

The inclusion of watermarks based on QR Code analysis

Jumana Waleed^{1*}, Sarah Saadoon Jasim², Thekra Abbas³

¹ Department of Computer Science, College of Science, University of Diyala, Iraq

² Middle Technical University, Computer Science/Technical college of Management – Baghdad/Dept. of IT, Iraq

³ Department of Computer Science, College of Science, Almustanseriya University, Iraq

*Corresponding author E-mail: jumanawaleed@sciences.uodiyala.edu.iq

Abstract

With the quick arrival of high grades of acceptance, the QR Code technology is becoming more adaptable and usable every day by more and more people. Lately, the utilization of two dimensional QR codes for the powerful encoding of information with a big capacity becomes significantly growing; especially for encoding the identity. Because the identity is private information, it requires being further authenticated. This paper firstly works on analyzing the content of the QR Code and finding the unimportant locations inside it; Then, a combination of the digital watermarking concept and QR Code technology is used to substitute the founded unimportant locations by the watermark for the purpose of increasing the authenticity for the owner of the identity. The experimental results show that the inclusion of watermark bits inside the detected unimportant locations did not affect the process of decoding and retrieving information from the QR Code.

Keywords: QR Code Analysis; Unimportant Locations; Digital Watermarking; QR Code Ridge; Hidden Capacity.

1. Introduction

QR Codes have become vastly used via the rapid growing of mobile Internet and the expansion of QR Codes applications. There are many advantages that characterize the QR Code; Firstly, it is readable very faster even under low contrast conditions and this feature allows the direct marking, also it can be scanned at any angle; Secondly, it has a high data capacity of storage since it takes minimum space; Thirdly, it can handle different kinds of data; Fourthly, the QR Codes enable powerful encoding even if up to 30 % of the data is damaged; Finally, the availability of QR Codes for everyone led to wide range of utilizations and applications [1].

Digital watermarking takes an important area due to the challenges accomplished in the efficient inclusion of the watermark bits on a limited size of image intensity values [2], [3], [4]. The digital image watermarking process starts with hiding a secret or private information via the owner into the host image, that is extracted at the recipient side to identify or verify the ownership [5]. The QR Code can be combined with the digital watermarking process for storing private information, and this combination is going to generate another type of QR Codes called the watermarked QR Code. Essentially, a watermarked QR Code includes embedded data which is hidden by using an embedding technique. After that, this QR Code is scanned and decoded by QR Codes decoder, and the embedded data is restored by using an extraction technique.

The organization of this paper is as follows: This section presents a brief introduction, and the next section summarizes the recently existing related works. In the third section, the core structure of QR Code is explained; then section four presents the proposed combination of the QR code analysis and digital watermarking process; the experimental results are given in section five; finally, conclusions are shown in the last section.

2. Related work

In the recent years, the utilization of QR Codes become extremely common in digital watermarking. There are a lot of researchers that have been used the QR Codes as a watermark images representation and these watermark images can provide a wide range of applications and purposes. Weijun et al. [6] present a robust algorithm against synchronization attacks in which the QR code is utilized as a watermark for the purpose of copyright protection. This robustness is achieved by using the positioning and correction of the Code. Jumana et al. [7] propose a highly robust type of watermarking technique in which the QR Code is used as a watermark for the purpose of improving the security of commercial activity on media and Internet. For the purpose of owner verification, a blind digital watermarking approach based on QR Code is proposed by Pillai et al. [5]. In this approach, the QR Code has been used as a watermark to ensure the extended usability. Kavitha et al. [8] propose a system for protecting the medical information against different illegal activities.

Also, many researchers used the QR Codes with the process of digital watermarking as a carrier image. And those researchers are working on embedding data with sizes should not be greater than the size of the QR Code. Ari [9] proposes a watermarking technique for the purpose of further authenticity to include a biometric characteristic inside the data and error correcting codeword of the QR Code. For the purpose of solving the problem of counterfeiting, H. Phuong et al. [10] present an inexpensive and efficient watermarking approach for securing printed QR Code. In this approach, a particular random texture is included with the background of the standard QR code for creating a secure one. With the application of Online Banking, an authenticated watermarking technique is proposed by Aayushi et al [11] in which a visual wa-

termark is included in the QR Code for ensuring a good security policy and suitability.

In this paper, the proposed approach utilizes the QR code as a carrier image for including a private information in the application of providing the authenticity of owner identity.

3. The core structure of QR Code

The symbol of QR Code ought to be built in a square module regulated in a square array. This Symbol should contain function patterns and encoding region and its four sides should be surrounded by a border of the quiet zone. Figure 1 demonstrates the core structure of the QR Code.

