

Fakultät für Informatik und Mathematik

Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung

PROF. DR. TOMAS SAUER

Masterarbeit

Versteckte Codierung von Bildinformationen

Zsuzsanna Huber

Studienfach: Master Informatik mit Schwerpunkt Informations- und Kommunikationssysteme

ERSTPRÜFER: Pro Zweitprüfer: Pro Betreuer: Pro Ansprechpartner: Dr.

Prof. Dr. Tomas Sauer Prof. Dr. Brigitte Forster-Heinlein Prof. Dr. Tomas Sauer Dr. Alexander Zimmermann

Datum: 10. April 2018

Eine wirklich gute Idee erkennt man daran, dass ihre Verwirklichung von vorne herein ausgeschlossen erscheint. (Albert Einstein)

Dieser Gedanke ist zweifach zutreffend:

Zum einen kann ich ihn allgemein auf mein Zweitstudium an der FIM¹ der Universität Passau beziehen. Schwanger und mit Baby habe ich vor $5\frac{1}{2}$ Jahren angefangen zu studieren; an eine erfolgreiche Verwirklichung war nicht wirklich zu denken.

Ein herzliches Dankeschön an meinen Mann Florian Huber, an meine Kinder David und Anna, an meine Eltern Eva und Josef Koman, an meine Schwiegermutter Ingrid und an ihren Mann Max Müller für ihre Unterstützung!

Zum anderen passt der Spruch selbst zum Thema der Arbeit: Wie kann eine "unsichtbare", versteckte Codierung verwirklicht werden? Das kann doch nicht möglich sein…

Die Neugier hat mich stets motiviert – so auch bei der vorliegenden Thematik. Auch hier ein Dank an Herrn Prof. Dr. Tomas Sauer und an FORWISS fürs Ermöglichen.

¹Fakultät für Informatik und Mathematik

Zusammenfassung

Die digitale und automatisierte Verarbeitung von Produktionslinien bildet einen Aufgabenbereich von Industrie 4.0. In diesem Kontext analysiert die Arbeit, inwieweit im Prozess einer Druckerei Bilder versteckt codiert werden können. Die Codierung kann zur automatischen Identifikation von Dokumenten genutzt werden. Der Stand der Technik bietet eine Fülle von Möglichkeiten, um in digitale Dateien nicht wahrnehmbare Informationen einzubetten. Bestehende Techniken werden vorgestellt und es wird validiert, inwieweit eine Umsetzung in vorliegendem Fall stattfinden kann. Im Druckprozess entstehen durch Codierung, Speicherung, Entwicklung und Reproduzieren der Bilder Fehler, die der Identifikation standhalten müssen.

Dazu werden unterschiedliche Domänen wie Steganographie, Bildbearbeitung im Ortsraum, Frequenzraum und Zeit-Frequenzraum betrachtet.

Simulationen und ein praktischer Test können abhängige Parameter und Anforderungen ermitteln. Eine mögliche Ableitung eines geeigneten Vorgehens ist das Ziel dieser Arbeit.

Inhaltsverzeichnis

1	Einf	ührung 4
	1.1	Motivation
	1.2	Inhaltsübersicht
2	Star	nd der Technik 8
	2.1	Steganographie
		2.1.1 Historische Entwicklung
		2.1.2 Terminologie \ldots 9
		2.1.3 Verfahren
	2.2	Bildbearbeitung im Ortsraum
		2.2.1 Historische Ursprünge
		2.2.2 Begriffsbestimmungen
		2.2.3 Segmentierung
	2.3	Bildbearbeitung im Frequenzraum
		2.3.1 Historische Ursprünge
		2.3.2 Mathematische Grundkonzepte
		2.3.3 Filterung
	2.4	Bildbearbeitung im Zeit-Frequenzraum
		2.4.1 Historische Entwicklung
		2.4.2 Wavelets
		2.4.3 Filterung
	2.5	Resümee
2	Läs	41
3	2 1	Teilprehleme 41
	5.1	1 Parkalierung der reproduzierten Dekumente 42
		2.1.2 Codiemung und Decodiemung
		$3.1.2$ Coderung und Decoderung $\ldots \ldots \ldots \ldots \ldots \ldots \ldots 44$
	29	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
	0.2	$\begin{array}{c} 3 2 1 \text{Algorithmug} \\ \end{array} $
		3.2.2 Algorithmus
		$3.2.2 \text{Bernwachsteinen} \dots \dots \dots \dots \dots \dots \dots \dots \dots $
	22	$5.2.5 \text{Resume} \qquad 5.2.5 \text{Resum} 5.2.5 \text{Resume} \qquad 5.2.5$
	0.0	$3 3 1 \qquad \text{Algorithmus} \qquad 54$
		$3.3.9 \text{Simulation} \qquad 55$
		$3.3.3 \text{Praktischer Test} \qquad \qquad$
		$3.3.4 \text{Regimon} \qquad \qquad$
	3/	7.014 Tresume 119 7.014 Frequenzraum 191
	0.4	3 4 1 Algorithmus
		3 4 9 Validierung 191
		9.7.2 vanutorung

Inhalts verzeichnis

3.4.3 Resümee	127
4 Fazit und Ausblick	128
Anhang A Überblick über die verwendeten Dateiformate	129 129 130
Abbildungsverzeichnis	139
Tabellenverzeichnis	141
Literaturverzeichnis	143

1 Einführung

1.1 Motivation

Hinsichtlich Industrie 4.0 findet die Validierung der Realisierbarkeit einer versteckten Codierung von Informationen bei Bildern statt. Die eingebetteten Daten können etwa zur Prüfung von Eintrittskarten genutzt werden. Als eine weitere Anwendung ist die Automatisierung der Produktionslinie bei einer Druckerei denkbar.

Für letzteren Gebrauch wird folgendes Szenario erstellt: In einer Druckerei (Bilder, Flyer, Plakate u.v.m) ist der Produktionsprozess von der Auftragsannahme bis zum Drucken und Zuschneiden der Druckerzeugnisse digitalisiert, die Zuordnung der Bestellungen erfolgt anschließend manuell. Es gilt, diesen digitalen Bruch zu schließen und die Zuordnung zu den Auftragsadressen mit Hilfe von eingebetteten Information zu automatisieren. Die Aufträge liegen im digitalen Grafikformat vor, die Text, Bild(er) oder eine Kombination daraus enthalten können. Im Folgenden wird allgemein von Dokument oder Bildern gesprochen. Aufgabe ist nun auf jedem einzelnen, digitalen Druckerzeugnis vor dem Druck eine versteckte Codierung einzubetten, die durch das menschliche Sehvermögen nicht wahrnehmbar ist. Die Bilder sollen nach dem Druck mit einer Standard-Hardware reproduziert (digitalisiert) werden, um die eingebettete Codierung zu decodieren. Das Extrahieren der hinzugefügten Daten soll unter unterschiedlichen Licht- und Umgebungsverhältnissen gewährleistet werden.

Abbildung 1.1 stellt diesen Prozess dar¹.



Abb. 1.1: Prozess von Auftragannahme bis zum Versand

¹Der Prozess ist mit *ADONIS*, http://en.adonis-community.com/ erstellt.

Die Umsetzung setzt eine Robustheit trotz Störfaktoren wie Rauschen, der Beleuchtung oder Quantisierungsfehler voraus.

In vorliegender Arbeit werden zur Realisierung mögliche Verfahrensansätze analysiert. In 1.2 Inhaltsübersicht werden vorab Verfahren der Bereiche Bildverarbeitung und Information Hiding auf Eignung geprüft.

Im Bezug auf die Methodikauswahl werden nachstehende Kriterien abgeleitet und bei der Analyse mitberücksichtigt:

- 1. In das digitale Dokument werden Daten derart eingebettet bzw. modifiziert, dass sie für das menschliche Auge nicht wahrnehmbar sind.
- 2. Die hinzugefügten bzw. veränderten Informationen lassen sich durch Reproduzieren (wie Einscannen oder Photographieren) des Dokuments wieder entschlüsseln.

1.2 Inhaltsübersicht

In der digitalen Kommunikation existieren zur Einbettung, Codierung oder Modifikation von Daten bereits zahlreiche Verfahren in der Domäne von Information Hiding und Bildverarbeitung. In diesem Abschnitt wird geklärt, welche ihrer Teilbereiche von Bedeutung sind und welche von vornherein ausgeschlossen werden können.

Information Hiding Diese Domäne, wie in *Abbildung 1.2* dargestellt, gliedert sich in verschiedene Bereiche.



Abb. 1.2: Information Hiding nach [39, S. 86]

Anonymität spiegelt Identitätsmanagement im Internet und den Umgang mit personenbezogenen Daten wider und kann in der Arbeit ausgeschlossen werden.

1.2 Inhaltsübersicht

Ebenso vernachlässigbar sind Verdeckte Kanäle, die Kommunikationskanäle darstellen. Durch sie können Informationen übertragen werden, ohne dass es auffällt [8, S.106].

Bei Wasserzeichen handelt es sich um die Einbettung von Daten bei urheberrechtlich geschützten Werken, die abhängig vom Werk sind. Diese können Dokumente, digitale Medien, Bücher, Gemälde oder ähnliches sein. Die Wasserzeichen, wie *Abbildung 1.2* zeigt, teilen sich in unterschiedliche Gebiete, von denen die robusten, nicht wahrnehmbaren Wasserzeichen von Bedeutung sind.

Bei den digitalen Wasserzeichen werden in die vorliegenden Coverdaten C durch unterschiedliche Verfahren Daten m eingebettet und ergeben ein Werk W mit Wasserzeichen versehen: $f(m, C) \rightarrow W$. Dabei werden zur Sicherheit gegen Manipulation kryptographisch sichere Schlüssel verwendet [25, S.102].

Im Allgemeinen gibt es zwei Verfahrensbereiche, wie Wasserzeichen eingefügt werden können:

Einerseits im Bildraumverfahren, bei dem das Einbetten von Informationen durch die Veränderung einzelner Pixel durchgeführt wird: Algorithmen nach Langelaar, Lagendijk und Biemond sind Beispiele, die unter anderem mit der Helligkeit von Pixeln arbeiten. Diese Verfahren können nicht genutzt werden, da sie im niederwertigen Bitbereich Informationen verstecken und lediglich in digitalen Medien eine große Rolle spielen [13]. Fehlererkennende und -korrigierende Maßnahmen fehlen, welche die abgebildeten Verarbeitungsschritte überstehen. Die unter 1.1 Motivation genannten Kriterien werden nicht abgedeckt.

Zum anderen werden Frequenzraumverfahren herangezogen. Anforderungen bezüglich Urheberrechtsschutz liegen nicht vor, sodass diese Domäne unabhängig davon in 2.3 Bildbearbeitung im Frequenzraum analysiert wird.

Zum Schluss steht das Gebiet der Steganographie im Bereich Information Hiding aus. Analog zu Wasserzeichen werden bei den meisten Verfahren in digitale Coverdaten Informationen im niederwertigen Bitbereich hinzugefügt. In 2.1 Steganographie wird dieser Bereich überblicksartig dargestellt. Im Rahmen dieser Arbeit eignen sich unter Umständen zwei Verfahren zur Codierung von formatierten Texten.

Bildverarbeitung Es wird zwischen lokalen und globalen Modifikationen unterschieden. Die Idee bei lokalen Veränderungen ist eine Art Fingerabdruck eines Druckerzeugnisses zu bilden und abhängig der immanenten Charakteristika Daten einzubinden. In *2.2 Bildbearbeitung im Ortsraum* wird dieses Vorgehen detailliert geschildert.

Im Kontext von globalen Modifikationen im Frequenzraum wird insbesondere die Fourier-Transformation in 2.3 Bildbearbeitung im Frequenzraum behandelt. Weitere Verfahren in der Frequenz-Domäne wie beispielsweise Sinus- oder Cosinus-Transformationen können vernachlässigt werden, da sie vergleichbare Techniken darstellen.

In 2.4 Bildbearbeitung im Zeit-Frequenzraum findet die Betrachtung von Wavelets statt. Neben den Frequenzinformationen sind die zeitlichen zugänglich, durch die eine unsichtbare Codierung möglich sein kann.

Bei Verfahren im Kontext mit Bildverarbeitung werden 2D-Graustufenbilder betrachtet.

Die für das Vorhaben qualifizierte Techniken und Methodiken aus diesen Bereichen wer-

1.2 Inhaltsübersicht

den in 2 Stand der Technik vorgestellt.

Dies beinhaltet die Betrachtung von Zielstellung, Merkmalen, Anforderungen und Verfahren.

In 3 Lösungsansätze wird die Anwendbarkeit dieser Domänen in 3.2 Ortsraum, 3.3 Frequenzraum und 3.4 Zeit-Frequenzraum validiert. Dazu werden zunächst Algorithmen formuliert, die anschließend entweder analysiert oder simuliert werden. Die Simulationen finden mit GNU Octave² und Matlab R2016a³ statt.

Eine Besonderheit stellt die Durchführung eines praktischen Tests (siehe 3.3.3) dar, bei der geprüft wird, ob die Codierung modifizierter Bilder im Frequenzraum erkennbar ist.

Ein Ziel ist, aus den gewonnenen Erkenntnissen der Verfahrensanalysen, Simulationen und dem praktischen Test mögliche Anforderungen für eine spätere Implementierung der Software abzuleiten.

²GNU Octave ist Open Source (https://www.gnu.org/software/octave/). Das Programm wurde genutzt, um Simulationen und den praktischen Test im Frequenzraum durchzuführen und auszuwerten.
³Die Software ist unter https://www.mathworks.com/products/new_products/release2016a.html zu finden. Sie wurde für die Simulationen im Zeit-Frequenzraum genutzt.

2 Stand der Technik

Dieses Kapitel beschäftigt sich mit bestehenden Techniken der Bildverarbeitung. Zusammenfassend werden die Konzepte und ein kurzer historischer Umriss in 2.1 Steganographie, 2.2 Bildbearbeitung im Ortsraum, 2.3 Bildbearbeitung im Frequenzraum und 2.4 Bildbearbeitung im Zeit-Frequenzraum vorgestellt.

2.1 Steganographie

2.1.1 Historische Entwicklung

Ursprünge Die Anfänge steganographischer Kommunikation gehen mehrere tausend Jahre zurück. Vor allem zu Kriegszeiten war der Austausch geheimer Nachrichten entscheidend. Dabei wurde der Einfallsreichtum an Ideen belohnt; hier eine Auswahl an Beispielen: Herodotus (486-425 vor Christus) berichtet in seinem Buch *The Histories* [19] um 440 v.C. von seinem Sklaven, dem er die Haare abrasierte, eine Nachricht auf die Kopfhaut tätowierte und die Haare nachwachsen ließ. So konnte er eine Revolte gegen die Perser initiieren. Des Weiteren schildert Herodotus wie Demeratus, ein Grieche am persischen Hof, Sparta vor einer Invasion durch Xerxes warnte: Er entfernte das Wachs von der hölzernen Schreibtafel, schrieb die Nachricht auf das Holz und bedeckte es wieder mit Wachs [23, S.3].

Bekannt ist auch das Buch *Hypnerotomachia Poliphili* [1] von 1499, wobei der erste Buchstabe jeden Kapitels die geheime Botschaft ergibt: *Father Colona passionately loves Polia*. Während den Weltkriegen wurden viele Nachrichten mit vertraulichen Daten versehen. Beispielsweise hat ein deutscher Spion im Weltkrieg I die Bestellung von Zigarren vorgetäuscht, welche unterschiedliche Typen von britischen Kriegsschiffen repräsentiert haben [24].

Steganographie in der Kunst [38, S.69f.] Selbst im Bereich der bildenden Kunst greifen Künstler auf steganographische Möglichkeiten zurück, um Informationen einzubetten. Hans Holbein der Jüngere beispielsweise verwendete in seinem Werk *Die Gesandten* die anamorphische Verzerrung, wie in *Abbildung 2.1* zu sehen. Darin ist ein Totenschädel im Winkel von 27° erkennbar.



(a) Die Gesandten, Sicht von vorne [9]



(b) Der Totenkopf im Winkel von 27° [10]

Abb. 2.1: Die Gesandten von Hans Holbein d.J., 1533

Steganographie wird auch in musikalischen Werken anwendet. Geheime Botschaften können unter anderem durch Rückwärtsspulen des Stückes identifiziert werden. Lässt der Hörer die Musik normal vorwärts ablaufen, sind die geheimen Nachrichten durch bloße Wahrnehmung nicht zu filtern.

2.1.2 Terminologie

Definition

Der Begriff Steganographie kommt aus dem Griechischen und bedeutet so viel wie *die Lehre vom verdeckten Schreiben*. Dabei werden vertrauliche Daten m in Coverdaten (Hülldaten) C eingebettet und dadurch Stegodaten S erzeugt [16, S.183].

f(m,C) o S

Die Steganographie wird in *technische* und *linguistische* klassifiziert und ist ein Teilbereich von *Information Hiding*. Wenn physikalische Mittel zu Hilfe genommen werden, um Informationen zu verdecken, wie Unsichtbare Tinte oder Mikropunkte, so handelt es sich um *technische* Steganographie [39, S.85].

Im Gegensatz dazu definiert sich die *linguistische* durch Einbetten von Nachrichten im Text bzw. einem Dokument – den Coverdaten. Beispielsweise werden in einem Bild Pixel oder in einem Covertext Buchstaben-Positionen vertauscht [39, S.85] oder die ersten Buchstaben jeden Kapitels ergeben die geheime Botschaft (siehe in 2.1.1 Historische Entwicklung).

Eigenschaften

Steganographie wird oft in Zusammenhang mit Kryptographie genannt, wobei beide zum Ziel die *Vertraulichkeit von Daten* haben. Durch Kryptographie wird allerdings nicht eine zusätzliche Nachricht verdeckt, sondern die Informationen selbst durch unterschiedliche

2.1 Steganographie

Algorithmen zu unverständlichen Daten umgewandelt und dadurch der Inhalt geschützt. Bei Steganographie wird die Existenz der versteckten Daten geheim gehalten [16, S.183]. Wenn eine steganographische Nachricht kryptographisch verschlüsselt wird, so kann ein Angreifer trotz einer Entschlüsselung die versteckten Daten nicht zwangsweise entdecken. Nur derjenige, der den Algorithmus kennt, kann die geheime Botschaft entschlüsseln.

Es handelt sich also um ein Sicherheitskonzept, welches "unsichtbare" Daten enthält. Zur Einbettung von Nachrichten können die Coverdaten verwendet (adaptive Algorithmen) oder ohne Berücksichtigung dieser eingebunden werden (nichtadaptive Algorithmen) [12, S.456]. Meist sind die versteckten Daten viel kleiner als die Hülldaten.

2.1.3 Verfahren

Im Bereich der linguistischen Steganographie gibt es eine Vielfalt an Möglichkeiten, um zusätzliche Daten einzubetten. Dazu können digitale Dateien im Text-, Audio-, Bild- oder Videoformat als Coverdaten verwenden werden [16, S. 184].

Eine Klassifizierung nach Modifikationsart des Trägers ist folgende: [23, S.43].

- (1) **Substitutionssysteme**: Sie ersetzen redundante Coverdaten mit einer geheimen Information. Dazu zählen *Least significant Bit* (LSB), Pseudozufällige Permutationen, Bearbeitung von paletten-basierten Bildern, Quantisierung und Dithering. Des Weiteren können ungenutzter bzw. reservierter Speicher in Rechnersystemen, Bildmanipulation oder Kompression des Bildes genutzt werden, um spezielle Eigenschaften wie Helligkeit zu ändern. Bei all diesen Verfahren werden zusätzliche Informationen durch "geheime" Bits in unbedeutenden Teilen der Coverdaten substituiert. Der Empfänger des manipulierten digitalen Mediums kennt die genaue Position der eingebetteten Daten. Der Sender geht davon aus, dass ein passiver Angreifer diese nicht bemerkt [23, S.45ff.].
- (2) Transformations-Domäne-Techniken: Sie binden geheime Informationen in einen Transformationsraum des Signals ein, beispielsweise im Frequenzbereich. Daten lassen sich robuster hinzufügen, wenn sie nicht in der Orts-Domäne, sondern in einer transformierten geschehen, wie in der Frequenz-Domäne. Transformations-Domäne-Methoden betten zusätzliche Informationen in signifikante Bereiche der Coverdaten ein, sodass Angriffe, die Kompression, Bildmanipulation wie LSB oder ähnliches betreffen, erschwert sind. Für das menschliche Sehvermögen bleiben die Veränderungen unter der Wahrnehmungsschwelle. Beispiele sind Diskrete Cosinus-, Frequenz- oder Wavelet-Transformation [23, S.56].
- (3) **Frequenzspreizung-Techniken**: Es werden Konzepte von Frequenzspreizung in der Nachrichtentechnik adaptiert.

Diese Verfahren haben sich vor allem seit den 50er Jahren entwickelt, um die Wahrscheinlichkeit von Abfangen der Nachrichten zu minimieren (*Low-probability-of-intercept*) und eine störungsfreie Kommunikation zu ermöglichen. Frequenzspreizung definiert sich als Übertragungsmittel, in welcher das Signal eine Bandbreite einnimmt, die über dem Minimum liegt. Dies ist notwendig, um die Informationen zu senden. Die Bandspreizung wird durch unabhängigen Code von den Daten erreicht. Zur Entspreizung bzw. Datenwiederherstellung reicht ein synchronisierter Empfang beim Empfänger. Es gibt das *Direct sequence Spread Spectrum (DSSS)* und das *Frequenzsprungverfahren (Frequency Hopping Spread Spectrum (FHSS)* [23, S.64] [36].

2.1 Steganographie

(4) **Statistische Methoden**: Sie verschlüsseln Informationen, indem sie mehrere statistische Eigenschaften der Coverdaten ändern und beim Entschlüsseln Hypothesentests nutzen.

Im Allgemeinen werden bei dieser Technik einzelne Bits einer Nachricht m der Länge l in die Coverdaten eingebettet, wobei diese in l Blöcke geteilt werden. Beträgt der Wert m_i an der *i*-ten Stelle der Information eine 1, so wird im *i*-ten Block ein Bit hinzugefügt [23, S.67].

- (5) **Distortionstechniken**: Es werden Informationen durch Signalverzerrung gespeichert und beim Entschlüsseln die Abweichung von den Originaldaten berechnet. Nachstehend werden zwei Verfahren vorgestellt.
- (6) **Coverdaten-Generierungs-Methoden**: Es werden digitale Coverdaten eigens für die einzubettenden Daten erzeugt [23, S.71f.].

Distorsionstechniken [23, S.69f.] In diesem Abschnitt wird das Verschlüsseln von Information in formatierten Texten behandelt. Dabei werden Textinhalte nach bestimmten Algorithmen verändert, sodass eine Distorsion stattfindet. Beim Entschlüsseln werden im Gegensatz zu den meisten anderen Verfahren die Original-Coverdaten benötigt. Dabei werden die Differenzen des ursprünglichen und dem modifizierten Dokument herausgearbeitet und ergeben letztendlich die eingefügten Informationen.

Formatierte Texte werden beim Verschlüsseln von Informationen in formatierten Texten als binäres Bild interpretiert. Es wird der Abstand zwischen aufeinanderfolgenden Textzeilen oder Worten verwendet, um Daten einzupflegen.

Bei der Zeilen-Abstand-Kodierung werden die Positionen der Linien im Dokument entsprechend der Nachricht m der Länge l hoch- oder hinunter gesetzt. Die restlichen Linien bleiben fest. Ein Bit m_i der einzubettenden Nachricht wird in einer Zeile folgendermaßen verarbeitet:

$$\boldsymbol{m}_{i} = \begin{cases} 1 & \text{die Line wird nach oben bewegt} \\ 0 & \text{die Linie wird nach unten bewegt} \end{cases}$$
(2.1)

Beim Entschlüsseln wird der Schwerpunkt durch Berechnung des Schwerpunktes um eine horizontale Achse ermittelt. Anschließend werden Distanzen wie folgt berechnet: $m = \frac{\Delta_{R+} + \Delta_{R-}}{\Delta_{R+} + \Delta_{R-}}$

$$n \coloneqq \frac{\Delta_{R+} + \Delta_{R-}}{\Delta_{R+} - \Delta_{R-}}, \quad o \coloneqq \frac{\Delta_{X+} + \Delta_{X-}}{\Delta_{X+} - \Delta_{X-}}$$

$$(n \ge o, dio Distanz wurdo orhöht)$$

$$\begin{cases} \boldsymbol{n} > \boldsymbol{o} & \text{die Distanz wurde erhoht} \\ \boldsymbol{n} < \boldsymbol{o} & \text{die Distanz wurde erniedrigt} \end{cases}$$
(2.2)

Dies ist ein Text.	Dies ist ein Text.
Dies ist ein Text.	Dies ist ein Text.
Dies ist ein Text.	[®] Dies ist ein Text.

Abb. 2.2: Zeilen-Abstand-Kodierung: Verzerrung eines formatierten Textes. Links ist der Originaltext abgebildet. Auf der rechten Seite befindet sich der modifizierte Text. In der Abbildung 2.2 wurde die erste Zeile nach unten und die dritte Zeile nach oben verschoben, demnach ist n < o.

Eine weitere Möglichkeit ist die Wort-Abstand-Kodierung, bei der der Abstand zwischen den Worten verändert werden kann. Prinzipiell kann dieser überall variieren, einzige Bedingung ist, dass die Summe der Verschiebungen in einer Zeile gleich 0 ist und damit die Zeilenausrichtung gleich bleibt.

Abbildung 2.3 zeigt im unten stehenden Satz mehrere Verschiebungen. Die Worte "ist" und "einem" sind um ein Pixel nach links (-1), "Beispiel" und "längeren" ein Bildpunkt nach rechts (+1) verrückt.

Die Addition beträgt: (-1) + (+1) + (-1) + (+1) = 0

Hier ist ein Beispiel mit einem längeren Satz. Hier ist ein Beispiel mit einem längeren Satz.

Abb. 2.3: Wort-Abstand-Kodierung: Verzerrung eines formatierten Textes

2.2 Bildbearbeitung im Ortsraum

Dieses Kapitel betrachtet Bildverarbeitung von lokalen Bildinformationen.

Nach einem kurzen historischen Abriss werden neben Begriffsbestimmungen Methoden vorgestellt, die sich zur Merkmalsextraktion eignen. Spezifische Merkmale werden anschließend modifiziert.

2.2.1 Historische Ursprünge

1858 wurde das erste transatlantische Kabel zwischen Europa und Nordamerika verlegt [37, S.18]. Dies war der Grundstein für viele Technologien, wie auch zur Übertragung digitaler Bilder.

