Improved QR Codes: Increasing Data Capacity through Dynamic Resizing and Multicolor Encoding

Research Innovations

Chandan Singh

January 29, 2014

Abstract

The project aims to transmit the greatest amount of data possible by scanning the surface of a phone. To do this, the project explores two methods: it modifies the existing QR Code standard to elongate and fill any rectangular surface and it utilizes multiple colors to codes to encode more data. The project concludes that elongating black-and-white codes is currently the most effective way to store more data in a code, but multicolored codes will likely become more useful in the future.

Background

Quick-Response codes (known as QR Codes) have gained widespread popularity in today's world as a way to quickly and conveniently transfer data. QR Codes are square, two-dimensional barcodes that are scanned by a phone (or another scanning device) and transfer a small amount of data. Often, this data contains only a URL to a webpage where a user can get more data. This project aims to increase the amount of data a QR code can encode by adding multiple colors to it. This would allow a user to easily transmit data either from phone to phone, or by scanning hard copies of printed codes.[1]

The QR Code standard includes several features that are necessary in order to properly send information to a scanner (shown in Figure 1). First, a quiet zone must be located around the code for it to be properly detected. Second, position patterns must be located at the corners of the code for it to be properly detected. Then, alignment patterns and timing patterns must be located at particular positions on the code to ensure the code is properly aligned. Fourth, the code must have information on its version and its format.[1]

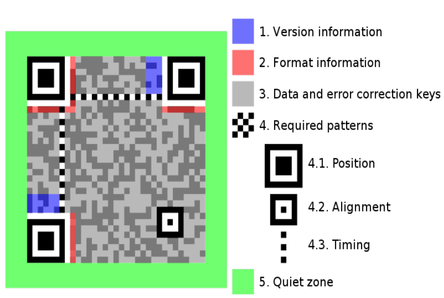

Figure 1. Important background parts of QR code standard [2]

Further complexities are added to the code because of the QR Code's error correction bits (shown in Figure 2.) The code's error correction bits use Reed-Solomon error correction algorithms to store extra bits that help to scan the code. Even if part of the code is misread or damaged, the entire message can still be transmitted because the error correction bits convey the missing info. Adding color to these codes would corrupt this process because the Reed-Solomon error correction algorithm is based on ones and zeroes. When part of the code is misread in color, more bits are missing, and it is difficult for the error correction codes to fully recuperate the missing values.[1]

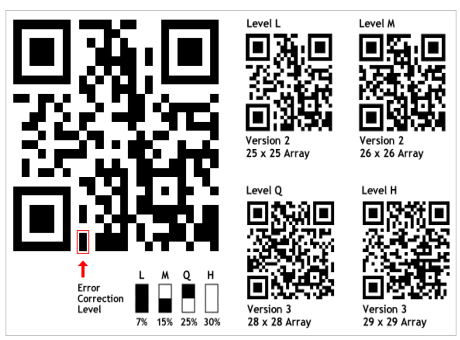

Figure 2. Placement of error correction bits according to QR standard[4]

Methodology

The project was started using a pre-existing library for generating and reading QR Codes from Google called Zebra Crossing. Zebra Crossing is written in Java for the Android operating system, and this project follows that framework.[3] The project uses two methods to add more data to the QR Codes: it add multiple colors to the codes and it elongates their square structures.

