# Entwurf und Implementierung eines mobilen Stereovision-Vorlesegerätes

---

*Am Fachbereich Mathematik und Informatik*

*in der Arbeitsgruppe "Intelligente Systeme und Robotik"*

*der Freien Universität Berlin*

***Dissertation***

*zur Erlangung des akademischen Grades*

*eines Doktors der Naturwissenschaften*


*vorgelegt von Roman Guilbourd*

*Berlin, Januar 2013*

**Betreuer:**

Prof. Dr. Raúl Rojas

Arbeitsgruppe "Intelligente Systeme und Robotik"
Institut für Mathematik und Informatik
Freie Universität Berlin
Arnimallee, 7
14195 Berlin
Deutschland

**Gutachter:**

Prof. Dr. Raúl Rojas, Freie Universität Berlin
Prof. Dr. Marco Block-Berlitz, Hochschule für Technik und Wirtschaft Dresden

**Disputation:**

08.11.2013

*Zusammenfassung*

Ziele dieser Promotionsarbeit sind Entwurf und Implementierung eines tragbaren Vorlesegerätes für blinde und sehbehinderte Menschen. Die diesem Vorhaben inhärente Innovation besteht in erster Linie darin, die erforderliche Mitwirkung des Anwenders bei der Lokalisierung und Erkennung des Textes durch den Einsatz intelligenter Algorithmen aus dem Bereich maschinelles Sehen zu minimieren. Neben den ingenieurtechnischen Herausforderungen bei der Konstruktion des Geräts steht die Integration von Stereovision-Verfahren in die Dokumentverarbeitungskette im Mittelpunkt der Betrachtung. Angesichts der mobilitätsbedingten Ressourcenbeschränkungen gelten dabei hohe Anforderungen an die Effizienz und Robustheit der Algorithmen in allen Teilschritten der Verarbeitung – Echtzeittextdetektion, textspezifischer Bildoptimierung, Layoutanalyse und Korrektur der Verzerrungsartifakte. Die praktische Orientierung der vorliegenden Arbeit spiegelt sich in der engen Zusammenarbeit mit den Betroffenen aus der Zielgruppe bei der Ermittlung der Anforderungen und in der Evaluierungsphase wider.

*Abstract*

The aim of this thesis is to develop a mobile reading device for blind and visually impaired people. The key innovation of the system consists in its ability to assist the user during the image capturing phase utilizing CV algorithms and methods. In addition to the challenge of designing and building the device this work is focused on the pre-processing of document images under hardware limitations of a mobile platform as well as integration of stereo vision techniques into the processing chain. It is shown, that additional depth information can be beneficial for solving some major problems in the field of document analysis such as text-specific image enhancement, layout recognition and de-warping of document images. Due to the practical nature of this work close collaboration with the targeted users has been maintained throughout all phases of the project.

# Abkürzungsverzeichnis

2D, 3D – zweidimensional, dreidimensional

Abb. – Abbildung

CV – maschinelles Sehen *(engl. Computer Vision)*

CG – Computergrafik *(engl. Computer Graphics)*

dpi – Maßeinheit für Punktdichte im Druck *(engl. dots per inch)*

DSP – digitaler Signalprozessor *(engl. Digital Signal Processor)*

fps – Maßeinheit für Bildrate *(engl. frames per second)*

FWT – schnelle Wavelet-Transformation *(engl. Fast Wavelet Transformation)*

GUI – grafische Benutzeroberfläche *(engl. Graphical User Interface*)

MUR – minimal umgebendes Rechteck

OCR – Zeichenerkennung *(engl. Optical Character Recognition)*

PDA – kompakter, tragbarer Computer *(engl. Personal Digital Assistant)*

px, Mpx – Pixel, Megapixel

SDK – Software-Entwicklungswerkzeuge *(engl. Software Development Kit)*

TTS – Sprachsynthese *(engl. Text-To-Speech)*

VGA – Bildauflösung 640x480px (Maximalauflösung von alten VGA-Karten)

# Inhaltsverzeichnis

# 1. Kapitel

# Einführung

## 1.1 Einführung und Motivation

Rund 125.000 Menschen erhielten in Deutschland im Jahr 2008 das Blindengeld, die Gesamtzahl der Blinden und hochgradig Sehbehinderten wird von dem Deutschen Blinden- und Sehbehindertenverband (DBSV) auf ca. 150.000[*] geschätzt. Insgesamt leiden in Deutschland etwa 500.000 Menschen an einer Sehbehinderung. Weltweit wird von über 160 Millionen Betroffenen ausgegangen [1]. Trotz der klar definierten Sehrestgrenzen wirken sich darüber hinaus verschiedene Augenerkrankungen sehr unterschiedlich auf die Lesefähigkeit des Erkrankten aus. Ein Sehrest von unter 5% kann bedeuten, dass der Mensch ein Objekt erst aus 5 m Entfernung erkennen kann, während ein normal Sehender das gleiche Objekt aus 100 m Entfernung erkennt. Das kann aber auch bedeuten, dass der Mensch nur 5% des normalen Gesichtsfeldes deutlich wahrnehmen kann [1]. Je nach Krankheitsbild verwenden die Betroffenen dementsprechend auch unterschiedliche Lesehilfsmittel: Von digitalen Luppen über elektrische Braillezeilen bis hin zu vollautomatischen flachbettscanner-, handscanner- und kamerabasierten Vorlesegeräten – das Angebot ist vielfältig.

---

[*] Als Blind gilt jemand, der auf dem besser sehenden Auge selbst mit Hilfsmitteln nicht mehr als 2% von dem sieht, was ein Mensch mit normaler Sehkraft erkennt. Hochgradig sehbehindert ist jemand, dessen sog. Sehrest unter 5% liegt [1].