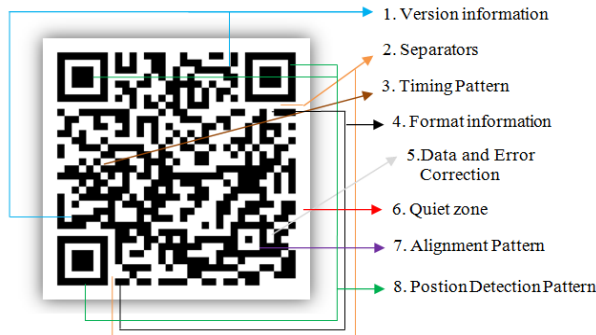


Fig. 1: The Core Structure of QR Code.

The function patterns are the forms which should be located in certain zones of QR Code for ensuring the correct scanning to decode the QR Code. These patterns contain four kinds; Firstly, position detection patterns are the unlikely finder patterns located in upper right, upper left, and lower left corners; Secondly, the separators between the finder patterns and the encoding region; Thirdly, two-timing patterns, horizontal which is located in the sixth row of QR Code, and vertical which is placed in the sixth column of QR code; and Fourthly, alignment patterns which is depending on the version of the symbol. The encoding region consists of the data, that symbolizes version information, format information, data and error correction codes. The quiet area is a four modules wide zone concluding no data and is utilized for ensuring that the surrounding markings ought not to mislead the data of QR code [12].

4. The proposed approach of combination

The proposed combination of QR code analysis and digital watermarking can be characterized into two phases; The QR Code analysis and watermark bits' inclusion and extracting.

4.1. The QR Code analysis

This phase consists of several steps for detecting the unimportant locations which are utilized to include the watermark without affecting the QR code, and these steps are as follows:

Step 1: load or create the QR code image;

Step 2: Perform the histogram equalization and binarization processes to remove the noise existing in the loaded QR Code image; as illustrated in the following pseudocode;

Pseudocode 1: QR Code Histogram equalization and Binarization

```

Input: QR code loading with noise
Output: QR code loading after noise removing
Histogram_Equalization(im As Image) As Image
Dim GrHst(256), GrSum(256) As Long
Dim R1, G1, B1 As Integer
Dim HstValue As Long = 0
Dim SumValue As Long = 0
Dim bmp As Bitmap = DirectCast(im, Bitmap)
Dim b As Bitmap = DirectCast(im, Bitmap)

```

```

Dim b2 As New Bitmap(im)
For row As Integer = 0 To bmp.Height - 1
For col As Integer = 0 To bmp.Width - 1
HstValue = CDbl(255 * bmp.GetPixel(col, row).GetBrightness())
GrHst(HstValue) = GrHst(HstValue) + 1
Next
Next
For level As Integer = 0 To 255
SumValue = SumValue + GrHst(level)
GrSum(level) = SumValue
Next
ToolStripProgressBar1.Maximum = bmp.Height
For row As Integer = 0 To bmp.Height - 1
For col As Integer = 0 To bmp.Width - 1
Dim clr As System.Drawing.Color = bmp.GetPixel(col, row)
HstValue = CInt(255 * clr.GetBrightness())
HstValue = CInt(255.0F / (bmp.Width * bmp.Height) * GrSum(HstValue) - HstValue)
R1 = CInt(Math.Min(255, clr.R + HstValue / 3))
G1 = CInt(Math.Min(255, clr.G + HstValue / 3))
B1 = CInt(Math.Min(255, clr.B + HstValue / 3))
b2.SetPixel(col, row, Color.FromArgb(Math.Max(R1, 0), Math.Max(G1, 0), Math.Max(B1, 0)))
Next
ToolStripProgressBar1.Value = row
Next
Return DirectCast(b2, Image)
End Function
Public Function Binarization(im As Image) As Image
On Error Resume Next
Dim bmp As Bitmap = DirectCast(im, Bitmap)
Dim b2 As New Bitmap(im)
Dim Temp As Long
ToolStripProgressBar1.Maximum = bmp.Width
xix = bmp.Width
yiy = bmp.Height
ReDim data(xix, yiy)
For i As Integer = 0 To bmp.Width - 1
For j As Integer = 0 To bmp.Height - 1
Temp = (CInt(bmp.GetPixel(i, j).R) + CInt(bmp.GetPixel(i, j).G) + CInt(bmp.GetPixel(i, j).B))
Temp = (Temp / 3)
If Val(Temp) >= 130 Then
b2.SetPixel(i, j, Color.White)
data(i, j) = 1
Else
b2.SetPixel(i, j, Color.Black)
data(i, j) = 0
End If
Next
Next
ToolStripProgressBar1.Value = i
Next
Return DirectCast(b2, Image)
End Function