Mit dem Internet, welches 1969 durch die Bereitstellung von ARPANET (Advanced Research Projects Agency-Net) und dem ersten *Interface Message Processor (IMP)* "geboren" wurde, war der Zugang auch für die allgemeine Nutzung von digitaler Kommunikation gelegt [33, 25f.]. Die digitale Kamera entwickelte sich ebenfalls in den 60er Jahren. Die rasante Entwicklung der Bildverarbeitung, des Internets und der leistungsfähigeren, erschwinglichen Rechner übernimmt in der Industriegesellschaft sehr viele Aufgaben, wie unten stehend dargestellt wird. Im Gegensatz zu der Ausführung durch einen Menschen sind die automatisierten Prozesse nicht fehleranfällig, ermüdend oder subjektiv [14].

2.2.2 Begriffsbestimmungen

Bildverarbeitung setzt sich aus mehreren Teilbereichen wie der Bildbearbeitung, Bildauswertung und Bildkompression zusammen und wird in Abhängigkeit des Kontextes verwendet [14, S.2].

Sie findet Anwendung in den unterschiedlichsten Bereichen: angefangen in der Automobil-, Elektro-, Nahrungs-, und Genussmittelindustrie zur Produktionskontrolle und Qualitätssicherung. Weiterhin wird im Verkehr die Bildverarbeitung genutzt, um z. B. Kennzeichen von Fahrzeugen zu erfassen oder an Flughäfen Sicherheitskontrollen durchzuführen. Banken (automatisches Lesen von Belegen), Internet (Archivierung, Bildanalyse), Telekommunikation, Fernsehen, Multimedia-Anwendungen, Kriminologie (Fingerabdruck, Gesichtserkennung), Kartographie (Identifikation natürlicher Erdformationen), Ökologie (Verzeichnen von Umweltschäden aus Satellitenbildern), Medizin (Tomographie, Röntgenbilderauswertung), Astronomie belegen die vielfältige Nutzung [14, S.4f.].

Nachfolgend werden grundlegende Begriffe im Kontext der Bildverarbeitung geklärt.

- **Bildverstehen** Es ist ein Prozess in mehreren Schichten (siehe Darstellung in Schichtenmodell), wobei ein Bild rechnergestützt vorverarbeitet, erkannt (identifiziert) und letztendlich verstanden (interpretiert) wird [27, S.21]. Das Verstehen im letzten Schritt ist in der Bildverarbeitung eine eindeutige Zuordnung zu einem Objekt wie Tisch, Haus oder Ähnliches.
- **Bildobjekt** Ein Objekt auf einem Bild wird als ein Gegenstand angesehen, welches strukturiert sein kann.
- **Bildstruktur** Sie wird durch Bildprimitive (Menge an Bildpunkten) oder Bildkomponenten gebildet.
- Merkmal Bildmerkmale stellen charakteristische Eigenschaften einer Menge von Bildpunkten, Bildprimitiven, Bildkomponenten oder Bildstrukturen dar. Jede Stufe des Bildverstehens hat ihre eigenen Merkmale wie statistische Mittelwerte, Schwerpunkte, Formmerkmale. Dadurch können Bildelemente identifiziert werden. Merkmalsextraktion entspricht einer Messung.
- Schichtenmodell Bildverstehen wird mit Modellen veranschaulicht. Levi stellt es in einem Vier-Schichten-Modell dar [27, S.27ff.], wobei bis zum Bildverstehen alle Stufen durchlaufen werden:

Bildvorverarbeitung Das Eingangsbild wird digitalisiert und kann danach entzerrt, komprimiert, restauriert, rekonstruiert und normiert werden.

₩

Komponentenerkennung Extrahierte Bildkomponenten nach der Bildvorverarbeitung ergeben ein Komponentenbild. Dazu gehört als Teilbereich die Segmentierung, die vor der Komponentenerkennung durchgeführt wird. Dabei werden Bildprimitive durch klassifizierte Bildpunkte (Pixel) gebildet.

↓

Objekterkennung Aus den Komponenten der zweiten Schicht sollen Objekte erkannt und eindeutig bezeichnet werden. Hier beginnt man mit dem Bildverstehen.

∜

Szenenerkennung Objekte aus der dritten Schicht werden gruppiert bzw. zueinander in Beziehung gebracht und dadurch mehr vom Bild erkannt und verstanden.

Operationen Allgemein formuliert transformiert ein Operator \mathcal{H} ein Bild \mathbf{G} zu \mathbf{G}' [22, S.113f.] mit $\mathbf{G}' = \mathcal{H}\mathbf{G}$.

Sie werden in zwei unterschiedliche Klassen geteilt: zum einen die Extrakationsoperationen, bei denen gemessen wird, zum anderen die Vergleichsoperationen, die für die Auswertung der Messdaten verwendet werden.

Sie können linear oder nichtlinear sein. Bei letzteren ist die Reihenfolge der Ausführung relevant [27, S.23].

Numerische Differentitation [32, S.81f.] Lineare Operatoren – wie Masken, unter *Be-griffsbestimmungen* beschrieben – werden für die näherungsweise Differentiation erstellt. Es wird die eindimensionale Ableitung der Bildfunktion I(x) verwendet:

$$I'(x) = \lim_{h \to 0} \frac{I(x+h) - I(x-h)}{2h}$$
(2.3)

Es wird h = 1 gesetzt (Bildpunkt-Abstand zu 1) für $x \in \mathbb{Z}$. Eine Näherung $\tilde{I}(x)$ für I'(x) ist demnach:

$$\tilde{I}(x) = \frac{1}{2}(I(x+1) - I(x-1))$$
(2.4)

Die Abbildung 2.4 stellt den Zusammenhang zwischen I'(x) und $\tilde{I}(x)$ dar:



- Abb. 2.4: Zur Annäherung der Ableitung $I'(x_2)$ einer abgetasteten Funktion. Für die Näherung $\tilde{I}(x_2)$ werden die Nachbarn x_1 und x_3 herangezogen und ist unabhängig von x_2 (nach [32, S.81]).
- Filtern Aus mathematischer Sicht findet beim Filtern eine Faltung statt [14, S.144], welche in der Abbildung 2.5 dargestellt wird. Die Filtermasken werden durch die Diskretisierung der partiellen Ableitungen errechnet. In Kantendetektion wird die Verwendung des Differenzierens näher betrachtet.



Abb. 2.5: Skizze einer Faltung¹

¹Die Skizze ist erstellt mit *Inkscape* nach [14, S.145]. Ist g(x, y) die diskrete Bildfunktion des Eingangsbildes, so entsteht mit dem Filterkern 3×3 ein geglättetes Bild g'(x,y) durch die Mittelung des Bildpunktes an der Stelle (x,y) mit seinen acht Nachbarn [14, S.145].

Die Faltung eines Eingangsbildes mit der diskreten Bildfunktion g(x,y) mit einem Filterkern h(x,y) der Größe 3×3 wird folgendermaßen definiert [14, 153]:

$$g'(x,y) = g(x,y) * h(x,y) = \sum_{v,u=-1}^{1} \sum_{u=-1}^{1} g(x-u,y-v)h(u,v)$$
(2.5)

Unter unterschiedlichen Dimensionen für den Filterkern wird im Folgenden ein 2D-Filter der Größe 3×3 ausgewählt [22, S.3124ff.]. Der Filterkern h(x,y) variiert dabei je nach Filterart. Man unterscheidet allgemein unter Tief- und Hochpassfilter. Tiefpassfilter sind durchlässig für tiefe, Hochpassfilter für hohe Frequenzen [14, S.153].

Die Segmentierung der Komponentenerkennung ist für die vorliegende Anwendung ein relevanter Aspekt und wird nachstehend näher betrachtet.

2.2.3 Segmentierung

In dieser Teilschicht werden Bildpunkte klassifiziert und dadurch Bildprimitive gebildet. Das Bild wird dabei in Bereiche geteilt [27, S.23].

Beim Segmentieren werden Merkmale aus dem vorverarbeiteten Bild extrahiert und anschließend Bildkomponenten durch die Messdatenanalyse erkannt.

In der vorliegender Arbeit werden pixelorientierte, kantenorientierte und modellbasierte Verfahren vorgestellt.

Pixelbasierter Ansatz Er stellt die einfachste Herangehensweise dar, um ein Bild zu segmentieren.

Thresholding ist ein Schwellwertverfahren, welches sich besonders gut für Bilder mit kontrastreichem Hintergrund eignet.

Beim globalen Thresholding wird ein konstanter Schwellwert \mathcal{T} definiert und kann vor allem für Detektion von Objekträndern verwendet werden. Alle Bildpunkte mit den Grauwerten unter \mathcal{T} fallen aus dem Objekt [6, S.452].

In adaptiven Verfahren werden Bilder berücksichtigt, die nicht kontrastreich sind. Durch die Analyse von Histogrammen können regional unterschiedliche Schwellwerte bestimmt werden, um Objekte zu erkennen [6, S.452ff.].



Tabelle 2.1: Eingangsbild g(x, y) wird mit dem Schwellwert \mathcal{T} verarbeitet.

Kantendetektion Eine Kante wird definiert als eine *Diskontinuität der räumlichen Grauwertfunktion* $g(\mathbf{x})$ *der Bildebene.* Die Kantendetektion basiert auf Ableitungen [22, S.345f.].

Erste Ableitung Mit der partiellen Ableitung erster Ordnung werden die lokalen Maxima einer Bildfunktion f(x, y) über zwei Variablen x und y gebildet, die eine Kante repräsentieren [28, S.108f.].

$$\nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right)$$
(2.6)

Daraus ergeben sich für die partiellen Ableitungen erster Ordnung [14, S.154]:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x,y)}{\Delta x}$$
(2.7)

$$\frac{\partial f(x,y)}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y, y) - f(x,y)}{\Delta y}$$
(2.8)

Die Diskretisierung mit Δx und $\Delta y = 1$ ergeben beispielhaft [14, S.155]:

$$\frac{\partial f(x,y)}{\partial x} = f(x,y) - f(x-1,y) \tag{2.9}$$

$$\frac{\partial f(x,y)}{\partial y} = f(x,y) - f(x,y-1) \tag{2.10}$$

In *Tabelle 2.2* sieht man an den Plots, wie die erste Ableitung an den Kanten Extremwerte zeigt. Sie ist rauschanfällig und nicht optimal für die Kantendetektion [14, S.155f.].

Differenzoperator: Es werden die partiellen Ableitungen erster Ordnung gebildet. Ein Beispiel für eine Faltungsmaske der Größe 3×3 zum Filtern der vertikalen Kanten (erste Ableitung nach x) [14, S.155f.]:

0	0	0
-1	1	0
0	0	0

Zur partiellen Ableitung erster Ordnung nach y mit Filtern der horizontalen Kanten wird beispielsweise folgende Maske verwendet [14, S.155]:

0	-1	0
0	1	1
0	0	0



Tabelle 2.2: Ein Objekt und seine ersten Ableitungen

Zweite Ableitung Mit der zweiten Ableitung I''(x) einer eindimensionalen Bildfunktion I(x) können stufenförmige Kanten als Wendepunkte lokalisiert werden. Weiterhin gilt I''(x) = 0, dritte Ableitung $I'''(x) \neq 0$ [32, S.80].

> Diese Eigenschaften lassen sich auf Zweidimensionalität übertragen. Dabei werden Richtungsableitungen senkrecht zur Kantenorientierung betrachtet. In diesem Fall stellt die Nullstelle der Ableitung zweiter Ordnung den Punkt der größten Steigung dar [32, S.92]. Aus diesem Grund existieren Filtermasken für jede Richtung.



Tabelle 2.3: Beispiel der Anwendung des Laplacefilters²

Regularisierte Kantendetektion Sie zeichnet sich durch eine Kombination von Differenzierung in eine Richtung und Glättung in alle anderen Direktionen aus. Optimale Kantendetektoren müssen den Frequenzbereich berücksichtigen, vor allem bei Signalen höherer Dimensionen zahlt sich eine Glättung aus [22, S.364].

Zuerst werden Tiefpassfilter beschrieben, anschließend folgt eine Darstellung von Hochpassfiltern.

Tiefpassfilter Sie glätten ein Bild, wobei Rauschen, Grauwertkanten und -spitzen im hochfrequenten Bereich entfernt werden. In der *Tabelle 2.4* werden einige Beispiele dazu gelistet.

Mittelwertfilter: Sie stellt die einfachste Form eines Glättungsfilters dar. Je nach Größe des Filternkerns ist der Koeffizient unterschiedlich groß. Bei einem Kern der Dimension 3×3 beträgt er $\frac{1}{9}$ [14, nach S.145].

g(x,y)





Gaussfilter: In der Mitte des linearen und verschiebungsinvarianten [27, S.112] Filterkerns wird der Wert des Bildpunktes mehrfach gewichtet. Die Werte der Filtermaske wird abhängig von der Größe $m \times m$ über die Binomialkoeffizienten eines Binoms der Ordnung m - 1 berechnet. Durch ihre Verwendung ergibt sich eine Verteilungsfunktion, wenn die Binomialkoeffizienten als Häufigkeiten betrachtet werden [14, S.149ff.].

Dem Gaussfilter liegt die Impulsantwort zu Grunde, welches durch die erste Ableitung der Sprungantwort nach der Zeit definiert wird. Sie stellt eine Funktion dar und enthält den Abstand σ vom Symmetriezentrum dieser Glockenkurve. Als Tiefpassfilter ist der Gaussfilter nahezu optimal [28, S.111ff.].

²Die mit Octave durchgeführte Filterung verwendet zur besseren Sichtbarkeit der Unterschiede colormap(pink). Dies findet bei mehreren Beispielen Anwendung.



Tabelle 2.4: Beispiele von Tiefpassfiltern

Hochpassfilter Die Gradientenfilter nach Sobel, Prewitt, Roberts, die auf den ersten Ableitungen beruhen, und der Laplaceoperator als Differenzenoperator zweiter Ordnung sind Beispiele für Hochpassfilter [14, S.154ff.].

Der **Gradient** ist ein Differentialoperator und stellt mit der Bildung der ersten Ableitung die lokalen Maxima dar, die eine Kante repräsentieren. Er ist richtungsabhängig und linear [28, S.108f.]. Viele Kantenberechnungen verwenden den Gradienten, da sie weniger empfindlich gegenüber Rauschen sind und oft die Richtung benötigt wird [28, S.110]. Allgemein entspricht Rauschen Störungen, die nicht zum eigentlichen Bild gehören.

In Tabelle 2.5 wird eine Auswahl an Hochpassfiltern vorgestellt. Die Filterkerne beinhalten eine Diskretisierung der ersten Ableitungen und eine Mittelung. Für die Darstellung sind jeweils zwei von mehreren möglichen Filterkernen verwendet worden. Als Eingangsbild g(x, y) wird Abbildung 2.6 benutzt. Jeder vorgestellte Filter wird folgendermaßen verarbeitet: Zuerst wird ein Filterkern $h_x(x, y)$ mit Ableitungsrichtung x und Glättung auf das Eingangsbild g(x, y), analog dazu $h_y(x, y)$ in Richtung y auf das originale Bild g(x, y) angewendet. Aus beiden Zwischenresultaten $g'_x(x, y)$ und $g'_y(x, y)$ wird das Ergebnisbild g'(x, y) erzeugt.



Abb. 2.6: Eingangsbild g(x, y) für die Filter in Tabelle 2.5



Diagonaler Filter 1	Diagonaler Filter 2	Ergebnisbild $g'(x, y)$
g(x,y)	g(x,y)	
\Downarrow	\Downarrow	
0 0 0	0 0 0	ALL BANK
0 -1 0	-1 0 0	
1 0 0	0 1 0	
$h_1(x,y)$	$h_2(x,y)$	
\Downarrow	\Downarrow	
$g_1'(x,y)$	$g_2'(x,y)$	

Prewittfilter: Die Filterkerne des Prewitt-Operators bewirken eine Ableitung in eine Richtung (hier x, y-Richtung). Vertikal dazu erfolgt die Mittelung durch ein Mittelwertfilter [14, S.157].

Filter Richtung x	Filter Richtung y	Ergebnisbild $g'(x, y)$
g(x,y) IL	g(x,y) IL	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	
$\frac{1}{h_x(x,y)}$	$ \begin{array}{c c} & & \\ \hline & & \\ & & \\ & & \\ & & \\ & & \\ & \\$	Contraction of the second s
$q'_x(x,y)$	$q'_u(x,y)$	

Sobelfilter: Wie mit dem Prewittfilter löst der Sobelfilter die partielle Ableitung in eine Richtung (hier x, y-Richtung) aus. Für die Mittelung orthogonal dazu wird ein Gaussfilter verwendet [14, S.157] [22, S.365].



Laplace of Gauss-Filter (LoG): Da Laplacefilter rauschanfällig sind, wird zuerst das Eingangsbild g(x, y) mit einem Gaussfilter geglättet und anschließend der Laplacefilter verwendet [14, S.368] [28, S.111ff.]. Die Filterkernwerte sind auf zwei Kommastellen gerundet und Octave entnommen.

Filter Richtung x	Filter Richtung y	Ergebnisbild $g'(x, y)$
$\begin{array}{c} g(x,y) \\ \Downarrow \\ 0.17 & 0.43 & 0.17 \\ \hline 0.43 & -3.15 & 0.43 \\ \hline 0.17 & 0.43 & 0.17 \\ \hline h_x(x,y) \\ \Downarrow \\ g'_x(x,y) \end{array}$	$g(x, y)$ \downarrow $h_y(x, y)$ $=$ $h_x(x, y)$ \downarrow $g'_y(x, y)$	

Tabelle 2.5: Beispiele von Hochpassfiltern

Canny-Operator [4, S.103ff.] [5] Die regularisierten Kantendetektoren haben zum Nachteil, dass sie nicht zuverlässig bzw. fälschlicherweise Kanten erkennen, vor allem bei verrauschten Bildern [46, S.1265]. Der Canny-Kantendetektor versucht reale Kanten genau zu lokalisieren. Grundlage bietet auch hier der Gradient mit Anwendung eines Gaußfilters, isoliert aber im nächsten Schritt die lokalen Maxima. Anschließend erfolgt mit der Definition von zwei unterschiedlichen Schwellwerten \mathcal{T}_l und \mathcal{T}_h eine Bündelung von Kanten-Bildpunkten (Hysterese-Schwellwertverfahren). Abbildung 2.7 demonstriert zwei Beispiele.



(a) $\mathcal{T}_l = 0.0147$ und $\mathcal{T}_h = 0.0368$, Va- (b) $\mathcal{T}_l = 0.03$ und $\mathcal{T}_h = 0.075$, $\sigma = 8$ rianz des Gaussfilters $\sigma = \sqrt{2}$

Abb. 2.7: Canny-Kantendetektion mit dem Eingangsbild Abbildung 2.6

Modellbasierte Verfahren [6, S.467f.] [17, S.799ff.] Die Hough-Transformation ist ein Beispiel für ein modellbasiertes Verfahren, welche zu den Vergleichsoperatoren zählt. Damit können die Grenzen von Objekten erkannt werden.

Im 2D-Koordinatensystem kann eine Gerade als y = mx + b im Polarkoordinatenraum als $\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$ definiert werden.

Das Polarkoordinaten-System wird mit den Parametern ρ und θ konstruiert. Dabei werden alle Geraden im x, y-Raum zu einem Punkt im ρ, θ -Raum abgebildet, siehe Abbildung 2.8. Der Vektor (θ, ρ) ist orthogonal zur Geraden durch den Ursprung. Jeder Punkt $x_i, y_i \in g, i \in \mathbb{Z}$ wird in einer sinusförmigen Kurve im Parameterraum wiedergegeben.



Abb. 2.8: Skizze einer Hough-Transformation am Beispiel einer Geraden

In Abbildung 2.9 ist ein Beispiel einer Hough-Transformation dargestellt.



Abb. 2.9: Durchführung einer Hough-Transformation (erstellt mit GNU Octave)

Nach der Durchführung der Hough-Transformation wird das Histogramm des Parameterraums (θ, ρ) nach den lokalen Maxima abgesucht. Dadurch werden die linearen Grenzen dieses Segments definiert.

2.3 Bildbearbeitung im Frequenzraum

Bei einer Verarbeitung in der Frequenzdomäne wird das Bild transformiert, wobei die Lokalisierung von Bildpunkten nicht direkt zugänglich ist.

Die menschliche Sehkraft ist empfindlicher gegenüber niederfrequenten Koeffizienten, also wahrnehmungssignifikant im Gegensatz zu hochfrequenten. Tieffrequente Koeffizienten können in diesem Fall eine Distorsion des Original-Bildes bedeuten [2, S.1321].

Unter den zahlreichen Verfahren wird anhand der Fourier-Transformation die Bildbearbeitung im Frequenzraum dargestellt. Nach kurzem historischen Abriss werden zur Fourier-Analyse wesentliche Konzepte wie Lebesguesche Räume, Fourier-Transformation, die Diskrete Fourier-Transformation und Faltung beschrieben.

2.3.1 Historische Ursprünge

Das Filtern in der Bildverarbeitung hat in der Frequenzdomäne relevante Impulse auf dem Gebiet der Wärmeausbreitung und -diffusion erhalten. Der Mathematiker Jean Baptiste Joseph Fourier, geboren 1768 in der französischen Stadt Auxerre, legte dazu bereits 1807 die Grundsteine und veröffentlichte sie 1822 in seinem Buch *La Théorie Analitique de la Chaleur*. Seine Idee war, für den Wärmefluss spezielle Differentialgleichungen zu formulieren. Mit der schnellen Fourier-Transformation (Fast Fourier Transform (FFT)) aus den frühen 1960er Jahren entwickelten sich bedeutende Singalverarbeitungs-Technologien im Bereich von medizinischen Monitoren und Scannern bis hin zu moderner elektronischer Kommunikation.

Fourier sagt, dass jede periodische Funktion als eine Summe von Sinus und/oder Cosinus von unterschiedlichen Frequenzen ausgedrückt werden kann, welche mit verschiedenen Koeffizienten multipliziert wird, bekannt heutzutage als die Fourierreihe [17, S.250ff.].

2.3.2 Mathematische Grundkonzepte

Lebesgue-Räume

Eine Grundvoraussetzung für die Lebesgue-Integrierbarkeit ist die Messbarkeit beliebiger Mengen [45, S.3ff.].

Definition 2.1 ((Lebesgue'sche) L^p -Norm). [45, S.8] L^p -Normen $\|\cdot\|_p$ für $1 \le p \le \infty$ sind für meßbare Funktionen $f : \mathbb{R}^n \to \mathbb{C}$ definiert durch

$$||f||_p = \begin{cases} \sqrt[p]{\int_{\mathbb{R}^n} |f(t)|^p \, dt}, & \text{falls } 1 \le p < \infty;\\ \text{ess sup}|f| := \inf\{r : \alpha_u(r) = 0\}, & \text{falls } p = \infty. \end{cases}$$

Definition 2.2 (Lebesgue'sche Raum $L^p(\mathbb{R}^n)$). [45, S.8] [15, S.173] Die Menge der entsprechenden Äquivalenzklassen von meßbaren Funktionen mit $||f||_p < \infty$ wird Lebesgue-Raum $L^p(\mathbb{R}^n)$ genannt: $L^p(\mathbb{R}^n) = \{f : \mathbb{R}^n \to \mathbb{C} \mid f \text{ meßbar und } ||f||_p < \infty\}$

Hilbert-Raum

Für die Fourier-Analyse wird ein besonderer Lebesque'scher Raum, der Hilbert-Raum L^2 benötigt. Der Hilbert-Raum ist eine direkte Generalisierung von endlich dimensionalen Euklidschen Räumen [15, S.163f.].

Definition 2.3 (Hilbert-Raum). [15, S.163f.] Sei *H* ist ein komplexer Vektorraum. Das Skalarprodukt (oder inneres Produkt) von *H* ist eine Abbildung $H \times H \to \mathbb{C}$, $(x, y) \to \langle x, y \rangle$ mit der Norm:

$$||x|| = \sqrt{\langle x, x \rangle} \text{ für } x \in H.$$
(2.11)

Dieser Vektorraum H heißt Hilbert-Raum, wenn er bezüglich dieser Norm vollständig ist.

 $L^2(\mathbb{R}^n)$ ist ein Hilbert-Raum. Die Norm und das innere Produkt werden über das Lebesgue-Integral definiert [45, S.8]:

$$\|\cdot\|_{2}: L^{2}(\mathbb{R}^{n}) \to \mathbb{R}_{\geq 0}, \, \|f\|_{L^{2}(\mathbb{R}^{n})} := \sqrt{\int_{\mathbb{R}^{n}} |f(x)|^{2} \, dx} < \infty;$$
(2.12)

$$\langle \cdot, \cdot \rangle : L^2(\mathbb{R}) \times L^2(\mathbb{R}) \to \mathbb{C}, \ \langle f, g \rangle_{L^2(\mathbb{R}^n)} := \int_{\mathbb{R}^n} \overline{f(x)} g(x) \, dx$$
 (2.13)

Für den Umgang im Zahlensystem \mathbb{Z} werden quadratisch-summierbare Folgen genutzt. Sie definieren den Hilbert-Raum $\ell^2 = \ell^2(\mathbb{Z})$ mit nachstehender Norm und innerem Produkt [45, S.8] [15, S.169]:

$$\|\cdot\|_{2}: \ell^{2}(\mathbb{Z}) \to \mathbb{R}_{\geq 0}, \|f\|_{\ell^{2}}:=\sqrt{\sum_{n=-\infty}^{\infty} |f(n)|^{2}} < \infty;$$
 (2.14)

$$\langle \cdot, \cdot \rangle : \ell^2(\mathbb{Z}) \times \ell^2(\mathbb{Z}) \to \mathbb{C}, \ \langle f, g \rangle_{\ell^2} := \sum_{n = -\infty}^{\infty} \overline{f(n)} g(n).$$
 (2.15)

Fouriertransformation

Definition 2.4 (Kontinuierliche Fourier-Transformation). [17, S.277] Sei $f \in L^2(\mathbb{R}^2)$. Die Fourier-Transformation wird wie folgt berechnet:

$$\hat{\cdot}: L^2(\mathbb{R}^2) \to L^2(\mathbb{R}^2), \ \hat{f}(\mu, \nu) = \int_{\mathbb{R}} \int_{\mathbb{R}} f(x, y) e^{-2\pi i (\mu x + \nu y)} dx \ dy$$
(2.16)

Die Inverse Transformation wird definiert mit:

$$\vdots: L^2(\mathbb{R}) \to L^2(\mathbb{R}), \ \check{f}(x,y) = \int_{\mathbb{R}} \int_{\mathbb{R}} f(\mu,\nu) e^{2\pi i(\mu x + \nu y)} d\mu \ d\nu$$
 (2.17)

Definition 2.5 (Diskrete Fourier-Transformation (DFT)). [17, S.286] Sei $v \in \mathbb{C}^{M \times N}$, für $M, N \in \mathbb{N}$. Bei einer $M \times N$ Matrix wird die Diskrete Fourier-Transformation im 2D-Raum definiert als:

$$\hat{\cdot}: \mathbb{R}^{M \times N} \to \mathbb{C}^{M \times N}, \ \hat{v}(\mu, \nu) := \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} v(x, y) e^{-2\pi i \left(\frac{\mu x}{M} + \frac{\nu y}{N}\right)}.$$
(2.18)

Die Inverse Diskrete Fourier-Transformation (IDFT) [17, S.286] lautet:

$$\vdots: \mathbb{C}^{M \times N} \to \mathbb{C}^{M \times N}, \ \check{v}(x, y) := \frac{1}{MN} \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} v(\mu, \nu) e^{2\pi i \left(\frac{\mu x}{M} + \frac{\nu y}{N}\right)}.$$
 (2.19)

Bemerkung 2.6 (Schnelle Fourier-Transformation (FFT)). [45, S.65ff.] Die FFT (Fast Fourier Transform) ist ein Algorithmus zur effizienten Berechnung der Diskreten Fourier-Transformation.