Ein Blick auf die Statistik zeigt, dass die altersbedingte Makuladegeneration (s. Abb. 1.1.1) die mit Abstand häufigste Ursache von Neuerblindungen in Deutschland ist [2]. Dies lässt zweierlei Schlussfolgerungen zu: Zum einen muss angesichts der demografischen Situation damit gerechnet werden, dass die Anzahl der Menschen mit einer hochgradigen Sehbehinderung in Deutschland in den nächsten Jahren stark zunehmen wird (der DBSV erwartet einen 30% Zuwachs bis 2030 [2]), zum anderen wird auch das Durchschnittsalter der Betroffenen höher, denn während die Anzahl der Geburtsblinden dank des medizinischen Fortschritts stetig abnimmt, gibt es für viele Formen der altersbedingten Makuladegeneration nach wie vor keine allgemein akzeptierte Behandlung [2]. Die Gruppe der im Alter erblindeten Menschen zeichnet sich indes dadurch aus, dass die Betroffenen mit der neuen Situation oftmals stark überfordert sind. Das Erlernen von speziellen Schriftsystemen wie der Blindenschrift, die die Orientierung für die Blinden erleichtern können, wird mit dem zunehmenden Alter immer problematischer, sodass ganz alltägliche Tätigkeiten wie der Einkauf im Supermarkt oder die Nutzung von öffentlichen Verkehrsmitteln für späterblindete Menschen oft plötzlich eine in vielfacher Hinsicht unüberwindliche Hürde darstellen können. Das führt u. U. dazu, dass die Betroffenen sich von der Außenwelt abschotten und auch Depressionen sind möglich [3]. Ein tragbares und einfach zu bedienendes Vorlesegerät soll ihnen einen Zugang zu den in unserem Leben allgegenwärtigen textuellen Informationen und damit ein selbstbestimmteres Leben ermöglichen. Darüber hinaus kommen auch Menschen mit Leseschwächen und Analphabeten als potentielle Nutzer des Geräts in Frage, allerdings gehören sie nicht zur primären Zielgruppe des Projekts, da die angestrebte Entscheidungsselbstständigkeit des Systems von Normalsehenden als lästige Bevormundung empfunden werden könnte. Schließlich ist die Verwendung der im Rahmen dieser Arbeit entwickelten Textlokalisierungsalgorithmen bei den AR–Systemen (engl. *Augmented Reality*) vorstellbar, die mit ähnlichen Problemstellungen konfrontiert sind.


*Abb. 1.1.1: Makula-Degeneration, Simulation aus [2].*

## 1.2 Geschichte und Vorarbeiten

Die Idee, ein Gerät zur Bildaufnahme, ein Buchstabenerkennungsprogramm (OCR*) zur Texterkennung und ein Programm zur Sprachsynthese zu einem vollautomatischen Vorlesegerät für Blinde zu kombinieren, ist nicht neu. Bereits 1949 entwickelten Ingenieure der amerikanischen Elektronik-Firma *RCA Corporation* ein entsprechendes System. Das Gerät konnte vorgelegte Texte buchstabieren, war damals allerdings noch viel zu teuer für den Privatgebrauch und wurde deswegen nie in Serie produziert [4]. Signifikante Fortschritte in der Entwicklung der OCR-Technologie gab es Ende der 70er Jahre als Postämter mehrerer Staaten Systeme für eine automatische Postleitzahlerkennung einführten [5]. Allerdings waren die ersten OCR-Algorithmen für bestimmte, speziell für die maschinelle Erkennung konzipierte Schriftarten ausgelegt.

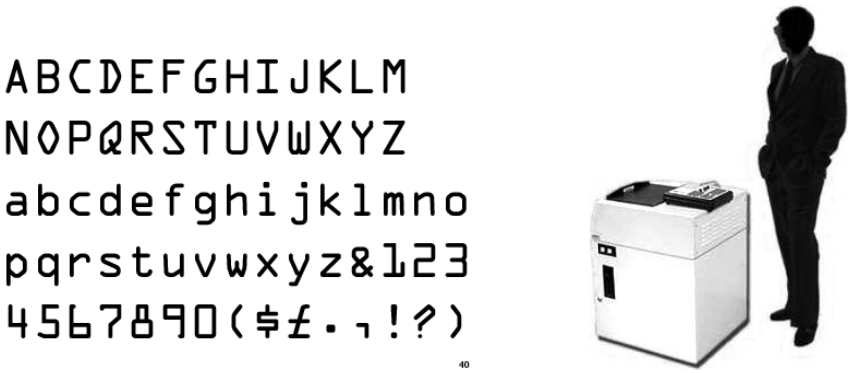


*Abb. 1.2.1: (links) OCR-A - die erste standartisierte maschinenlesbare Schrift der Welt. (rechts) Kurzweil Reading Machine aus [6].*

Das vom Forscherteam um Raymond Kurzweil im Jahr 1974 (s. Abb. 1.2.1 rechts) entwickelte Verfahren zur optischen Zeichenerkennung gehörte zu den ersten s. g. omni-font OCR-Programmen, die robuste Merkmalsextraktionsalgorithmen verwendeten und dadurch in der Lage waren, verschiedene Schriftarten zu erkennen [6]. Seine erste große Anwendung fand der Algorithmus in einem

* Texterkennung und OCR werden im deutschen Sprachraum oft synonym verwendet, so wie auch in dieser Arbeit. Der Vorgang, bei dem die extrahierten Bildsegmente mit den Mustererkennungsalgorithmen untersucht werden, wird hier dagegen als „Zeichenerkennung“ bezeichnet.

Vorlesesystem für Blinde - der berühmten „*Kurzweil Reading Machine*“ (*Kurzweil's Lesemaschine*) [6]. Eine weitere Innovation, die in dem Gerät von Kurzweil Verwendung fand, war der Flachbettscanner [6], wobei das damals entwickelte Konzept auch heute noch in zahlreichen auf dem Markt verfügbaren Vorlesegeräten implementiert ist. Ungeachtet des stolzen Preises von bis zu 50.000 $ gilt die Maschine heute als einer der wichtigsten Beiträge zur Integration von blinden Menschen seit der Erfindung der Braille-Schrift.

