Enhancing QR Code Error Correction using LDPC code:
A Comparative Study

Jiasheng Yun

May 2022

Abstract

Error correction is one of the most important parts of Automatic Identification and Data transmission. And Quick Response code (QR code) is one of the most popular types of two-dimensional bar-codes. It has the advantages of high density, fast recognition and mass storage. However, due to factors such as ambient noise and image damage, the reliability and robustness of QR Code becomes a key issue. The current mainstream error correction code of QR code is Reed-Solomon code, which is a kind of forward error correction channel code. In this paper, an error correcting code scheme based on LDPC code is proposed and compared with the coding algorithm of conventional error detection.

Keywords: QR code Information theory, LDPC code, Reed-Solomon code, Channel coding

1 Introduction

Bar-codes are widely used because of its advantages of automatic identification. But with the use of bar-codes spreading, their limitations became apparent as well. The most prominent was the fact that a bar-code can only hold 20 alphanumeric characters or so of information. The two-dimensional code came into being in this case. Compared with one-dimensional code, two-dimensional code has high data capacity, no additional storage and great error correction ability. Quick Response code (QR code) is one of the most

popular types of two-dimensional bar-code developed in Japan by DENSO WAVE INCORPORATED. Although QR code was first designed for the automotive industry, they have become ubiquitous in various domains, including advertising, packaging, transportation, and digital content.

However, even with the advantages of two-dimensional codes, challenges related to error correction in QR codes persist. Factors like environmental noise, print quality, and image distortion during scanning can lead to errors in decoding QR codes. Therefore, the development of robust error correction techniques is essential to ensure accurate and reliable data retrieval from QR codes.

The current mainstream error correction code for QR code is Reed-Solomon code[3]. However, In this paper, we use the LDPC code as the error-correcting code. LDPC code is a powerful and widely used error correcting code, known for its ability to detect and correct many kinds of errors. By integrating LDPC encoding into the QR code framework, we aim to improve its error correction capabilities, make it more resistant to data corruption, and improve its reliability under harsh conditions.

The primary object of this paper is to explore the application of LDPC code as a robust error correction mechanism for QR code and compare it with existing error correction code.

The structure of this paper is as follows: Section 2 provides an overview of QR code and LDPC code fundamentals. Section 3 details the proposed LDPC coding-based error correction scheme for QR codes. Section 4 shows the error correction effect of our code and the comparison of conventional algorithm. Section 5 outlines directions for future research. . Finally, Section 6 concludes the paper.

By using LDPC code as vulnerability detection code, this study helps to provide a new idea in the field of QR code error correction.

2 overview of QR code and LDPC code fundamentals

2.1 overview of QR code

The next few sections will briefly cover the generation of QR code. They are Data analysis, Data encoding, Error correction coding, Structure final message, Module placement in matrix, Data masking, Format and version information.

2.2 overview of LDPC code

Low-Density Parity Check (LDPC) code represents a class of error correcting codes that may be employed for providing error correction of transmission errors in communication systems. Using LDPC code, channel capacities that are close to the theoretical Shannon Limit can be achieved[4]. Moreover, LDPC code as an error detection code also has certain advantages compared with other coding methods.

LDPC codes are usually expressed in the form of (n,j,k). In Gallager's definition of LDPC code in 1962, the parity matrix of LDPC code expressed as (n,j,k) has the following structural characteristics:

1) There are k ones in each row (k is a fixed value, k is also called the weight of the row, and other elements are 0)

2) Each column has j ones (j is a fixed value,j is also known as the weight of the row, and other elements are 0)

3) The number of positions in any two rows (columns) with a common "1" shall not exceed 1;

4)k and j are small compared with the number of columns and rows in H, and H is the check matrix.

An LDPC code can be uniquely defined by its parity matrix H. Each row of the parity matrix corresponds to a parity equation, and each column corresponds to a bit of the code word. For example: Consider the source code

$$s = (c_1, c_2, c_3)$$

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} = \begin{cases} c_1 + c_2 + c_4 = 0 \\ c_2 + c_3 + c_5 = 0 \\ c_1 + c_2 + c_3 + c_6 = 0 \end{cases}$$

The corresponding generating matrix is:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

The LDPC code is:

$$C(s) = (c_1, c_2, c_3, c_4, c_5, c_6) = (c_1, c_2, c_3) \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

It can be found by observation

$$HC^T(s) = 0$$

The reasons for choosing LDPC code are as follows:

In terms of anti-interference: LDPC code for communication channel interference and noise has better anti-interference performance compared with Reed-Solomon code. This means that under poor channel conditions, LDPC codes can provide more reliable error correction and decoding performance, and enhance the reliability and robustness of two-dimensional code.

