bits binary data are concatenated into one binary data. Lastly, the concatenated binary data are used to construct the image and store it in the memory.

V. RESULTS

A. Experimental Setup

The developed McC is printed on the standard white A4 paper with a thickness of 80 grams per square meter (gsm). The printer used to print out the McC is the Canon PIXMA E410 inkjet color printer. The maximum printing resolution of the printer is 4800 dots per inch (dpi) on horizontal and 600 dpi on vertical. The McC is printed in several dimensions: 100 mm x 100 mm, 80 mm x 80 mm, 60 mm x 60 mm, 50 mm x 50 mm, 40 mm x 40 mm, 30 mm x 30 mm.

The printed McC is captured with a mobile phone whereby its model is Huawei Mate 9. The camera resolution of the phone is 20 megapixels (20 MP). In order to ensure constant lighting conditions, the printed McC is captured in a box whereby the box is equipped with a white LED strip on top. A ruler is placed on the bottom of the box to measure the captured distance between the printed McC and the mobile phone. Figure 14 shows the hardware setup for capturing the printed McC.

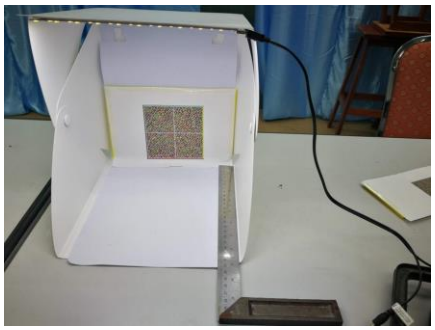

FIGURE 14. Hardware setup for capturing printed McC.

B. Performance of Adaptive Piecewise Transformed Normalisation

The performance of the developed contrast enhancement method namely Adaptive Piecewise Transformed Normalisation is also assessed by using the two parameters namely Similarity Index Measurement (SSIM) and Peak Signal to Noise Ratio (PSNR). The technique is compared with histogram equalization and contrast stretching (also known as normalization). Figure 15 shows the visualized SSIM values for contrast enhancement techniques and Figure 16 shows the visualized PSNR values for contrast enhancement techniques.

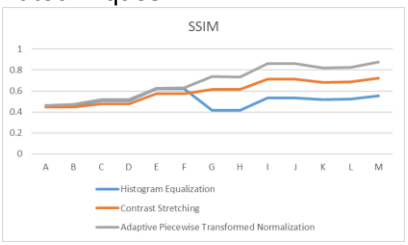

FIGURE 15. Visualized SSIM values in line graph for contrast enhancement methods.

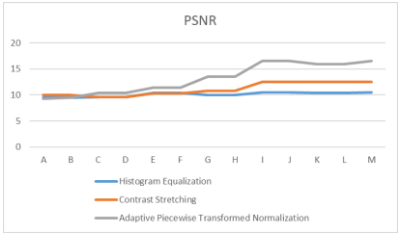

FIGURE 16. Visualized PSNR values in line graph for contrast enhancement methods.

The higher value in SSIM and PSNR signifies the higher performance yield in contrast enhancement. Thus, the data shows that the developed Adaptive

Piecewise Transformed Normalisation yields the best performance in contrast to the other two contrast enhancement techniques in terms of SSIM and PSNR values.

C. Performance for Lanczos-8 Interpolation Technique

The performance of the developed interpolation method for image scaling namely the Lanczos-8 Interpolation Technique is also assessed by using SSIM and PSNR. The developed technique is compared with the conventional nearest neighbor, bilinear, and bicubic interpolation methods. Figure 17 shows the visualized SSIM values for interpolation techniques and Figure 18 shows the visualized PSNR values for interpolation techniques.

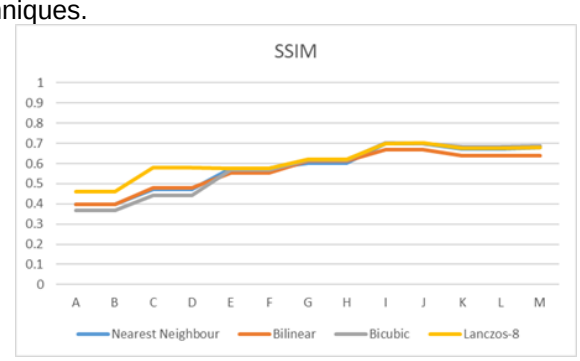

FIGURE 17. Visualized SSIM values in line graph for interpolation method.

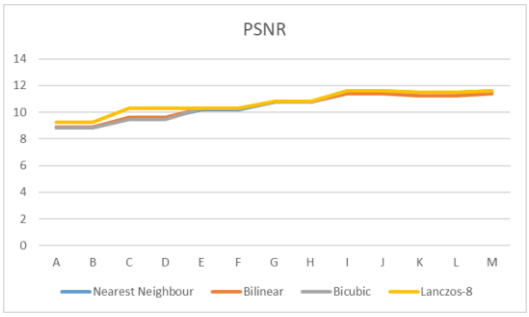

FIGURE 18. Visualized PSNR values in line graph for interpolation methods.