In Abbildung 2.10 wird eine FFT dargestellt.



(a) Eingangsbild f(x, y)

(b) Fouriertransformierte $\hat{f}(x, y)$

Abb. 2.10: Beispiel einer FFT

Faltung

Faltungen können durch die Multiplikation der Fourier-Transformierten berechnet werden.

Folgen mit endlichen Trägern werden ähnlich behandelt, die Ränder bedürfen dabei besonderer Beachtung [45, S.26f.].

Definition 2.7 (**Diskrete Faltung im 2D-Raum**). [17, S.299f.] Für zwei Funktionen f und h lautet die diskrete Faltung:

$$(f*h)(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m,y-n),$$
(2.20)

für x = 0, 1, 2, ..., M - 1 und y = 0, 1, 2, ..., N - 1.

2.3.3 Filterung

Unter einer Filterung im Frequenzraum ist ein Prozess mit folgendem Ablauf zu verstehen: eine Fourier-Transformation eines Eingangsbildes f(x, y), eine Modifizierung der Transformierten mit einer Transferfunktion $\hat{h}(\mu, \nu)$ und die Rekonstruktion mit der inversen Fourier-Transformation.

Definition 2.8 (Filterung). [17, S.307] Eine Bildfunktion g(x, y) wird mit einer Funktion h gefiltert. Es findet dabei eine Faltung der Fourier-Transformierten statt.

$$g: \mathbb{C} \times \mathbb{C} \to \mathbb{C} \times \mathbb{C}, \ g = \operatorname{Re}(\check{}(\hat{h} \cdot \hat{f})).$$
(2.21)

Niedrige Frequenzen in der Transformation hängen mit langsam variierenden Intensitätskomponenten in einem Bild zusammen. Im Gegensatz dazu verursachen hohe Frequenzen scharfe Intensitätsübergänge wie Kanten und Rauschen [17, S.308].

Am Beispiel von $Abbildung \ 2.6$ wird ein Gaußscher Tiefpass- und Hochpassfilter dargestellt.

Die Transferfunktion $\hat{h}(\mu, \nu)$ des Tiefpassfilters beträgt [17, S.323]:

$$\hat{h}_{TP}(\mu,\nu) = e^{-D^2(\mu,\nu)/2\sigma^2}.$$
(2.22)

 $D(\mu, \nu)$ repräsentiert dabei die Distanz vom Zentrum eines $M \times N$ großen Rechtecks zu jedem Punkt (μ, ν) , σ den Abstand vom Zentrums mit $\sigma \in \mathbb{R}_{\geq 0}$. Der Filter hat dabei die identische Größe wie das zu filternde Bild. $\hat{h}_{TP}(\mu, \nu)$ und der Filter sind in Abbildung 2.11 illustriert.



Abb. 2.11: Gaußsche Transferfunktion (a) und Filter (Tiefpass) (b)

Die Faltungsoperation mit dem Tiefpassfilter ergibt eine Glättung des Bildes, siehe Abbildung 2.12.



Abb. 2.12: IDFT der Gaußschen Tiefpassfilterung

Korresponierend dazu lässt sich ein Hochpassfilter in der Frequenzdomäne berechnen [17, S.330] und ist in *Abbildung 2.13* dargestellt:

$$\hat{h}_{HP}(\mu,\nu) = 1 - \hat{h}_{TP}(\mu,\nu).$$
(2.23)



Abb. 2.13: Gaußsche Transferfunktion (a) und Filter (Hochpass) (b)

Mit dem Hochpassfilter werden die Kanten des Bildes detektiert, siehe Abbildung 2.14.



Abb. 2.14: IDFT der Gaußschen Hochpassfilterung

2.4 Bildbearbeitung im Zeit-Frequenzraum

Werden im Frequenzraum lokale Informationen zusätzlich zu den Frequenzinformationen benötigt, so sind diese in der Phase versteckt und nicht zugänglich. Im Zeit-Frequenz-Raum bieten Gefensterte FourierTransformation (WFT)³ und Wavelets eine Vielzahl von Möglichkeiten, Signale im \mathbb{R}^n -Bereich $(n \in \mathbb{N})$ zu transformieren, zu analysieren und zu bearbeiten. Sie ergeben gruppentheoretisch das Gleiche und werden über dieselbe Konstruktion durch Gruppen eingeführt [30, S.32].

Definition 2.9 (Gefensterte Fourier-Transformation (WFT)). Sei $f \in L^2(\mathbb{R}^2)$. Die WFT ist folgendermaßen definiert [30, S.32]:

 $^{^3\}mathrm{Nach}$ der englischen Abkürzung WFT für "Windowed Fourier Transform".

$$\mathcal{F}_{\psi}f(p,q) = \left(\frac{1}{2\pi}\right)^{\frac{n}{2}} \int_{\mathbb{R}^n} f(x)e^{-i\langle q,x\rangle}\psi(x-p)dx \tag{2.24}$$

Die optimale Fensterfunktion dazu lautet [30, S.37]:

$$\psi(x) = \pi^{-1/4} e^{-x^2/2}.$$
(2.25)

Allerdings ist die Auflösung durch die Fensterfunktion sowohl im Zeit- als auch im Frequenzbereich eingeschränkt [30, S.37]. Die Wavelets bieten im Gegensatz dazu mit zunehmender Frequenz eine bessere Zeitauflösung [30, S.38].

Dieses Kapitel stellt nach einem kurzen geschichtlichen Abriss in 2.4.1 Historische Entwicklung die Wavelets allgemein vor (siehe 2.4.2 Wavelets) und fasst die Möglichkeiten der Analyse in 2.4.3 Filterung zusammen. Es werden lediglich wesentliche Konzepte dargestellt [30], [34], [11], [31].

2.4.1 Historische Entwicklung

Die Zeit-Frequenz-Transformationen mit den mathematischen Konzepten ist ein eher junges Forschungsgebiet. Die mathematischen Grundkonzepte sind auf Fourier zurückzuführen. Es entwickelte sich das Bestreben weg von der frequenzbasierten hin zu einer skalenbasierten Analyse von Signalen, welche weniger empfindlich auf Rauschen reagiert. 1909 hat Alfred Haar das erste Wavelet entwickelt, aber noch nicht unter dieser Bezeichnung [35, S.35].

Dennis Gábor hat 1946 die Gefensterte Fourier-Transformation eingeführt und dadurch eine Lokalisierung im Frequenzraum ermöglicht [30, S.32].

Der Begriff der Wavelettransformation mit dem Konzept der Zeit-Frequenz-Lokalisierung entstand in den frühen 80ern durch Alex Grossmann und Jean Morlet. Die Reproduktion der Wavelettransformation wurde durch Y. Meyer mit der Fourieranalysis im Caldorónischen Lemma in Zusammenhang gebracht. Ein Fortschritt in der Diskretisierung von Wavelets gelang S. Mallat und Y. Meyer durch die Einführung der Multiskalenanalyse. Ingrid Daubechies (geboren 1954) konstruierte anschließend orthogonale Wavelets mit kompaktem Träger [40, S.13f.].

Die Anwendungen sind vielfältig: Sie reichen von EKG-Analyse und Qualitätsbeurteilung von Gewebe (wie z. B. menschlichem) bis hin zu Datenkompression und Analyse bzw. Verfahren bei Randwertproblemen [30, S.6f.].

2.4.2 Wavelets

Dieser Teil zeigt wesentliche Konzepte von Wavelets und Wavelet-Transformation. Um darzustellen, dass die Konstruktion der Fourier-Transformation (siehe *Gleichung 2.4*) ähnlich ist, wird zunächst die Kontinuierliche Wavelet Transformation $(CTW)^4$ definiert. Die Diskretisierung wird über die Multiskalenanalyse hergeleitet, dessen wichtige Grundzüge aufgezeigt werden. Anschließend folgt zusammenfassend das Prinzip der diskreten Wavelet-Transformation und dessen Rekonstruktion.

⁴Die Abkürzung CWT orientiert sich am englischen Begriff *Continuous Wavelet Transform*.

Definition 2.10 (Wavelet). [30, S.18] Eine Funktion $\psi \in L^2(\mathbb{R})$ heißt Wavelet, wenn die Zulässigkeitsbedingung

$$0 < c_{\psi} := 2\pi \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty$$
(2.26)

erfüllt ist.

Ein Signal $f \in L^2(\mathbb{R})$ wird mit Wavelets ψ analysiert. Es werden dabei die L^2 -Skalarprodukte mit dem translatierten Parameter t und dem dilatierten s von ψ gebildet.

Definition 2.11 (Kontinuierliche Wavelet-Transformation (CWT)). [30, S.18] Sei $f \in L^2(\mathbb{R})$. Die Kontinuierliche Wavelet-Transformation $\tilde{f}(s,t)$ wird definiert als

$$\tilde{f}(s,t) := |s|^{-1/2} \int_{\mathbb{R}} f(u)\psi\left(\frac{u-t}{s}\right) du.$$
(2.27)

 ψ stellt da
bei das Mutterwavelet dar. tverschiebt das Wavelet und zum Zeitpunk
tu=tsind in $\tilde{f}(s,t)$ die lokalen Zeit
informationen enthalten.

Die Wavelet-Transformation misst die Korrelation zwischen einem Signal und einem Wavelet. In der Zeit-Frequenz-Domäne wird dabei auch der Ort berücksichtigt. Eine wichtige Eigenschaft der Wavelet-Transformation ist die Unschärferelation.

 $\{(t,\omega)|t,\omega \in \mathbb{R}\}$ gibt eine Menge von Paaren an. Sei $g_{t_0\omega_0}$ eine Funktion, die sich im Zeitbereich um t_0 und deren Fourier-Transformierte um ω_0 konzentriert [30, S.30].

Satz 2.12 (Heisenbergsche Unschärferelation). [30, S.31] Sei $g \in L^2(\mathbb{R}), ||g||_{L^2} = 1$. Für alle $t_0, \omega_0 \in \mathbb{R}$ gilt:

$$\int_{\mathbb{R}} (t - t_0)^2 |g(t)|^2 dt \int_{\mathbb{R}} (\omega - \omega_0)^2 |\hat{g}(\omega)|^2 d\omega \ge \frac{1}{4}.$$
(2.28)

Die Heisenbergsche Unschärferelation besagt, dass t_0 und ω_0 nicht mit beliebiger Genauigkeit gemessen werden können. Das heißt, dass wenn bessere Auflösung von ω_0 angestrebt wird, die Genauigkeit von t_0 leidet und umgekehrt. Bedeutend ist dieser Aspekt bei der Auswahl eines spezifischen Wavelets, um ein Signal f in der Zeit-Frequenz-Domäne zu analysieren. Die Unschärferelation gilt ebenso für Wavelet-Transformationen [7].

Multiskalen-Analyse

In der Darstellung der Multiskalen-Analyse als Grundlage für eine Diskretisierung erfolgt eine Einschränkung auf den mehr- bzw. in den zweidimensionalen Raum, welche zur Bildanalyse benötigt wird⁵.

Definition 2.13 (Multiskalen-Analyse (MRA)⁶). [30, S.128f.] Eine Multiskalen-Analyse des $L^2(\mathbb{R}^n)$ setzt sich aus einer aufsteigenden Folge abgeschlossener Unterräume $\{V_m\}_{m\in\mathbb{Z}} \subset L^2(\mathbb{R}^n)$ mit

⁵Es existieren Unterschiede zwischen ein- und mehrdimensionaler Multiskalen-Analyse, auf die hier nicht näher eingegangen wird.

⁶Die Abkürzung MRA orientiert sich an der englischen Bezeichnung Multiresolution Analysis.

2.4 Bildbearbeitung im Zeit-Frequenzraum

$$\{0\}... \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset ... \subset L^2(\mathbb{R}^n)$$

zusammen, sodass gelten:

$$\overline{\bigcup_{m\in\mathbb{Z}}V_m} = L^2(\mathbb{R}^n) \tag{2.29}$$

$$\bigcap_{m\in\mathbb{Z}} V_m = \{0\}\tag{2.30}$$

$$f(\cdot) \in V_m \iff f(A \cdot) \in V_{m-1}.$$
 (2.31)

A ist dabei eine reguläre Matrix mit ganzzahligen Einträgen, die Dilatationsmatrix genannt wird. Sie ist für die Translationen zuständig.

Die Skalierungsfunktion $\varphi \in L^2(\mathbb{R}^n)$ erzeugt den Grundraum V_0 und

$$\{\varphi_{m,k}(\cdot) = |\det A|^{-m/2}\varphi(A^{-m}\cdot -k)|k \in \mathbb{Z}^n\}$$
(2.32)

bildet eine Orthonormalbasis von V_m [30, S.128]⁷.

Die mehrdimensionalen Wavelets spannen das orthogonalen Komplement von V_m in V_{m-1} auf. Es gilt [30, S.129]:

$$W_{m,j} = \overline{span\{\psi_j(\cdot - k) | k \in \mathbb{Z}^n\}}, \ j \in \{1, ..., |\det A| - 1\}.$$
(2.34)

Die Anzahl der Wavelets $\psi_1, \psi_2, ..., \psi_{|\det A|-1}$, die das orthogonale Komplement bilden, ist $|\det A| - 1$.

$$\{\psi_{j,m,k}(\cdot) = |\det A|^{-m/2}\psi_j(A^{-m}\cdot -k)| j = 1, ..., \det |A-1|, m \in \mathbb{Z}, k \in \mathbb{Z}^n\}$$

ist eine Orthonormalbasis des $L^2(\mathbb{R}^n)$ [30, S.130]⁸. Die *Abbildung 2.15* skizziert die Zerlegung bei einer Multiskalen-Analyse.

$$\varphi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k).$$
(2.33)

 $\{h_k\}$ repräsentiert die Folge der Skalierungskoeffizienten.

$$\psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \varphi(2x - k) = \sum_{k \in \mathbb{Z}} g_k \varphi_{-1,k}(x)$$
(2.35)

 $\{g_k\}$ repräsentiert die Folge der Wavelet-Koeffizienten.

⁷Bei der Darstellung der schnellen Wavelet-Transformation (*FWT*) wird die eindimensionale Skalierungsfunktion φ verwendet, durch die eine Folge $\{h_k\}_{k\in\mathbb{Z}}$ gebildet wird. Diese Skalierungsgleichung lautet [30, S.113]:

⁸Bei der Darstellung der schnellen Wavelet-Transformation (*FWT*) wird die eindimensionale Wavelet-Funktion $\psi \in V_{-1}$ benötigt. $\{V_m\}_{m \in \mathbb{Z}}$ sei eine MSA erzeugt von der orthogonalen Skalierungsfunktion $\varphi \in V_0$ [30, S.122]:



Abb. 2.15: Skizze der mehrdimensionalen Multiskalen-Analyse [30, nach S. 129].

In Bezug auf Kantenerkennung bei Bildern können separable und anisotrope Wavelets verwendet werden. Wie in späteren Abbildungen zu sehen ist, bevorzugen diese Wavelets durch die Eigenschaft der Anisotropie die x-, y-Richtung und die Diagonale [30, S.130].

Schnelle Wavelet-Transformation

Die diskrete Wavelet-Transformation und ihre Rekonstruktion wird anhand der schnellen (eindimensionalen) Wavelet-Transformation (FWT)⁹ präsentiert [30, S.132ff.]. Für die zweidimensionale wird jeder Schritt des Algorithmus auf jede Zeile und jede Spalte angewandt.

Dekomposition Die Grundfunktion f in V_0 entspricht dem Grundraum der Multiskalen-Analyse zu einer orthogonalen Skalierungsfunktion φ . f ist definiert als:

$$f(x) = \sum_{k \in \mathbb{Z}} c_k^0 \varphi(x - k)$$

mit Entwicklungskoeffizient:

$$c^0 = \{c_k^0 \mid k \in \mathbb{Z}\}.$$

 ψ stellt, wie bereits eingeführt, das orthogonale Komplement zur Skalierungsfunktion φ dar.

Die Orthonormalbasis des $L^2(\mathbb{R})$ ist

$$\{\psi_{m,k} = 2^{-m/2}\psi(2^{-m}\cdot -k) \mid m, k \in \mathbb{Z}\}$$

Die Diskretisierung beginnt mit den Skalarprodukten

⁹Die Abkürzung FWT orientiert sich an der englischen Bezeichnung Fast Wavelet Transform.

$$\sqrt{c_{\psi}}\tilde{f}(2^m, 2^m k) = \langle f, \psi_{mk} \rangle_{L^2}, \ m \in \mathbb{N}_0, \ k \in \mathbb{Z}.$$

Rekursiv findet die Zerlegung, wie in *Abbildung 2.15* zu sehen ist, durch diskrete Faltungen statt. Dabei entstehen folgende Elemente:

$$d_k^m = \langle f, \psi_{m,k} \rangle_{L^2}, d^m = \{ d_k^m \mid k \in \mathbb{Z} \} \in \ell^2(\mathbb{Z})$$

$$c_k^m = \langle f, \varphi_{m,k} \rangle_{L^2}, c^m = \{ c_k^m \mid k \in \mathbb{Z} \} \in \ell^2(\mathbb{Z}).$$

Anhand der Folgen der Skalierungs- und Wavelet-Koeffizienten g und h (siehe *Gleichung 2.33* und *Gleichung 2.35*) können d_k^m und c_k^m folgendermaßen berechnet werden:

$$\begin{aligned} d_k^m &= \langle f, \psi_{m,k} \rangle_{L^2} = \sum_{\ell \in \mathbb{Z}} g_\ell \langle f, \varphi_{m-1,2k+\ell} \rangle_{L^2} = \sum_{\ell \in \mathbb{Z}} g_{\ell-2k} c_\ell^{m-1}, \\ c_k^m &= \langle f, \varphi_{m,k} \rangle_{L^2} = \sum_{\ell \in \mathbb{Z}} h_\ell \langle f, \varphi_{m-1,2k+\ell} \rangle_{L^2} = \sum_{\ell \in \mathbb{Z}} h_{\ell-2k} c_\ell^{m-1}. \end{aligned}$$

Die Zerlegungsoperatoren H und G werden definiert als¹⁰:

$$H: \ell^{2}(\mathbb{Z}) \to \ell^{2}(\mathbb{Z}), \ c \mapsto Hc = c *_{2} h = \{ (Hc)_{k} = \sum_{\ell \in \mathbb{Z}} h_{\ell-2k} c_{\ell} \},$$
(2.36)

$$G: \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z}), \ c \mapsto Gc = c \ast_2 g = \{ (Gc)_k = \sum_{\ell \in \mathbb{Z}} g_{\ell-2k} c_\ell \}.$$

$$(2.37)$$

Rekonstruktion Sei die Skala einer diskreten Wavelet-Transformation bezeichnet mit M. Es wird die Ausgangsfolge c^0 aus den Koeffizientenfolgen $\{c^M, d^m \mid m = 1, ..., M\}$ berechnet.

Beispielsweise errechnet sich c^0 aus c^1 und d^1 mit:

$$c_k^0 = \sum_{\ell \in \mathbb{Z}} c_\ell^1 h_{k-2\ell} + \sum_{\ell \in \mathbb{Z}} d_\ell^1 g_{k-2\ell}$$

Rekursiv kommen die Zerlegungskoffizienten d^m von M - 1, ..., 1 hinzu. Die adjungierten Operatoren H^* und G^* ergeben:

$$H^*: \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z}), \ c \mapsto \{ (H^*c)_k = \sum_{\ell \in \mathbb{Z}} h_{k-2\ell} c_\ell \},$$

$$(2.38)$$

$$G^*: \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z}), \ c \mapsto \{ (G^*c)_k = \sum_{\ell \in \mathbb{Z}} g_{k-2\ell}c_\ell \}.$$
 (2.39)

Ein Rekursionsschritt ergibt:

$$c^{m-1} = H^* c^m + G^* d^m.$$

¹⁰Bei dieser Form der diskreten Faltung wird mit $*_2$ jeder zweite Index ausgewertet und die Indizes $\ell - 2k$ vertauscht (Faltung mit "Sub-Sampling").
In Abbildung 2.16 wird das bekannte Eingangsbild Abbildung 2.6 verwendet, um eine schnelle, diskrete Wavelet-Tranformation zu zeigen. Dabei wurde das biorthogonale Wavelet 3.7 benutzt. Im Quadranten vier (unten rechts) sind je Level drei d_k^m zu sehen, welche die herausgefilterten, hochfrequenten Anteile in horizontaler, vertikaler und diagonaler Richtung der Ebene darstellen. An diesem Bild ist deutlich erkennbar, dass die Skalierungsfunktion φ die tiefen Frequenzen des Bild-Signals erfasst und die hohen (feinen) Details "ausschließt". Somit ist das Bild in unterschiedlichen Auflösungsstufen erfasst. Im zweiten Quadranten (oben rechts) werden die hochfrequenten Details des dritten Levels in horizontaler Richtung vergrößert. "dwt" entspricht der Diskreten Wavelet-Transformation, "idwt" der Inversen Diskreten Wavelet-Transformation.



Abb. 2.16: Beispiel einer Wavelet-Transformation. Es findet eine rekursive Zerlegung bis zum dritten Level statt.¹¹

2.4.3 Filterung

Die Wavelet-Transformation kann als Filter selbst interpretiert werden¹² [30, S.29]. Dazu werden der Dilatations- (D^s) und Translations-Operator (T^t) des Wavelets ψ formuliert [30, S.26]:

¹¹Die mit Matlab R2016a erstellte Wavelet-Transformation verwendet zur besseren Sichtbarkeit der Unterschiede colormap(pink).

¹²Dieser Unterabschnitt orientiert sich an den Ausführungen der Kapitel 1.2 und 1.3 von [30]. Auf weitere Referenzierungen wird hier im Folgenden verzichtet.

$$D^{s}: L^{2}(\mathbb{R}) \to L^{2}(\mathbb{R}), f \mapsto D^{s}f = |s|^{-1/2}f(\cdot/s), \ s \neq 0$$
 (2.40)

$$T^{t}: L^{2}(\mathbb{R}) \to L^{2}(\mathbb{R}), f \mapsto T^{t}f = f(\cdot - t).$$

$$(2.41)$$

Die Wavelet-Transformation von f entspricht

$$\tilde{f}(s,t) = \frac{1}{\sqrt{c_{\psi}}} (D^{-s}\psi * f)(t), \qquad (2.42)$$

für festes s einer Filterung mit $\psi(\cdot/s)$. ψ repräsentiert einen Bandfilter zwischen zwei Frequenzbereichen.

In der *Tabelle 2.6* werden auszugsweise einige Wavelets vorgestellt (die Beispiele sind aus Matlab R2016a).



Mexikanischer Hut: Dieses Wavelet hat keine Skalierungsfunktion. Es wird von einer Funktion abgeleitet, die proportional zur Funktion der zweiten Ableitung der Gaußschen Wahrscheinlichkeitsdichtefunktion ist.



Daubechies: Es sind kompakt tragende, orthonormale Wavelets und werden bezeichnet mit dbN, wobei N die Ordnung darstellt. Es folgt ein Beispiel der Ordnung 2.



Biorthogonal: Die Wavelets dieser Familie haben eine lineare Phase, welche für Signal- und Bildrekonstruktion benötigt wird. Für die Zerlegung und die Synthese werden unterschiedliche Wavelets verwendet.



Coiflets: Daubechies hat diese Wavelets auf Anforderung von R. Coifman entwickelt. Die Wavelet-Funktion hat 2N Momente gleich 0, die Skalierungsfunktion 2N - 1 Momente gleich 0. Das unten stehende Beispiel ist *Coiflet* 1.



Symlets: Sie sind annähernd symmetrisch, die von Daubechies als Modifikation der db-Familie entwickelt wurden. Ihre Eigenschaften sind ähnlich. Es folgt ein *Symlets 8*.



Meyer: Das Meyer-Wavelet und die Skalierungsfunktion sind in der Frequenzdomäne definiert.



Tabelle 2.6: Wavelet-Familie mit Beispielen [35, S.1-38 ff.]

Weiterhin gibt es reelle, komplexe, umgekehrt biorthonormale Wavelets.

2.5 Resümee

Steganographie

Es wird die Relevanz der jeweiligen Bereiche für die vorliegende Arbeit validiert.

Die Verfahren der Substitutionssysteme führen geringfügige Modifikationen der digitalen Coverdaten, wie z. B. im niederwertigsten Bit eines Pixels, durch. Die eingebetteten Daten können in diesem Anwendungsfall durch den Speicher-, Ausdruck- und Reproduktionsvorgang nicht mehr rekonstruiert werden, da unter anderem keine Kontrollstellen bei der Veränderung enthalten sind, die zur robusten Verarbeitung notwendig sind.

Die Transformations-Domäne-Techniken werden nicht ausschließlich zum Zwecke der Steganographie verwendet und werden im nächsten Kapitel gesondert behandelt.

In den Methoden der Frequenzspreizung werden zwar Nutzdaten hinzugefügt, aber sie zielen auf die Nutzung von größerer Bandbreite bei der Kommunikation. Die Daten können in diesem Fall beim Photographieren der gedruckten Dokumente nicht mehr rekonstruiert werden.

Statistische Verfahren, wie sie bei Steganographie verwendet werden, können für die vorliegende Thematik vernachlässigt werden. Ähnlich wie bei Substitutionssystemen, sind die eingebetteten Daten im niederwertigen Bitbereich, sodass eine fehlerkorrigierende und robuste Entschlüsselung nicht möglich ist. Es ist allerdings vorstellbar, dass eine statistische Analyse der digitalen Druckerzeugnisse integriert werden kann.

Das Einbetten von Informationen durch Distorsion bei formatierten Texten stellt eine Möglichkeit der versteckten Codierung in textuellen Dokumenten dar.

Eine Anwendung von Coverdaten-Generierungs-Methoden ist hier irrelevant.

Zusammenfassend können aus dem Gebiet der Steganographie Distorsionstechniken genutzt werden, um Informationen in formatierte Texte einzubetten. Verfahren in Transformationsräumen werden in 2.3 Bildbearbeitung im Frequenzraum und 2.4 Bildbearbeitung im Zeit-Frequenzraum behandelt.

Ortsraum

Die Herangehensweise zur Bildbearbeitung im Ortsraum geht über Kantendetektion mit anschließender Detektion von Linie bzw. zusammenhängenden Punkten mit der Hough-Transformation. Durch diese Erkenntnis ist es möglich, die Dokumente abhängig von ihren Merkmalen an diesen Stellen zu codieren. Eine detaillierte Betrachtung ist im Kapitel 3 Lösungsansätze unter 3.2 Ortsraum abgebildet.

