Thema:

QR-Codes

Autor
Martin Lehmann (3390563)

Lehrbeauftragte
Dr.-Ing. Dagmar Schönfeld

Inhaltsverzeichnis

1 Einleitung

Im Rahmen der Lehrveranstaltung “Kanalkodierung” wird in dieser Arbeit der **Q**uick **R**esponse Code (QR-Kode) vorgestellt. Bei dem QR-Kode handelt es sich um einen zweidimensionalen Kode, der im Jahr 1994 von der japanischen Firma “Denso Wave” zur Markierung von Baugruppen und Komponenten in der Automobilproduktion des Toyota-Konzerns entwickelt wurde. Der QR-Kode ist eine Weiterentwicklung des eindimensionalen Barcodes, mit dem Informationen lediglich in horizontaler Ebene gespeichert und ausgelesen werden können. Im Gegensatz zu den Barcodes, erfolgt die Kodierung der Information in horizontaler und in vertikaler Richtung, wodurch ein Vielfaches an Daten gespeichert werden kann [Alb11].

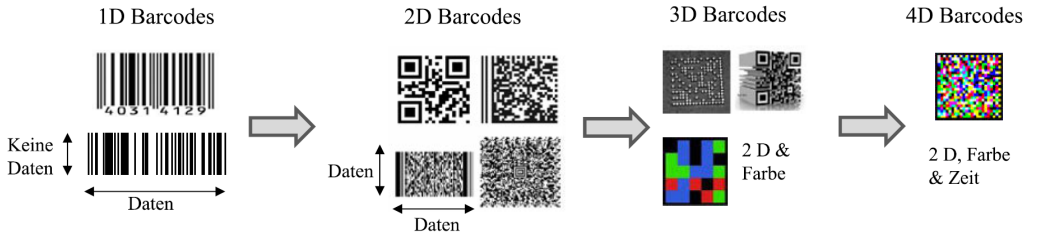

Abbildung 1.1: Entwicklung der Barcodes [uMH12]

Durch die Verbreitung von Smartphones sowie die Entwicklung von leistungsfähiger Sensortechnik in diesen Endgeräten, durchdringen Barcodesysteme nahezu alle Bereiche unseres täglichen Lebens (Abbildung 1.1). Elektronische Bordkarten [IAT12], Eintrittskarten für Kino oder Theater sowie die Verwendung im Mobile Marketing [Gmb12] sind einige der Anwendungsbereiche von QR-Kodes.

QR-Kodes weisen folgende Merkmale auf [Wav10a]

- **High Capacity Encoding of Data** - Kodierung von unterschiedlichen Datentypen bis zu 7089 Zeichen in einem Symbol
- **Small Printout Size** - Horizontale und vertikale Kodierung ermöglicht die Speicherung von Informationen auf engstem Raum
- **Kanji and Kana Capability** - Möglichkeit der Kodierung von logografischen Schriftzeichen
- **Dirt and Damage Resistant** - Korrekturfähigkeit von unvollständig ausgelesenen QR-Kodes
- **Readable from any direction in** 360° - Richtungunabhängiges Lesen (omnidirektional) der QR-Kodes durch Orientierungspunkte
- **Structured Append Feature** - Einteilung eines QR-Kodes in mehrere kleine QR-Kode-Symbole