```

Step 3: Remove the white zones surrounding the QR Code;

Step 4: Find the QR Code Ridge as represented in Equation (1):

$$\text{QR Code Ridge} = \text{Width} \times \text{Height} \quad (1)$$

Step 5: Depending on the computed QR Code ridge, draw the pixels for determining the number of blocks;

Step 6: Deconstruct the QR Code by employing a mask of 3×3 to gather all the possible probabilities in QR code;

Step 7: Find the number of unimportant locations for the process of watermark inclusion by using the following Equation:

$$\text{No. of Hidden Blocks} = \text{QR Code Ridge} \times \text{No. of Blocks} \quad (2)$$

Step 8: Remove these unimportant locations to proper the QR Code for the next Phase, the processes of deconstruction and construction are illustrated in the Pseudocode 2.

Pseudocode 2: QR Code deconstruction and construction processes

```

Input: QR Code before eliminating the irrelevant positions
Output: Reconstructed QR Code
Private Sub DeConstructingToolStripMenuItem_Click(sender As Object, e

```

```

As EventArgs Handles DeConstructingToolStripMenuItem.Click
Dim i, j, cbit
Dim Offset As Integer = 1
Dim xcount As Integer = 0
Dim RedSum As String
ReDim m_DataValues(28)
For i = LBound(m_DataValues) To UBound(m_DataValues)
    m_DataValues(i) = 0
Next i
Dim xx As Integer = 1
Dim chwck As Boolean = False
ToolStripProgressBar1.Maximum = CWxixF - Offset - 1
For ii = Offset To CWxixF - Offset - 1 Step 2
    For jj = Offset To CWyiyF - Offset - 1 Step 2
        RedSum = ""
        For fi = -Offset To Offset
            For fj = -Offset To Offset
                RedSum = RedSum + Trim(Str(TCWdata(ii + fi, jj + fj)))
            Next
        Next
        chwck = False
        Select Case RedSum
            Case "011011110"
                m_DataValues(1) = m_DataValues(1) + 1
                RedSum = "011111110"
            Case "110111011"
                m_DataValues(2) = m_DataValues(2) + 1
                RedSum = "110111011"
            Case "100010100"
                m_DataValues(3) = m_DataValues(3) + 1
                RedSum = "100110100"
            Case "001010001"
                m_DataValues(4) = m_DataValues(4) + 1
                RedSum = "001011001"
            Case "101010000"
                m_DataValues(5) = m_DataValues(5) + 1
                RedSum = "111010000"
            Case "000010101"
                m_DataValues(6) = m_DataValues(6) + 1
                RedSum = "000010111"
            Case "010101010"
                m_DataValues(11) = m_DataValues(11) + 1
                RedSum = "010111010"
            Case "111101111"
                m_DataValues(12) = m_DataValues(12) + 1
                RedSum = "111111111"
            Case "011001011"
                m_DataValues(13) = m_DataValues(13) + 1
                RedSum = "011011011"
            Case "110100110"
                m_DataValues(14) = m_DataValues(14) + 1
                RedSum = "110110110"
            Case "000101111"
                m_DataValues(15) = m_DataValues(15) + 1
                RedSum = "000111111"
            Case "111101000"
                m_DataValues(16) = m_DataValues(16) + 1
                RedSum = "111111000"
            Case "101000101"
                m_DataValues(17) = m_DataValues(17) + 1
                RedSum = "101010101"
            Case "010101010"
                m_DataValues(18) = m_DataValues(18) + 1
                RedSum = "010111010"
            Case "011001000"
                m_DataValues(19) = m_DataValues(19) + 1
                RedSum = "011011000"
            Case "000100110"
                m_DataValues(20) = m_DataValues(20) + 1
                RedSum = "000110110"
            Case "001001111"
                m_DataValues(21) = m_DataValues(21) + 1
                RedSum = "001011111"
            Case "100100111"
                m_DataValues(22) = m_DataValues(22) + 1
                RedSum = "100110111"
            Case "111001001"
                m_DataValues(23) = m_DataValues(23) + 1
                RedSum = "111011001"
            Case "111100100"
                m_DataValues(24) = m_DataValues(24) + 1
                RedSum = "111110100"
            Case "001001001"
                m_DataValues(25) = m_DataValues(25) + 1
                RedSum = "001011001"
            Case "100100100"
                m_DataValues(26) = m_DataValues(26) + 1
                RedSum = "1001110100"
            Case "000000111"
                m_DataValues(27) = m_DataValues(27) + 1
                RedSum = "000010111"
            Case "111000000"
                m_DataValues(28) = m_DataValues(28) + 1
                RedSum = "111010000"
            Case Else
                chwck = True
        End Select
        cbit = 1
        If Not (chwck) Then
            For fi = -Offset To Offset
                For fj = -Offset To Offset
                    TCWdata(ii + fi, jj + fj) = CInt(Mid(RedSum, cbit, 1))
                Next
            Next
            RedSum = ""
        End If
        Next
        ToolStripProgressBar1.Value = ii * 100 / (Y - 1)
    Next
    Dim sum As Integer = 0
    For i = 1 To 28
        sum += m_DataValues(i)
    Next
    DataGridView1.Rows(0).Cells(i - 1).Value = m_DataValues(i)
    Next
    TextBox1.Text = sum
End Sub

Private Sub ConstructingToolStripMenuItem_Click(sender As Object, e
As EventArgs) Handles ConstructingToolStripMenuItem.Click
    Dim Newb As Bitmap = New Bitmap(Cxix + 1, Cyiy + 1)
    Dim dx As Integer = CInt(TextBox4.Text)
    Dim dy As Integer = CInt(TextBox5.Text)
    Dim Offset As Integer = CInt(TextBox5.Text) / 2
    Dim ii As Integer = 0
    Dim jj As Integer = 0
    For i = Offset To Cxix Step dx
        jj = 0
        For j = Offset To Cyiy Step dy
            For fi = -Offset To Offset
                For fj = -Offset To Offset
                    Cdata(i + fi, j + fj) = TCWdata(ii, jj)
                    If Cdata(i + fi, j + fj) = 1 Then
                        Newb.SetPixel(i + fi, j + fj, Color.White)
                    Else
                        Newb.SetPixel(i + fi, j + fj, Color.Black)
                    End If
                Next
            Next
            jj = jj + 1
        Next
        ii = ii + 1
    Next
    PictureBox6.Image = Newb
    PictureBox6.Refresh()
    GroupBox4.Text = ii.ToString() + "," + jj.ToString()
    Dim g As Graphics = Graphics.FromImage(Newb)
    Dim blackPen As New Pen(Color.Red, 3)
    For i = 0 To Cxix - 1 Step dx
        jj = 0
        For j = 0 To Cyiy - 1 Step dy
            Dim rect As New Rectangle(i, j, dx, dy)
            g.DrawRectangle(blackPen, rect)
            jj = jj + 1
        Next
        ii = ii + 1
    Next
    PictureBox5.Image = Newb
End Sub