Frequenzraum

In der Bildverarbeitung im Frequenzraum werden als Grundlage mehrere mathematische Konzepte benötigt. Um messbare Funktionen $f \in L^2(\mathbb{R})$ in den Frequenzraum zu transformieren, wird das Skalarprodukt des Hilbert-Raumes zu Hilfe genommen.

2.5 Resümee

Filterung im Frequenzraum ist analog zu 2.2 Bildbearbeitung im Ortsraum sowohl durch Tiefpass- als auch durch Hochpassfilter möglich. Im Vergleich zur Modifikation im Ortsraum ergeben sich im Frequenzraum unter anderem folgende Vorteile:

- Zum einen ist die Laufzeit bei FFT mit $\mathcal{O}(N \log_2 N)$, $N \to \infty$ deutlich besser gegenüber $\mathcal{O}(N^2)$. Hinzu kommen im Frequenzraum zwar die Laufzeit in Höhe von $\mathcal{O}(N \log_2 N)$ für die IFFT und die Faltungsoperation, die Performanz bleibt jedoch deutlich überlegen.
- Die Veränderungen durch Filter werden im Frequenzraum global im Bild verteilt, sodass das entstandene Bild eher "nicht wahrnehmbare" Modifikationen vermuten lässt, als welche im Ortsraum mit lokalen Änderungen.

Das Kapitel 3.3 Frequenzraum adaptiert die hier vorgestellten Konzepte und zeigt einen möglichen Lösungsansatz.

Zeit-Frequenzraum

Im Vergleich zur schnellen Fourier-Transformation, welche in $\mathcal{O}(N \log N)$ liegt, beträgt die Laufzeit der schnellen Wavelet-Transformation $\mathcal{O}(N)$ [30, S.136]. Die Komplexität ist damit deutlich schneller.

Diese Analyse gibt im Vergleich zu Fourier-Transformation mehr Informationen preis, nämlich die lokalen Daten zu der dazugehörigen Frequenz.

Inwieweit für vorliegenden Untersuchungsgegenstand diese Verfahren genutzt werden können, wird in 3.4 Zeit-Frequenzraum geprüft.

3 Lösungsansätze

Dieses Kapitel widmet sich möglichen Lösungsansätzen, die zur Codierung der Druckerzeugnisse und der Decodierung der eingebetteten Daten dienen.

Die Codierung betreffend muss zwischen der Codegenerierung bezüglich der einzubettenden Daten und dem Einbettungsverfahren selbst unterschieden werden. Letzteres betreffend finden bestehende Technologien aus 2 Stand der Technik in den unten dargestellten Algorithmen Anwendung.

3.2 Ortsraum, 3.3 Frequenzraum und 3.4 Zeit-Frequenzraum validieren die Realisierbarkeit einer versteckten Codierung, weisen auf Schwachstellen hin und identifizieren abhängige Parameter.

Die Lösungsverfahren haben dabei einen ähnlichen Ablauf:

- (1) Digitale Daten werden transformiert. (optional)
- (2) Daten (\doteq Codewort) werden im digitalen Dokument eingebettet.
- (3) Die modifizierten Dokumente werden rücktransformiert. (optional)
- (4) Dokumente werden gedruckt.
- (5) Dokumente werden zugeschnitten und reproduziert¹ (wie durch Einscannen oder Photographieren).
- (6) Codewort wird extrahiert und decodiert.
- (7) Decodierte Information des Codewortes und damit die Identifikationsnummer kann zugeordnet werden.

In den nachfolgenden Lösungsansätzen differieren die Schritte (1), (2), (3) und (6).

Generell gibt es im abgebildeten Prozess (*Abbildung 1.1*) Probleme, die alle Lösungsverfahren enthalten, nämlich die Reskalierung der reproduzierten Dokumente, die Codegenerierung und dessen Decodierung. Wie letztendlich eine Decodierung umgesetzt wird, ist kein Bestandteil dieser Arbeit. Es gilt aber darzustellen, welche Verfahren dazu geeignet sein könnten und diese in Simulationen zu testen. Dabei spielt das menschliche (subjektive) Erkennen der Codierung die Rolle des Decodierers, in der Simulation findet das Code-Extrahieren statt.

¹Es wird der Begriff der *Reproduktion* verwendet, um vor reinem Digitalisieren eines Bildes abzugrenzen. Denn beim Ablichten des Dokuments ist die Papierbeschaffenheit beispielsweise selbst Teil des neuen digitalen Bildes.

Bei den nachfolgenden Lösungsansätzen existieren generelle Teilprobleme, die zunächst betrachtet werden müssen. In diesem Unterkapitel ist die Bezeichnung "Codierung" auf "Codegenerierung" bezogen. Damit soll die Bezeichnung der Literatur der Codierungstheorie-Domäne angeglichen sein. Ist explizit das Codieren im Sinne des Einbettungsverfahrens gemeint, wird darauf hingewiesen.

- 1. Die einzubettenden Daten betragen eine spezifische Größe. Um nach dem Reproduzieren Daten der gleichen Größe extrahieren zu können, muss das digitale Dokument dieselben Maße haben wie das ursprüngliche Ausgangsbild. Unter 3.1.1 Reskalierung der reproduzierten Dokumente wird darauf näher eingegangen.
- 2. Die Codierung und vor allem die Decodierung stellen die größte Herausforderung des Prozesses dar.

Das Herauslesen der eingebetteten Daten beim Decodieren soll eindeutig sein. Das bedeutet, dass fehlererkennende bzw. -korrigierende Maßnahmen vorhanden sein müssen, falls durch die Reskalierung oder anderweitige Einflüsse von außen beim Scannen (wie Belichtung) der extrahierte Code nicht dem ursprünglich eingebetteten entspricht.

Aus der Codierungstheorie werden in 3.1.2 Codierung und Decodierung zusammenfassend Aspekte dargestellt.

 Abhängig vom Codierungs- und Decodierungsverfahren müssen die einzubettenden Daten (≜ Codewort) und ihre Größe ermittelt werden. In 3.1.3 Codewortbildung werden mögliche Parameter zur Festlegung vorgestellt.

3.1.1 Reskalierung der reproduzierten Dokumente

Bei der Reproduktion können neben äußeren Bedingungen wie Tagesbelichtung auch die Lage des Dokuments beeinflussend sein und Fehler entstehen, die korrigiert werden müssen. Es wird davon ausgegangen, dass viele Faktoren kontrolliert werden können, kleine Korrekturmaßnahmen sollen nichtsdestotrotz automatisch geschehen.

Aus diesem Grund wird in diesem Abschnitt übersichtlich dargestellt, wie eine Anpassung bzw. Reskalierung zu benötigten Maßen der digitalisierten, gedruckten Dokumente vorgenommen werden kann.

Zum Teilproblem der Reskalierung gibt es Voraussetzungen bzw. Vorbedingungen, die zur einwandfreien Verarbeitung zunächst formuliert werden.

Bedingung 3.1 (Vordefinierte Formate). Die Formate der digitalen Druckerzeugnisse sind vordefiniert, wie beispielsweise die Bildpunktmaße einer Visitenkarte, von Bildern, Plakaten, die unter Umständen nach DIN-Normen ausgerichtet sind.

Bedingung 3.2 (Bounding Boxen). Die Hardware, die die ausgedruckten Dokumente ablichtet, enthält für die vordefinierten Formate beispielsweise "Bounding Boxen", um eine Reskalierung in Echtzeit durchführen zu können und die benötigten Maße zu kennen.

Die digitalisierten Dokumente entsprechen womöglich nicht den benötigten Bildpunktmaßen. Zur Berechnung dieser sind Transformationen erforderlich. Neben linearen Transformationen, die unten stehend definiert werden, ist möglicherweise eine Transformation der Translation (Verschiebung) notwendig.

Definition 3.3 (Lineare Transformation). [20, S.221ff.] Eine lineare Transformation \mathcal{L} wird definiert durch:

$$\mathcal{L}: \mathbb{R}^2 \to \mathbb{R}^2, \, \mathcal{L}(v + \alpha w) = \mathcal{L}(v) + \alpha \mathcal{L}(w), \, v, w \text{ sind Vektoren in } \mathbb{R}^2 \text{ für } \alpha \in \mathbb{R}.$$
(3.1)

Rotation, Skalierung, Scherung und Spiegelung sind lineare Transformationen. Die Skalierung mit negativen Werten führt zu einer Degenerierung eines Objekts.

Zur Korrektur der Bildmaße des reproduzierten Dokuments werden unter Umständen Rotation, Skalierung und Translation als Transformationen benötigt, wie die *Abbildung 3.1* veranschaulicht.



Abb. 3.1: Skizze der Transformationen zur Reskalierung eines reproduzierten Dokuments

Für die Translation muss zur Berechnung der Verschiebungswerte ein Fixpunkt F definiert sein. Im Folgenden wird F mit dem Ursprungspunkt (x, y) = (0, 0) einer Bounding Box festgelegt.

Die Rotation erfolgt mittels Cosinus und Sinus eines Winkels α . Dieser kann durch die Abweichung von der Grenze der Bounding Box und dem Rand des Dokuments errechnet werden.

Eine Skalierung ist für die Streckung bzw. Degenerierung eines Objekts zuständig.

Bemerkung 3.4. [20, S.235] Die unterschiedlichen Transformationen sind nicht kommutativ, sodass die Reihenfolge der Ausführungen ausschlaggebend ist.

Zur Durchführung der Transformationen wird die Euklidsche Ebene (entspricht dem reproduzierten Dokument) zu einem xyw-Raum erweitert. Die w-Koordinate wird auf 1 gesetzt, sodass die Punkte der Ebene als x, y, 1 definiert sind. Mit Hilfe der hinzugefügten

Dimension können Transformationen durchgeführt werden, wobei die Vektoren $\begin{bmatrix} & \\ y \\ 1 \end{bmatrix}$ mit

einer 3 × 3 Matrix $\mathbf{M} = \begin{bmatrix} a & b & c \\ d & e & f \\ p & q & r \end{bmatrix}$ multipliziert werden. Die Transformation \mathcal{T} wird wie folgt berechnet [20, S.234]:

$$\mathcal{T}: \mathbb{R}^3 \to \mathbb{R}^3, \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} a & b & c \\ d & e & f \\ p & q & r \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}.$$
(3.2)

Definition 3.5 (**Translationsmatrix**). [20, S.234] Die 3×3 Matrix **M**, Translationsmatrix genannt, beinhaltet alle Transformationsarten – neben den linearen Transformationen auch die Translationen (Verschiebung) und können mit \mathcal{T} errechnet werden.

Die Matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathbf{M}$ definiert lineare Transformationen. Des Weiteren werden c, f für eine Translation benötigt. p und q werden gleich 0 und r wird 1 gesetzt, um px + py + r = 1 = w zu erhalten. Gründe für die Verwendung der dritten Dimension liegen unter anderem in der Vereinheitlichung aller Transformationen und um zusammengesetzte Transformationen leichter berechnen zu können. Dadurch wird auch die Fehleranfälligkeit minimiert [20, S.235f.].

Durch die Reskalierung wurde die gleiche Größe des digitalen Ausgangbildes erstellt. So ist sichergestellt, dass die zu extrahierenden Daten ebenfalls dasselbe Format haben wie bei der Einbettung. Allerdings können Rundungen bei den Transformationen dazu führen, dass beim Filtern der eingebetteten Daten Ungleichheit mit den ursprünglich hinzugefügten Informationen vorliegt.

Im Folgenden wird analysiert, inwieweit eine Fehlererkennung bzw. -korrektur möglich ist.

3.1.2 Codierung und Decodierung

Um die einzubettenden Informationen korrekt zu codieren, ist es in erster Linie wichtig, fehlererkennende und -korrigierende Verfahren beim Decodierern zu kennen. Daraus können Parameter zur Codierung abgeleitet werden.

Der Bereich der Codierungstheorie kann für den vorliegenden Untersuchungsgegenstand ein Anreiz sein und Lösungen zur Codierung liefern, die in diesem Unterkapitel überblicksartig vorgestellt werden.

Im dargestellten Ablauf in *Abbildung 1.1* können an unterschiedlichen Stellen Fehler entstehen. Die Übertragung von codierten Daten findet im Gegensatz zum üblichen Kommunikationstransport nicht über einen (Funk-)Kanal statt, sondern wird digital im Druckerzeugnis eingebettet, gespeichert, ausgedruckt, reproduziert und (optional) reskaliert. Darauf folgt das "empfängerseitige" Decodieren, wobei eine möglichst hohe Fehlertoleranz vorhanden und fehlererkennend bzw. -korrigierend sein sollte bzw. muss. Allein bereits beim Speichern der digitalen, codierten Dokumente treten Quantisierungsfehler auf, die unumgänglich sind.

Um diesen Anforderungen gerecht zu werden, muss die Quellencodierung in eine redundante Darstellung systematisch übergeführt werden [43, S.13].

Der Decodierer muss fehlerhaften Code entdecken und den Code finden, der mit größter Wahrscheinlichkeit in das ursprüngliche digitale Dokument hinzugefügt wurde [43, S.14]. Relevant sind in diesem Fall zyklische Codes (\subset lineare systematische Codes [43, S.37]), die mehr Fehler korrigieren können als beispielsweise nicht-zyklische, lineare systematische Codes. Letztere geben das Prüfschema n = m + k wieder, wobei m den Daten mit |m| Stellen und k den Kontrollstellen entsprechen [42, S.38] [43, S.42]. Der lineare systematische Hamming-Code beispielsweise kann einen Fehler korrigieren [42, S.34].

Das Prinzip eines zyklischen Codes wird nachfolgend beispielhaft dargestellt.

Sei f = 0001011 eine binäre Folge, bestehend aus m Informationsbits und k Paritätskontrollbits nach obigem Schema n = m + k [42, S.54]. Die entsprechenden zyklischen Verschiebungen lauten:

Folge f	0001011
1. Verschiebung	0010110
2. Verschiebung	0101100
3. Verschiebung	1011000
4. Verschiebung	0110001
5. Verschiebung	1100010
6. Verschiebung	1000101

f = 0001011 kann als Polynom formuliert werden [42, S.49]. Jede Eins repräsentiert einen Term des Polynoms. Das Polynom zu f entspricht $g(X) = X^3 + X + 1$.

Abhängig von n werden linear unabhängige Generatorfolgen definiert. Diese Folgen können aus Addition der jeweiligen Verschiebungen nicht hergestellt werden [42, S.49]. In Bsp. f sieht die Generator-Matrix \mathcal{G} folgendermaßen aus:

$$\mathcal{G} = \begin{bmatrix} 1011000\\0101100\\0010110\\0001011 \end{bmatrix}$$

Zur Codierung werden primitive Polynome als Generator verwendet, beispielsweise für einen zyklischen Hamming-Code. Eine gleichwertige, aber performantere Art stellt die

Berechnung mit Schieberegistern und Rückkopplung in einem Blockschaltkreis dar [42].

Die Fehlererkennung beim Decodieren wird durch den Begriff des Syndroms festgehalten. Durch zyklische, lineare Codes ist nicht nur die Erkennung, sondern auch die Korrektur mehrerer Fehler möglich [42, S.62ff.]. Das Hinzufügen weiterer Bits in ein Codewort erhöht die Sicherheit [42, S.66ff.].

Um eine höhere Fehlertoleranz zu erlauben, können binäre Folgen erweitert werden. Zu ihrer Berechnung werden in Galois-Feldern GF(p), p ist prim oder eine geradzahlige Potenz einer Primzahl, irreduzible Polynome und Wurzeln verwendet [42, S.71ff.].

Die Familie der zyklischen BCH-Codes, benannt nach Bose, Chaudhuri und Hocquenghem, behandelt neben Einzelfehler- auch Korrektur von mehreren Zufallsfehlern. Enthalten sind auch welche, die nicht nur binär sind, sodass $GF(q^m)$ mit q ist prim, m ist gerade, zum Einsatz kommen [42, S.68].

Ein t-fehlerkorrigierender BCH-Code besitzt 2t aufeinanderfolgende Wurzeln im $GF(q^m)$ und hat die Länge q^{m-1} , $t \in \mathbb{Z}$ [42, S.82]. Primitive, binäre BCH-Codes können im Frequenzbereich abgebildet werden, bei der spektral 2t aufeinanderfolgende Komponenten Null ergeben. Unter anderem gehören Reed-Solomon-Codes zu dieser Gruppe. Bei der Decodierung werden die Wurzeln der Polynome zur Positionsbestimmung angewandt [42, S.87ff.]. Der Ablauf ist folgender [42, S.90f.]:



Abb. 3.2: Codierung und Decodierung im Frequenzraum nach Reed-Solomon-Code [42, S.91]

In der Literatur zu Codierungstheorie finden sich zahlreiche zyklische, lineare Block-Co-

des, die im Rahmen dieser Arbeit nicht dargestellt werden. Zu dieser Gruppe soll der Reed-Solomon-Code (siehe oben) ein repräsentatives Beispiel sein. Diese Domäne bietet einen guten Ansatzpunkt zur weiteren Recherche zur Weiterentwicklung bzw. Anpassung einer bestehenden Codierung.

Unter 3.3.2 Simulation wird eine Codierung und Decodierung an Beispielen mit einem online generierten 2D-Barcode (Data Matrix)² angewandt, bei dem der Reed-Solomon-Code Anwendung findet.

3.1.3 Codewortbildung

Ausstehend ist die Bestimmung der einzubettenden Daten und der Codewortgröße. Wie in *3.1.2 Codierung und Decodierung* beschrieben, ist die Codierung vom ausgewählten Verfahren abhängig.

Zur Festlegung der Informationen, die den Bildern bzw. Dokumenten hinzugefügt werden, können unterschiedliche Parameter dienen.

Szenario Wie in *Abbildung 1.1* dargestellt, werden mehrere Dokumente auf einem Druckbogen gebündelt. Im Folgenden wird angenommen, dass bis zu 150 Druckerzeugnisse unterschiedlicher Größe (Visitenkarten, Plakate, Bilder, Flyer) darauf platziert werden. Dadurch ist die Anzahl der einzubettenden Information eingegrenzt.

Eine Möglichkeit ist, eine Adresse mit einer eindeutigen Identifikationsnummer abzubilden. Dabei kann ein Code mehrfach in die digitalen Dokumente eingebettet sein. Vorteil ist dadurch nicht nur ein kürzerer Code, sondern damit zusammenhängend ein einfacheres und fehlerresistenteres Decodierungsverfahren. Denn je länger eine einzubettende Information ist, umso länger muss das Codewort lauten. Dies hat jedoch wiederum zur Folge, dass es fehleranfälliger ist als ein kürzeres Codewort und darüber hinaus eine höhere Fehlerkorrektur notwendig ist.

Zusätzliche Parameter zur Bestimmung einer Identifikationsnummer (ID) können Format, Größe, aber auch ein digitaler Fingerabdruck wie ein Histogramm oder Ähnliches sein.

Im Folgenden werden exemplarisch zwei mögliche Codewort-Bildungen dargestellt.

- (Codewort 1) Es wird ein primitiver, binärer BCH-Code verwendet. Es reichen acht Bit mit m = 8 als die Größe der einzubettenden Daten mit der Voraussetzung, dass maximal 150 Dokumente eindeutig identifiziert werden müssen. Es liegt $GF(2^8)$ mit der Codewortlänge $2^m - 1 = 2^8 - 1 = 255$ vor. Wenn wie oben beschrieben zusätzliche Parameter des Dokuments in der ID mitberücksichtigt werden, minimiert sich die Codewortgröße z. B. auf m = 6, sodass die Länge $2^6 - 1 = 63$ beträgt. Dabei muss miteinkalkuliert werden, dass zur Korrektur 2t aufeinanderfolgende Wurzeln in $GF(q^m)$ enthalten sein müssen, um t Fehler zu korrigieren.
- (Codewort 2) Es wird ein einfaches Codewort, wie in *Abbildung 3.3* zu sehen, verwendet, in dem drei Kombinationsmöglichkeiten in der Verteilung von schwarzen und weißen Streifen gegeben sind. Wird es im digitalen Dokument zwei

²https://racoindustries.com/barcodegenerator/2d/datamatrix/

mal angewendet und codiert, so ergeben sich damit $2^6 = 64$ mögliche Codierungen von Adressen³.



Abb. 3.3: Ein Beispiel eines einfach strukturierten Codewortes der Maße 9×9 Pixel

Die Kontrollstellen dieses Codewortes ergeben sich aus der Struktur, bei der jeder Teilbereich 3×3 Pixel innehat: im ersten Teil sind schwarze Farbcodewerte, im zweiten weiße und im letzten Teil wieder schwarze. Bei der Extraktion können der Einsatz eines Threshold-Verfahrens und die Berechnung eines Durchschnitts der Farbcodewerte in den einzelnen Bereichen beispielsweise zu einer Erkennung und damit zur Decodierung des Codewortes führen.

In den Simulationen wird das in *Abbildung 3.3* dargestellte Codewort benutzt und als eine Identifikationsnummer für eine Adresse eingesetzt.

 $^{^3 \}rm Die$ Anzahl der Identifikationsmöglichkeiten ist für vorliegendes Beispiel nicht vorgegeben. Diese kann nach Szenario maximal 150 betragen.

3.2 Ortsraum

Es wird in diesem Kapitel geklärt, inwieweit eine Verarbeitung im Ortsraum für den vorliegenden Anwendungsfall durchführbar und geeignet ist.

3.2.1 Algorithmus

In 2.2 Bildbearbeitung im Ortsraum wurden Verfahren vorgestellt, die sich zur Merkmalsextraktion eignen. Ein mögliches Vorgehen im Ortsraum unter Anwendung dieser ist folgendes:

Algorithmus 1: Verarbeitung im Ortsraum
Input : digital Documents of size k Output : printed And Assigned Documents of size l
Data: digitalDocuments of size k , documentsWithDetectedEdges of size k , codedDocuments of size l , printedDocuments of size l , scanedDocuments of size l , decodedData of size l , printedAndAssignedDocuments of size l , $l \ge k, l, k \in \mathbb{N}$
1 for $\underline{i=1 \text{ to } k} \mathbf{do}$
2 documentsWithDetectedEdges $(i) = process$ edge detection and Hough
transformation at digitalDocuments (i) ;
while exists multiple addresses for i do
4 codedDocuments.add($modify$ coherent pixel at digitalDocuments(i) with information of documentsWithDetectedEdges(i));
5 printedDocuments = $print$ codedDocuments;
\mathbf{s} scanedDocuments = $scan$ printedDocuments;
7 for $j = 1$ to l do
8 decodedData $(j) = compare \text{ scanedDocuments}(j)$ with digitalPrintProducts and identify the most similar one;
9 $map \ decodedData(j) \ to \ address;$
o printedAndAssignedDocuments $(j) = assign \text{ address to scanedDocuments}(j);$
1 return printedAndAssignedDocuments;

Bemerkung 3.6 (Variablen k, l). Die Anzahl der gedruckten Bilder der Größe l beträgt mehr oder gleich der digitalen Dokumente der Größe k, da sie mehrfach in Druckauftrag gegeben werden können.

Zunächst wird eine Merkmalsextraktion in Zeile 2 durch Kantendetektion durchgeführt und eine Hough-Transformation angewandt. Dadurch werden (krumme) Linien bzw. zusammenhängede Bildpunkte erkannt, die anschließend im Originaldokument an den ermittelten Positionen modifiziert werden (Zeile 4). Das veränderte Druckerzeugnis wird zum Decodieren nochmals benötigt und gespeichert.

Nach dem Druck-, Zuschnitt- und Reproduktions-Prozess wird das digitalisierte Dokument mit gespeicherten, codierten Dokumenten verglichen (*Zeile 8*) und dasjenige modifizierte ausgewählt, das am ähnlichsten ist (*Zeile 9*). In Schritt 10 kann die dazugehörige Adresse aufgrund der Selektion zugeordnet und der Prozess abgeschlossen werden.

3.2.2 Schwachstellen

Der dargestellte Algorithmus im Ortsraum stellt eine mögliche Lösung dar. Es treten an mehreren Stellen, die nachfolgend analysiert werden, allerdings Probleme auf. Es geht dabei um die Belange der *Robustheit* der Codierung und Decodierung, der visuellen *Wahr-nehmbarkeit* der Modifikation, der *Distorsion* des Dokuments und der *technischen Her-ausforderungen* bei der Kantendetektion.

Der Begriff des "Codewortes" mit den Identifikationsdaten wird im Ortsraum als Modifikationen von (krummen) Linien bzw. zusammenhängenden Bildpunkten verstanden.

Es werden Schwachstellen $S_x, x \in \mathbb{N}$ beim Codieren und Decodieren beschrieben, die sich auf die Robustheit, Wahrnehmbarkeit und Qualität durch die Codierung im Ortsraum auswirken.

Robustheit In diesem Abschnitt werden Aspekte angesprochen, die alle Bilder mit unterschiedlichen Bildobjekten betrifft - mit textuellem Inhalt, Konturen, homogenen Flächen, Bildern mit Textur.

 $(S_1)\,$ Die Codewortgenerierung kann nicht bei allen Druckerzeugnissen durchgeführt werden.

Bei Bildern mit ausschließlich homogenen Flächen (z. B. einem blauen Himmel) schlägt dieses Verfahren im Ortstraum fehl. Es existieren keine zusammenhängenden Bildpunkte, die eine Linie beschreiben und die zu modifizieren sind, sodass keine Codierung (Zeile 2 bis 4) erzeugt werden kann. Das heißt, dass ein zusätzliches Verfahren benötigt wird, um die Verarbeitung unterschiedlicher Bildobjekte abdecken zu können.

 (S_2) Hinsichtlich Fehlererkennung und -korrektur gilt es festzustellen, ob eine robuste Codierung bzw. Decodierung möglich ist, da eine Abhängigkeit von lokalen Informationen besteht.

Dazu wird für das oben beschriebene *Szenario* ein Beispiel eines Bildobjekts mit einer Linie (siehe *Abbildung 3.4*) verwendet. Die Grenzen der Linie wurden durch eine Hough-Transformation erkannt. Welches Objekt sie im Gesamtbild wiedergibt, ist hier nicht relevant. Weiterhin ist in der Skizze irrelevant, wie viele Pixel verändert wurden – sie dient zur visuellen Veranschaulichung.

Es wird außerdem angenommen, dass es $150 \times$ (ein Druckbogen) ausgedruckt und an unterschiedliche Adressen (10) verschickt wird. Für die Verschlüsselung müssen zehn Codeworte bzw. zehn Modifikationen an der Linie vorgenommen werden. Abbildung 3.4 stellt mit lediglich vier Codeworten (Modifikation $M_x, x \in \{0, 1, 2, 3\}$) eine mögliche Codewortgenerierung dar.