2 Aufbau

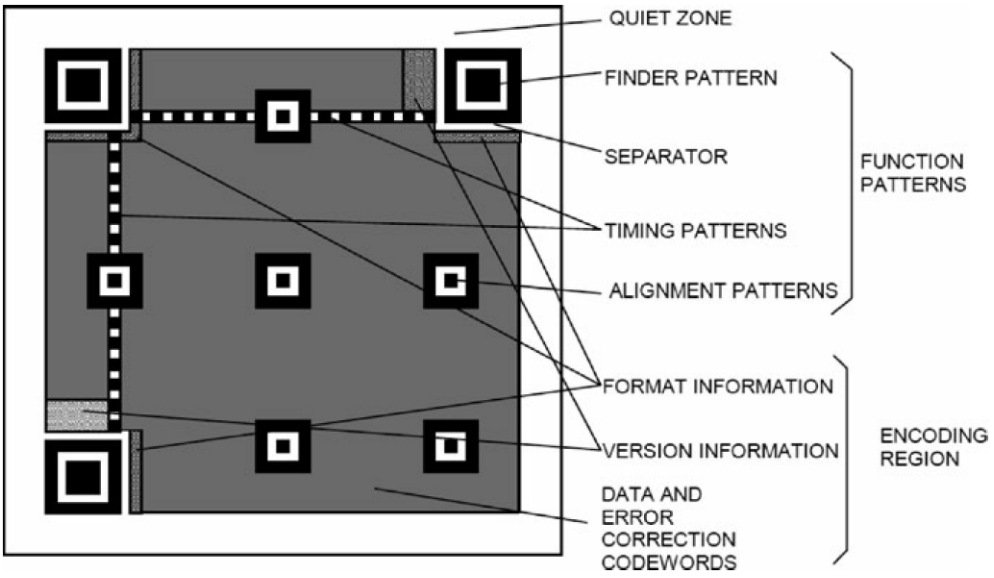

Abbildung 2.1: Struktur eines QR-Kodes [uMH12]

Der QR-Kode gehört zu der Klasse der Matrixkodes, welche "die Dateninhalte über die Position der dunklen Elemente in einer Matrix gleichmäßig verteilt" [uMH12]. Mit anderen Worten besteht die graphische Oberfläche eines QR-Kodes aus einer Matrix mit schwarzen und weißen Punkten, welche die kodierten Daten in Form der binären Darstellung (schwarz oder weiß) enthalten. Die Daten werden dabei redundant gespeichert, um sie mit Hilfe des fehlerkorrigierenden Reed-Solomon-Kodes (RS-Kode) bei einer Fehlerrate von bis zu 30% wiederherstellen zu können. Neben den reinen Daten und den Formatinformationen über den Kode ("ENCODING REGION"), wie die Version und das benutzte Datenformat, werden auch Orientierungspunkte ("FUNCTION PATTERNS") in dem QR-Kode gespeichert [uMH12].

An drei der vier Ecken der Matrix existieren spezielle Markierungen, die Hauptpositionsmarkierungen ("FINDER PATTERN"), welche die Orientierung beim Auslesen des QR-Kodes vorgeben. Hierdurch "kann die Position, die Größe und

der Winkel des Symbols erkannt werden" [uMH12]. Sofern der QR-Kode eine bestimmte Größe überschreitet, werden weitere Erkennungsmuster hinzugefügt ("ALIGNMENT PATTERNS"), um die Ausrichtung zu erleichtern und Verzerrungen durch Kippen des Lesegerätes zu korrigieren. Abschließend befindet sich zwischen den Hauptpositionsmarkierungen des QR-Kodes eine abwechselnd komplementäre Bitfolge, das Zeitraster ("TIMING PATTERNS"), an dem sich die Matrix erstreckt. Das Zeitraster dient der Synchronisation und identifiziert die Koordinaten jeder Zelle [uMH12].

2.1 Dimensionierung

Die Matrix des QR-Kodes besteht aus mindestens 21 * 21 und maximal 177 * 177 Elementen (Modulen). Durch die Anzahl der Elemente sind ebenfalls die verfügbaren Fehlerkorrektur-Level definiert, die eine Rekonstruktion der Informationen bei 7% bis zu 30% fehlerhaft ausgelesener Daten ermöglicht. Je höher das Fehlerkorrektur-Level ist, desto weniger Speicherkapazität besitzt der QR-Kode.

Allgemein wird durch den Informationsgehalt, respektive die Länge der Zeichenfolge, den Grad der Fehlerkorrektur sowie durch den Datentyp, die Dimensionierung des QR-Kodes bestimmt. Der Datentyp meint hier die Verwendung des Zeichensatzes (ausschließlich Ziffern, alphanumerische Zeichen oder komplexere Zeichensätze), der entscheidend für die Anzahl der Bitlänge bei der Kodierung und Dekodierung eines Zeichen ist.

2.2 Speicherung der Dateninhalte

Im Datenbereich des QR-Kodes ("DATA AND ERROR CORRECTION CODEWORDS") werden zum einen die Informationen und zum anderen die redundanten Stellen des Reed-Solomon-Kodes binär in den schwarzen und weißen Feldern der Matrix kodiert.