```

4.2. Watermark bits' inclusion and extracting

The process of watermark bits' inclusion is depicted as; firstly, reformulate the watermark into a vector of bits; secondly, use the unimportant locations that are founded in the phase of QR Code analysis for including the watermark bits; and finally, reconstruct the watermarked QR Code.

Whereas the process of extracting the watermark bits is done by utilizing the unimportant locations that are detected within the process of QR Code analysis as an extraction key. After that, the watermark bits are extracted from the watermarked QR Code to construct the extracted watermark. Figure 2 shows the overall structure of the approach of the combination.

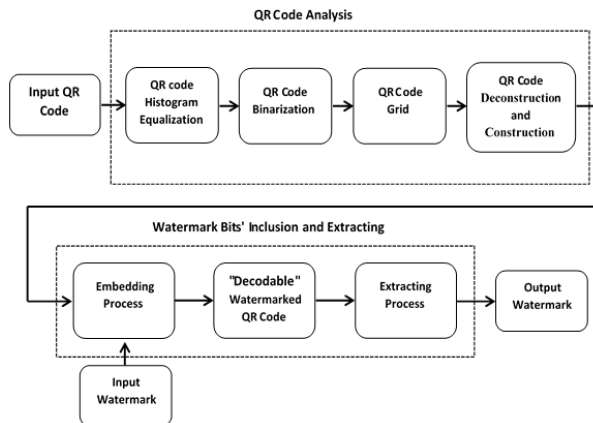


Fig. 2: The Overall Structure of the Proposed Approach.

5. The experimental results

For evaluating the proposed combination of QR Code analysis and digital image watermarking, the loaded QR Codes which decode the title and the keywords of this paper are used. The size of the used QR Codes, the number of founded blocks, and the hidden capacity are demonstrated in table 1, and table 2 and 3 show the details of the proposed approach for the given example.