Obwohl scannerbasierte Vorlesegeräte gegenwärtig eine beherrschende Marktstellung haben, werden in letzter Zeit immer öfter auch Videokameras zur Dokumentaufnahme eingesetzt, da sie i. Allg. kleiner und schneller sind. So erschien im Jahr 2006, 32 Jahre nach der waschmaschinengroßen „*Kurzweil Reading Machine*“, ein mobiles PDA-integriertes Vorlesegerät von *Kurzweil Technologies*, genannt *K-NFB Reader* [7]. Die rasante Entwicklung der letzten Jahre auf dem PDA/Smartphone-Markt führte zu einer ganzen Reihe von mobilen Smartphone-basierten Hilfsmitteln für Blinde und Sehbehinderte. Bereits im Jahr 2002 implementierten Forscher der *HP Laboratories Bristol* ein Vorlesegerät auf dem 133 MHz schnellen und mit einer VGA-Kamera ausgerüsteten *HP Jornada 568* PDA [8]. Seitdem haben sich die Rechenkapazitäten von Smartphones mehr als verzehnfacht und auch die Bildqualität der eingebauten Kameras ist inzwischen mit der von Kompaktkameras aus dem Industriebereich vergleichbar geworden. Dank des Fortschritts im Bereich des mobilen Internets haben internetfähige Smartphones zudem die Möglichkeit, rechenintensive Verarbeitungsschritte ins Netz auszulagern und dadurch die Latenzzeiten zu reduzieren (vgl. TextScout-System [9]). Trotz der angeführten Vorteile, die PDAs und Smartphones scheinbar als Hardwareplattform für Vorlesegeräte prädestinieren, spielen diese zurzeit noch keine allzu wichtige Rolle auf dem Markt der Hilfsmittel für Blinde (s. Anhang A). Die Gründe dafür liegen auch in den Schwierigkeiten, die Menschen mit einem eingeschränkten Raumgefühl bei der Handhabung der Geräte haben, insbesondere was die Gewährleitung der notwendigen Qualität von Dokumentaufnahmen angeht. Insbesondere die Ausrichtung einer Smartphone-Kamera auf die zu verarbeitende Textregion kann für die Betroffenen eine Herausforderung darstellen (s. Abb. 1.2.2 rechts).

So erklärt es sich, dass in den letzten Jahren gleich mehrere Firmen, darunter bspw. *Intel* mit dem *Intel Reader* [10] (s. Abb. 1.2.2 links), dedizierte kamerabasierte Vorlesegeräte auf den Markt brachten. Die Systeme zeichnen sich typi-

scherweise durch eine stabile Haltevorrichtung für die Kamera aus und müssen vor der Benutzung entsprechend eingerichtet werden. Eine bestimmte Minimalgröße dieser Geräte resultiert alleine schon aus der Tatsache, dass deren Konstruktion einen gewissen Abstand zwischen dem Dokument und der eingebauten Kamera gewährleisten muss, damit die Kamera ein komplettes Dokument mit einem bestimmten Format, typischerweise DIN A4 oder größer, aufnehmen kann. Die Größe hat unterdessen einen erheblichen Einfluss auf die Mobilität und Benutzbarkeit des Systems.


*Abb. 1.2.2: (links) Intel Reader aus [10]. (rechts) Ausrichtungsproblematik bei Smartphone-basierten Vorlesegeräten.*

Parallel zu den Entwicklungen auf dem Markt wurde im universitären Bereich an neuen intelligenten Vorlesesystemen geforscht, die möglichst selbstständig agieren sollten. 1996 schlugen Forscher des Projekts "Tyflos" [11] ein Vorlesesystem vor, bei dem eine helmmontierte Videokamera mit einem auf dem Rücken getragenen Rechner verbunden wurde. Wie auf der Abb. 1.2.3 zu sehen ist, war die Mobilität des Systems alleine schon durch die Größe des Rechners stark eingeschränkt. 2003 präsentierten S. Panchanatha et al. [12] ein im Rahmen des Projekts "iCare" entwickeltes Assistenzsystem, bei dem die Kamera erstmals in ein Brillengestell eingebaut wurde. Die Miniaturkamera wurde an eine PDA angeschlossen, sodass die Mobilität des Geräts kein Problem mehr darstellte. In ihrem Bericht beklagen die Forscher, keine Miniaturkamera mit einer ausreichenden Bildqualität gefunden zu haben, was die Robustheit der Zeichenerkennung beeinträchtigt haben soll.

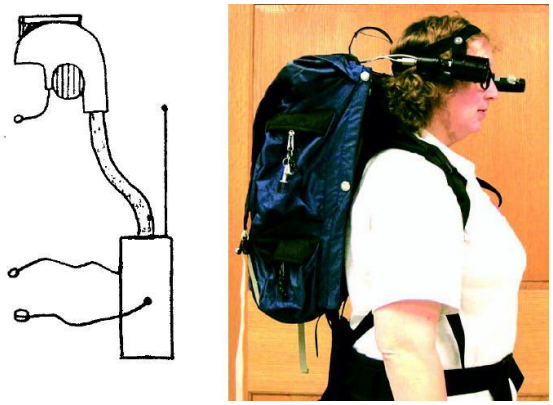

*Abb. 1.2.3: Die ersten Prototypen mobiler kamera-basierter Hilfsmittel für Blinde aus [11].*

Nobuo Ezaki et al. beschreiben in [13] ein PDA-basiertes Vorlesesystem bei dem die Kamera auf der Schulter des Blinden getragen wird, allerdings wird die Bilanz des Projekts an der Stelle nicht präsentiert. In [14] wird ein tragbares System vorgestellt, das in der Lage ist, eine Echtzeit-Textdetektion vorzunehmen, die Angaben zum weiteren Vorgehen des Geräts, insbesondere der Texterkennungsprozedur, konnten jedoch nicht gefunden werden. Schließlich wurden in den letzten Jahren diverse weitere kamerabasierte Assistenzsysteme für Blinde entwickelt, deren Kamera-Komponenten mittels eines Brillengestells bzw. eines Kopfhörerbügels auf dem Kopf des Anwenders befestigt werden [15][16][17][18][19][20] (s. Abb. 1.2.4). Allerdings handelt es sich bei diesen Systemen um Navigationshilfen, die den Blinden Orientierungshinweise sowie Hinderniswarnungen liefern. Die Vorlesefunktionalität wird, wenn überhaupt, nur am Rande als ein theoretisch möglicher Zusatzdienst erwähnt und nicht weiter diskutiert [20]. Trotz intensiver Suche konnte keine Beschreibung eines nachweislich funktionierenden, tragbaren und weitgehend selbstständig agierenden Vorlesegerätes gefunden werden, was nicht zuletzt auf die hohen Mindestanforderungen an die Hardware zurückzuführen ist. Mittlerweile hat die rasante technische Entwicklung der letzten Jahre Miniaturkameras und mobile Rechensysteme hervorgebracht, die diese Anforderungen erfüllen.