Bedingt durch die Blöcke von Symbolen aus 8 Bit bei der Verwendung des RS-Kodes (RS-Kodes sind über $GF(2^8)$ definiert), kann innerhalb eines Blockes keine Zahl größer als 255 dargestellt werden. Dadurch, dass ein QR-Kode ebenso größere Blocklängen verarbeiten können soll, wird die Zeichenfolge auf mehrere Blocks aufgeteilt.

3 Kodierungs- und Dekodierungsverfahren

Mit Hilfe der Fehlerkorrektureigenschaft des RS-Kodes sowie der Kodierung der Formatinformationen durch einen BCH-Kode, können verschmutzte oder unvollständige QR-Kodes unter Berücksichtigung des Fehlerkorrektur-Levels wiederhergestellt werden.

Abbildung 3.1: Verschmutzter und unvollständiger QR-Kode

Die nachstehende Auflistung der Kodeparameter dienen in den folgenden Kapiteln der Beschreibung des RS-Kodes und des BCH-Kodes:

n - Gesamtkodewortlänge / Blocklänge

l - Informationslänge

k - Anzahl redundanter Stellen

d_{min} - minimaler Hamming-Abstand

grad g(x) - Grad des Generatorpolynoms

3.1 Anwendungsbeispiel

Die Zeichenfolge “KANALKODIERUNG” soll in einem QR-Kode kodiert gespeichert werden. Dabei wird ein QR-Kode Version 1 (21 * 21 Pixel) mit dem Fehlerkorrektur-Level Q (Fehlerkorrektur bis 25% fehlerhafter Dekodierung) verwendet.

3.2 Kodierungsverfahren

Der erste Schritt bei der Erstellung eines QR-Kodes ist die Generierung einer Bitfolge aus den Informationen. Zum einen enthält die Bitfolge die Zeichen der Originalnachricht, die kodiert werden soll. Zum anderen werden Informationsbits für den Typ des QR-Kodes hinzugefügt.

Im zweiten Schritt wird die generierte Bitfolge verwendet, um die Fehlerkorrektur-Kodewörter mit Hilfe eines RS-Kodes zu erzeugen. Anschließend werden die Kodewörter als Bitfolge der bereits generierten Bitfolge aus dem ersten Schritt angehangen.

Mit der vollständigen Bitfolge aus Informations- und Fehlerkorrekturbits können in dem dritten Schritt 8 verschiedene QR-Kodes, abhängig von dem verwendeten Maskenmuster, erzeugt werden. Maskenmuster definieren, welche Pixel in der Matrix hell und welche dunkel dargestellt werden.

Im letzten Schritt werden alle erzeugten QR-Kodes miteinander verglichen, um das bestmögliche QR-Symbol auszuwählen. Auf diese weise können für den Decoder schwer lesbare Muster, wie beispielsweise große zusammenhängende Blöcke einer gleichen Farbe, minimiert werden. Die grafischen Erstellung des QR-Kodes sowie die Bewertung der unterschiedlichen QR-Kodes werden in [Eby12a] ausführlich erläutert.

3.2.1 Reed-Solomon-Kodes

Definition 1 *Ein RS-Kode ist ein nichtbinärer zyklischer Kode, bei dem die Koeffizienten der Kodepolynome Elemente eines Grundkörpers $GF(q) = GF(2^{k_1})$) sind, der durch ein primitives Modularpolynom M(x) über GF(2) erzeugt wird [HK12].*

RS-Kodes können Fehler in einer übertragenen Nachricht erkennen und diese mit Hilfe der redundanten Daten korrigieren. Dabei werden RS-Kodes besonders zur Korrektur von Burstfehlern (zusammenhängende fehlerhafte Bits) eingesetzt. RS-Kodes sind zyklische Kodes, die mit Blöcken von Symbolen aus 8 Bit arbeiten.