3.2 Ortsraum



Abb. 3.4: Bild mit einer Linie: Skizze von vier Codeworten⁴

Im Decodierverfahren muss berücksichtigt werden, dass der Prozess mehrere Fehlerquellen (Speichern der Modifikationen, Ausdruck, Scannen, (optionale) Reskalierung) aufweist, wie bereits in 3.1.2 Codierung und Decodierung beschrieben. Dazu muss das Codewort Kontrollstellen enthalten und Konzepte aus 3.1.2 Codierung und Decodierung finden. Nur dann wird eine hohe Fehlertoleranz erreicht, sodass eine Fehlererkennung und -korrektur möglich ist.

 (S_3) Eine mehrfache Codewortgenerierung ist unter Umständen nicht möglich. Als Beispiel wird ein Bild mit einem Punkt verwendet, siehe *Abbildung 3.5*⁵.



Abb. 3.5: Bild mit einem Punkt

Sollte dieses Dokument mit unterschiedlichen Codeworten versehen werden, ist die Anzahl der Generierung eingeschränkt, ohne dass es womöglich Auswirkungen auf die Wahrnehmbarkeit hat.

Wahrnehmbarkeit Abschnitt *Robustheit* stellt die Probleme bei der Codierung und Decodierung dar. In diesem Zusammenhang wurde bereits die nachstehende Schwachstelle identifiziert.

 (S_4) Die Modifikationen am Bild führen gegebenenfalls zu visueller Wahrnehbarkeit der Veränderungen. Das Beispiel in (S_3) verdeutlicht es. Dies widerspricht der essenzi-

⁴Bei der Modifikation M_1 wurde die Linie um eine Pixelreihe nach oben, bei M_2 nach unten verstärkt. M_3 enthält bei der ersten Hälfte der Linie eine duplizierte Pixelreihe.

⁵Auch hier spiegelt das Beispiel das Resultat nach Kantendetektion wider. Welches Objekt der Punkt im Gesamtbild wiedergibt, ist hier nicht relevant.

ellen Anforderung der Aufgabenstellung, nämlich der "versteckten" Codierung der Druckerzeugnisse.

Distorsion Im Hinblick auf visuell wahrnehmbare Modifikationen bei der Codierung entstehen qualitative Einbußen. Qualität bezieht sich hier nicht auf die Bildqualität und deren Auflösung, sondern auf inhaltliche Veränderungen der Bildobjekte, welches eine Distorsion des Originals zur Folge hat. Deren menschliche Interpretation entspricht im Kontext womöglich nicht einer realen Darstellung.

 $(S_5)\,$ Eine Codierung mit erkennbaren Veränderungen führt zu verminderter Qualität und Distorsion des Bildes.

Als Beispiel wird ein kreisförmiges Bildobjekt (Ball) in *Abbildung 3.6* derart modifiziert bzw. codiert, dass es nicht mehr zur Bildperspektive passt und zu ovalförmig ist bzw. wirkt.



(a) Original bild



(b) Codiertes Bild (am Ball)

Abb. 3.6: Distorsion im Ortsraum: Modifikation verfälscht Realität

Abbildung 3.6 zeigt eine weitere Besonderheit. Der Ball und sein Schatten sind kausal zusammengehörige Bildobjekte, die eine rechnergestützte Bildverarbeitung schwer erkennt und als solches versteht. Zusätzlich zum Aspekt der Codierung und Decodierung ist diesbezüglich eine tiefer gehende Analyse von Bildverständnis notwendig.

Technische Umsetzung Neben Schwachstellen, die die Codierung (im Sinne der Codegenerierung) betreffen, gilt es bei diesem Lösungsansatz im Ortsraum auch genauer die technische Umsetzung zu betrachten.

 (S_6) Eine Herausforderung stellt die Binarisierung der Kanten und Anwendung der Hough-Transformation dar. Dabei geht es insbesondere um eine robuste Kantendetektion ohne Rauschanteile. Eine Herangehensweise ist die Verwendung von Eigenwertanalyse nach Guru, Shekar und Nagabhushan in [18]. Die Rechenzeit der vergleichsweise rechenintensiven Hough-Transformation ist in diesem Verfahren geringer.

3.2.3 Resümee

Anhand einiger Beispiele konnten herausfordernde Aspekte im Lösungsverfahren im Ortsraum identifiziert werden, die bei einer Umsetzung beachtet werden müssen. Um eine durchführbare Verarbeitung zu erhalten, bedürfen folgende Fragestellungen einer weiteren Analyse:

3.2 Ortsraum

- (1) Welche Dokumentvorlagen eignen sich für die Verarbeitung im Ortsraum? Was für ein Verfahren kann für die ausgeschlossenen verwendet werden?
- (2) Wie kann ein fehlererkennendes bzw. -korrigierendes Verfahren in die Codierung und Decodierung eingearbeitet werden?
- (3) Was ist der Schwellwert zu einer wahrnehmbaren Modifikation? Müssen womöglich mehrere Grenzen definiert werden, die in Abhängigkeit zu den Bildobjekten stehen?
- (4) Wie können kausal zusammenhängende Bildobjekte (wie Objekt und sein Schatten) erkannt und im Modifikationsprozess mitberücksichtigt werden?
- (5) Hinsichtlich einer robusten technischen Umsetzung muss die Binarisierung bei der Kantendetektion analysiert werden.

Zu einer robusteren Lösung im Ortsraum können weitere Parameter wie Bildformat oder digitaler Fingerabdruck (z. B. Histogramm) zusätzlich betrachtet und berücksichtigt werden. Bezüglich der grundlegenden Aspekte der Robustheit, Wahrnehmbarkeit und Distorsion existieren allerdings Schwachstellen in der Anwendbarkeit, die zunächst untersucht werden müssen.

Die dargestellten Variablen und Abhängigkeiten zeigen, dass die Komplexität in einer Umsetzung im Ortsraum hoch ist, nur eingeschränkt verwendbar und sogar fehlerbehaftet ist.

Ein effektives Lösungsverfahren bietet die Codierung im Frequenzraum. Dabei werden die einzubettenden Informationen global im digitalen Dokument verteilt. Somit besteht die Annahme, dass die Modifikation für das menschliche Sehvermögen im nicht wahrnehmbaren Bereich liegt.

Nach 3.3.1 Algorithmus, in welchem der Ablauf in Pseudocode formuliert ist, folgt die Ausführung und Beschreibung der Simulation (3.3.2) und eines praktischen Tests (3.3.3). In 3.3.4 Resümee werden abhängige Parameter daraus abgeleitet.

3.3.1 Algorithmus

Der nachstehende Algorithmus repräsentiert den Ablauf von der Codierung (Einbettung der Daten) bis zur Zuordnung der Druckerzeugnisse zu der dazugehörigen Adresse.

Algorithmus 2: Verarbeitung in der Frequenzdomäne

Input : digitalDocuments of size k**Output**: printedAndAssignedDocuments of size lData: digitalDocuments of size k, codedDocuments of size l, printedDocuments of size l, scanedDocuments of size l, decodedData of size *l*, printedAndAssignedDocuments of size *l*, $l > k, l, k \in \mathbb{N}$ 1 for i = 1 to k do while exists multiple addresses for i do $\mathbf{2}$ codedDocuments.add(embed codeword (= order information)) in 3 FFT(digitalDocuments(i)));4 printedDocuments = print codedDocuments; 5 scanedDocuments = scan printedDocuments; 6 for j = 1 to l do decodedData(j) = extract information of scanedDocuments(j) with FFT; 7 $map \ \texttt{decodedData}(j) \ \texttt{to} \ \texttt{address};$ 8 printedAndAssignedDocuments(j) = assign address to scanedDocuments(j);9 10 return printedAndAssignedDocuments;

Mit Hilfe der schnellen Fourier-Transformation (FFT) wird in Zeile 3 ein Druckerzeugnis zunächst in die Frequenzdomäne transformiert. Zur Codierung wird ein Codewort verwendet, das in einen Filter verarbeitet ist. Für das spätere Decodierverfahren muss die Abbildung (Codewort \rightarrow Adresse) gespeichert werden.

Im selben Schritt wird eine Rückführung in den Ortsraum mit der inversen schnellen Fouriertransformation (IFFT) durchgeführt (*Zeile 3*), um das digitale Druckerzeugnis auszudrucken zu können.

In Zeile 4 geschieht die Bündelung von den digitalen Bildern zu einem Bogen, deren Ausdruck und dem Zuschnitt der einzelnen Elemente.

Anschließend erfolgt die Reproduktion der gedruckten Dokumente (Zeile 5). In der Simulation bedeutet dies die Speicherung und das Wiedereinlesen der codierten Bilder, im

praktischen Feldtest werden ausgewählte, codierte Bilder entwickelt und reproduziert. Zeilen 7 bis 9 beschreiben das Extrahieren der eingebetteten Daten eines digitalisierten Bildes einschließlich der Decodierung und der Zuordnung zu der dazugehörigen Adresse. Um eine fehlerfreie Zuordnung und damit eine fehlererkennende und -korrigierende Decodierung zu gewährleisten, ist unter Umständen eine Speicherung des gefilterten Dokuments notwendig (Zeile 2). Auf diese Weise ist neben fehlerkorrigierenden Maßnahmen zusätzlich ein Vergleich mit den modifizierten Dokumenten möglich, wobei das passendste bzw. am meisten ähnliche der Treffer ist.

3.3.2 Simulation

Der vorgestellte *Algorithmus* wird anhand unterschiedlicher Dokumentvorlagen mit GNU Octave⁶ durchgeführt und dabei geprüft, ob die eingebetteten Daten nach der Reproduktion (hier: nach dem Speichern und Wiedereinlesen) zu erkennen sind. Die Simulation hat zur Intention, die Realisierbarkeit einer Codierung im Frequenzraum zu verifizieren.

In *Simulationsbedingungen* werden die Vorlagen vorgestellt und die Art der Codierung näher beschrieben. Es folgen in *Durchführung* die Simulationsdarstellung der Codierung und Extraktion der Codeworte. Abgerundet wird dieses Unterkapitel mit *Ergebnisse*. Darin werden nach einer Zusammenfassung der Resultate Beobachtungen und Schlussfolgerungen gezogen.

Simulationsbedingungen

Die Auswahl von Vorlagen und die Definition der Codierungsart stellen für die Simulation Voraussetzungen dar. Dabei wird eine breite Abdeckung angestrebt, um mögliche, abhängige Parameter identifizieren zu können.

Vorlagen Die Muster werden so ausgewählt, dass unterschiedliche Bildobjekte abgebildet sind. Es soll getestet werden, wie die Codierung bei hellen oder dunklen Bildern, Bildern mit Text, Kanten oder ohne Kanten im Frequenzraum verarbeitet werden. Die dargestellten Histogramme der Bilder verdeutlichen die unterschiedliche Verteilung der Farbcodewerte (der konvertierten Graustufenbilder). Zusätzlich variieren die Bildmaße, sodass eine möglichst breite Abdeckung gewährleistet ist. Die Maße der Dokumente sind in Höhe × Breite angegeben. Verwendet werden folgende Vorlagen⁷:

⁶https://www.gnu.org/software/octave/

 $^{^7\}mathrm{Mehr}$ Informationen zu den Datei-Formaten ist unter A Überblick über die verwendeten Dateiformate zu finden.

Vorlage (1) – Text (hell): Bildmaße: 628 × 992 Pixel Format: jpg Inhalt: Abbild von schwarzem Text mit Logo auf hellem Hintergrund⁸



Vorlage (2) – Text (dunkel): Bildmaße: 2000 × 2000 Pixel Format: png Inhalt: Abbild von rotem Text mit dunklem Hintergrund⁹



⁸Die urheberrechtlich geschützte Vorlage darf mit freundlicher Genehmigung von *prinux GmbH* (http://prinux.com) genutzt werden.

⁹Dieses Bild wurde mit *Inkscape* erstellt.

Vorlage (3) – Bild mit Konturen: Bildmaße: 529 × 611 Pixel Format: png Inhalt: Abbild einer Münze



Vorlage (4) – Bild mit Konturen Bildmaße: 2159 × 3359 Pixel Format: jpg Inhalt: Abbild einer Playmobil®-Figur mit Schatten und Ball



Vorlage (5) – Bild mit homogener Fläche: Bildmaße: 767 × 1024 Pixel Format: jpg Inhalt: Abbild eines Himmels



Tabelle 3.1: Vorlagen für die Simulation im Frequenzraum

Alle Muster liegen im png- oder jpg-Format vor, sodass nicht zwischen textuellen Bildobjekten oder photographischen Bildobjekten unterschieden wird. Die Verarbeitung aller Dokumente geschieht gleichermaßen. Eine Reskalierung ist hier nicht notwendig, da das gespeicherte und wieder eingelesene Dokument dieselbe Auflösung hat wie das Eingangsbild des Originaldokuments.

Codierung Die Codierung gliedert sich in zwei Teilaspekte. Zum einen bedarf sie der Bestimmung eines Codewortes. In der Simulation werden zwei einfache und ein komplexes Codewort dafür eingesetzt, um das Verhalten und die Grenzen des Lösungsansatzes zu erkennen. Zum anderen ist die Lage der Codeworte bedeutend. Diese befinden sich sowohl im Hoch- als auch im Tieffrequenzbereich.

In Abbildung 3.7 sind die Codeworte dargestellt. Es werden drei unterschiedliche Varianten verwendet, ein schwarzes Quadrat in $(a)^{10}$, ein geometrisches Objekt mit schwarzen und weißen Bereichen in (b) und ein 2D-Barcode (Data Matrix)¹¹ in (c) – als ein Beispiel für ein komplexes Codewort. Aufgrund von minimaler und maximaler Abweichung des codierten Bildes zum Original kann gemessen werden, wie sich die Codierung auf eine wahrnehmbare Modifikation auswirkt.



Abb. 3.7: Verwendete Codeworte für Simulation im Frequenzraum

Abbildung 3.8 skizziert die Lage der Codeworte sowohl im hochfrequenten als auch im tieffrequenten Bereich der Filter. Die exakte Positionierung ist von den Maßen der einzelnen Bildern abhängig. In (c) Codierung im Tieffrequenzbereich mit verlustfreier Speicherung wird näher darauf eingegangen.

 $^{^{10}\}mathrm{Dieses}$ Codewort dient als Veranschaulichung bei der ersten Vorlage.

¹¹Der 2D-Barcode ist unter https://racoindustries.com/barcodegenerator/2d/datamatrix/ generiert worden. Als Barcode-Text wurde ABC 123456789 verwendet.

 $^{^{12}}$ Unter 3.1.3 Codewort
bildung sind zu diesem Codewort mehr Informationen zu finden.



im Hochfrequenzbereich an Positionen Nr. 1, 2, 3 und 4.

Tieffrequenzbereich.



Abb. 3.8: Lage der Codeworte

Die Begründung, dass die Codierung im Hochfrequenzbereich vier Mal platziert ist, ist in Tabelle 3.2 visualisiert. Es wird davon ausgegangen, dass in der Simulation nach der Extraktion eine gute Decodierung möglich sein muss, damit in der Praxis nach der Reproduktion des Dokuments eine Chance besteht, das eingebettete Codewort zu decodieren. Es eignet sich die Vorlage (3) mit dem Bildobjekt der Münze zur Demonstration. Dabei wird das Bild mit **I** codiert und im png-Format gespeichert. Es liegt der Schwerpunkt auf dem Extrahieren und der visuellen Erkennbarkeit des Codewortes.



ten bei der Bildung der FFT erscheint nach dem Einlesen die Codierung redundant. Das hat zur Auswirkung, dass sich die Codierung nicht nur im unteren rechten Eck des Bildes (viertes Quarant) wiederfindet, sondern auch oben links im ersten Quadranten. Es kann eine Optimierung stattfinden, indem beide Stellen mit dem Codewort extrahiert werden. In (a) ist das Codewort im FFT nicht zu erkennen. In (c) ist das Codewort zu erahnen, jedoch nicht signifikant.



Tabelle 3.2: Begründung der Codierung im Hoch- und Tieffrequenzbereich

Wie das folgende Unterkapitel zeigt, kann festgestellt werden, dass selbst die Redundanz der Codeworte bei der Codierung im Hochfrequenzbereich beim Extrahieren nicht immer zu einem Erkennen des Codewortes führt.

Die Lage des eingebetteten Codewortes im Tieffrequenzbereich ist entscheidend, ob die Codierung eine visuell wahrnehmbare Modifikation am Bild als Auswirkung hat. Eine redundante Codierung bei (b) (siehe *Abbildung 3.8*) hätte zur Folge, dass das Dokument

sichtbar verändert ist. Zur Verdeutlichung des Effekts einer Codierung im Tieffrequenzbereich wird die Simulation sowohl mit Filter der Art (b) als auch (c)(*Abbildung 3.8*) durchgeführt.

Speichervorgang Um die Relevanz des Speicherformats zu validieren, findet die Persistierung einerseits verlustbehaftet, andererseits verlustfrei statt. Die verlustfreien Formate haben unterschiedliche Eigenschaften, sodass hier eine zufällige, mehrfache Auswahl getroffen wird. Die Verwendung von diversen verlustfreien Formaten geschieht ebenfalls im Hinblick auf das Speicherformat als einen potentiellen abhängigen Parameter bei der Verarbeitung der Codierung mit FFT.

Mehr Informationen zu den Datei-Formaten ist unter A Überblick über die verwendeten Dateiformate zu finden.

- (a) Die Muster werden im Hochfrequenzbereich modifiziert und verlustfrei im bmp-, pngund jp2-Format¹³ gespeichert.
- (b) Die Muster werden im Hochfrequenzbereich modifiziert und verlustbehaftet im jpg-Format gespeichert.
- (c) Die Muster werden im Tieffrequenzbereich modifiziert und verlustfrei im bmp-, pngund jp2-Format gespeichert.
- (d) Die Muster werden im Tieffrequenzbereich modifiziert und verlustbehaftet im jpg-Format gespeichert.

Bemerkung 3.7 (Speicherung). Beim Speichervorgang der codierten Bilder findet eine Konvertierung in uint 16^{14} statt. Dies ist eine wichtige Voraussetzung für die nachfolgenden Simulationen, da mit dieser Intensität der Bilder die extrahierten Codeworte besser erkannt werden als mit uint 8^{15} .

Durchführung

Die Bilder liegen im *png-* oder *jpg-*Format vor, werden im hochfrequenten Frequenzbereich codiert und im verlustfreien bmp- , png- und jp2-Format gespeichert, sodass hochfrequente Informationen der Bilder nicht verloren gehen. Die Dokumente werden als Graustufenbild verarbeitet.

Es folgt eine tabellarische Darstellung der Codierung und der Extraktion der eingebetteten Daten. Auf die Dekodierbarkeit wird detaillierter in *3.3.2 Ergebnisse* Bezug genommen. Die Notation und weitere Hinweise zu den Ergebnistabellen sind in *Tabelle 3.3* gelistet.

 $^{^{13}}$ Dieses Format kann bis zu einer verlustfreien Qualität Rastergrafiken speichern. Hier findet die Speicherung verlustfrei statt.

¹⁴Die Funktion unit16 konvertiert in Matlab einen Vektor zu einem Integer, der 2¹⁶ mögliche Werte zwischen 0 und 65.535 annehmen kann. unit steht für unsigned integer.

 $^{^{15}}$ Die Funktion unit
8 konvertiert in Matlab einen Vektor zu einem Integer, der
 2^8 mögliche Werte zwischen 0 und 255 annehmen kann.

$\mathbf{D}\mathbf{A}:=$ Durchschnittliche Abweichung zwischen Original und codiertem Dokument
$\mathbf{MA} := $ Maximale Abweichung zwischen Original und codiertem Dokument
$Dokument \Leftrightarrow Bild$
Codewort redundant := Multiplikation der extrahierten, redundanten Code- worte (Beschreibung siehe <i>Tabelle 3.2</i>)
$[\mathbf{X}]$ [21] := Das extrahierte Codewort ist nicht erkennbar ¹⁶ .
\checkmark [21] := Das extrahierte Codewort ist erkennbar.
$\boxed{?}$ [21] := Das extrahierte Codewort ist nicht deutlich erkennbar.
Tabelle 3.3: Notation und Hinweise zu den Ergebnistabellen

(a) Codierung im Hochfrequenzbereich mit verlustfreier Speicherung

Die Codierung bei der ersten Mustervorlage mit dem einfachen, schwarzen Quadrat wird zur Veranschaulichung die Größe 50×50 verwendet. Dabei wird der Ablauf nochmals visuell verdeutlicht. Anschließend werden der codierte Filter, FFT des Dokuments und das gefilterte Dokument nicht mehr abgebildet.

Vorlage (1): Visitenkarte In *Abbildung 3.9* sind die einzelnen Schritte der Codierung dargestellt¹⁷.



Abb. 3.9: Vorlage (1): Simulation der Codierung im Frequenzraum

 $^{^{16}\}mathrm{Die}$ Bewertung beruht auf Erkennbarkeit der Struktur der Codierung.

¹⁷Um die Unterschiede besser sichtbar zu machen, wird in diesem Kapitel meist colormap(pink) verwendet.

In den Tabellen ist an der Abbildung der Unterschiede zwischen Original und codiertem Dokument zu erkennen, wie die Codierung im Frequenzraum verteilt wird. Die durchschnittlichen und maximalen Abweichungen, die durch die Modifikationen entstehen, können als Indiz für visuelle Wahrnehmbarkeit verwendet werden.

Exemplarisch wird die Codierung mit dem 2D-Barcode bei Vorlage (1) gelistet¹⁸.

 $^{^{18}\}mathrm{Die}$ restlichen Simulationen mit dem 2D-Barcode befinden sich im Anhang unter Tabelle 4.1



werden. Das Resultat in (c) zeigt eine sichtbare Verbesserung.



Der Code ist darauf hell und sichtbar. (c) zeigt, dass die schwarze: Bereiche des Codewortes im bmp-Format umgekehrt sind.



¹⁹Abweichungen und Unterschiede siehe Vorlage (1) mit demselben Codewort. Des weiteren ist das codierte Dokument vor der Speicherung identisch, weswegen auf eine zusätzliche Vergleichs-Darstellung von Original und codiertem Bild bei allen folgenden Vorlagen im jp2- und png-Format verzichtet wird.



ist und nicht decodiert werden kann.



Tabelle 3.5: Vorlage (1): Simulation der Codierung im Frequenzraum im hochfrequenten Bereich (verlustfreie Speicherung)







Tabelle 3.6: Vorlage (2): Simulation der Codierung im Frequenzraum im hochfrequenten Bereich (verlustfreie Speicherung)




Tabelle 3.7: Vorlage (3): Simulation der Codierung im Frequenzraum im hochfrequenten Bereich (verlustfreie Speicherung)





Tabelle 3.8: Vorlage (4): Simulation der Codierung im Frequenzraum im hochfrequenten Bereich (verlustfreie Speicherung)





Tabelle 3.9: Vorlage (5): Simulation der Codierung im Frequenzraum im hochfrequenten Bereich (verlustfreie Speicherung)

(b) Codierung im Hochfrequenzbereich mit verlustbehafteter Speicherung

Die Auswirkungen einer verlustbehafteten Speicherung im jpg-Format zeigen sich in Tabelle 3.10.

Wie bereits bei der verlustfreien Speicherung zu erkennen war, ist die Decodierung der extrahierten Daten mit dem 2D-Barcode schwer möglich. Dieses Unterkapitel beschränkt sich auf die Illustration mit dem bekannten schwarz-weiß Codewort (Maße 9×9 Pixel)

Da die Einbettung der Daten dieselben sind wie bei der verlustfreien Codierung im Hochfrequenzbereich, werden die Unterschiede zum Original, die minimalen und maximalen Abweichungen und das codierte Dokument nicht mehr dargestellt.



Ausschnitt 4. Quadrant

×

×





Tabelle 3.10: Simulation der Codierung im Frequenzraum im hochfrequenten Bereich (verlustbehaftete Speicherung)

(c) Codierung im Tieffrequenzbereich mit verlustfreier Speicherung

Die Codierung findet im tieffrequenten Bereich der Bilder statt. Die Bilder werden im verlustfreien bmp-, png- und jp2-Format gespeichert.

Auch hier erfolgt die Einschränkung zur Codierung mit dem schwarz-weiß Codewort (Maße 9×9 Pixel)

Wie in *Abbildung 3.8* dargestellt, wird zum einen in der Mitte des transformierten Bildes codiert, zum anderen im vierten Quadranten. Dabei ist die untere, rechte Ecke des Codewortes vom unteren und rechten Rand jeweils $\frac{1}{3}$ der Breite bzw. der Höhe des Bildes entfernt.

Die Codierung in Zentrum des FFT-Bildes soll lediglich die hohe Intensität der Modifikation widerspiegeln. Eine Darstellung der Codewort-Extraktion ist daher nicht notwendig.







Tabelle 3.12: Vorlage (1): Simulation der Codierung im Frequenzraum im tieffrequenten Bereich (verlustfreie Speicherung)









Tabelle 3.14: Vorlage (2): Simulation der Codierung im Frequenzraum im tieffrequenten Bereich (verlustfreie Speicherung)







Tabelle 3.16: Vorlage (3): Simulation der Codierung im Frequenzraum im tieffrequenten Bereich (verlustfreie Speicherung)









Tabelle 3.18: Vorlage (4): Simulation der Codierung im Frequenzraum im tieffrequenten Bereich (verlustfreie Speicherung)









Tabelle 3.20: Vorlage (5): Simulation der Codierung im Frequenzraum im tieffrequenten Bereich (verlustfreie Speicherung)

(d) Codierung im Tieffrequenzbereich mit verlustbehafteter Speicherung

In der Simulation werden die Bilder das schwarz-weiß Codewort (Maße 9×9 Pixel) \blacksquare eingebettet und im jpg-Format gespeichert. Wie beim bmp-Format sind bei einem erkennbaren Codewort die Farbbereiche vom Wert her umgekehrt: Überwiegend helle Farbcodes entsprechen einem schwarzen Bereich und analog bei extrahierten, dunklen Farbcodes dem weißen Teil des Codewortes. Nachstehende *Tabelle 3.21* bietet eine Übersicht über die Resultate der Simulationen der Codierung im tieffrequenten Bereich.









Tabelle 3.21: Simulation der Codierung im Frequenzraum im tieffrequenten Bereich (verlustbehaftete Speicherung)

Ergebnisse

Es wird zu den Simulationsergebnissen der Codierung mit dem Codewort \blacksquare Bezug genommen. In *Tabelle 3.23* und *Tabelle 3.24* werden die gemessenen Ergebnisse der Simulationen zusammenfassend gelistet, um konkreter Beobachtungen machen und Schlussfolgerungen ziehen zu können. Die Legende dazu ist in *Tabelle 3.22* abgebildet.