Dokumentverarbeitung ist bereits seit Jahrzehnten ein Objekt intensiver Forschung. Moderne Programme für die Digitalisierung von Dokumenten können mit diversen Schriftarten, -größen sowie verschiedenen Layout-Typen mittlerweile sehr gut arbeiten (s. dazu Abschnitt 3.5). Demgegenüber stehen Systeme für die Verkehrsüberwachung, die zwar „nur“ speziell formatierte Kennzeichen erkennen

können, allerdings eine besondere Robustheit in Bezug auf die Aufnahmebedingungen aufweisen. Das zu entwickelnde Vorlesegerät soll die Robustheit eines Systems für die Verkehrsüberwachung mit der Leistung einer modernen Anwendung für die Dokumentendigitalisierung so gut wie möglich zu einer mobilen und zuverlässig funktionierenden Lösung kombinieren.


*Abb. 1.2.4: Brillenförmige Blindenhilfsmittel: a."vOICe" Projekt aus [17] b."Virtual acoustic space" Projekt aus [19] c. "DORA" Projekt aus [18]. d. "Tyflos" Projekt aus [11] e. ein Prototyp des Geräts.*

## 1.3 Aufbau der Arbeit

Der ingenieurwissenschaftliche Beitrag dieser Arbeit besteht in der Entwicklung und Umsetzung eines mobilen Vorlesegerätes für Blinde. Der praktische Hintergrund des Projekts schlägt sich auch in der Gliederung der Arbeit nieder, die vom Wasserfallmodell [21] des Softwareentwicklungsprozesses inspiriert wurde. Nach der Erläuterung der theoretischen Grundlagen in Kapitel 2 wird in Kapitel 3 eine ausführliche *Anforderungsanalyse* basierend auf den Ergebnisse des InformA-Projekts [22] vorgenommen. Im darauffolgenden Kapitel 4 wird ein *Systementwurf* präsentiert, das Verarbeitungskonzept diskutiert und die Hardwarekomponenten festgelegt. Die im Laufe des Entwurfsprozesses identifizierten Aufgaben werden in den folgenden Kapiteln in der Reihenfolge ihres Auftretens thematisiert. In Kapitel 5 werden Aufgaben zusammengefasst, die zur Vorbereitung der Bildaufnahme und Sicherung der notwendigen Aufnahmequalität dienen. Methoden zur *schnellen Textlokalisierung* und *automatischen Einstellung der Kamera* werden vorgestellt. Das Thema des 6. Kapitels ist die Erkennung des physischen Dokumentlayouts, insbesondere *Segmentierung* von Dokumentaufnahmen und *sorgfältige Textlokalisierung*. Kapitel 7 behandelt die *Entzerrung* von Dokumentbildern auf Basis von Stereoaufnahmen sowie *Dokument-Stitching* – eine Zusammenführung von mehreren Teilen einer Aufnahme zu einem einzigen Dokument. In Kapitel 8 geht es um die logische Struktur eines Dokuments und die Bestimmung der Vorlesereihenfolge für die erkannten Textstellen. Schließlich wird in Kapitel 9 das Gesamtergebnis bewertet und eine Zusammenfassung der wissenschaftlichen Beiträge der Arbeit gegeben. Die Kapitel 5 bis 8, in denen die Teilaufgaben der Dokumentverarbeitung behandelt werden, haben eine einheitliche Struktur: Nach einer Erläuterung der jeweiligen Problemstellung werden Vorarbeiten diskutiert, die für die im Anschluss vorgestellte Lösung relevant sind. Abschließend wird eine quantitative Auswertung der jeweiligen Implementierung präsentiert.

# 2. Kapitel

# Notation und theoretische Grundlagen

In diesem Kapitel wird ein Überblick über die grundlegenden Konzepte der Bildverarbeitung gegeben, die eine Relevanz für die vorliegende Arbeit haben.

## 2.1 Bildrepräsentation

Unter dem Begriff "maschinelles Sehen" wird im Rahmen dieser Arbeit eine Anwendung von Technologien und Methoden der digitalen Bildverarbeitung zur Nachbildung von Fähigkeiten des menschlichen, visuellen Systems verstanden. Der Begriff "Bild" ist somit von zentraler Bedeutung und bezeichnet in dem gegebenen Zusammenhang eine flächenhafte Verteilung der Bestrahlungsstärke in einer Ebene. Aus mathematischer Sicht handelt es sich dabei um eine kontinuierliche Funktion von zwei räumlichen Variablen: $I_{cont}(x, y)$ [23]. Während eines Digitalisierungsvorgangs wird aus einem kontinuierlichen Signal $I_{cont}(x, y)$ mittels Abtastung und Quantisierung ein zeit- und wertdiskretes Signal $I(x, y)$ gewonnen, wobei ein Abtastwert eines Bildsignals als *Pixel* bezeichnet wird. Dementsprechend kann ein *idealisiertes*, digitales Bildsignal als Produkt des kontinuierlichen Signals mit dem Dirac-Kamm $Ш_{\Delta}(x) = \sum_{n \in Z} \delta(x - n\Delta)$ [23] beschrieben werden:

$$I(x,y) = I_{cont}(x,y) \sum_{n \in Z} \sum_{m \in Z} \delta(x - m\Delta_x, y - n\Delta_y)$$

Mit $\delta$ wird hier die Dirac-Funktion, $\Delta_x, \Delta_y$ – zwei Perioden, je eine pro Bilddimension, und mit $m, n \in Z$ – ganzzahlige Indizien notiert. Eine solche Darstellung eines Bildsignals mit Hilfe ortsabhängiger Basisbilder wird als räumliche Darstellung eines Bildes oder *Ortsdarstellung* bezeichnet. Manche Lösungsansätze dieser Arbeit basieren auf einer anderen Darstellungsform – der *Frequenzdarstellung* – bei der periodische Funktionen ohne Ortsbezug als Basis dienen. Während die räumliche Bildrepräsentation zur Darstellung von lokalen Merkmalen eines Bildes besonders gut geeignet ist, werden im Frequenzraum die globalen Eigenschaften hervorgehoben [23].