Zur Kodierung der Informationsbits in den redundanten Datenbits werden zwei Verfahren bei der RS-Kodierung unterschieden. Zum einen kann das Multiplikationsverfahren zur Anwendung kommen, indem das Nachrichtenpolynom mit dem Generatorpolynom multipliziert wird, wodurch sich das Kodepolynom ergibt:

$$a(x) = a*(x) \cdot g(x)$$

Zum anderen kann das Divisionsverfahren zur Erstellung des Kodepolynoms verwendet werden. Dabei besteht die Berechnungsformel des Divisionsverfahrens aus zwei Summanden. Der erste Summand ist das Produkt des Nachrichtenpolynoms mit x^k (Verschiebung nach links), wobei k die Anzahl der redundanten Stellen ist. Der zweite Summand ergibt sich aus dem Rest der Division des "nach links verschobenen" Nachrichtenpolynoms durch das Generatorpolynom:

$$a(x) = a*(x) \cdot x^k + ((a*(x) \cdot x^k) mod g(x))$$

In Abschnitt 3.2.3 werden die in diesem Abschnitt verwendeten Begrifflichkeiten an dem Anwendungsbeispiel näher erläutert und das Divisionsverfahren zur Generierung der redundanten Stellen angewandt.

3.2.2 Generierung der Informationsbitfolge

"KANALKODIERUNG" ist eine alphanumerische Zeichenfolge, die nach der QR-Spezifikation durch die Bitfolge **0010** repräsentiert wird.

Zu dieser Bitfolge wird die Länge der Beispielzeichenfolge als Binärzahl hinzugefügt. 1110 ist die Binärdarstellung der 14 Zeichen langen Zeichenfolge "KANALKODIERUNG". Durch die Verwendung der QR-Kode Version 1 muss, bezüglich der Spezifikation, die Länge der Eingabezeichenfolge durch 9 Bits kodiert werden. Folglich wird die "linke Seite" mit Nullen aufgefüllt, wodurch die Bitfolge

000001110 entsteht. Zusammen mit der Bitfolge des Datentyps ergibt sich **0010 000001110**.

Um die Zeichenfolge kodieren zu können, wird sie in Zeichenpaare zerlegt (“KA”, “NA”, “LK”, “OD”, “IE”, “RU”, “NG”). Für jedes Paar wird der ASCII-Wert des ersten Zeichens mit 45 multipliziert und der ASCII-Wert des zweiten Zeichens zu diesem Produkt addiert. Daraufhin wird das Ergebnis zu einer 11 stelligen Bitfolge konvertiert. Hat die Zeichenfolge eine ungerade Anzahl von Zeichen, so wird der ASCII-Wert des letzten Zeichens in eine 6-Bitfolge konvertiert. Für das Anwendungsbeispiel können folgende Bitfolgen ermittelt werden:

KA: (45 * 20) + 10 = 910 -> **01110001110**

NA: (45 * 23) + 10 = 1045 -> **10000010101**

LK: (45 * 21) + 20 = 965 -> **01111000101**

OD: (45 * 24) + 13 = 1093 -> **10001000101**

IE: (45 * 18) + 14 = 824 -> **01100111000**

RU: (45 * 27) + 30 = 1245 -> **10011011101**

NG: (45 * 23) + 16 = 1051 -> **10000011011**

Nach der Kodierung der Zeichenfolge ergibt sich die Bitfolge:

0010 000001110
01110001110 10000010101 01111000101 10001000101 01100111000
10011011101 10000011011

Durch die Auswahl des QR-Kodes Version 1 mit dem Fehlerkorrektur-Level Q ist die Anzahl der Datenbits durch die Spezifikation der QR-Kodes vorgegeben [Wav10b]. Ist die generierte Bitfolge für diese Konstellation kleiner als 104 Bitstellen, so werden an dem “rechten Ende” bis zu vier Nullen aufgefüllt, ohne die Gesamtzahl von 104 zu überschreiten. Beispielsweise werden bei einer Länge von 90 Bitstellen (wie in diesem Beispiel) vier, jedoch bei 102 Bitstellen nur zwei weitere Nullen hinzugefügt. Die Beispielbitfolge wird demnach um vier weitere Nullen ergänzt:

0010 000001110
01110001110 10000010101 01111000101 10001000101 01100111000
10011011101 10000011011
0000

Da der zyklische RS-Kode mit Blöcken von 8 Bit arbeitet, erfolgt die Einteilung in 8 Bit lange Bitfolgen, um die Redundanzberechnung mit Hilfe des RS-Kodes zu ermöglichen. Dabei werden fehlende Nullen (in diesem Beispiel zwei Nullen) dem letzten Block hinzugefügt:

00100000 01110011 10001110 10000010 10101111 00010110 00100010
10110011 10001001 10111011 00000110 11000000