In terms of error correction ability, LDPC code has higher error correction ability compared with Hamming code. LDPC code can detect and correct multiple errors simultaneously, whereas the Hamming code can only correct single-bit errors.

3 QR code correction scheme based on LDPC code

According to the current generation algorithm of QR code error correction[2], we design the error correction code based on LDPC code.

1) In the data encoding step of QR code generation algorithm, the information to be transmitted is converted into a series of 8 bits of binary code. Therefore, the information bit length of the final compiled LDPC code in our algorithm is 8. 2) In the error correction coding step of QR code generation algorithm, a series of correction code for the series of 8 bits of binary code will be generated and each each error correction code is 8 bits long. In order to simplify the generation of QR code, we keep the number of bits of the error correcting code unchanged. Therefore, The length of the parity bit of LDPC code is 8.

The following will explain the idea of constructing LDPC codes:

The information bit and check bit of the LDPC code are both 8. We can construct 8 linear independent equations which are calibration equations.

According to the definition of H matrix introduced before, we can construct an 8x16 parity matrix H by taking these 8 check equations as rows. Then, through H matrix, it is easy

to get the generated matrix G.(The reasons will be explained later)In this way, the transmission code block can be obtained through the information bit and the generation matrix. Therefore, the process of constructing linear block code is the process of constructing check matrix H or generating matrix G.

For the generation matrix G and the parity matrix H, the relationship is as follows:

Assuming that the sparse matrix H has the following form H=[C1 | C2], where C2 is a sparse square matrix of

$$m * m$$

and is a non-singular matrix in the binary field, then G can be given:

$$G^T = \begin{bmatrix} I_k \\ C_2^{-1} C_1 \end{bmatrix}$$

. It is easy to verify that such G satisfies

$$HG^T = 0$$

.

3.1 Encode algorithm

The encoding algorithm of LDPC code is very similar to general linear block code, which requires the generation matrix. Using the knowledge in Section 2.2, we can see that C=S*G which is the encoded code. After obtaining the checksum matrix H, using the orthogonality between G and H, the generated matrix H can be obtained by Gaussian elimination method.

3.1.1 construction of parity matrix

According to IEEE802.16e, there are many ways to construct parity matrix. Here we use the Gallager construction method to construct an 8 by 16 parity matrix H.[1] The parity matrix

H is:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

I wrote a program to implement the above algorithm

```
import numpy as np
def GallagerH(CodeLength, dv, dc):
    SubRow = CodeLength // dc
    SubCol = dc
    SubH = np.zeros((SubRow, SubCol))
    SubH[0, :] = 1
    SubOneH = SubH.copy()
    for i in range(1, SubRow):
        MidH = np.roll(SubH, i, axis=0)
        SubOneH = np.hstack((SubOneH, MidH))
    GallagerH = SubOneH
    SubReH = np.zeros((SubRow, CodeLength))
    for j in range(1, dv):
        randomV = np.random.permutation(CodeLength)
        for k in range(CodeLength):
            SubReH[:, k] = SubOneH[:, randomV[k]]
        GallagerH = np.vstack((GallagerH, SubReH))

    return GallagerH
CodeLength = 16
dv = 4
dc = 8
GaH = GallagerH(CodeLength, dv, dc)
# print(len(GaH))
print(GaH)
```

3.1.2 construction of generation matrix

Based on the relationship between H and G discussed in Section 2.1, We Use Gaussian elimination to obtain the generating matrix G and design an algorithm to construct generation

matrix.

$$H = \begin{bmatrix} I_{(n-k)*(n-k)} & P_{(n-k)*k} \\ 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} P^T_{k*(n-k)} & I_{k*k} \end{bmatrix}$$

3.2 Error correction by parity matrix

If the result of the transfer is correct,

$$s * H = 0$$

is displayed. In actual channels, information may generate errors due to various reasons. This will result in

$$s * H \neq 0$$

. LDPC code can detect vulnerabilities and make modifications. The decoding method we use here is the bit flipping algorithm.

3.2.1 Algorithm process

1. Verify the received c, i.e. calculate the verification vector

   $$s = H * c^T$$

   . If s calculates a zero vector, that is, the received information meets all the check equations, the decoding is successful, and the receiver has received the correct information; Otherwise, proceed to step 2.