## 2.2 Farbräume

Die Festlegung einer Basis macht es möglich, digitale Bilder in Form von Matrizen zu repräsentieren. Ein Wechsel zwischen verschiedenen Darstellungsformen entspricht dann einer Koordinatentransformation und viele Bildmanipulationen lassen sich in dem Fall als Matrix-Operationen beschreiben. Die räumliche Darstellungsform kann als besonders intuitiv und anschaulich ausgezeichnet werden, wobei Elemente einer Matrix die Helligkeitswerte der Pixel repräsentieren. Die Kodierung der Helligkeitswerte hängt maßgeblich von dem gewählten Farbraum ab, wobei im Rahmen dieser Arbeit folgende Farbräume eine besondere Bedeutung haben:

$$RGB - Bilder: I^{RGB}(x,y) \in [0{,}255]^3$$

$$Grauwertbilder: I^{G}(x,y) \in [0{,}255]$$

$$Binärbilder: I^{B}(x,y) \in \{0{,}1\}$$

Im Falle des RGB-Farbraums besitzt jeder Pixel drei Farbkanäle, je einen für Rot $I_{rot}(x,y)$, Grün $I_{grün}(x,y)$ und Blau $I_{blau}(x,y)$. Die von einer Farbkamera gelieferten Aufnahmen liegen häufig als RGB-Bilder vor. Da Farbinformationen i. d. R. keine Relevanz für die Zeichenerkennung haben, kann zu Beginn der OCR-Verarbeitung eine Konvertierung der Aufnahmen zu Grauwertbildern vorgenommen werden. Dabei werden die Helligkeitswerte aus den drei Kanälen in Anleh-

nung an die menschliche Farbwahrnehmung gewichtet zu einem einzigen Intensitätswert zusammengefasst [26]:

$$I(x,y) = 0.299 \cdot I_{rot}(x,y) + 0.587 \cdot I_{grün}(x,y) + 0.114 \cdot I_{blau}(x,y)$$

Das auf diese Weise produzierte Grauwertbild kann immer noch Informationen enthalten, die für eine Zeichenerkennung überflüssig oder sogar hinderlich sind, wie die Informationen über die Hintergrundgestaltung und die Beleuchtungsunterschiede in verschiedenen Bereichen der Aufnahme. Um OCR-relevante Bildmerkmale noch deutlicher hervorzuheben, wird mit *Binarisierung* häufig ein weiterer Abstraktionsschritt vor der eigentlichen Zeichenerkennung vorgenommen. Binarisierungsmethoden für Dokumentaufnahmen werden nach einer formalen Einführung der Bildverarbeitungsoperationen ausführlich diskutiert.

## 2.3 Faltungsfilter

Die im Rahmen dieser Arbeit verwendeten Bildverarbeitungsoperationen lassen sich aufgrund ihrer Lokalitätseigenschaften in *Punktoperationen*, *Nachbarschaftsoperationen* und *globale Operationen* unterteilen. Punktoperationen arbeiten mit den Helligkeitswerten einzelner Pixel und werden zur Lösung von vielen Standardaufgaben gebraucht. Konvertierung von Bilddaten zwischen verschiedenen Farbräumen stellt bspw. eine typische Punktoperation dar. Im Gegensatz zu den Punktoperationen ziehen die Nachbarschaftsoperationen eine gewisse lokale Umgebung von Pixeln in die Berechnung mit ein.

Eine Anwendung von Nachbarschaftsoperationen im Ortsraum setzt die Festlegung einer Nachbarschaftrelation zwischen den Pixeln voraus. Unter den wesentlichen Nachbarschaftskonzepten sind insbesondere Vierer- und Achter-Nachbarschaft zu nennen (s. Abb. 2.3.1). Die Größe sowie die Position des Referenzpunktes sind zwei weitere charakteristische Merkmale einer Nachbarschaftsoperation.

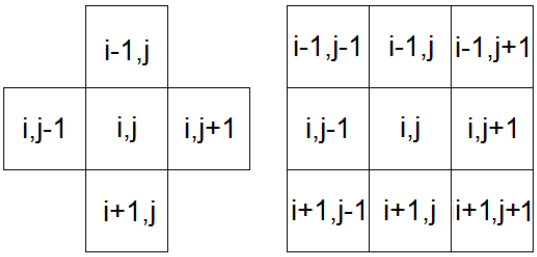


*Abb. 2.3.1: (links) 4-er Nachbarschaft, (rechts) 8-er Nachbarschaft*

*Faltungsoperationen* sind typische Vertreter von Nachbarschaftsoperationen, wobei die Verknüpfung der Pixelwerte aus einer vordefinierten Umgebung nach dem Vorbild der (diskreten) mathematischen Faltung stattfindet:

$$(I * F)(n) = \sum_{k \in D} F(k) I(n - k)$$

Der Faltungsoperator $F$ wird häufig auch *Maske*, *Kernel*, *Kern* oder einfach nur *Filter* genannt. In der Praxis wird der Definitionsbereich $D$ in Abhängigkeit von der Nachbarschaftsumgebung festgelegt, sodass die Filtergröße mit der Größe der Umgebung übereinstimmt (engl. *finite impulse response filter (FIR-Filter)*). Aufgrund des beschränkten Definitionsbereichs realer Bilder führt die Anwendung von Faltungsfiltern zu Problemen im Randbereich einer Aufnahme. Im einfachsten Fall werden die Helligkeitswerte außerhalb der Bildgrenzen pauschal auf 0 gesetzt. Alternativ werden die Werte außerhalb der Bildgrenzen unter Einsatz verschiedener Methoden [26] extrapoliert.

Lineare Faltungsfilter spielen im Bereich der Bildverarbeitung eine wichtige Rolle. Damit lassen sich u. a. grundlegende Operationen wie Glättung (engl. *smoothing*), Schärfen (engl. e*dge enhancement*), Kantendetektion und Rauschreduktion durchführen:

- Glättung, Rauschunterdrückung: *Mittelwert-Filter, Gauß-Filter, Binomial-Filter [41]*
- Schärfen, Kantendetektion: *Laplace-Filter, Sobel-Filter, Scharr-Filter [41]*

Die oben aufgeführten Operatoren besitzen Tiefpass- bzw. Bandpassfilter-Eigenschaften, sodass ihre Anwendung bei der Dokumentverarbeitung häufig mit spezifischen Problemen verbunden ist. So bewirken Glättungsoperationen eine gewisse Reduktion des hochfrequenten Rauschens, verschmieren jedoch gleichzeitig die Buchstabenkonturen, während Schärfungsfilter eine gegenteilige Wirkung haben.