Um die Anzahl von 104 Datenbits nach der Spezifikation der QR-Kodes zu erreichen, werden der Bitfolge in dem letzten Schritt Spezialwörter angehangen. Die Bitfolgen 11101100 und 00010001 werden der Folge alternierend hinzugefügt, um eine Datenbitlänge von 104, mit gleichzeitiger, eindeutiger Kennzeichnung von Eingabewörtern und Spezialwörtern, zu erreichen. Die Generierung der Bitfolge aus den Informationen ist mit diesem Schritt abgeschlossen:

00100000 01110011 10001110 10000010 10101111 00010110 00100010
10110011 10001001 10111011 00000110 11000000 11101100

3.2.3 Erstellung der Fehlerkorrektur-Kodewörter

QR-Kodes beinhalten neben den reinen Informationen zusätzlich Blöcke von Fehlerkorrekturwörtern. Diese Blöcke von redundanten Daten garantieren die fehlerfreie Dekodierung bis zu einem gewissen Grad, selbst wenn ein Teil der Daten nicht gelesen werden kann. Dabei verwenden QR-Kodes das RS-Fehlerkorrekturverfahren.

Abhängig von dem Fehlerkorrektur-Level und der Version des QR-Kodes, kann mit Hilfe der QR-Spezifikation die Anzahl der Fehlerkorrekturwörter ermittelt werden. Für die QR-Kode Version 1 mit dem Fehlerkorrektur-Level Q werden 13 Fehlerkorrekturwörter benötigt, welche in den nachfolgenden Schritten erstellt werden.

Da das RS-Fehlerkorrekturverfahren auf Basis von Polynomen operiert, wird im ersten Schritt das “Nachrichtenpolynom” erstellt. Dieses repräsentiert die Bitfolge des vorangegangenen Abschnitts in polynomialer Darstellung. Die Bitfolge

selbst, wurde in 13 Blöcke mit je 8 Bitstellen unterteilt. Folglich besteht das Nachrichtenpolynom aus 13 Symbolen, wobei der dezimale Wert jedes Blockes dem Koeffizienten des jeweiligen Symbols entspricht. Die dezimale Darstellung der Beispielbitfolge ist definiert durch:

32 115 142 130 175 22 34 179 137 187 6 192 236

Der Exponent des ersten Symbols kann wie folgt berechnet werden:

(Anzahl der Datenblöcke) + (Anzahl der Fehlerkorrekturwörter) - 1:
= 13 + 13 - 1 = **25**

Aus der dezimalen Darstellung der Datenblöcke, der Anzahl der Datenblöcke und dem Exponenten des ersten Symbols ergibt sich das Nachrichtenpolynom:

$$32x^{25} + 115x^{24} + 142x^{23} + 130x^{22} + 175x^{21} + 22x^{20} + 34x^{19} + 179x^{18} + 137x^{17} + 187x^{16} + 6x^{15} + 192x^{14} + 236x^{13}$$

Im nachfolgenden Schritt wird ein "Generatorpolynom" erzeugt, welches als Divisor (Anwendung des Divisionsverfahrens) für das Nachrichtenpolynom fungiert und die Menge der Kodewörter vollständig beschreibt. Mit Hilfe des Ergebnisses dieser Division können die Fehlerkorrekturwörter berechnet werden. Das Generatorpolynom entspricht dabei einem "Galoiskörper" $GF(2^m)$ mit einer endlichen Anzahl von Elementen ($GF(2^8)$ = 256 Elemente) auf dem die Grundoperationen Addition und Multiplikation definiert sind.

Das Generatorpolynom hat die Form $(x - \alpha)(x - \alpha^2)...(x - \alpha^t)$, wobei t den Grad des Generatorpolynoms beschreibt, der mit der Anzahl der redundanter Symbole gleichzusetzen ist. In diesen Beispiel ist t als 13 definiert. Nach der Ausmultiplizierung der $(x - \alpha)$-Faktoren (Minimalpolynome) ergibt sich das Generatorpolynom:

$$x^{13} + \alpha^{74}x^{12} + \alpha^{152}x^{11} + \alpha^{176}x^{10} + \alpha^{100}x^9 + \alpha^{86}x^8 + \alpha^{100}x^7 + \alpha^{106}x^6 + \alpha^{104}x^5 + \alpha^{130}x^4 + \alpha^{218}x^3 + \alpha^{206}x^2 + \alpha^{140}x + \alpha^{78}$$