The higher value in SSIM and PSNR indicates better performance in interpolation. Thus, the data shows that the developed Lanczos-8 interpolation yields the best results as compared to the other three interpolation methods. This indicates that the developed Lanczos-8 interpolation method is the best technique to perform interpolation for color images.

D. Decoding Accuracy

The decoding accuracy of the decoder application is assessed by the similarity between the decoded data as compared to the encoded data in character form. The value of the accuracy in percentage is calculated based on the average accuracy of total 50 samples for each captured distance. All the 50 samples are captured in random orientation. Figure 19 shows the visualized data for the McC decoding accuracy.

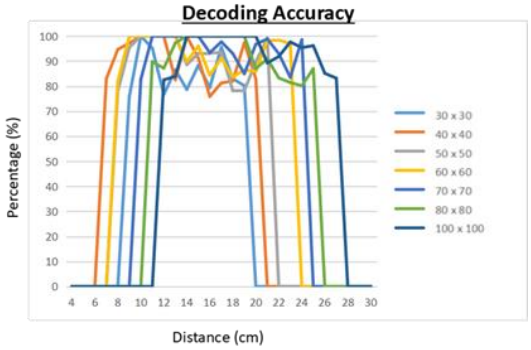

FIGURE 19. Visualized data for McC decoding accuracy.

The results show that the McC is able to achieve 100 percent with the contribution of the Reed Solomon error correction algorithm, reference colors, and timing pattern. The performance has proven the reliability of the Colour Code system.

E. Data Capacity and Data Density

The performance of the developed McC is examined by benchmarking with the data capacity of QR code, HCCB, HCC2D, CQR Code-5, and CQR Code-9. The results are computed in Table 1.

TABLE 1. Data capacity of various code.

Code	Data Capacity in Byte
QR Code	3916
HCCB	91
HCC2D	784
CQR Code-5	600
CQR Code-9	1200
Multi-color Code	10710

The performance of the implemented Colour Code is also examined by the data density. The data density is defined as the number of data in byte form per square inch which can be stored and decoded accurately. The data density of the QR code, HCCB, HCC2D, CQR Code-5, CQR Code-9, and McC are computed in Table 2.

TABLE 2. Data density of various code.

Code	Data Capacity in Byte
QR Code	0.627
HCCB	2.000
HCC2D	1.811
CQR Code-5	2.057
CQR Code-9	3.086
Multi-color Code	7.677

The results show that the McC has the highest data capacity and data density as compared to all the developed codes.

VI. CONCLUSION

In conclusion, the Multi-color Code (McC) system is successfully developed. The system consists of the personal computer-based encoder application and decoder application. The encoder application is able to encode either ASCII characters or an image file up to 7Kb whereby the total capacity is 10Kb. The result proves that the encoding method using 8 colors is able to triple the data capacity compared to the ordinary QR code. The decoder application is able to decode the encoded information back to ASCII data or image form from the captured image. With the contribution of the developed image quality enhancement techniques, auto rotation feature, and Reed Solomon error correction algorithm, the decoder application is able to achieve up to a 100% decoding accuracy rate. This shows that the hypothesis is proven true that the developed image processing techniques and error correction algorithm allow the decoder to decode the McC in high accuracy manner. The results have justified the innovativeness and reliability of the developed McC system. As result, the developed McC has breakthrough the data capacity bottleneck of QR codes and other codes as well. Besides, this project also contributes new and effective image processing techniques which are useful in color image processing. As this technology is developed using a computer system, one future work recommendation is to implement this technology into the mobile phone application. It is strongly affirmed that the developed McC system has the potential to be commercialized and generally utilized in our future daily life.

ACKNOWLEDGMENT

In this research, we would like to thanks Multimedia University to provide Graduate Research Assistant scholarship scheme to support the research scholar to conduct this research work.

AUTHOR CONTRIBUTIONS

Zheng You Lim: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Kok Swee Sim: Project Administration, Supervision, Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

REFERENCES

[1] F. Charles, "The Killer App - Bar None," *Am. Way*, 2001. URL: https://web.archive.org/web/20100112043409/. (Accessed: 2 April 2022)

[2] E. Ohbuchi, H. Hanaizumi, and L.A. Hock, "Barcode Readers Using the Camera Device in Mobile Phones," in *Proc. 2004 Int. Conf. Cyberworlds*, pp. 260–265, 2004. DOI: https://doi.org/10.1109/CW.2004.23

[3] S. Tiwari, "An Introduction to QR Code Technology," in *Proc. 2016 Int. Conf. Inf. Technol. (ICIT)*, pp. 39–44, 2016. DOI: https://doi.org/10.1109/ICIT.2016.021 (Accessed: 2 April 2022)

[4] "History of QR Code," *QRcode.com*, Denso Wave, 2019. URL: https://www.qrcode.com/en/history/. (Accessed: 2 April 2022)