Viele wichtige Bildverarbeitungsalgorithmen basieren auf linearen Faltungsoperationen oder setzen diese im Laufe der Berechnung ein. Dazu gehören Kantenextraktion, Berechnung von Bildpyramiden, Bildsegmentierungsmethoden sowie

viele Mustererkennungsalgorithmen [41]. Darüber hinaus werden sie aufgrund ihrer Effizienz als integrale Bestandteile komplexerer nicht-linearer Operatoren wie bspw. bilateraler Filter [24] verwendet.

## 2.4 Morphologische Filter

Morphologische Filterungsmethoden gehören ebenfalls zur Klasse der Nachbarschaftsoperationen und können als eine Art binäre Faltung aufgefasst werden [23]. Ursprünglich zur Formanalyse von Objekten in Binärbildern $I^B$ entwickelt, basiert die morphologische Bildverarbeitung auf den Mengenoperationen der mathematischen Morphologie *Erosion* und *Dilatation*:

$$Erosion: \quad (I^B \ominus \mathcal{M})(x,y) = \bigcap_{(i,j)\in\mathcal{M}} I^B(x+i, y+j)$$

$$Dilatation: \quad (I^B \oplus \mathcal{M})(x,y) = \bigcup_{(i,j)\in\mathcal{M}} I^B(x+i, y+j)$$

Hier bezeichnen ∪,∩ – Vereinigungs- bzw. Schnittmengenzeichen. Der Filter $\mathcal{M}$ wird in dem Fall oft als *strukturierendes Element* oder auch *Strukturmaske* genannt. Die zu detektierenden Form- und Struktureigenschaften werden durch die Wahl der Form und der Größe von der Strukturmaske festgelegt. Eine Erosionsoperation bewirkt, dass alle Vordergrundobjekte, die die Maske nicht vollständig ausfüllen, aus dem Bild entfernt werden, wohingegen eine Dilatation das Gleiche mit den „Löchern“ (zusammenhängende Hintergrundbereiche) der entsprechenden Form macht. Auch der Rand von den übrig bleibenden Strukturen wird dabei verändert, sodass die beiden Basisoperationen häufig aufeinanderfolgend angewendet werden, um die Größe von Objekten aufrechtzuerhalten. Folgende morphologische Operatoren spielen bspw. in der Praxis eine wichtige Rolle [23]:

$$Öffnen: \quad I^B \circ \mathcal{M} = (I^B \ominus \mathcal{M}) \oplus \mathcal{M}$$

$$Schließen: \quad I^B \bullet \mathcal{M} = (I^B \oplus \mathcal{M}) \ominus \mathcal{M}$$

$$Hit-Miss-Operator: \quad I^B \odot (\mathcal{M}_1, \mathcal{M}_2) = (I^B \ominus \mathcal{M}_1) \cap ((I^B)^{-1} \ominus \mathcal{M}_2)$$

$$Randextraktion: \quad dI^B = I^B \cap ((I^B)^{-1} \oplus \mathcal{M}),$$

Dabei bezeichnet $(I^B)^{-1}$ das Komplementärbild von $I^B$. Der Anwendungsbereich der morphologischen Verarbeitung erstreckt sich von Merkmalsdetektion und Rauschunterdrückung bis hin zu Bildsegmentierung und Formanalyse. Das Konzept der morphologischen Verarbeitung kann in einer verallgemeinerten Form auch für Grauwertbilder angewandt werden [41].

## 2.5 Geometrische Bildoperationen

Sowohl die Punkt- als auch die Nachbarschaftsoperationen transformieren die Helligkeitswerte der Pixel ohne ihre Positionen zu verändern. Geometrische Transformationen haben indes einen gegenteiligen Effekt und können Modifikationen der Bildgröße und der grundlegenden Struktur eines Bildes hervorrufen. In den meisten Fällen handelt es sich dabei um globale Operationen, die als Funktionen $G$ auf der Menge der Bildkoordinaten $p = (x, y)$ definiert sind:

$$(x', y') = G(x, y)$$

Hier werden mit $(x', y')$ die transformierten Koordinaten von $p$ notiert. Infolge der diskreten Natur von Digitalaufnahmen kann es im Zuge einer geometrischen Transformation zu unerwünschten Effekten wie Lücken oder Überlappungen im transformierten Bild kommen, sodass in der Praxis häufig die Umkehrabbildung (engl. *inverse mapping*) $G^{-1}$ der Bildpunkte berechnet wird:

$$(x, y) = G^{-1}(x', y')$$

Pixelwerte, die außerhalb der Gitterpunkte liegen, müssen für eine solche Umrechnungsoperation interpoliert werden, wofür unterschiedlich Methoden wie z. B. Nächster-Nachbar-Interpolation oder verschiedene Formen von Polynominterpolation – insbesondere bilineare und bikubische Interpolation [41] – in Frage kommen. Da eine perfekte Rekonstruktion von digitalisierten Signalen prinzipiell nicht möglich ist [23], führen geometrische Transformation häufig zu einer Verschlechterung der Bildqualität (Verwischung, Aliasing-Effekte).

Wichtige Anwendungsbereiche für geometrische Bildoperationen sind die Modellierung des Aufnahmevorgangs sowie die Korrektur von geometrischen Verzerrungen, die von dem optischen System einer Kamera verursacht werden. Affine Abbildungen wie *Translation, Rotation, Scherung und Skalierung* gehören zu den

geometrischen Elementaroperationen und können in homogenen Koordinaten wie folgt in Matrixform dargestellt werden [23]:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Hier bezeichnen $(t_x, t_y)^T$ die beiden Komponenten des Translationsvektors, während die restlichen Koeffizienten $(a_{11}, a_{12}, a_{21}, a_{22})$ die vier Freiheitsgrade einer affinen Transformationen repräsentieren [23]. Die affine Geometrie wird u. a. zur Berechnung von Parallelprojektionen angewandt, wobei eine idealisierte Objektdarstellung auf einer Abbildungsebene ohne Beachtung von perspektivischen Effekten erfolgt.