Das Ergebnis der Polynomdivision des Nachrichtenpolynoms mit dem Generatorpolynom ist im letzten Schritt:

$$220x^{12} + 231x^{11} + 187x^{10} + 20x^9 + 188x^8 + 190x^7 + 246x^6 + 211x^5 + 205x^4 + 71x^3 + 194x^2 + 59x^1 + 175x^0$$

Die Koeffizienten des Ergebnispolynoms entsprechen dann den Fehlerkorrekturwörtern. Diese werden der dezimalen bzw. der binären Darstellung der Beispielbitfolge hinzugefügt:

32 115 142 130 175 22 34 179 137 187 6 192 236 220 231 187 20 188
190 246 211 205 71 194 59 175

00100000 01110011 10001110 10000010 10101111 00010110 00100010
10110011 10001001 10111011 00000110 11000000 11101100 11011100
11100111 10111011 00010100 10111100 10111110 11110110 11010011
11001101 01000111 11000010 00111011 10101111

Nach diesem Schritt wurde dem QR-Kode ausreichend Redundanz hinzugefügt, um eine Fehlerkorrektur bis zu 25% zu ermöglichen. Die Parameter für den vorliegenden QR-Kode können mit $(n, l, d_{min}) = (26, 13, 14)$ beschrieben werden. Folgender Zusammenhang liegt den Kodeparametern des RS-Kode zugrunde:

- $n = l + k = 13 + 13 = 26$
- $k = gradg(x) = 13$
- $d_{min} = k + 1 = 14$
- $f_k = \frac{d_{min}-1}{2} = \frac{k}{2} = 6$

Hinsichtlich der Fehlerkorrektureigenschaften von RS-Kodes lässt sich aus $d_{min} = 2f_k + 1$ ableitet, dass mit Sicherheit $f_k = 6$ Einzelfehler zu je $k_1 = 5$ Bit korrigierbar sind. Zusätzlich können Bündelfehler der Länge $(f_k - 1)k_1 + 1 = 26$ Bit (verteilt auf $f_k = 6$ Elemente) korrigiert werden. k_1 entspricht dabei dem Grad des Modularpolynoms, mit dem das Generatorpolynom erzeugt wird.

Der vorliegende RS-Kode erfordert einen Grad $k_1 = 5$ des Modularpolynoms, da den RS-Kodes folgende Formel zur Berechnung der Gesamtkodewortlänge n zugrunde liegt:

$$n = 2^{k_1} - 1 \geq l + k$$

Die Parameter “l” und “k” sind in diesem Beispiel je 13 Bit, wodurch nach obiger Berechnungsformel ein (31, 18, 14) RS-Kode beschrieben wird. Entsprechend der Vorlage von “l” wird dieser auf einen (26, 13, 14) RS-Kode verkürzt. Hierbei

bleiben die Korrektureigenschaften erhalten, jedoch geht die zyklische Eigenschaft verloren.

3.2.4 Auswahl des Maskenmusters

Es stehen acht verschiedene Maskenmuster zur Verfügung, um die ausgegebene QR-Matrix bezüglich besserer Lesbarkeit weiterführend zu bearbeiten. Dabei verändert ein gewähltes Muster die Bits abhängig von der jeweiligen Position in der Matrix. Die Maskenmuster können anhand der nachstehenden Formeln berechnet werden [Eby12b].

Maskennummer	**Formel**
0	(y + x) mod 2 == 0
1	y mod 2 == 0
2	x mod 3 == 0
3	(y + x) mod 3 == 0
4	((y / 2) + (x / 3)) mod 2 == 0
5	((y * x) mod 2) + ((y * x) mod 3) == 0
6	(((y * x) mod 2) + ((y * x) mod 3)) mod 2 == 0
7	(((y + x) mod 2) + ((y * x) mod 3)) mod 2 == 0

Durch die gewählte Maskenformel wird für jedes Bit in der Matrix entschieden, ob der enthaltene Wert negiert wird oder nicht (Änderung der Farbe von weiß zu schwarz oder umgekehrt). Im Speziellen werden die Platzhalter “x” und “y” in der Formel durch die Koordinaten eines betrachteten Bits ersetzt (“x” und “y” Koordinate in der QR-Matrix). Entspricht das Ergebnis der Maskenformel gleich 0, erfolgt eine Negation der Wertes. Beispielsweise wird der Wert 0 an der Koordinate (1,3) gespeichert, sofern der Wert des Bits an der Koordinate (1,3) gleich 1 und das Ergebnis der Maskenformel gleich 0 ist.