Method 1: Color

The methodology to be used bypasses many of the intricacies involved in creating QR Codes by using an idea from a previous attempt to add multiple colors to QR Codes (shown in Figure 3.) Instead of tampering with their error correction bits, the method generates three different QR Codes each encoding on third of the data. Then, it assigns each of these three codes one of the basic color channels: red, green, or blue. These colors are chosen because these are the channels that the Android camera and hardware uses as a standard. It combines each of those single-colored codes into one larger, multicolored code that encompasses the data of all the codes. This would effectively triple the amount of data a QR Codes could store while maintaining the error-correction bits of the code and its other necessary parts.[2]

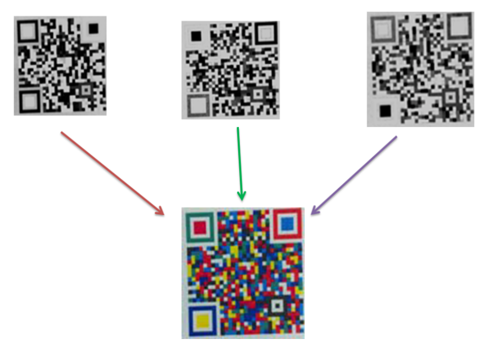

Figure 3. Proposed method to utilize colors to triple the data capacity of QR code. [2]

The bottom of Figure 3 shows the final proposed multicolored QR Code of this project. Scanning and separating the code into its separate color channels presents a significant problem. There are effectively eight different colors located in the QR Code, and the android camera has difficulty in distinguishing between them. [2] Figure 4 below shows the red, green, and blue versions of a color wheel separated by the android color channels on a Galaxy Note 2. The red channel separates fairly cleanly, but the green and blue channels have varying shades of gray. Because the camera cannot readily recognize whether these pixels are black or white, the pixels can easily be misread.

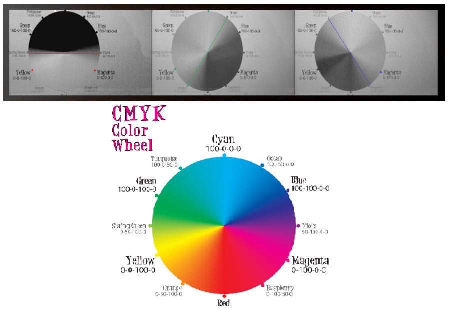

Figure 4. Ability of Android camera to separate red, green, and blue color channels.

Another problem arises if a code were to be scanned under a colored light. For example under a bright-yellow lamp, the code should still be readable. The proposed method for dealing with changing lighting conditions is to calibrate the code by reading colors in the code that are already known. The scanner reads the colors from the corners of the QR Code and using these values determines thresholds for red, green, and blue.

However, this leads to a new problem: if shadows or lights are passed over part of the QR code but not others, the code will be rendered unreadable. For example, a shadow across the red alignment corner would make the entire red part of the code unreadable. Black and white codes solve this problem by taking small chunks of the code and averaging the values in that chunk. Then, the code checks if pixels are black or white by determining whether or not they are darker than this local average. Thus, even if a chunk of the code was in a shadow, the black and white pixels would still be able to read. This algorithm does not work for colored codes because the average of multiple colors is meaningless to determining the color of an individual pixel. The inability to locally threshold pixels is a major weakness of the colored codes.[2]

Method 2: Elongated Codes

In addition to making the codes multicolored, the project also elongates them to fill any rectangular area rather than the square they currently occupy. This is useful for reading codes off of platforms that are constrained in their dimensions. Most phones and tablets have a rectangular screen, and a square QR code wastes much of their screen real estate. By creating rectangular QR codes that can change size to match the dimensions of any device, the codes can encode more data in a code and still remain readable.

The elongated codes require a large restructuring of the QR standard (shown in Figure 5. below). The principal challenge in elongating the codes is having the ability to properly transform the data into a grid. In a normal QR code, the three alignment patterns form the corners of a square, so when the camera finds them it can linearly transform them into a grid. With an elongated code, however, the scanning app has no way to discern what the height of the elongated code should be and how many pixels it contains. For this reason, the top of the elongated code is essentially a normal QR code, having three corners equally spaced apart. It uses these to do a linear transformation on the top half of the data and then reads a few important bits located adjacent to the corners. These bits encode the ratio of the code's height to its width. The scanning app then uses this ratio to accurately transform the data using the top two alignment corners and the bottom two alignment corners. The ratio has no rounding error because the ratio is rounded to the nearest fortieth and only codes with ratios divisible by forty are produced.