 $\mathbf{CW}:=\mathrm{Codewort}$

 $\mathbf{D}\mathbf{A}:=\operatorname{Durchschnittliche}$ Abweichung zwischen Original und codiertem Dokument

 $\mathbf{MA} := Maximale Abweichung zwischen Original und codiertem Dokument²⁰$

 $\mathbf{u.r.}{:=}$ Extraktion des Codewortes im vierten Quadranten

 $\mathbf{red.} := \mathbf{Extraktion} \text{ des Codewortes redundant} (alle extrahierten Codeworte multipliziert)}$ $\mathbf{KB} := \mathbf{Kilobyte}$

 $V(x), x \in \{1, 2, 3, 4, 5\} :=$ Vorlage (x), gemäß Tabelle 3.1:

- V(1) Abbild der Visitenkarte, 992 × 628 Pixel

- V(2) Abbild des Textes mit dunklem Hintergrund, 2000 × 2000 Pixel
- V(3) Abbild der Geldmünze, 611 × 529 Pixel
- V(4) Abbild der Playmobil[®]-Figur, 3359 × 2159 Pixel
- V(5) Abbild des Himmels, 1024×767 Pixel

Tabelle 3.22: Notation und Hinweise zu der Übersichtstabelle

ile	$\mathbf{V}(\mathbf{x})$	CW	МА	DA	E	Extraktion		KB
Z	V (A)			DA	Format	u.r.	red.	ND
1	V(1)		0,250827	0,025399	bmp			1.826
2					png			225
3					jp2	5		150
4					jpg		8	40

Codierung im Hochfrequenzbereich

²⁰Die Werte sind bei DA und MA auf sechs Stellen hinter dem Komma gerundet.

5	V(2)	0,029434	0,003896	bmp	5		11.719
6				png	i k		521
7				jp2	1	5	514
8				jpg	23	쮏	131
9	V(3)	0,273671	0,048157	bmp	嚻		949
10				png	dire.	202	369
11				jp2			120
12				jpg	凝	쮋	20
13	V(4)	0,007881	0,001697	bmp	뽩		21.253
14				png		100	3.694
15				jp2	1	100	2.601
16				jpg			196

17	V(5)	0,005590	0,000787	bmp	1	1	2.302
18				png	1	2	69
19				jp2	纐		213
20				jpg	£.	醚	9

Tabelle 3.23: Zusammenfassende Auflistung der Simulationen mit Codierung im hochfrequenten Bereich

eile	$\mathbf{V}(\mathbf{x})$	CW	МА	DA	E	Extraktion		KB
Z	V (X)			DA	Format	u.r.	red.	
21	V(1)		0,858132	0,113535	bmp	59		1.826
22					png	200 100	3	503
23					jp2		3 2	186
24					jpg	100	3	41

Codierung im Tieffrequenzbereich

25	V(2)	0,012935	0,002363	bmp	鑻	簚	11.719
26				png	鍜	10	570
27				jp2	韗		516
28				jpg	1	5	131
29	V(3)	0,208946	0,035096	bmp	鯼	洒	949
30				png		505	330
31				jp2		1	120
32				jpg		鶅	21
33	V(4)	0,015441	0,003608	bmp	簫	簚	21.253
34				png	油 的		4.834
35				jp2	建設		2.644
36				jpg			237

37	V(5)	0,006076	0,001156	bmp			2.302
38				png	63	8	108
39				jp2	62	8	217
40				jpg	霺	8.0	9

Tabelle 3.24: Zusammenfassende Auflistung der Simulationen mit Codierung im tieffrequenten Bereich

Simulationsbeobachtungen Die Beobachtungen der Simulation werden als $SB_y(x), x \in \mathbb{N}, y \in \{HB, TB, \star\}$ dokumentiert. HB bezieht sich ausschließlich auf die Codierung im Hochfrequenz-, TB auf die im Tieffrequenzbereich und \star auf beide Bereiche.

- $SB_{HB}(1)$ Bei mindestens einem der verlustfrei gespeicherten Vorlagen (1), (2), (3), (4) und (5) ist das extrahierte Codewort mit der Redundanz wieder zu erkennen, siehe *Tabelle 3.23*, Zeilen 1-3, 5-7, 9-11, 13-15, 17-19.
- SB_{HB}(2) Die extrahierte Codierung ist bei Vorlage (2) in png- und jp2-Format in der Spalte u.r., Zeilen 6 und 7 (*Tabelle 3.23*) deutlich. Dies kann daran liegen, dass das Muster lediglich zwei Farbcodewerte besitzt und zusätzlich dunkel ist.
- $SB_{HB}(3)$ Welches verlustfreie Format zum Speichern zu bevorzugen ist, kann aus der Simulation nicht herausgelesen werden. Es scheint abhängig von der Vorlage zu sein, welches besser geeignet ist.
- $SB_{HB}(4)$ Im verlustbehaftet gespeicherten jpg-Format ist die Codierung nach dem Wiedereinlesen nicht sichtbar, wie in *Tabelle 3.23* in Zeilen 4, 8, 12 und 16 zu sehen.
- $SB_{HB}(5)$ Wird die *Tabelle 3.23* betrachtet, so benötigt bmp mehr Speicherplatz als png und jp2, bedingt durch die jeweiligen Eigenschaften der Formate.
- $SB_{HB}(6)$ Die Abweichungen zwischen Original und codierten Bildern deuten darauf hin, dass die Codierung nicht zu entschlüsseln ist, wenn sich die durchschnittliche Abweichung im 10.000stel Bereich befindet wie bei Vorlage (5), Zeilen 17-20 in *Tabelle 3.23*. Die höchste, durchschnittliche Abweichung ist bei Vorlage (3) mit 0,048157 (Zeile 9), die niedrigste bei Vorlage (5) mit 0,000787 (Zeile 17).

- $SB_{HB}(7)$ Liegt die durchschnittliche Abweichung im 100stel oder sogar im 1.000stel Bereich, so kann das extrahierte Codewort erkannt werden²¹, *Tabelle 3.23* Zeilen 1, 2, 3, 5, 7, 9, 10, 11, 13, 14, 15 (Multiplikation redundanter Codeworte).
- SB_{HB}(8) Ob der Wert der maximalen Abweichung allerdings von den Maßen des Bildes oder der Verteilung der Farbcodewerte im Histogramm oder eventuell von beiden Parametern abhängig ist, ist hier nicht herauszulesen und müsste in einer weitergehenden Untersuchung ermittelt werden.
- $SB_{TB}(9)$ Werden die Bilder im tieffrequenten Bereich codiert, so ist das Codewort bei Vorlage (2), (3), (4) und (5) bei mindestens einem der verlustfrei gespeicherten Formate zu erkennen, siehe *Tabelle 3.24* Zeilen 25-27, 29-31, 33-35, 37-39.
- SB_{TB}(10) Aufgrund der Eigenschaften und Bildinhalte der Vorlagen ist es möglich, dass ein bestimmtes verlustfreies Speicherformat besser geeignet ist: Bei Vorlage (2), (4), (5) ist das Codewort nach dem Speichern im bmp-Format, bei Vorlage (3) im jp2-Format am deutlichsten, nach *Tabelle 3.24* Zeilen 25, 31, 33, 37.
- $SB_{TB}(11)$ Es kann an der durchschnittlichen Abweichung liegen, dass das Codewort bei Vorlage (5) im gespeicherten bmp-Format deutlicher zu erkennen ist. Diese befindet sich mit 0,001156 (Zeile 35 in *Tabelle 3.24*) im 1.000stel Bereich und nicht – wie bei der hochfrequenten Codierung – im 10.000stel.
- SB_{TB}(12) Die Codeworte bei Vorlage (4) und (5) im verlustbehaftet gespeicherten Format können identifiziert werden, wie in den Zeilen 36 und 40 Spalte **red.** abgebildet. Sogar bei der Extraktion der Codierung im vierten Quadranten sind sie zu erkennen.
- $SB_{TB}(13)$ Bei Vorlage (1) ist die Codierung nicht herauszulesen, wie in den Zeilen 21-24 (*Tabelle 3.24*) zu sehen ist. Dies kann an der hellen Vorlage liegen.
- $SB_{TB}(14)$ Die höchste durchschnittliche Abweichung zwischen Original und codiertem Dokument ist bei Vorlage (1) mit 0,113535, die niedrigste bei Vorlage (5) mit 0,001156. Die Abweichung liegt auch bei Vorlage (2) und(4) im 1000stel Bereich, siehe *Tabelle 3.24*, Zeilen 21, 25, 29, 33, 37.
- SB_{*}(15) Nach der detaillierten Darstellung der Simulationsergebnisse ist beim Codieren am Abbild der Unterschiede die "globale" Verteilung im Bild zu sehen und nach der Rekonstruktion nicht wahrnehmbar.
- $SB_{\star}(16)$ Das Codewort darf nicht zu fein strukturiert sein wie ein 2D-Barcode. Eine Erkennung nach dem Extrahieren bleibt aus, siehe *Tabelle 4.1*.
- $SB_{\star}(17)$ Die Decodierung bei Vorlage (2) ist deutlicher bei einer Codierung im Tieffrequenz- als im Hochfrequenzbereich (Zeilen 5-7 bzw. 25-27 in *Tabelle 3.23* und *Tabelle 3.24*).

²¹Dies ist durchaus eine subjektive Bewertung.

- SB_{*}(18) Das Codewort ist bei Vorlage (3) im Hochfrequenzbereich nach dem Speichern im bmp- und png-Format am deutlichsten erkennbar, tieffrequent codiert im jp2-Format (Zeilen 9-11 bzw. 29-31 in *Tabelle 3.23* und *Tabelle 3.24*).
- $SB_{\star}(19)$ Bei Vorlage (4) ist eine hochfrequente Codierung in allen verlustfrei gespeicherten Formaten nach dem Extrahieren zu erkennen, eine tieffrequente im gespeicherten bmp- und jpg-Format (Zeilen 13-15 bzw. 33 und 36 in *Tabelle 3.23* und *Tabelle 3.24*).
- $SB_{\star}(20)$ Die extrahierte Codewort bei Vorlage (5) ist nach einer tieffrequenten Codierung mit Speicherung im bmp-Format deutlicher zu erkennen als nach einer hochfrequenten Codierung, ebenfalls im bmp-Format persistiert (Zeilen 17 bzw. 37 in *Tabelle 3.23* und *Tabelle 3.24*).

Schlussfolgerungen Zusammenfassend zeigen die Simulationen im Frequenzbereich, dass der vorgestellte Algorithmus zur Codierung von versteckten Informationen in *3.3.1 Algorithmus* eine mögliche Lösung darstellt.

Bezüglich der Einbettung von Informationen im Frequenzraum werden für eine optimiertere Lösung folgende Annahmen und Schlussfolgerungen getroffen, die als Schlussfolgerungen $SF_S(x), x \in \mathbb{N}, S :=$ "Simulation" festgehalten sind.

- $SF_S(1)$ Die Bildung des Codewortes kann abhängig von der Größe des Druckerzeugnisses sein. Je größer das Dokument ist, verträgt es ein umso größeres Codewort.
- $SF_S(2)$ Die durchschnittliche Abweichung zwischen Original und codiertem Dokument gilt als ein Indiz für eine gute Codierung bzw. Decodierung.
- $SF_S(3)$ Die Qualität bzw. Erkennbarkeit der Codierung und Decodierung ist abhängig von den enthaltenen Farbcodewerten und deren Verteilung bei den Bildern.
- $SF_S(4)$ Zur Fehlerkorrektur beim Decodiervorgang besteht folgende Optimierungsoption: Das codierte Druckerzeugnis wird gespeichert und beim Decodieren miteinbezogen. Schlägt das Decodieren des Codewortes eines reproduzierten Bildes fehl, so kann es mit dem gesicherten, codierten Dokument verglichen werden. Das am Ähnlichste kann als Ergebnis verwendet werden.

3.3.3 Praktischer Test

Nach der Durchführung der Simulationen erfolgt ein erster praktischer Test. Die Codierung erfolgt gemäß den Ausführungen in *Simulationsbedingungen*. Beim vorgestellten Algorithmus in 2 werden die Schritte 4 und 5 ausgeführt. Es wird geprüft, ob nach der digitalen Reproduktion durch Ablichten ausgedruckter Bilder die extrahierte Codierung zu erkennen ist.

Die Intention beim Test ist, dass ein ähnlicher Ablauf wie bei den Simulationen stattfindet, das bedeutet, dass das Verhalten der Extraktion sowohl im verlustfrei als auch im verlustbehaftet gespeicherten Format validiert wird. Deswegen erfolgt die Speicherung beim Ablichten zum einen im cr 2^{-22} , zum anderen im jpg-Format.

In diesem Unterkapitel werden nach den *Testbedingungen* die *Durchführung* präsentiert und anschließend in *Ergebnisse* die Beobachtungen und Schlussfolgerungen festgehalten.

Testbedingungen

Da der praktische Test lediglich einen ersten Eindruck vermitteln soll, wird eine Selektion der Bilder verwendet. Sie ist relevant zur Hypothesenbildung, sodass sie zuvor vorgestellt wird. Danach folgen in diesem Unterabschnitt *Versuchsaufbau* und *Ablichtbedingungen*.

Auswahl der Bilder Es eignen sich diejenigen codierten Bilder, bei denen nach dem Speichervorgang das extrahierte Codewort zu erkennen war. Es geschieht dadurch automatisch die Einschränkung auf die Codierung mit dem schwarz-weiß Codewort (Maße 9×9 Pixel) \blacksquare , da bei der Codierung mit dem 2D-Barcode eine erkennbare Decodierung in den Simulationen ausbleibt.

Die Selektion der Bilder in Übersicht:

Codierung im Hochfrequenzbereich, verlustfreie Speicherung

- 1. Vorlage (3), gespeichert im bmp-Format
- 2. Vorlage (3), gespeichert im png-Format

Codierung im Tieffrequenzbereich, verlustfreie Speicherung

- 3. Vorlage (4), gespeichert im bmp-Format
- 4. Vorlage (5), gespeichert im bmp-Format

Codierung im Tieffrequenzbereich, verlustbehaftete Speicherung

- 5. Vorlage (4), gespeichert im jpg-Format
- 6. Vorlage (5), gespeichert im jpg-Format

Hypothesen Nachfolgend werden die Hypothesen $H(x), x \in \mathbb{N}$ vorgestellt.

 ${\rm H}(1)$ Die Codierung der Bilder, die beim Ablichten im verlustbehafteten jpg-Format gespeichert werden, ist nach der Extraktion nicht erkennbar.

 $^{^{22}}$ Mehr Informationen unter Canon RAW version 2 (CR2).

- H(2) Das extrahierte Codewort bei Bildern, die nach der Codierung im jpg-Format (vor dem Ausdruck) gespeichert werden (Vorlage (4) und (5)), ist nicht erkennbar.
- H(3) Die Codierung im Tieffrequenzbereich ist im Gegensatz zu einer hochfrequenten Codierung nach dem Extraktionsvorgang der photographierten Bilder zu erkennen.
- H(4) Das extrahierte Codewort ist bei einer verlustfreien Speicherung beim Photographieren abhängig von der Vorlage erkennbar: Es besteht die Hypothese, dass bei einem Bild mit vorhandenen Kanten (Vorlage (4)) die Codierung wahrnehmbar ist.
- ${\rm H}(5)\,$ Die extrahierte Codierung ist bei einer Photographie mit der Einstellung des Bildsensors auf ISO 500 deutlicher erkennbar als auf ISO 400.

Versuchsaufbau Der praktische Test hat folgenden Ablauf:

- 1. Durchführung der Simulationen an ausgewählten Bildern
- 2. Qualitativ hochwertige Entwicklung der Bilder in zwei Photo-Ateliers Passau^{23} auf mattem Photopapier
- 3. Photographie der entwickelten Bilder mit Canon EOS 70D
- 4. Extraktion des Codewortes mit Matlab R2016a

Bemerkung 3.8 (Entwicklung der Bilder). In den Photo-/Ateliers liegen Einschränkungen bei der Entwicklung bezüglich der vorliegenden Datei-Formate vor. Aus diesem Grund werden codierte Bilder, die im bmp-, png- und jpg-Format gespeichert sind, ausgedruckt. Es wird darauf geachtet, dass die digitalen Dokumente im Prozess nicht komprimiert werden, sodass hier weitgehend ein Informationsverlust auszuschließen ist.

Bemerkung 3.9 (Papierbeschaffenheit). Das qualitativ hochwertige, matte Papier der entwickelten Bilder weist eine leichte Struktur auf, die selbst beim reproduzierten Bild durch Vergrößern erkennbar ist.

Ablichtbedingungen Beim Photographieren der Bilder sind nachfolgende Bedingungen gegeben.

		Eigenschaft	Wert bzw. Beschreibung	
	1.	Тур	Canon EOS 70D	
sra	2.	Objektiv	Canon EF-S 18-55mm 1:3,5-5,6 IS STM Objektiv (58mm Filtergewinde)	
Came	3.	Lichtempfindlichkeit des Bildsensensors	ISO 500, ISO 400	
	4.	Blende	4.0	
	5.	Verschlusszeit	0,01 Sekunden	

²³Kaps, Lichtbox

sse	6. Tag und Uhrzeit	23. Februar 2018, 12:30 bis 13:30 Uhr
hältnis	7. Räumlichkeit	Audimax-Cafeteria Universität Passau, keine direkte Sonneneinstrahlung
Ver	8. Wetter	teils sonnig, bewölkt

Tabelle 3.27: Ablichtbedingungen beim praktischen Test

Die Abbildung 3.10 visualisiert die Testbedingungen in der Audimax-Cafeteria der Universität Passau. Die Camera ist an einem Stativ fixiert. Des Weiteren wurde beim Photographieren zusätzlich in der Displayanzeige eine Wasserwaage und ein Gitter zur Ausrichtung eingeblendet, um Verzerrungen vorzubeugen. Wie in der Auswertung zu sehen, besteht eine maximale Korrektur bei der Rotation von $-0, 5^{\circ}$ und 0.45° .

Beim Ablichten eines Photos speichert die Camera gleichzeitig eine Datei im cr2- und im jpg-Format.



Abb. 3.10: Praktische Testbedingungen

Durchführung

In nachfolgender Auflistung (beginnend mit *Vorlage (3)*) werden die Ergebnisse dargestellt. Wie in den Simulationen wird eine FFT des Bildes durchgeführt und anschließend die Positionen, an denen die Codeworte bei der Codierung platziert wurden, extrahiert. Es findet bei jedem Bild eine Reskalierung (siehe *3.1.1 Reskalierung der reproduzierten Dokumente*) statt. Statt der Translation erfolgt ein digitaler Zuschnitt der Bilder. Die Wertigkeit der jeweiligen Aktion wird in der Tabelle ebenfalls dokumentiert.

Bemerkung 3.10 (Notation der Maße). Die Bildmaßangaben werden in der Auflistung folgendermaßen notiert: Höhe \times Breite des Bildes.

Bemerkung 3.11 (Abweichungen der Maße). Der Zuschnitt und die Rotation der Bilder geschieht in der Auswertung "manuell", d.h. es wird keine Formel, sondern reines Augenmaß verwendet. Das hat zum einen den Grund, dass beim Ablichten kein Fixpunkt gegeben war, zum anderen weist das Papier der entwickelten Bilder Wölbungen auf. Dadurch sind gerade Zuschnitte an den Rändern nicht möglich. In diesem Fall wird die Gerade gewählt, die den äußersten Punkt des Bildes (höchster bzw. minimaler Wert der Wölbung) noch enthält.

Bemerkung 3.12 (Codewort im bmp- und jpg-Format). Wie bei den Simulationen bereits erwähnt, ist das Verhalten der Farbwerte einer Codierung im bmp- und jpg-Format gespeicherten Bild konträr – die weißen Farbwerte sind schwarz und umgekehrt.

Vorlage	(3),	\mathbf{ISO}	500
---------	------	----------------	-----

– Speicherung nach Codierung im bmp-Format

	Eckdaten		
Bildmaße (bei Entwicklung)	$18,5\times21,5~\mathrm{cm}$		
Bildmaße Original	529×611 Pixel		
	jpg	cr2	
Bildgröße (digital)	4,88 MB	26,4 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2947 \times 3390 \text{ Pixel} \\ (5,55 \times \text{ Original}) \end{array}$	$\begin{array}{l} 2940 \times 3395 \text{ Pixel} \\ (5,57 \times \text{ Original}) \end{array}$	
Rotation	$-0,5^{\circ}$	$-0,5^{\circ}$	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)







(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)







(f) Codewort redundant

Tabelle 3.28: Vorlage(3), Codierung im Hochfrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 500

Vorlage	(3),	\mathbf{ISO}	400
---------	------	----------------	-----

– Speicherung nach Codierung im bmp-Format

	Eckdaten		
Bildmaße (bei Entwicklung)	$18,5\times21,5~\mathrm{cm}$		
Bildmaße Original	529×611 Pixel		
	jpg	cr2	
Bildgröße (digital)	4,99 MB	25,3 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2947 \times 3390 \text{ Pixel} \\ (5,55 \times \text{ Original}) \end{array}$	$\begin{array}{l} 2940 \times 3395 \text{ Pixel} \\ (5,57 \times \text{ Original}) \end{array}$	
Rotation	$-0,5^{\circ}$	$-0,5^{\circ}$	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)







(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)







(f) Codewort redundant

Tabelle 3.29: Vorlage(3), Codierung im Hochfrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 400

Vorlage	(3),	\mathbf{ISO}	500
---------	------	----------------	-----

– Speicherung nach Codierung im $\mathit{png}\text{-}\mathsf{Format}$

Eckdaten			
Bildmaße (bei Entwicklung)	$18,5\times21,5~\mathrm{cm}$		
Bildmaße Original 529×611 Pixel			
	jpg	cr2	
Bildgröße (digital)	4,81 MB	26,3 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2940 \times 3396 \text{ Pixel} \\ (5,55 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2942 \times 3393 \text{ Pixel} \\ (5,56 \times \text{ Original}) \end{array}$	
Rotation	$-0,5^{\circ}$	$-0,5^{\circ}$	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)

(c) Codewort

redundant





(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)







Tabelle 3.30: Vorlage(3), Codierung im Hochfrequenzbereich, Speicherung im png-Format, Ablichtung mit ISO 500

Vorlage	(3),	\mathbf{ISO}	400
---------	------	----------------	-----

– Speicherung nach Codierung im png-Format

Eckdaten			
Bildmaße (bei Entwicklung)	$18,5\times21,5~\mathrm{cm}$		
Bildmaße Original 529×611 Pixel			
	jpg	cr2	
Bildgröße (digital)	4,94 MB	25,2 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2940 \times 3396 \text{ Pixel} \\ (5,55 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2942 \times 3393 \text{ Pixel} \\ (5,56 \times \text{ Original}) \end{array}$	
Rotation	$-0,5^{\circ}$	$-0,5^{\circ}$	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)







(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)



(e) Codewort4. Quadrant



(f) Codewort redundant

Tabelle 3.31: Vorlage(3), Codierung im Hochfrequenzbereich, Speicherung im png-Format, Ablichtung mit ISO 400

Vorlage	(4),	\mathbf{ISO}	500
---------	------	----------------	-----

– Codierung im Tieffrequenzbereich

– Speicherung nach Codierung im bmp-Format

Eckdaten			
Bildmaße (bei Entwicklung)	$19\times29~{\rm cm}$		
Bildmaße Original 2159×3359 Pixel			
	jpg	cr2	
Bildgröße (digital)	5,42 MB	26,6 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2918 \times 4519 \text{ Pixel} \\ (1,35 \times \text{Original}) \end{array}$	$\begin{array}{l} 2923 \times 4519 \text{ Pixel} \\ (1,35 \times \text{ Original}) \end{array}$	
Rotation	$-0, 2^{\circ}$	$-0, 3^{\circ}$	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)



(b) Codewort4. Quadrant



(c) Codewort redundant



(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)



(e) Codewort4. Quadrant



(f) Codewort redundant

Tabelle 3.32: Vorlage(4), Codierung im Tieffrequenzbereich, Speicherung bmp-Format, Ablichtung mit ISO 500

Vorlage	(4),	\mathbf{ISO}	400
---------	------	----------------	-----

– Codierung im Tieffrequenzbereich

– Speicherung nach Codierung im bmp-Format

Eckdaten			
Bildmaße (bei Entwicklung)	$19\times29~{\rm cm}$		
Bildmaße Original 2159×3359 Pixel			
	jpg	cr2	
Bildgröße (digital)	5,63 MB	26,0 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2961 \times 4580 \text{ Pixel} \\ (1, 36 \times \text{ Original}) \end{array}$	$\begin{array}{l} 2966 \times 4578 \text{ Pixel} \\ (1, 36 \times \text{ Original}) \end{array}$	
Rotation	-0,1°	$-0, 1^{\circ}$	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)



(b) Codewort4. Quadrant



(c) Codewort redundant



(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)



4. Quadrant



(f) Codewort redundant

Tabelle 3.33: Vorlage(4), Codierung im Tieffrequenzbereich, Speicherung bmp-Format, Ablichtung mit ISO 400

Vorlage	(4),	\mathbf{ISO}	500
---------	------	----------------	-----

– Codierung im Tieffrequenzbereich

– Speicherung nach Codierung im jpg-Format

Eckdaten			
Bildmaße (bei Entwicklung)	$19\times29~{\rm cm}$		
Bildmaße Original 2159×3359 Pixel			
	jpg	cr2	
Bildgröße (digital)	4,99 MB	26,7 MB	
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel	
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{l} 2963 \times 4581 \text{ Pixel} \\ (1, 36 \times \text{ Original}) \end{array}$	$\begin{array}{l} 2925 \times 4516 \text{ Pixel} \\ (1, 34 \times \text{Original}) \end{array}$	
Rotation	0,4°	0,4°	



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)



(b) Codewort4. Quadrant



(c) Codewort redundant



(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)



(e) Codewort4. Quadrant

83

(f) Codewort redundant

Tabelle 3.34: Vorlage(4), Codierung im Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 500
Vorlage	(4),	\mathbf{ISO}	400
---------	------	----------------	-----

– Speicherung nach Codierung im jpg-Format

Eckdaten						
Bildmaße (bei Entwicklung)	$19\times29~{\rm cm}$					
Bildmaße Original	2159×3359 Pixel					
	jpg	cr2				
Bildgröße (digital)	5,45 MB	25,9 MB				
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel				
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2961 \times 4580 \text{ Pixel} \\ (1, 36 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2963 \times 4581 \text{ Pixel} \\ (1, 36 \times \text{ Original}) \end{array}$				
Rotation	$0, 2^{\circ}$	$0,15^{\circ}$				



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)





(c) Codewort redundant



(d) FFT des Bildes im cr2-Format mit Reskalierung (Originalmaße)



(e) Codewort 4. Quadrant



(f) Codewort redundant

Tabelle 3.35: Vorlage(4), Codierung im Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 400

Vorlage	(5),	\mathbf{ISO}	500
---------	------	----------------	------------

– Speicherung nach Codierung im bmp-Format

Eckdaten					
Bildmaße (bei Entwicklung)	$14\times18,5~\mathrm{cm}$				
Bildmaße Original	767×1024 Pixel				
	jpg	cr2			
Bildgröße (digital)	4,86 MB	24,5 MB			
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel			
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2587 \times 3471 \text{ Pixel} \\ (3, 37 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2588 \times 3469 \text{ Pixel} \\ (3, 37 \times \text{ Original}) \end{array}$			
Rotation	$0, 2^{\circ}$	$0, 2^{\circ}$			



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)





a) FF1 des Blides im cr2-Format m Reskalierung (Originalmaße)



(e) Codewort4. Quadrant



(f) Codewort redundant

Tabelle 3.36: Vorlage(5), Codierung im Tieffrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 500

Vorlage	(5),	\mathbf{ISO}	400
---------	------	----------------	-----

– Speicherung nach Codierung im bmp-Format

Eckdaten					
Bildmaße (bei Entwicklung)	$14\times18,5~\mathrm{cm}$				
Bildmaße Original	767×1024 Pixel				
	jpg	cr2			
Bildgröße (digital)	4,86 MB	23,9 MB			
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel			
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2803 \times 3756 \text{ Pixel} \\ (3,65 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2802 \times 3752 \text{ Pixel} \\ (3,65 \times \text{Original}) \end{array}$			
Rotation	0,1°	0,1°			



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)





redundant



(d) FFT des Bildes im cr2-Format mi Reskalierung (Originalmaße)



(e) Codewort4. Quadrant



(f) Codewort redundant

Tabelle 3.37: Vorlage(5), Codierung im Tieffrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 400

Vorlage	(5),	\mathbf{ISO}	500
---------	------	----------------	-----

– Speicherung nach Codierung im jpg-Format

Eckdaten					
Bildmaße (bei Entwicklung)	$14\times18,5~\mathrm{cm}$				
Bildmaße Original	767×1024 Pixel				
jpg cr2					
Bildgröße (digital)	4,76 MB	24,8 MB			
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel			
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2616 \times 3509 \text{ Pixel} \\ (3,41 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2617 \times 3510 \text{ Pixel} \\ (3, 41 \times \text{Original}) \end{array}$			
Rotation	$0,45^{\circ}$	$0,45^{\circ}$			



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)



4. Quadrant





Reskalierung (Originalmaße)



(e) Codewort4. Quadrant



(f) Codewort redundant

Tabelle 3.38: Vorlage(5), Codierung im Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 500

Vorlage	(5),	\mathbf{ISO}	400
---------	------	----------------	-----

– Speicherung nach Codierung im jpg-Format

Eckdaten					
Bildmaße (bei Entwicklung)	$14\times18,5~\mathrm{cm}$				
Bildmaße Original	767×1024 Pixel				
	jpg	$\mathrm{cr}2$			
Bildgröße (digital)	4,86 MB	23,9 MB			
Bildmaße (digital) ohne Reska- lierung	3648×4864 Pixel	3648×5472 Pixel			
Bildmaße (digital), nach Zu- schnitt, ohne Reskalierung	$\begin{array}{c} 2831 \times 3796 \text{ Pixel} \\ (3,70 \times \text{ Original}) \end{array}$	$\begin{array}{c} 2830 \times 3800 \text{ Pixel} \\ (3,70 \times \text{Original}) \end{array}$			
Rotation	$-0,15^{\circ}$	$-0,15^{\circ}$			



(a) FFT des Bildes im jpg-Format mit Reskalierung (Originalmaße)





redundant



(d) FFT des Bildes im cr2-Format mi Reskalierung (Originalmaße)



(e) Codewort4. Quadrant



(f) Codewort redundant

Tabelle 3.39: Vorlage(5), Codierung im Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 400

Zusätzlich zu dieser Ergebnisdarstellung werden vergrößerte Bereiche der FFTs präsentiert, bei denen die Existenz der Codeworte vorliegt, jedoch bei der Extraktion der Codierung untergeht bzw. nicht deutlich erkennbar ist. Zunächst werden die hochfrequent codierten, anschließend die tieffrequent codierten Bilder in *Abbildung 3.12* und *Abbildung 3.13* behandelt.