3.3 Speicherung der Formatinformationen durch BCH-Kodes

Die Informationen über das Fehlerkorrektur-Level und das verwendete Maskenmuster wird ebenfalls in redundanter Form im QR-Kode gespeichert. Zum einen wird die 15 Bit lange Folge unterhalb der beiden oberen Hauptpositionsmarkierungen in horizontaler Richtung gespeichert. Zum anderen erscheint die Bitfolge rechts neben den beiden linken Hauptpositionsmarkierungen in vertikaler Richtung.

Hinsichtlich des Anwendungsbeispiels und der Spezifikation der QR-Kodes wird das Fehlerkorrektur-Level Q mit der Bitfolge **11** kodiert. Weiterhin soll das Maskenmuster 3 als optimales QR-Muster zur Anwendung kommen. Daraus folgt die Bitfolge **11011**.

Für die Kodierung der Formatinformationen kommt ein zyklischer dreifach-fehlerkorrigierenden BCH-Kode (Bose-Chaudhuri-Hocquenghem-Code) zur Anwendung, der mehrere 1-Bit-Fehler korrigieren kann. Die Bitfolge 11011 soll durch einen (15, 5, 7) BCH-Kode beschrieben werden, um Lesefehler zu korrigieren. Um das Divisionsverfahren zur Berechnung des Kanalkodewortes verwenden zu können, werden dieser Bitfolge 10 weitere Bitstellen (x^k) hinzugefügt (um k-Stellen nach links verschoben), wodurch die Bitfolge **110110000000000** entsteht. Hieraus lässt sich folgendes Nachrichtenpolynom ableiten:

$$x^{14} + x^{13} + x^{11} + x^{10}$$

Wie auch bei dem RS-Kode, der ein spezieller BCH-Kode ist, wird ein Generatorpolynom, mit dem Grad gleich der Anzahl an redundanten Bitstellen (k = 10), verwendet. Das Generatorpolynom ist für diesen Kode ($k_1 = 4$) wie folgt definiert.

$$x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

Der Rest der Polynomdivision des Nachrichtenpolynoms mit dem Generatorpolynom ergibt $x^9 + x^4 + x^2$. Durch Addition des um k-Stellen verschobenen Quellkodewortes mit dem Rest der Polynomdivision, entsteht das Kanalkodewort, das in dem QR-Kode gespeichert werden kann:

$$x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^4 + x^2 \text{ — } \mathbf{110111000010100}$$

Das kodierte Kanalkodewort enthält nun ausreichend Redundanz, um mehrere 1-Bit-Fehler ($f_k = 3$) korrigieren zu können. Aus den Parametern des (15, 5, 7) BCH-Kodes kann nachstehender Zusammenhang abgeleitet werden:

- $n = l + k = 5 + 10 = 15$
- $k = gradg(x) = 10$
- $d_{min} = 7$
- $f_k = \frac{d_{min}-1}{2} = 3$

Wie bereits bei dem erzeugten RS-Kode aufgezeigt, gelten folgende Formeln:

$$n = 2^{k_1} - 1 \geq l + k$$

$$k \geq d_{min} - 1$$

Letztere Formel ist auch als Singleton-Schranke bekannt, welche die obere Schranke für die minimale Hamming-Distanz eines Blockkodes definiert. Der RS-Kode ist der einzige Kode, der die Singleton-Schranke auf Gleichheit erfüllt (setzt am effizientesten d_{min} um). Bei dem vorliegenden BCH-Kode wird die obere Singleton-Schranke jedoch nicht erreicht.