[5] "What is a QR Code?," *QR Code*, 2019. URL:https://www.qrcode.com/en/about/. (Accessed: 2 April 2022)

[6] L. Jenny, "Tesco's Cool QR Code Advertising Campaign," *Vancouver Sun*, 2012. URL:https://vancouversun.com/news/staff-blogs/tescos-cool-qr-code-advertising-campaign. (Accessed: 2 April 2022)

[7] I. Tkachenko, W. Puech, O. Strauss, J.M. Gaudin, C. Destruel, and C. Guichard, "Improving the Module Recognition Rate of High Density QR Codes (Version 40) by Using Centrality Bias," in *Proc. 2014 4th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, pp. 1–6, 2014. DOI: https://doi.org/10.1109/IPTA.2014.7001950

[8] "QR Codes For Sharing Videos," *QR Stuff*, 2018. URL: https://web.archive.org/web/20201026213309/https://blog.qrstuff.com/2018/02/26/qr-codes-for-video. (Accessed: 2 April 2022)

[9] N. Taveerad and S. Vongpradhip, "Development of Color QR Code for Increasing Capacity," in *Proc. 11th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, pp. 645–648, 2015. DOI: https://doi.org/10.1109/SITIS.2015.42

[10] A. Grillo, A. Lentini, M. Querini, and G.F. Italiano, "High Capacity Colored Two Dimensional Codes," in *Proc. Int. Multiconf. Comput. Sci. Inf. Technol.*, pp. 709–716, 2014. DOI: https://doi.org/10.1109/IMCSIT.2010.5679869

[11] "High Capacity Color Barcodes (HCCB)," *Microsoft Research*, 2007. URL:https://www.microsoft.com/en-us/research/project/high-capacity-color-barcodes-hccb/. (Accessed: 2 April 2022)

[12] A. Grillo, A. Lentini, M. Querini, and G.F. Italiano, "High Capacity Colored Two Dimensional Codes," in *Proc. Int. Multiconf. Comput. Sci. Inf. Technol.*, pp. 709–716, 2014. DOI: https://doi.org/10.1109/IMCSIT.2010.5679869

[13] R. Rahim, et al., "Combination Base64 Algorithm and EOF Technique for Steganography," *J. Phys.: Conf. Ser.*, vol. 1007, p. 12003, 2018. DOI: https://doi.org/10.1088/1742-6596/1007/1/012003

[14] D. Kythe and P. Kythe, "Reed-Solomon Codes," *Algebraic Stochastic Coding Theory*, vol. 1, no. 3, pp. 257–286, 2012. DOI: https://doi.org/10.1201/b11707

[15] R.A.S. Ferreira and P.S. André, "Colour Multiplexing of Quick-Response (QR) Codes," *Electron. Lett.*, vol. 50, no. 24, pp. 1828–1830, 2014. DOI: https://doi.org/10.1049/el.2014.2501

[16] K.T. Tan, C. Douglas, K. Hiroko, and S.K. Ong, "Designing a Color Barcode for Mobile Applications," *IEEE Pervasive Comput.*, vol. 11, no. 2, pp. 50–55, 2012. DOI: https://doi.org/10.1109/MPRV.2010.67

[17] B. Tepekule, U. Yavuz, and A.E. Pusane, "On the Use of Modern Coding Techniques in QR Applications," in *Proc. 2013 21st Signal Process. Commun. Appl. Conf. (SIU)*, pp. 1–4, 2013. DOI: https://doi.org/10.1109/SIU.2013.6531318

[18] Z. Yang, H. Xu, J. Deng, C.C. Loy, and W.C. Lau, "Robust and Fast Decoding of High-Capacity Color QR Codes for Mobile Applications," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6093–6108, 2018. DOI: https://doi.org/10.1109/TIP.2018.2855419

[19] Z.Y. Chen, B.R. Abidi, D.L. Page, and M.A. Abidi, "Gray-Level Grouping (GLG): An Automatic Method for Optimized Image Contrast Enhancement—Part II: The Variations," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2303–2314, 2006. DOI: https://doi.org/10.1109/TIP.2006.875201

[20] K.S. Sim, C.K. Toa, and C.W. Ho, "Cubic Spline Hermite Interpolation with Linear Least Square Regression for Single Scanning Electron Microscope Image Signal-to-Noise Ratio Estimation," *Eng. Lett.*, vol. 47, no. 4, pp. 1–13, 2020. URL: https://www.iaeng.org/IJCS/issues_v47/issue_4/IJCS_47_4_15.pdf. (Accessed: 2 April 2022)

[21] K.S. Sim, C.C. Lim, and S.C. Tan, "Contrast Enhancement in Endoscopic Images Using Fusion Exposure Histogram Equalization," *Eng. Lett.*, vol. 28, no. 3, pp. 1–7, 2020. URL: https://www.engineeringletters.com/issues_v28/issue_3/EL_28_3_09.pdf. (Accessed: 2 April 2022)