Table 1: Finding the Hidden Capacity of the Watermark Bits

QR Code Size	No. of Blocks	Hidden Capacity
370*370	6	10*10*6=600 bits
400*400	8	9*9*8=648 bits
410*410	12	10*10*12=1200 bits

Table 2: An Example of Covers and the Watermarks of Suitable Capacity







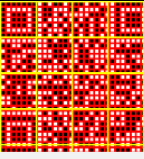

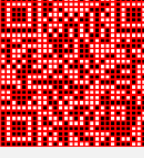
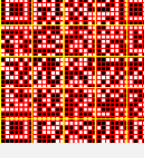
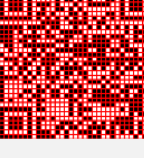
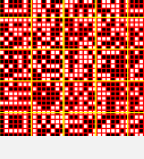
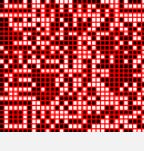
The QR Codes	Watermarks
370*370 	24*24 
400*400 	25*25 
410*410 	32*32 

Table 3: The Results of QR Code Analysis and Watermarking Processes for the Used Example

QR Code Analysis	Decodable Watermarked QR Code
	
	
	
	
	
	

6. Conclusion

In this paper, it is clear that the carrier QR Code images have remained decodable even when the watermark bits are included inside it, and this achievement is accomplished by analyzing the content of QR Code and detecting the suitable locations for the process of embedding. The obtained results as depicted in the earlier presented tables have been shown clearly that the QR code won't be affected.

References

- [1] Muming Li, Peng Cao, Liuping Feng, Lifang Yu, Jianbo Chen and Jing Wang, (2017), "The research of QR code image correction based on image gray feature", 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), Harbin, China, pp. 1-5. <https://doi.org/10.1109/EIIS.2017.8298583>.
- [2] Jumana Waleed, Huang Dong Jun, and Saad Hameed, (2014), "A robust Optimal Zero-Watermarking Technique for Secret Watermark Sharing," International Journal of Security and Its Applications Vol.8, No.5, pp.349-360. <https://doi.org/10.14257/ijisa.2014.8.5.31>.
- [3] Jumana Waleed Huang Dong Jun, and Saad Hameed., (2015), "An optimized digital image watermarking technique based on cuckoo search (CS)", ICIC Express Letters, Part B: Applications, Vol. 6, No. 10, pp. 2629-2634.
- [4] Abhinaya. M, A. Umamakeswari, (2018), "A Robust Image Watermarking Method based on Contourlet Transform for Correcting

- Geometric Attacks", International Journal of Engineering and Technology(UAE), Vol. 7, No. 2.24, pp. 24-28.
- [5] Pillai Praveen Thulasidharan, Madhu S. Nair, (2015), "QR code based blind digital image watermarking with attack detection code", AEU - International Journal of Electronics and Communications, Vol. 69, No. 7, PP. 1074-1084. <https://doi.org/10.1016/j.aeue.2015.03.007>.
- [6] Weijun Zhang and Xuettian Meng, (2015), "An improved digital watermarking technology based on QR code," 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, pp. 1004-1007.
- [7] Jumana Waleed, Huang Dong Jun, Sarah Saadoon, Saad Hameed, Hiyam Hatem, (2015), "An Immune Secret QR-Code Sharing based on a Twofold Zero Watermarking Scheme", International Journal of Multimedia and Ubiquitous Engineering Vol.10, No.4, pp.399-412. <https://doi.org/10.14257/ijmue.2015.10.4.38>.
- [8] K. J. Kavitha and B. P. Shan, (2017), "Implementation of DWM for medical images using IWT and QR code as a watermark," 2017 Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, pp. 252-255. <https://doi.org/10.1109/ICEDSS.2017.8073698>.
- [9] A. M. Barmawi and F. A. Yulianto, (2015), "Watermarking QR Code," 2015 2nd International Conference on Information Science and Security (ICISS), Seoul, pp. 1-4. <https://doi.org/10.1109/ICISSEC.2015.7371041>.
- [10] H. P. Nguyen, A. Delahaies, F. Retraint, D. H. Nguyen, M. Pic and F. Morain-Nicolier, (2017), "A watermarking technique to secure printed QR codes using a statistical test," 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, pp. 288-292. <https://doi.org/10.1109/GlobalSIP.2017.8308650>.
- [11] A. Mishra and M. Mathuria, (2017), "Multilevel security feature for online transaction using QR code & digital watermarking," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, pp. 48-51. <https://doi.org/10.1109/ICECA.2017.8212709>.
- [12] S. Tiwari, (2016), "An Introduction to QR Code Technology," 2016 International Conference on Information Technology (ICIT), Bhubaneswar, pp. 39-44. <https://doi.org/10.1109/ICIT.2016.021>.