Perspektivische Projektionen stellen eine Generalisierung der affinen Transformationen dar und haben in homogenen Koordinaten (mit der homogenen Komponente $w'$) die Form [23]:

$$\begin{pmatrix} w'x' \\ w'y' \\ w' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Sie bilden eine Grundlage für das einfache Modell optischer Abbildungssysteme, mit dessen Hilfe ein Bildaufnahmevorgang als perspektivische Projektion aus dem 3D-Raum der Weltkoordinaten auf die 2D-Sensorebene formalisiert werden kann [23]. Mit Hilfe von perspektivischen Transformationen lassen sich insbesondere perspektivische Verzerrungseffekte beschreiben.

Transformationen, die als Hintereinanderausführung zweier und mehrerer perspektivischen Projektionen repräsentiert werden können, spielen für das Stereokamera-Modell eine besondere Rolle und werden im Rahmen dieser Arbeit als *homographische Abbildungen* oder *Homographien* bezeichnet. Mit Hilfe von Homographien lassen sich Beziehungen zwischen Punkten zweier Projektionsebenen und damit auch zwischen zwei verschiedenen Ansichten einer Szene beschreiben [25]. Eine Homographie kann in homogenen Koordinaten durch eine reguläre 3x3 Abbildungsmatrix mit acht Freiheitsgraden $(h_{11}, \ldots, h_{32})$ dargestellt werden, wobei es sich beim neunten Matrixelement $h_{33}$ um einen Skalierungsfak-

tor handelt, der in einem homogenen Koordinatensystem frei wählbar ist [26]:

$$\begin{pmatrix} w'x' \\ w'y' \\ w' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Zur Schätzung einer homographischen Abbildung, sind dementsprechend mindestens acht Abbildungskoordinaten von vier Punkten der Ausgangsebene notwendig.

Neben den beschriebenen linearen und perspektivischen Transformationen kommen auch komplexere nicht-lineare Abbildungen bei der Definition von geometrischen Operationen zum Einsatz. So wird die rotationssymmetrische Verzeichnung eines Linsensystems häufig durch einen an dem Hauptpunkt zentrierten 2D-Polynom modelliert. Die (Ent-)Verzerrung eines Bildes unter Verwendung von geometrischen Transformationen wird im Rahmen dieser Arbeit auch als *(De-) Warping* (engl. für (Ent-)Krümmung) bezeichnet.

## 2.6 Kalibrierung von Stereokameras

Die Verwendung einer Stereokamera für die OCR-bezogenen Aufgaben gehört zu den wichtigsten Beiträgen dieser Arbeit. Im Folgenden werden die Grundlagen von stereovisionbasierten Methoden unter Verwendung von geometrischen Bildoperationen und der Multiple-View- (engl. für *Mehrsichten*-) Geometrie [25] vorgestellt. Die Abb. 2.6.1 c zeigt eine schematische Darstellung eines einfachen Models der stereokamerabasierten Distanzmessung mit Hilfe der Triangulation. Alle Punkte im Weltkoordinatensystem, die auf dem Hauptstrahl einer Kamera liegen, werden auf einen Punkt in den Kamerakoordinaten abgebildet, sodass nicht mehr zwischen verschiedenen Objektebenen unterschieden werden kann. Diese Unterscheidung wird jedoch mit einer weiteren Aufnahme derselben Szene aus einer anderen Raumposition möglich, da die Hauptstrahlpunkte von der ersten Kamera in der zweiten Aufnahme entlang einer Geraden – der *Epipolarlinie* – liegen [26]. Für die Berechnung der Kamerakoordinaten eines Punktes im 3D-Raum müssen die Bildkoordinaten der beiden Bildhauptpunkte $(c_x, c_y)$, die fokalen Längen der Kameras $f$ sowie die beiden Entfernungen von dem jeweiligen Bildzentrum bekannt sein. Anhand der Entfernungsdifferenz, auch Disparität genannt, lässt sich der Abstand zur Bildebene unter Verwendung der Triangulation

ermitteln (s. Abb. 2.6.1 a).

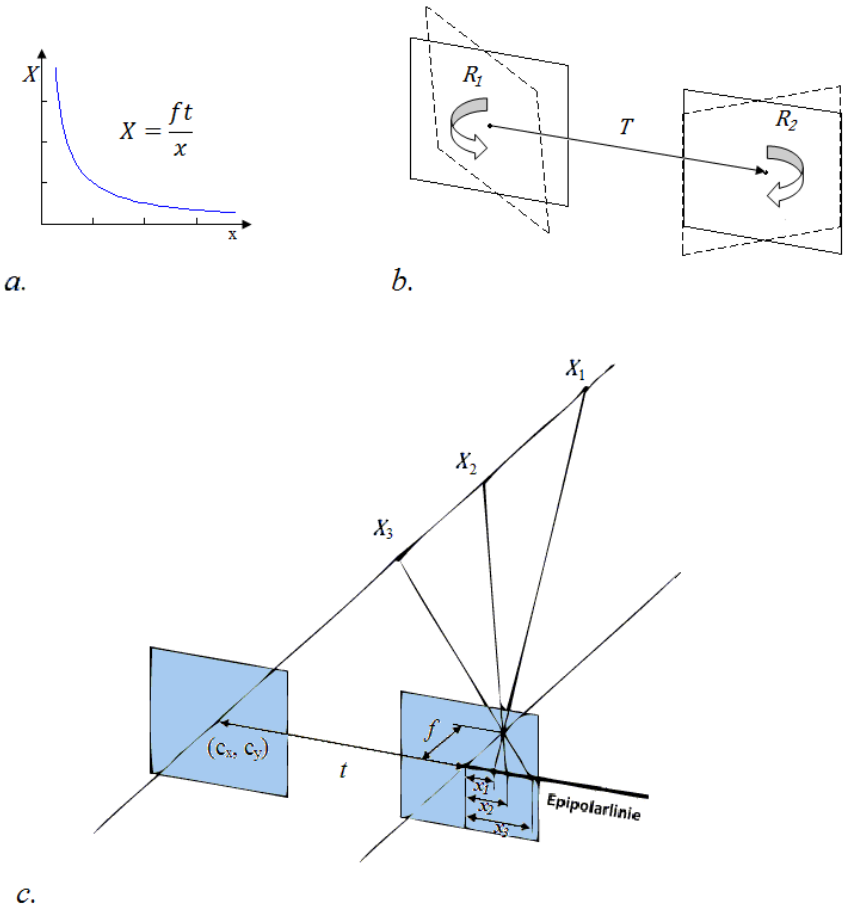


*Abb. 2.6.1: a. Abfall der Genauigkeit mit steigender Distanz b. Rektifizierung und Zeilenausrichtung c. Distanzmessung mit Triangulation.*