(a) Vorlage (3): FFT des 1. Quadranten, ISO 500 (nach Codierung gespeichert im png-Format)



(c) Vorlage (3): FFT des 1. Quadranten, ISO 500 (nach Codierung gespeichert im bmp-Format)



(b) Vorlage (3): FFT des 4. Quadranten, ISO 500 (nach Codierung gespeichert im png-Format)



(d) Vorlage (3): FFT des 4. Quadranten, ISO 500 (nach Codierung gespeichert im bmp-Format)

Abb. 3.11: Vorlage (3), Codierung im Hochfrequenzbereich, Speicherung nach Ablichten im jpg-Format: Vergrößerung der FFTs

An den äußersten Ecken wird die Platzierung der Codeworte durch die hellen (nach Speicherung im bmp-Format) und dunklen (nach Speicherung im png-Format) Stellen angedeutet, allerdings nicht signifikant. Die FFTs der im cr2-Format gespeicherten Bilder sind durch die Reskalierung insgesamt verdunkelt, sodass dort keine Aussage getroffen werden kann.



(a) Vorlage (4): FFT des 1. Quadranten, ISO 500



(c) Vorlage (4): FFT des 1. Quadranten, ISO 400



(b) Vorlage (4): FFT des 4. Quadranten, ISO 500



(d) Vorlage (4): FFT des 4. Quadranten, ISO 400

Abb. 3.12: Vorlage (4), Codierung im Tieffrequenzbereich, Speicherung nach Codierung im bmp-Format, Speicherung nach Ablichten im cr2-Format: Vergrößerung der FFTs

In (a) und (c) ist an der oberen, linken Ecke und in (b) und (d) an der unteren, rechten Ecke die Codierung erkennbar, nach der Extraktion allerdings nicht. Weiterhin ist das Codewort "verschwommen". Beim Speichern der Photographie im jpg-Format ist das Codewort auf dem FFT nicht sichtbar.



(a) Vorlage (5): FFT des 1. Quadranten, ISO 500



(c) Vorlage (5): FFT des 1. Quadranten, ISO 400



(b) Vorlage (5): FFT des 4. Quadranten, ISO 500



(d) Vorlage (5): FFT des 4. Quadranten, ISO 400

Abb. 3.13: Vorlage (5), Codierung im Tieffrequenzbereich, Speicherung nach Codierung im bmp-Format, Speicherung nach Ablichten im cr2-Format: Vergrößerung der FFTs

Bei Vorlage (5) sind die Codierungen deutlich. Wird das Bild nach der Ablichtung im jpg-Format gespeichert, so sind auch bei einer Vergrößerung der FFTs die Codeworte nicht erkennbar.

Ergebnisse

Tabelle 3.41 bietet zunächst einen Überblick über die extrahierten Codierungen vom praktischen Test. Dabei werden die Ergebnisse der Simulationen gegenübergestellt, sodass die Veränderung durch den Entwicklungs- und Verarbeitungsprozess deutlich wird. Die Legende ist in Tabelle 3.40 abgebildet.

Anschließend werden die Hypothesen geprüft und anhand der Resultate Schlussfolgerungen gezogen.

u.r.:= Extraktion des Codewortes im vierten Quadranten
red. := Extraktion des Codewortes redundant (alle extrahierten Codeworte multipliziert)
Mult. := Vielfaches vom Original
Ext. Simulation := Extraktion des Codewortes in der Simulation

 $\mathbf{Gesp.}\ \mathbf{Format}:=\mathbf{Gespeichertes}\ \mathbf{Format}\ \mathbf{nach}\ \mathbf{der}\ \mathbf{Codierung}$

 $V(x), x \in \{3, 4, 5\} :=$ Vorlage (x), gemäß Tabelle 3.1:

- $\mathbf{V(3)}$ Abbild der Geldmünze, codiert im Hochfrequenzbereich
- V(4) Abbild der Playmobil[®]-Figur, codiert im Tieffrequenzbereich
- $\mathbf{V(5)}$ Abbild des Himmels, codiert im Tieffrequenzbereich

Tabelle 3.40: Notation und Hinweise zu der Übersichtstabelle

eile	$\mathbf{V}(\mathbf{x})$	Format	Ext. Simulation		Extraktion Feldtest				
Z	• (A)	Gesp.]	u.r.	red.	ISO	u.r.	red.	Photo	Mult.
1	V(3)	bmp	8		500	5	教	jpg	5,55
2							32	cr2	5,57
3					400	民	8	jpg	5,55
4						33	22	cr2	5,57
5		png	du.	212	500	10	<u>%</u>	jpg	5,55
6							202	cr2	5,56
7					400	20	꽱	jpg	5,55
8								cr2	5,56

9	V(4)	bmp	鏩	鏩	500		22	jpg	1,35
10							24	cr2	1,35
11					400	23	諁	jpg	1,36
12								cr2	1,36
13		jpg	1		500			jpg	1,36
14								cr2	1,34
15					400	諁	鱵	jpg	1,36
16							193	cr2	1,36
17	V(5)	bmp		2.5	500	Ξ.	÷	jpg	3,37
18						R		cr2	3,37
19					400			jpg	3,65
20						躛		cr2	3,65



Tabelle 3.41: Übersicht der Extraktion des Tests im Vergleich zu den Simulationen

Testbeobachtungen Die aufgestellten Hypothesen werden im Folgenden geprüft und als Testbeobachtung $TB(x), x \in \mathbb{N}$ dokumentiert:

- TB(1) Die Hypothese H(1) kann weder verifiziert noch falsifiziert werden. Werden bei der Ablichtung die Bilder im jpg-Format gespeichert, so kann das Codewort nach der Extraktion bei Zeile 17 erkannt werden. Bei allen anderen Codierungen (siehe Zeilen 1, 3, 5, 7, 9, 11, 13, 15, 19, 21, 23) ist dies nicht der Fall. Zusätzlich sei angemerkt, dass die geringe Anzahl der verwendeten Bilder für den Test keine ausreichenden Informationen liefert, um bzgl. H(1) eine eindeutige Aussage treffen zu können.
- TB(2) Die Hypothese H(2) ist durch die Darstellung der extrahierten Codeworte bestätigt: Die Bilder, die nach der Codierung im jpg-Format (vor dem Ausdruck) gespeichert wurden, sind nicht erkennbar. Dazu werden in der *Tabelle 3.41* die Zeilen 13 bis 16 und 21 bis 24 herangezogen.
- TB(3) H(3) wird ebenfalls verifiziert. Zum einen wird die Codierung im Tieffrequenzbereich bei der Vorlage (5) erkannt, siehe Zeilen 17 bis 20 in *Tabelle 3.41*. Zum anderen ist die Existenz der Codeworte sowohl auf der FFT-Vergrößerung der Vorlage (4) als auch der Vorlage (5) deutlich sichtbar (siehe *Abbildung 3.12* und *Abbildung 3.13*).
- TB(4) Die Hypothese H(4) ist durch den Test nicht eindeutig zu bestätigen oder zu widerlegen. Das Codewort kann bei Vorlage (4) (Speicherung nach Codierung im bmp-Format) mit vorhandenen Kanten nach der Extraktion nicht erkannt werden, siehe *Tabelle 3.41* bei Zeilen 9 bis 12. Allerdings ist die Codierung in den vergrößerten FFTs wahrnehmbar (*Abbildung 3.12*).
- TB(5) Nach dem Test kann ebenfalls hinsichtlich der Hypothese H(5) keine eindeutige Aussage getroffen werden. Eine Abhängigkeit der Bildsensoreinstellung auf ISO 500 oder ISO 400 ist nicht herauszulesen. Dabei wurden einerseits die vergrößerten FFTs der Vorlagen (4) und (5) in Abbildung 3.12 und Abbildung 3.13 betrachtet.

Andererseits bietet die *Tabelle 3.41* in den Zeilen 17 bis 20 eine Übersicht über die extrahierten Codierungen, bei denen auch nicht erkennbar ist, ob die Ablichtung mit ISO 400 oder mit ISO 500 eine bessere Decodierung zur Folge hat.

Schlussfolgerungen Die vorgestellten Resultate werden gedeutet und daraus Schlussfolgerungen $SF_T(x), x \in \mathbb{N}, T := ,Test"$ abgeleitet.

- $SF_T(1)$ Bei Vorlage (5) mit dem Abbild des Himmels ist die Codierung am deutlichsten wahrnehmbar. Daraus kann gefolgert werden, dass bei einem Bild mit homogenen Flächen die Fehlertoleranz hoch ist.
- $SF_T(2)$ Die Papierbeschaffenheit (siehe 3.9) wird beim Photographieren mit abgelichtet und ist damit ein Teil des reproduzierten Bildes. Wird eine Codierung – wie bei Vorlage (5) – erkannt und bei Vorlage (4) wahrgenommen, so besteht die Annahme, dass sie robust ist.
- $SF_T(3)$ Bei Vorlage (4) ist die Codierung in der FFT-Abbildung sichtbar. Das abgelichtete Bild ist ca. 1,35 × größer, Vorlage (5) im Durchschnitt ein Vielfaches in Höhe von 3,5. Es wird angenommen, dass das Vielfache des photographierten Bildes höher sein muss als 1,35 und im Bereich von 3,5 × liegt, damit das Codewort nach der Extraktion erkannt wird. Dies kann in weiteren Tests geprüft werden.

3.3.4 Resümee

Der Lösungsansatz im Frequenzraum hat in den Simulationen und im praktischen Test gezeigt, dass er ein mögliches Vorgehen darstellt und eine versteckte Codierung durchführbar ist.

Parameter Es werden aus den Abschnitten der Simulationsbeobachtungen, Schlussfolgerungen (Simulation), Testbeobachtungen und Schlussfolgerungen (Test) abhängige Parameter $P(x), x \in \mathbb{N}$ formuliert.

$\mathrm{P}(1)$ Speicherformat nach der Codierung

Die Simulation hat verdeutlicht, dass eine verlustfreie Speicherung benötigt wird. Welches Format (bmp, png oder jp2) bevorzugt wird, konnte dabei nicht herausgelesen werden und bedarf weiterer Analyse.

P(2) Durchschnittliche Abweichung zwischen Original und codiertem Bild (DA) Es konnte in der Simulation festgestellt werden, dass der Wert der DA im 100stel Bereich liegen soll. Damit ist zum einen die Codierung nicht wahrnehmbar, und zum anderen ist das Codewort decodierbar.

P(3) Lage der Codierung

Werden die Simulationen und der praktische Test zusammen betrachtet, so ist eindeutig eine Codierung im Tieffrequenzbereich notwendig, damit eine Decodierung möglich ist.

P(4) Codewortbildung

Das Codewort darf nicht zu komplex, das heißt nicht zu fein strukturiert sein. Der 2D-Barcode konnte nach der Extraktion nicht erkannt werden.

P(5) Größe des reproduzierten Bildes

Im praktischen Test konnte die versteckte Codierung bei der Vorlage extrahiert und erkannt werden, bei der das photographierte Bild mehr als die dreifache Auflösung des Originals betrug.

Ausblick In Bezug auf die Parameter sind weitergehende Analysen notwendig, um für Vorlagen mit unterschiedlichen Bildinhalten, Bildauflösungen und der Farbverteilung eine robuste Lösung zu erarbeiten:

- (1) In Bezug auf P(5) muss validiert werden, welchen Wert das Vielfache des reproduzierten Dokuments annehmen darf. Ist der Wert zu hoch, so zieht evtl. die Reskalierung ein Verdunkeln des Bildes und das Erschweren der Decodierung mit sich. Ist der Wert zu niedrig, so ist unter Umständen das Codewort undeutlich.
- (2) Ferner erfordert die Codierung weiterer Betrachtung und Erweiterung auf eine höhere Anzahl der Codierungen. Mit dem Codewort \blacksquare können lediglich $2^3 = 8$ mögliche Identifikationsnummern abgebildet werden.

Denkbar wäre beispielsweise die Verwendung zweier Codeworte, die an unterschiedlichen Positionen im FFT des Bildes platzieren werden. Demnach ist wiederum eine Redundanz beim Extrahieren im reproduzierten Bild gegeben, die zur einer robusten Decodierung notwendig ist.

- (3) Im praktischen Test wurden die Bilder auf mattem Photopapier entwickelt. Es gilt zu klären, inwieweit Papierarten unterschiedlicher Eigenschaften das Codierungs- und Decodierungsverfahren beeinflussen.
- (4) Der Test deckt weiterhin nicht eine Analyse des Codierungsverhaltens bei allen Vorlagen ab. In diesem Kontext muss validiert werden, ob das Codewort (eingebettet im Tieffrequenzbereich) bei sämtlichen Mustern erkannt wird.
- (5) Unter der Annahme, dass eine Umsetzung mit einer Codierung im Tieffrequenzbereich in Betracht gezogen wird, so muss geklärt werden, wie helle Vorlagen (wie hier Vorlage (1)) behandelt werden, da selbst bei der Simulation eine Erkennung des extrahierten Codewortes ausbleibt.

3.4 Zeit-Frequenzraum

Bisher wurden zwei unterschiedliche Codierungsmöglichkeiten betrachtet: zum einen lokale Codierung im Ortsraum und zum anderen global verteilte in der Frequenzdomäne. Im Frequenzbereich sind bei einer Fouriertransformation die Ortsinformationen nicht direkt zugänglich im Gegensatz zu einer Transformation im Zeit-Frequenzraum.

Die Realisierbarkeit der Codierung mit Hilfe von Wavelets und der Multiskalen-Analyse im Tiefpassbereich bei einer niedrigeren Auflösung wird validiert.

Nach 3.4.1 Algorithmus folgt die Prüfung der Machbarkeit in 3.4.2 Validierung. Das Unterkapitel 3.4.3 Resümee fasst anschließend die wichtigsten Erkenntnisse zusammen.

3.4.1 Algorithmus

Algorithmus 3: Verarbeitung in der Zeit-Frequenzdomäne

Input : digitalDocuments of size k **Output**: printedAndAssignedDocuments of size l

```
Data: digitalDocuments of size k, codedDocuments of size l,
         printedDocuments of size l, scanedDocuments of size l,
         decodedData of size l, printedAndAssignedDocuments of size l,
         l > k, l, k \in \mathbb{N}
1 for i = 1 to k do
      while exists multiple addresses for i do
\mathbf{2}
         codedDocuments.add(embed coding in rougher level of MRA at
3
         (digitalDocuments(i)));
4 printedDocuments = print codedDocuments;
5 scanedDocuments = scan printedDocuments;
6 for j = 1 to l do
      decodedData(j) = extract information of scanedDocuments(j) with wavelets
7
      and MRA;
      map decodedData(j) to address;
8
      printedAndAssignedDocuments(j) = assign \text{ address to scanedDocuments}(j);
9
10 return printedAndAssignedDocuments;
```

Der Ablauf ist den bereits vorgestellten Algorithmen ähnlich. Die Codierung geschieht in Zeile 3 in einem Level mit gröberer Auflösung einer Multiskalen-Analyse des Bildes.

3.4.2 Validierung

In diesem Kapitel wird geprüft, inwieweit Wavelets und MRA zu einer Codierung von Bildern genutzt werden können. Besonderes Augenmerk wird dabei auf das Einbetten bzw. Verändern von Daten gelegt.

Vorüberlegung Wird von bestehenden Anwendungen ausgegangen, so führen Analysetools oder Filter nicht zum gewünschten Ergebnis. Es muss nämlich eine Codierung hinzugefügt werden, um unterschiedliche Identifikationsnummer (hier: Adressen) abbilden zu können.

Folgende Fragestellungen $F(x), x \in \mathbb{N}$ werden untersucht:

- F(1) Wie ist eine "nicht wahrnehmbare" Codierung mit Wavelets und MRA möglich?
- F(2) Wo wird die Codierung eingebettet?
- F(3) Wie findet die Decodierung statt?

Codierungsverfahren

Tabelle 3.43 stellt die getesteten Modifikationen mit dem Eingangsbild Abbildung 3.14 dar. Es wird das Biorthogonale Wavelet 3.7 gewählt, welcher unter Tabelle 2.6 gelistet ist. Dabei werden F(1) und F(2) fokussiert und die Ergebnisse festgehalten.



- Abb. 3.14: Eingangsbild zur Analyse mit Abbild einer Playmobil®-Figur (Maße: 1007 \times 1007 Pixel)
- In Tabelle 3.42 sind die Hinweise zur Notation der nachfolgenden Darstellung.

Maße sind in Pixel (Höhe × Breite) gegeben. $V_m :=$ Unterraum V_m der MRA (gemäß *Definition MRA*), $W_{m,1} :=$ orthogonales Komplement von V_m in horizontaler Richtung (Index 1) mit $*_m, m \in \mathbb{N} := m$ -tes Level der MRA

Tabelle 3.42: Notation und Hinweise zu den Codierungsbeispielen im Zeit-Frequenzraum

3.4 Zeit-Frequenzraum

Bemerkung 3.13. In *Tabelle 3.43* wird in unterschiedlichen Level codiert. Dadurch ist eine Unvergleichbarkeit untereinander gegeben. Die Selektion der Level erfolgt jedoch nicht in Hinsicht auf Vergleichbarkeit, sondern bezüglich der Prüfung des Verhaltens und der "Nicht-Erkennbarkeit" der Codierungen.





B(6) Diese lokale Codierung verfälscht zusätzlich die Farbcodewerte auf dem ganzen Bild. In diesem Fall wird es aufgehellt.

C - Codierung: Skalierung des Bildes im achten Level der MRA von 18 × 18 auf 17 × 17. Das skalierte Bild wird in den Filter der Größe 18 × 18 im oberen linken Eck platziert. In die unterste Reihe des Filters wird ein schwarzes Codewort der Größe 1 × 18 eingebettet:



(a) Rekonstruktion nach Codierung in V_8 (Größe von $V_8 : 18 \times 18$)



(b) Unterschiede zwischen Original und codiertem, rekonstruierten Bild (a)

- B(7) Die Helligkeit des Gesamtbildes wird nicht dermaßen beeinflusst wie mit einer Codierung in B, allerdings sind lokale Aufhellungen im synthetisierten Bild zu erkennen – in (a) z. B. bei den Beinen und in (b) analog an dergleichen Position.
- **B(8)** Wird beim codierten, gespeicherten Bild (a) nach dem gleichen Verfahren das achte Level der MRA gebildet, ist das Codewort mit dem schwarzen Streifen nicht zu sehen, stattdessen ist V_8 dunkler geworden:



 ${\bf D}-{\bf Codierung:}$ Fourier-Transformation mit Einbettung des Codewortes (Maße $9\times9)$ im tieffrequenten Bereich^{24}

Zweck: Testen der Wahrnehmbarkeit der Codierung beim rekonstruierten Bild und Prüfung der Docodierbarkeit (in Hinsicht auf Robustheit)



(a) Rekonstruktion nach Codierung in V_2 (Größe von V_2 : 263 × 263)



(b) Rekonstruktion nach Codierung in V_4 (Größe von V_4 : 77 × 77)

B(9) Die extrahierte (redundante) Codierung bei (a) nach Speicherung mit im2unit8 und Wiedereinlesen des Bildes ist nicht erkennbar:



B(10) Die extrahierte (redundante) Codierung bei (b) nach Speicherung mit im2unit8 und Wiedereinlesen des Bildes ist deutlicher, aber die Modifikation sichtbar (digonale "Wellen" über das Bild verteilt):



Tabelle 3.43: Beispiele von Codierungen mit Wavelets und MRA

Decodierungsverfahren

Dieser Abschnitt zielt allgemein auf die Fragestellung F(3) und welche Aspekte beim Decodieren im Umgang mit Wavelets zu berücksichtigen sind. Die Beobachtungen bezüglich der Extraktionen, die in Codierungsverfahren gezeigt wurden, werden miteinbezogen.

Es bieten sich nachstehende Optionen O(x), $x \in \mathbb{N}$ an, die teils einer weitergehenden

²⁴Die Codierung im tieffrequenten Bereich erfolgt nach den Ausführungen in Simulationsbedingungen. Hier findet die Codierung im vierten Quadranten statt.

3.4 Zeit-Frequenzraum

Analyse bedürfen. Es wird jeweils die Art der Codierung und Decodierung vorgestellt und anschließend aufgeführt, welche Punkte dabei beachtet werden müssen.

- O(1) Es wird eine Codierung gefunden, die durch MRA in einer groben Auflösung im Tieffrequenzbereich (V_x) eingebettet wird und nach der Synthese nicht erkennbar ist. Die Decodierung findet auf die gleiche Weise statt, d.h. der Code wird wieder durch Bildung der MRA auf demselben Level herausgelesen. Dabei muss berechnet werden, dass bei der MRA-Zerlegung im Decodiervorgang relevante Informationen bereits herausgefiltert werden können und letztendlich beim Extrahieren nicht mehr zugänglich sind. Beispiel **B** in *Tabelle 3.43* zeigt dieses Verhalten.
- O(2) Es findet eine Codierung statt wie bei O(1). Zum Decodieren wird eine Formel entwickelt, welche die eingebettete Informationen extrahiert, ohne dass eine MRA-Zerlegung notwendig ist.
- O(3) Es werden eigens für das Codierverfahren Wavelets entwickelt, welche abhängig von Größe, Helligkeit, Struktur oder weiteren Parametern sein können. In der MRA findet mit der Filterung bei einer "Bruchstelle", d.h. an einem charakteristischen Punkt der Bildanalyse, die Codierung statt. In diesem Fall sollte die Codierung Eigenschaften besitzen, die beim Decodieren in der MRA-Zerlegung mit dem Wavelet keine Informationen verloren gehen. Möglich

der MRA-Zerlegung mit dem Wavelet keine Informationen verloren gehen. Möglich wäre beispielsweise die Entwicklung eines dazugehörigen "Decodier"-Wavelets, welche die hinzugefügten Daten fehlerfrei extrahieren kann.

3.4.3 Resümee

Im Zeit-Frequenzraum führten die getesteten Modifikationen durch MRA und Wavelets zu keinem erfolgreichen Ergebnis. Es konnte keine Codierung gefunden werden, welche sowohl nicht wahrnehmbar als auch nach der Simulation decodierbar ist.

Durch die höhere Komplexität dieser Domäne müssen mehrere Faktoren betrachtet und analysiert werden. Dazu müssen Fragen geklärt werden, wie eine versteckte Codierung umgesetzt werden kann, sodass eine Decodierung möglich ist. Beim Extrahieren muss verifiziert werden, ob eine MRA benötigt wird, oder im reproduzierten Bild ohne Transformation die Informationen herausgelesen werden können (siehe *Decodierungsverfahren*).

4 Fazit und Ausblick

Der Schwerpunkt der Arbeit lag auf der Validierung der Realisierbarkeit einer versteckten Codierung von Bildinformationen. Diese ist im vorliegenden Fall durch mehrere Verfahren möglich. In *Resümee* (Steganographie), *3.2.3 Resümee* (Ortsraum), *3.3.4 Resümee* (Frequenzraum) und *3.4.3 Resümee* (Zeit-Frequenzraum) wurden die jeweiligen Vorgehen abschließend betrachtet.

Aus dem Bereich der Steganographie ermöglichen *Distorsionstechniken* das Einbetten von Informationen bei formatierten Texten.