Another challenge when dealing with elongated codes is properly placing parts of the code. Timing patterns and all but one alignment patterns were eliminated to allow for growth of the codes and to encode more data. However, data bits and error correction bits were more difficult to change. The number of data bits and error correction bits is hard-coded by version in the QR standard, and so is their placement in the code. In order to be able to successfully incorporate as much data as could fit into the rectangular space, the elongated codes repeated the data and error-correction bit pattern of a version three QR code.

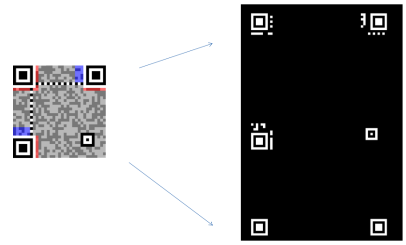

Figure 5. Dimensions of a particular elongated code.

Results

Result 1: Color

The colored codes had the potential to store triple the amount of data. However, because the phones were unable to clearly discern the differences between different colors in the codes, especially in smaller phones, the actual increase in data storage capacity was not 200%. For a phone the size of the Motorola Atrix (about the same size as an iPhone) or larger, the resulting increasing in data storage capacity was at least 20%, with the ability to store about 2.4 Kilobytes of data. For larger phones, the increase in data storage capacity grows. There is a very noticeable drop in performance speed.

Result 2: Elongated

This method increases data proportionally to the screen it is displayed on. For example, the Galaxy S has a 67% increase in data storage capacity, and the Galaxy Note 2 has a 77% increase. For a phone the size of the Motorola Atrix or larger, elongated codes can consistently store three Kilobytes of data. There is a small drop in performance speed.

Conclusion

A combination of both colored codes and elongated codes resulted in a code that was less effective than a normal code. Firstly, it had a very slow scanning speed, which partially defeats the purpose of a quick-response code. Secondly, the small imperfections in the transformation of the elongated code result in large discrepancies for the different colored pixels. If two differently colored pixels are blurred together, the resulting pixel is very difficult to discern.

The colored codes take much longer to decode because the scanning app essentially must decode three separate codes within the same image. It must also take the time to separately scan for the alignment corners that are different colors. The small increase in scanning time for elongated codes is because the scanning app must now find and identify five alignment patterns. Then, using linear regressions and cross products, it must identify their orientation in the code, a process which requires time.

While the colored codes could potentially store more data than the elongated codes, today's technology drastically limits the precision of color separation by phone cameras. As this technology progresses, colored codes will become a more feasible solution. In the present, however, elongated codes already function as an effective way to increase the amount of data storage possible for a given surface area.

Applications of this technology could have many potential uses. In addition to the simple phone-to-phone data transfer, these codes could be used to put data in places where it is normally inaccessible. For example, a large amount of medical information could be put onto a tag for people who have serious medical conditions that could interfere with their treatment. As technology progresses, QR codes will be able to store more and more data in smaller areas.

References

[1]"Information Technology --Automatic Identification and Data Capture Techniques." *ISO/IEC 18004:2006*. International Organization for Standardization, n.d. Web. 21 Oct. 2013.

[2] O. Bulan, H. Blasinski, and G. Sharma, "Color QR codes: Increased capacity via per-channel data encoding and interference cancellation," in Proc. IS&T/SID Nineteenth Color and Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications, San Jose, CA, 7-11 No*Multi-format 1D/2D Barcode Image Processing Library with Clients for Android, Java*. Google, n.d. Web. 21 Oct. 2013.

[3] "ZXing ("Zebra Crossing")." Zxing - Multi-format 1D/2D Barcode Image Processing Library with Clients for Android, Java. Google, n.d. Web. 21 Oct. 2013.

[4] http://www.qrstuff.com/blog/2011/12/14/qr-code-error-correction