2. Calculate the number of validation equations for each element of c that are involved in incorrect results.

3. If the corresponding fn of an element cn in c exceeds a certain design value which is set by the designer, flip the corresponding element

   $$f_n = \sum_{m=1}^{8} (s_m h_{mn}), n = 1, 2, 3, ...16$$

4. Repeat steps 1, 2, and 3 until decoding is successful.

4 Compare with Reed-Solomon algorithm

Reed-Solomon code:RS(D,K),form_0,.....m_{k-1},we can form: f(x)=

$$\sum_{i=0}^{k-1}$$

$m_i x^i$,for eacha_0,... $a_{n-1} \in$ **D**,c_i=F(a_i).define RS[F,D,k] is all the possible RS(D,k) code $word^{'}$s collection. The decoding algorithm knows D and receives a polynomial with a degree less than k that may contain errors such asb_0,...,b_{n_1}. We hope to decode the correct polynomial with a degree less than k.

Here is an example15

Figure 1: The QR code forming by The right code

its data code is 00100000 01011011 00001111011000110000 11010001 01110010110010 11011100 01001101 01000011 01000011 01000000 11101100 00010001 11101100

And its error correction polynomial is

$196x^9$+$35x^8$+$39x^7$+$119x^6$+$235x^5$+$215x^4$+$231x^3$+$226x^2$+$93x$+23

Figure 2: The QR code forming by The wrong code

its data code is 00100000 01011011 00001111011000110 00 11010001 01110010110010 11011100 01001101 01001001 01000000 11101100 00010001 11101100 00010001

And its error correction polynomial is

$196x^9+35x^8+39x^7+119x^6+235x^5+215x^4+231x^3+226x^2+86x+23$

Then the next step is to put them in the QR code matrix along with the required function patterns. A function pattern is a non-data element of the QR code that is required by the QR code specification, such as the three finder patterns in the corners of the QR code matrix.

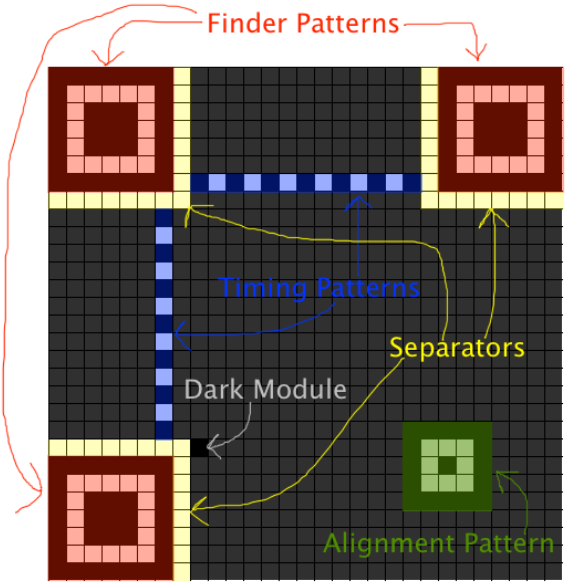

Figure 3: How to form a QR code

Figure 4: The QR code by the error correction code

We can see that though with the noise, the encode algorithm still can recover the QR code in different error correction level,which can prove our algorithm is effective.

5 Future Work

5.1 Optimize LDPC code design

Further research and optimization of LDPC code design to achieve better performance in specific QR code application scenarios. We can consider using different LDPC code construction methods, adjusting parameters such as code rate and code word length to find better error correction and decoding performance.

5.2 Code length and performance analysis

Explore the relationship between the performance of LDPC codes and code length. Through systematic experiments and analysis, study the error correction ability and decoding performance of LDPC codes under different codeword lengths, and evaluate their matching with QR codes. In addition, it is possible to explore how to balance the relationship between codeword length and decoding complexity to achieve a better balance between performance and practicality.

5.3 Algorithm optimization and acceleration

Improve the decoding algorithm of LDPC codes to improve decoding speed and performance. It can research and develop efficient LDPC code decoding algorithms, including research based on hardware acceleration, parallel computing and optimization algorithms. This can reduce the complexity of LDPC code decoding and improve its real-time performance and feasibility in practical applications.

6 Conclusion

In our algorithm design, in order to enhance the error correction ability of QR codes, we propose using LDPC code as the error correction code. Firstly, we introduced the principle of the LDPC algorithm and provided the advantages of using this algorithm. Next, we analyzed the encoding process of QR code and proposed an LDPC encoding method that can be applied to QR code.Afterwards, we compared it with the traditional Solomon algorithm and looked forward to the future applications of LDPC coding.

In the QR code problem, we can see that the encoding algorithm is worse comparing to the Reed-Solomon error correction code,but its error correction can be used in different error correction level.

[1] . "IEEE802.16eLDPC." , : , : , 201101

[2] Thonky.com. "QR Code Tutorial." https://www.thonky.com/qr-code-tutorial/

[3] "Reed–Solomon code." https://www.geeksforgeeks.org/what-is-reed-solomon-code/

[4] Paul H. Siegel. “An introduction to Low-Density Parity-Check Codes”