3.4 Dekodierungsverfahren

Damit der QR-Kode gelesen werden kann, wird ein Lesegerät, beispielsweise ein Smartphone mit Kamera, benötigt. Das Lesegerät nimmt den QR-Kode grafisch auf und analysiert ihn. Der erste Schritt der Analyse beschäftigt sich mit der korrekten Positionierung (“Image Conversion”). Mit Hilfe der Hauptmarkierungspunkte in dem QR-Kode erfolgt die Ausrichtung sowie die eindeutige Zuordnung jedes Pixels in ein Koordinatensystem (“QR Detector”).

Daraufhin werden die Formatinformationen ausgelesen bzw. übersetzt (grafische Punkte in Bitfolgen). Können die Formatinformationen nicht korrekt dekodiert werden, erfolgt eine Fehlerkorrektur mit dem BCH-Dekodierungsverfahren. Führt dies zum Erfolg, werden die Formatinformationen verwendet, um die Informationen in dem QR-Kode zu entschlüsseln. Werden weiterhin die Informationen fehlerhaft ausgelesen, kommt das RS-Dekodierungsverfahren zur Anwendung,

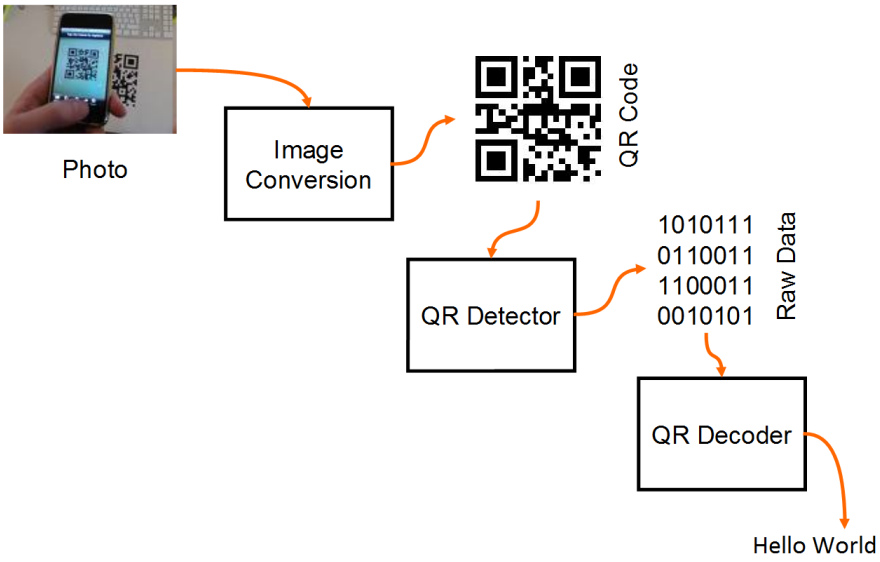

Abbildung 3.2: Dekodierungsschritte von QR-Kode Lesegeräten

mit dem bis zu 30% der Daten wiederhergestellt werden können. Die dekodierten Bitfolgen können anschließend mit Hilfe der Formatinformationen in Text übersetzt werden (“QR Decoder”).

Abbildungsverzeichnis

Literaturverzeichnis

[Alb11] P. K. Albrecht, T. und Plinkert. Qr-code - was ist das? *HNO - Springer-Verlag*, 2011.

[Eby12a] Carolyn Eby. Qr code tutorial. http://www.thonky.com/qr-code-tutorial/part-3-mask-pattern, 2012.

[Eby12b] Carolyn Eby. Qr code tutorial. http://www.thonky.com/qr-code-tutorial, 2012.

[Gmb12] Evolaris Next Level GmbH. Reality magazin & app: Interaktivierung vom feinsten! http://www.evolaris.net/reality-magazin-app-video/, 2012.

[HK12] Dagmar Schönfeld Herbert Klimant, Rudi Piotraschke. *Informations- und Kodierungstheorie*. Springer Vieweg, 2012.

[IAT12] IATA. Bar-coded boarding passes (bcbp). http://www.iata.org/whatwedo/stb/bcbp/Pages/index.aspx, 2012.

[uMH12] Iris Uitz und Michael Harnisch. Der qr-code – aktuelle entwicklungen und anwendungsbereiche. *Informatik-Spektrum - Springer-Verlag*, 2012.

[Wav10a] Denso Wave. Qr code features. http://www.qrcode.com/en/qrfeature.html, 2010.

[Wav10b] Denso Wave. Qr code versions. http://www.qrcode.com/en/vertable1.html, 2010.