Weiterhin erlauben Modifikationen im Ortsraum das Hinzufügen von Daten. Allerdings ist dies eingeschränkt und nicht bei allen Vorlagen möglich. Die Codierung ist abhängig vom Dokument und erschwert damit das Entwickeln eines robusten Algorithmus. Bezüglich der Fehlererkennung und -korrektur bei der Decodierung gilt zu klären, wie eine Codierung mit Kontrollstellen gebildet werden kann.

Abhängig von einem konkreten Anwendungsfall und wie viele Codierungsmöglichkeiten benötigt werden, kann eine Umsetzung im Ortsraum in Betracht gezogen werden.

Der Ansatz in der Zeit-Frequenz-Domäne verdeutlicht die Komplexität des Verfahrens mit Anwendung der Multiskalen-Analyse. Es konnten keine Parameter abgeleitet werden, die eine versteckte Codierung und Decodierung ermöglichen. Eine Weiterverfolgung wird nicht empfohlen.

Eine versteckte Codierung im Frequenzraum ist durchführbar, welche in Simulationen und einem praktischen Test verifiziert wurde. Der dargestellte Algorithmus mit der Anwendung der FFT zur Codierung und Decodierung beschreibt ein mögliches Verfahren. Es konnten relevante Parameter identifiziert werden, die zu einer praktischen Umsetzung notwendig sind.

Neben den bereits beschriebenen Aspekten, die einer weitergehenden Analyse bedürfen (siehe Abschnitt *Ausblick* (Frequenzraum)), muss darüber hinaus geklärt werden, inwieweit die vorgestellten Lösungsverfahren bei Farbbildern durchführbar sind, da in dieser Arbeit ausschließlich Graustufenbilder betrachtet wurden.

Anhang

A Überblick über die verwendeten Dateiformate

Bitmap Format (bmp) [3, S.733ff.] Das geräteunabhängige Format wird von Windows zum Speichern von Rasterbildern verwendet. Es gibt drei unterschiedliche Komprimierungsstufen, nämlich BI_RBG (unkomprimiert), BI_RLE8 (Run-Length-Encoding-Verfahren (RLE) mit 8 Bit pro Pixel) und BI_RLE4 (RLE mit 4 Bit pro Pixel). In der vorliegenden Arbeit liegt eine unkomprimierte Speicherung vor.

Portable Network Graphics (png) [26, S.111] Als ein Nachfolger des gif-Formats (Graphics Interchange Format (gif)) gehört png zu den Rastergrafik-Formaten und ist seit 1996 als ein W3C-Standard definiert. Mit png ist sowohl verlustfreie Speicherung bei palettenbasierten Bildern bis zu 256 Farben als auch bei nicht palettenbasierten mit einer Farbtiefe bis zu 48 Bit möglich.

JPEG2000 [44, S.399ff.] 1997 erschien der von *JPEG* (siehe unten) und JBIG (Joint Bilevel Imagery Experts Group) neueste, internationale Standard JPEG2000. Grundlage zur Komprimierung sind Wavelets. Im Gegensatz zu den meisten anderen Kompressionssystem und Standards, bei denen die Kompression und die Dekompression weitgehend eine Speicherfunktion darstellt, sind mit JPEG2000 nach der Komprimierung mehrere Arten der Dekomprimierung gegeben. So können unterschiedliche Auflösungen und Qualitäten je nach Bedarf genutzt werden. Die Kompressionstiefe ist im Algorithmus einstellbar und kann bis zu einer verlustfreien Qualität justiert werden. Weitere Eigenschaften sind unter [44, S.409ff.] zu finden. Es wird hier die Dateiendung *.jp2 verwendet.

Joint Photographic Experts Group (JPEG) Das Rastergrafik-Format wendet standardisierte Kompressionsalgorithmen an, die von JPEG entwickelt wurden (ISO 10918-1). Der Qualitätsverlust hängt von der Dateigröße ab, wobei eine Reduktion der Farbzahl durchgeführt wird. Dieses Format kann Bilder der Farbtiefe von 24 Bit speichern und ist nicht palettenbasiert. In der vorliegenden Arbeit haben diese Bilder die Dateiendung *.jpg [26, S.109f.].

Es ist sowohl eine verlustfreie als auch verlustbehaftete Komprimierung möglich. Bei ersterer lassen sich dich Ursprungsdaten wiederherstellen. Zur Komprimierung wird die diskrete Cosinus-Transformation (DCT) genutzt [29, S.504ff.].

Canon RAW version 2 (CR2) Beim verlustfreien RAW-Format werden zwölf Bit genutzt im Gegensatz zu acht bei JPEG. Es finden keine Korrekturen hinsichtlich *Weißpunkt, Farben, Schärfe, Rauschen, Tonwertumfang* und dergleichen statt. Durch die höhere Bitanzahl können Weißabgleich, Farb-, Kontrast- oder Perspektivoptimierungen vorgenommen werden [41, S.11].

B Simulation mit 2D-Barcode

Es folgt eine tabellarische Darstellung der Simulationen codiert mit dem 2D-Barcode im Hochfrequenzbereich.

In *Tabelle 3.4* ist die Veranschaulichung mit Vorlage (1) mit dieser Codierung bereits vollständig. Im Folgenden befindet sich die Abbildung der Simulationen für die Vorlagen (2), (3), (4) und (5) (siehe *Tabelle 3.1*).





¹Abweichungen und Unterschiede siehe *Vorlage (2)* mit demselben Codewort, Speicherung im bmp-Format. Des weiteren ist das codierte Dokument vor der Speicherung identisch, weswegen auf eine zusätzliche Vergleichs-Darstellung von Original und codiertem Bild bei allen folgenden Vorlagen im jp2- und png-Format verzichtet wird.













Tabelle 4.2: Simulation der Codierung mit 2D-Barcode bei Vorlagen (2), (3), (4), (5) im hochfrequenten Bereich (verlustfreie Speicherung)

Abbildungsverzeichnis

1.1	Prozess von Auftragannahme bis zum Versand	4
1.2	Information Hiding und seine Teildiszipline	5
01	Stagenegraphic Deigniel Die Coognation von H Helbein	0
∠.1 ೧೧	Steganographie: Delspiel Die Gesanaten von H. Holden	9
2.2 0.2	Steganographie. Zenen-Abstand-Kodierung bei formatierten Texten	11
2.5 9.4	Ortgraum: Annäherung der Ableitung einer abgetasteten Funktion	12
2.4 2.5	Ortsraum: Annaherung der Ableitung einer abgetasteten Funktion	15
$\frac{2.5}{2.6}$	Ortsraum: Fingangshild für Filterungen (Abhild einer Münze)	10
$\frac{2.0}{2.7}$	Ortsraum: Canny-Kantendetektion	$\frac{20}{23}$
2.8	Ortsraum: Skizze der Hough-Transformation	$\frac{20}{23}$
2.9	Ortsraum: Beispiel einer Hough-Transformation	$\frac{-9}{24}$
2.10	Frequenzraum: Beispiel einer FFT	26
2.11	Frequenzraum: Gaußsche Transferfunktion und Tiefpassfilter	28
2.12	Frequenzraum: IDFT der Gaußschen Tiefpassfilterung	28
2.13	Frequenzraum: Gaußsche Transferfunktion und Hochpassfilter	29
2.14	Frequenzraum: IDFT der Gaußschen Hochpassfilterung	29
2.15	Zeit-Frequenz-Raum: Skizze der mehrdimensionalen Multiskalen-Analyse .	33
2.16	Zeit-Frequenz-Raum: Beispiel einer Wavelet-Transformation mit drei Leveln	35
21	Lögunggvorfahren. Skizze der Beskelierung	12
3.1 3.2	Lösungsverfahren: Skizze der Reskalierung	43
$3.1 \\ 3.2$	Lösungsverfahren: Skizze der Reskalierung	43 1- 46
3.1 3.2 3.3	Lösungsverfahren: Skizze der Reskalierung	43 n- 46 48
3.1 3.2 3.3 3.4	Lösungsverfahren: Skizze der Reskalierung Lösungsverfahren: Teilproblem Codierung und Decodierung mit Reed-Solomon Code Lösungsverfahren: Beispiel eines einfach strukturierten Codewortes Ortsraum: Skizze von vier Codeworten zu einer Linie bzgl. Bobustheit	43 1- 46 48 51
3.1 3.2 3.3 3.4 3.5	Lösungsverfahren: Skizze der Reskalierung Lösungsverfahren: Teilproblem Codierung und Decodierung mit Reed-Solomon Code Lösungsverfahren: Beispiel eines einfach strukturierten Codewortes Ortsraum: Skizze von vier Codeworten zu einer Linie bzgl. Robustheit Ortsraum: Bild eines Punktes bzgl. Robustheit	43 1- 46 48 51 51
3.1 3.2 3.3 3.4 3.5 3.6	Lösungsverfahren: Skizze der Reskalierung Lösungsverfahren: Teilproblem Codierung und Decodierung mit Reed-Solomon Code Lösungsverfahren: Beispiel eines einfach strukturierten Codewortes Ortsraum: Skizze von vier Codeworten zu einer Linie bzgl. Robustheit Ortsraum: Bild eines Punktes bzgl. Robustheit Ortsraum: Beeinflussung der Distorsion	43 46 48 51 51 52
3.1 3.2 3.3 3.4 3.5 3.6 3.7	Lösungsverfahren: Skizze der Reskalierung	43 46 48 51 51 52 58
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8	Lösungsverfahren: Skizze der Reskalierung	$43 \\ 46 \\ 48 \\ 51 \\ 51 \\ 52 \\ 58 \\ 59$
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9	Lösungsverfahren: Skizze der Reskalierung	$43 \\ 46 \\ 48 \\ 51 \\ 51 \\ 52 \\ 58 \\ 59 \\ 62$
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10	Lösungsverfahren: Skizze der Reskalierung	$\begin{array}{c} 43 \\ 46 \\ 48 \\ 51 \\ 51 \\ 52 \\ 58 \\ 59 \\ 62 \\ 98 \end{array}$
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11	Lösungsverfahren: Skizze der Reskalierung	43 46 48 51 51 52 58 59 62 98
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11	Lösungsverfahren: Skizze der Reskalierung	$ \begin{array}{r} 43 \\ 46 \\ 48 \\ 51 \\ 52 \\ 58 \\ 59 \\ 62 \\ 98 \\ 112 \end{array} $
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12	Lösungsverfahren: Skizze der Reskalierung	43 46 48 51 52 58 59 62 98 112
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12	Lösungsverfahren: Skizze der Reskalierung	$\begin{array}{c} 43\\ 46\\ 48\\ 51\\ 51\\ 52\\ 58\\ 59\\ 62\\ 98\\ 112\\ \end{array}$
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12	Lösungsverfahren: Skizze der Reskalierung	$\begin{array}{c} 43\\ 46\\ 48\\ 51\\ 51\\ 52\\ 58\\ 59\\ 62\\ 98\\ 112\\ 113\end{array}$
$\begin{array}{c} 3.1\\ 3.2\\ 3.3\\ 3.4\\ 3.5\\ 3.6\\ 3.7\\ 3.8\\ 3.9\\ 3.10\\ 3.11\\ 3.12\\ 3.13\end{array}$	Lösungsverfahren: Skizze der Reskalierung	$\begin{array}{c} 43\\ 46\\ 48\\ 51\\ 51\\ 52\\ 58\\ 59\\ 62\\ 98\\ 112\\ 113\end{array}$
3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12 3.13	Lösungsverfahren: Skizze der Reskalierung	$\begin{array}{c} 43\\ -\\ 46\\ 48\\ 51\\ 51\\ 52\\ 58\\ 59\\ 62\\ 98\\ 1112\\ 1113\\ \end{array}$

Abbildungsverzeichnis

3.14	Zeit-Frequenzraum:	Eingangsbild zur	Analyse mit Abbild	einer Playmobil®-
	Figur			

Tabellenverzeichnis

2.1	Ortsraum: Beispiel zum Schwellwertverfahren	16
2.2	Ortsraum: Beispiel der ersten Ableitungen eines Bildes	18
2.3	Ortsraum: Beispiel der Anwendung des Laplacefilters	18
2.4	Ortsraum: Beispiele von Tiefpassfiltern	20
2.5	Ortsraum: Beispiele von Hochpassfiltern	22
2.6	Zeit-Frequenz-Raum: Wavelet-Familie mit Beispielen	38
3.1	Frequenzraum: Vorlagen für die Simulation	57
3.2	Frequenzraum: Begründung der Codierung im Hoch- und Tieffrequenzbereich	60
$3.3 \\ 3.5$	Frequenzraum: Notation und Hinweise zu den Ergebnistabellen (Simulation) Frequenzraum: Simulation der Codierung mit Vorlage <i>Visitenkarte</i> im hoch-	62
	frequenten Bereich mit verlustfreier Speicherung	68
3.6	Frequenzraum: Simulation der Codierung mit Vorlage <i>Text mit dunklem</i> Hintergrund im hochfrequenten Bereich mit verlustfreier Speicherung	70
3.7	Frequenzraum: Simulation der Codierung mit Vorlage Münze im hochfre-	10
	quenten Bereich mit verlustfreier Speicherung	72
3.8	Frequenzraum: Simulation der Codierung mit Vorlage <i>Playmobil</i> [®] im hoch- frequenten Bereich mit verlustfreier Speicherung	74
3.9	Frequenzraum: Simulation der Codierung mit Vorlage <i>Himmel</i> im hochfre-	• -
	quenten Bereich (verlustfreie Speicherung)	76
3.10	Frequenzraum: Simulation der Codierung im hochfrequenten Bereich mit	
	verlustbehafteter Speicherung	78
3.12	Frequenzraum: Simulation der Codierung mit Vorlage <i>Visitenkarte</i> im tief- frequenten Bereich mit verlustfreier Speicherung	80
3.14	Frequenzraum: Simulation der Codierung mit Vorlage Text mit dunklem	
3.16	<i>Hintergrund</i> im tieffrequenten Bereich mit verlustfreier Speicherung Frequenzraum: Simulation der Codierung mit Vorlage <i>Münze</i> im tieffre-	82
0.10	quenten Bereich mit verlustfreier Speicherung	83
3.18	Frequenzraum: Simulation der Codierung mit Vorlage <i>Playmobil</i> [®] im tief-	
	frequenten Bereich mit verlustfreier Speicherung	84
3.20	Frequenzraum: Simulation der Codierung mit Vorlage <i>Himmel</i> im tieffre-	0.0
0.01	quenten Bereich mit verlustfreier Speicherung	86
3.21	requenzraum: Simulation der Codierung im tieffrequenten Bereich mit verlustbehafteter Speicherung	88
3.22	Frequenzraum: Notation und Hinweise zu der Übersichtstabelle (Simulation)	89
3.23	Frequenzraum: Zusammenfassende Auflistung der Simulationen mit Codie-	
	rung im hochfrequenten Bereich	91
3.24	Frequenzraum: Zusammenfassende Auflistung der Simulationen mit Codie-	
	rung im tieffrequenten Bereich	93
3.27	Frequenzraum: Ablichtbedingungen beim praktischen Test	98

Tabellenverzeichnis

3.28 Frequenzraum: Resultat des praktischen Tests: Vorlage(3), Codierung im
Hochfrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 500100
3.29 Frequenzraum: Resultat des praktischen Tests: Vorlage(3), Codierung im
Hochfrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 400101
3.30 Frequenzraum: Resultat des praktischen Tests: Vorlage(3), Codierung im
Hochfrequenzbereich, Speicherung im png-Format, Ablichtung mit ISO 500 102
3.31 Frequenzraum: Resultat des praktischen Tests: Vorlage(3), Codierung im
Hochfrequenzbereich, Speicherung im png-Format, Ablichtung mit ISO 400 103
3.32 Frequenzraum: Resultat des praktischen Tests: Vorlage(4), Codierung im
Tieffrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 500 104
3.33 Frequenzraum: Resultat des praktischen Tests: Vorlage(4), Codierung im
Tieffrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 400 105
3.34 Frequenzraum: Resultat des praktischen Tests: Vorlage(4), Codierung im
Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 500 . 106
3.35 Frequenzraum: Resultat des praktischen Tests: Vorlage(4), Codierung im
Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 400 . 107
3.36 Frequenzraum: Resultat des praktischen Tests: Vorlage(5), Codierung im
Tieffrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 500 108
3.37 Frequenzraum: Resultat des praktischen Tests: Vorlage(5), Codierung im
Tieffrequenzbereich, Speicherung im bmp-Format, Ablichtung mit ISO 400 109
3.38 Frequenzraum: Resultat des praktischen Tests: Vorlage(5), Codierung im
Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 500 . 110
3.39 Frequenzraum: Resultat des praktischen Tests: Vorlage(5), Codierung im
Tieffrequenzbereich, Speicherung im jpg-Format, Ablichtung mit ISO 400 . 111
3.40 Frequenzraum: Notation und Hinweise zu der Übersichtstabelle (prakti-
scher Test) \ldots
3.41 Frequenzraum: Übersicht der Extraktion des Tests im Vergleich zu den
Simulationen
3.42 Zeit-Frequenzraum: Notation und Hinweise zu den Codierungsbeispielen 122
3.43 Zeit-Frequenzraum: Beispiele von Codierungen mit Wavelets und MRA 126
4.2 Frequenzraum: Simulation der Codierung mit 2D-Barcode bei Vorlagen (2),
(3), (4), (5) im hochfrequenten Bereich mit verlustfreier Speicherung 138
Literaturverzeichnis

- Anonymus: Hyperotomachia Poliphili (1499): The Dream Battles of Polia's Lover.
 Auflage.
- [2] Arunachalam, AR. (6. August 2013): Imperceptible Digital Image Watermarking. In: International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) Volume 1, Issue 6. S. 1321-1326.
- [3] Born, Günter (1993): Referenzhandbuch Dateiformate. Grafik, Text, Datenbanken, Tabellenkalkulation. 2. Auflage. Bonn u.a.: Addison-Wesley.
- [4] Burger, Wilhelm / Burge, Mark J. (2013): Principles of Digital Image Processing. Advanced Methods. London / Heidelberg / New York / Dordrecht: Springer.
- [5] Canny, John (November 1986): A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-8, No.6 S.679-698. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.
 1.420.3300&rep=rep1&type=pdf
- [6] Castleman, Kenneth R. (1996): Digital image processing. New Jersey u.a.: Prentice-Hall.
- [7] Dahlke, S. / Maass P. (1995): The affine uncertainty principle in one and two dimensions. In: Demkowicz, Leszek (Hg.): Computers and Mathematics with Applications, Vol.30, No. 3-6. Elsevier Science Ltd. S.293-305.
- [8] Desmedt, Yvo (2005): COVERT CHANNELS. In: Tilborg, Henk C.A. (Hg.): Encyclopedia of cryptography and security. New York: Springer. S. 106f.
- [9] Die Gesandten 1. https://de.wikipedia.org/wiki/Die_Gesandten#/media/ File:Hans_Holbein_the_Younger_-_The_Ambassadors_-_Google_Art_Project. jpg. (aufgerufen am 02.09.2017)
- [10] Die Gesandten 2. https://de.wikipedia.org/wiki/Die_Gesandten#/media/ File:Holbein_Skull.jpg. (aufgerufen am 02.09.2017)
- [11] Daubechies, Ingrid (1992): Ten Lectures on Wavelets. Philadelphia: Society for Industrial and Applied Mathematics.
- [12] Dittmann, Jana / Franz, Elke / Scheidewind, Antje (2005): Steganographie und Wasserzeichen. Aktueller Stand und neue Herausforderungen. Bode, Arndt (Hg.): Informatik-Spektrum. Bd. 28. Issue 6. Springer-Verlag. S. 453-461.

- [13] Dittmann, Jana (2000): Digitale Wasserzeichen. Grundlagen, Verfahren, Anwendungsgebiete. Berlin u.a.: Springer.
- [14] Erhardt, Angelika (2008): Einführung in die Digitale Bildverarbeitung. Grundlagen, Systeme und Anwendungen. Wiesbaden: Vieweg + Teubner.
- [15] Folland, Gerald B. (1984): Real Analysis. Modern Techniques and Their Applications. New York u.a.: John Wiley Sons.
- [16] Franz, Elke / Pfitzmann, Andreas (1998): Einführung in die Steganographie und Ableitung eines neuen Stegoparadigmas. Bode, Arndt (Hg.): Informatik-Spektrum. Bd. 21. Springer-Verlag. S. 183-193.
- [17] Gonzales, Rafael C. / Woods, Richard E. (2018): Digital Image Processing. New York: Pearson. 4. Auflage.
- [18] Guru, D.S. / Shekar, B.H. / Nagabhushan, P. (2004): A simple and robust line detection algorithm based on small eigenvalue analysis. In: Pattern Recognition Letters. Vol. 25, Issue 1. New York: Elsevier Science Inc. S. 1-13.
- [19] Herodotus: The Histories. London (1992): J.M. Dent & Sons, Ltd.
- [20] Hughes, John F. u.a. (2014): Computer Graphics. Principles and Practice. 3. Auflage. Upper Saddle River, NJ. u.a.: Addison-Wesley.
- [21] Icons: https://icons8.de/icon/set/. (aufgerufen 09.03.2018).
- [22] Jähne, Bernd (2005): Digitale Bildverarbeitung. 6. Auflage. Berlin, Heidelberg: Springer-Verlag.
- [23] Johnson, Neil F. / Katzenbeisser, Stefan C. (2000): A survey of steganographic techniques. In: Katzenbeisser, Stefan / Fabien A.P.Petitcolas (Hg.): Information Hiding Techniques for Steganography and Digital Watermarking. Bosten, Cambride, Massachusetts: Artech House Books. S. 43-78.
- [24] Kahn, David: The History of Steganography (1996). In: Anderson, Ross (Hg.): Information Hiding. First International Workshop Cambridge, U.K., May 30-June 1. S.1-5.
- [25] Kutter, Martin / Hartung, Frank (2000): Introduction to watermarking techniques. In: Katzenbeisser, Stefan / Fabien A.P.Petitcolas (Hg.): Information Hiding Techniques for Steganography and Digital Watermarking. Bosten, Cambridge, Massachusetts: Artech House Books. S. 97-120.
- [26] Lehner, Franz (2001): Einführung in Multimedia. Grundlagen, Technologien und Anwendungsbeispiele. Wiesbaden: Gabler.
- [27] Levi, Paul (1993): Grundlagen und Begriffsdefinitionen des Bildverstehens. In: Radig, Bernd (Hg.): Verarbeiten und Verstehen von Bildern. Handbuch der Informatik Band 6.4. München / Wien: Oldenburg Verlag. S. 21-65.

- [28] Levi, Paul (1993): Lineare Kantenoperatoren und Filter. In: Radig, Bernd (Hg.): Verarbeiten und Verstehen von Bildern. Handbuch der Informatik Band 6.4. München / Wien: Oldenburg Verlag. S. 107-163.
- [29] Lipp, Thomas W. (1997): Grafikformate. Aufbau und Programmierung aller wichtigen Grafikformate und sofort einsetzbare Bibliotheken unter Windows 95 und Windows NT. Unterschleißheim: Microsoft Press.
- [30] Louis, Alfred Karl / Maaß, Peter / Rieder, Andreas (1988): Wavelets. Theorie und Anwendungen. 2. Auflage. Stuttgart: B.G.Teubner.
- [31] Mallat, Stephane (2009): A Wavelet Tour Of Signal Processing. The Sparse Way. 3. Auflage. Amsterdam u.a.: Elsevier.
- [32] Mallot, Hanspeter A. (1998): Sehen und die Verarbeitung visueller Informationen. Eine Einführung. Braunschweig, Wiesbaden: Vieweg.
- [33] Meinel, Christoph / Sack, Harald (2004): WWW. Kommunikation, Internetworking, Web-Technologien. Berlin u.a.: Springer.
- [34] Meyer, Y. (1991): Wavelets and applications. Proceedings of the International Conference Marseille, France, May 1989. Paris u.a.: MASSON.
- [35] Misiti, Michel / Misiti, Yves / Oppenheim, Georges / Poggi, Jean-Michel (2006): Wavelet Toolbox. For Use with MATLAB. User's Guide. Version 3. Natick (Massachusetts): The MathWorks.
- [36] Pickholtz, Raymond L. / Schilling, Donald L. / Milstein, Laurence B. (1982): Theory of Spread-Spectrum Communications – A Tutorial. IEEE Transactions on Comuunication. Vol. 30, No.5. S.855-884.
- [37] Proakis, John G. / Salehi, Masoud (2004): Grundlagen der Kommunikationstechnik.2. Auflage. München u.a.: Pearson Studium.
- [38] Purgathofer, Peter (2010): Eine kurze Geschichte der Steganographie. In: Brandstetter, Manuela, Schmidberger Marina, Sommer, Sabine (Hg.): Die Funktion "verdeckter Kommunikation": Impulse für eine Technikfolgenabschätzung der Steganographie. Lit Verlag. S.65-74.
- [39] Rocha, Anderson / Goldenstein, Siome (2008): Steganography and Steganalysis in Digital Multimedia: Hype or Hallelujah? In: RITA. Volume XV. Numero 1. S. 83-110. http://www.ic.unicamp.br/~siome/papers/Rocha-Rita08.pdf.
- [40] Schneider, Reinhold (1998): Multiskalen- und Wavelet- Matrixkompression. Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme. Wiesbaden: Springer Fachmedien.
- [41] Steinmüller, Uwe / Gulbins, Jürgen (2006): Die Kunst der RAW-Konvertierung. RAW-Files bearbeiten mit führenden RAW-Konvertern. 2. Auflage. Heidelberg: dpunkt.verlag.

- [42] Sweeney, Peter (1992): Codierung zur Fehlererkennung und Fehlerkorrektur. München: Hanser.
- [43] Swoboda, Joachim (1973): Codierung zur Fehlerkorrektur und Fehlererkennung. München, Wien: R.Oldenburg Verlag.
- [44] Taubmann, David S. / Marcellin, Michael W. (2002): JPEG2000. Image Compression Fundamentals, Standards And Practice. Boston / Dordrecht / London: Kluwer Academic Publishers.
- [45] Wickerhauser, Mladen Victor (1996): Adaptive Wavelet-Analysis. Theorie und Software. Braunschweig / Wiesbaden: Vieweg.
- [46] Zhang, Zhen / Ma, Siliang / Liu, Hui / Gong, Yuexin (April 2009): An edge detection approach based on directional wavelet transform. In: Demkowicz, Lszek (Hg.): Computers and Mathematics with Applications. Volume 57, Issue 8. S. 1265-1271. http://www.sciencedirect.com/science/article/pii/S0898122109000492

Erklärung

Name, Vorname: Huber, Zsuzsanna

Universität Passau, Fakultät für Informatik und Mathematik

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt habe. Des Weiteren habe ich keine anderen als die angegebenen Quellenangaben oder Hilfsmittel benutzt. Wörtliche und sinngemäße Zitate sind auch als solche gekennzeichnet.

Datum

Unterschrift des Studierenden