Eine Voraussetzung für die stereokamerabasierte Distanzmessung ist eine im Vorfeld stattfindende Parametrisierung des Kameramodells. Intrinsische (innere) Parameter einer Kamera beschreiben den Zusammenhang zwischen dem 3D-Kamera- und dem 2D-Sensorkoordinatensystem, während extrinsische (äußere) Parameter die Weltkoordinaten-zur-Kamerakoordinaten-Abbildung formal darstellen. Die wichtigsten intrinsischen Parameter $c_x, c_y, f_x, f_y$ werden häufig in einer s. g. *Kameramatrix* $C$ zusammengefasst:

$$C = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

Die beiden Kameramatrizen $C_1, C_2$ einer Stereokamera werden unabhängig voneinander anhand von Aufnahmen mehrerer koplanarer Punkte mit einer bekannten Anordnung (*Schachbrettmuster*) berechnet [26]. Typischerweise werden parallel dazu die Entzerrungsparameter zur Korrektur der radialen und tangentialen Verzeichnungen [27] einer Kamera bestimmt.

Die relativen Position und Orientierung der beiden Sensorebenen zu einander werden durch eine *Essential-Matrix* $E$ beschrieben, die aus einer Verschiebungskomponente $T$ und einer Rotationskomponente $R$ besteht (s. Abb. 2.6.1b):

$$E = R \cdot T$$

Die Kombination der Essential-Matrix mit den beiden Kameramatrizen $C_1, C_2$ ergibt die *Fundamentalmatrix F*

$$F = (C_2^{-1})^T \cdot E \cdot C_1^{-1},$$

die die Beziehung zwischen den beiden Bildkoordinatensystemen einer Stereoaufnahme beschreibt. Insbesondere lässt sich mit Hilfe der Fundamentalmatrix jeder Punkt eines Bildes auf die zugehörige Epipolarlinie im anderen Bild abbilden. Für eine effiziente Bestimmung von Korrespondenzpunkten, auch Stereokorrespondenzproblem oder Stereokorrespondenzsuche genannt, ist es wichtig, dass alle Epipolarlinien waagerecht verlaufen und den gleichen Abstand zum jeweiligen Bildhauptpunkt haben, sodass die Suche auf eine Dimension beschränkt werden kann. Eine solche geometrische Transformation der Bildpunkte wird im Rahmen dieser Arbeit als *Rektifizierung* bezeichnet. Nach einer Rektifizierung der beiden Aufnahmen kann die Abstandsmessung für Bildpunkte mit der Triangulierungsformel erfolgen, sodass die Transformation eines Bildpunkte $p = (x, y)$ (s. Abb. 2.6.1 a) in das Kamerakoordinatensystem mit der Reprojektionsmatrix $Q_{rect}$ stattfinden kann [26]:

$$Q_{rect} = \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & f \\ 0 & 0 & -1/t & 0 \end{pmatrix}$$

sodass

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ W \end{pmatrix} = Q_{rect} \cdot \begin{pmatrix} x \\ y \\ d \\ 1 \end{pmatrix}$$

Hier bezeichnet $d$ die Stereo-Disparität für den Punkt $(x, y)$. Die Kamerakoordinaten $(X, Y, Z)$ des Punktes $p$ berechnen sich schließlich als [26]

$$(X, Y, Z) = (X'/W\,, Y'/W\,, Z'/W).$$

Die Parametrisierung der Fundamentalmatrix findet in der Praxis anhand bekannter Korrespondenzpunktpaaren $\vec{x}, \vec{x}'$ und unter Verwendung der Multiple-View-Geometrie statt, wobei sich die neun Koeffizienten von $F$ als Lösung des Gleichungssystems $(\vec{x}')^T F \vec{x} = 0$ ergeben. Da die Fundamentalmatrix einzelne Punkte auf Linien abbildet, hat die Koeffizientenmatrix des Gleichungssystems den höchstmöglichen Rang acht, sodass theoretisch acht Korrespondenzpaare für die Kalibrierung ausreichend sind. In der Praxis werden aufgrund der Messfehler i. d. R. jedoch weitaus mehr Korrespondenzpunkte berücksichtigt [26].

Die Genauigkeit der Distanzmessung $\sim 1/\Delta Z$ mit Hilfe der stereovisionbasierten Verfahren hängt von mehreren Parametern ab. Die Tiefenauflösung einer frontalparallel ausgerichteten Stereokamera ist in der näher der Kameras am größten und nimmt mit steigendender Distanz $Z$ quadratisch ab [26]:

$$\Delta Z = \frac{Z^2}{ft} \Delta d$$

$f$ bezeichnet hier die fokalen Längen, $t$ – den Abstand zwischen den Bildhauptpunkten der beiden Kameras (s. Abb. 2.6.1 c) und $\Delta d$ – den minimalen messbaren Disparitätswert, der mit der Bildauflösung zusammenhängt.

## 2.7 Binarisierung von Dokumenten

Unter *Binarisierung* $\mathcal{B}$ wird die Erzeugung eines *Binärbildes* $I^B$ verstanden, die mit einer Reduktion der Farbtiefe einhergeht. Für Grauwertbilder $I^G$ der ße $M \times N$ lässt sich die Reduktion wie folgt formal darstellen:

$$\mathcal{B}: [0{,}255]^{M \times N} \rightarrow \{0{,}1\}^{M \times N}$$

Im Rahmen dieser Arbeit wird Binarisierung vor allem für die Zwecke der Bildsegmentierung eingesetzt, wobei die Pixel einer Aufnahme ihrer aufgabenspezifischen Relevanz entsprechend als *Vordergrund* bzw. *Hintergrund* klassifiziert werden. Das resultierende Binärbild stellt das Ergebnis der Klassifikation in kodierter Form dar, wobei die relevanten Bildbereiche (engl. *region of interest (ROI)*) typischerweise mit "1" und der Rest durch "0" markiert werden.

*Schwellenwertverfahren* gehören zu den einfachsten Binarisierungsmethoden. Die Grenze zwischen den beiden Klassen wird dabei durch eine oder mehrere scharfe Trennlinien $\theta(x, y)$ definiert:

$$I^G(x, y) = \begin{cases} 1, falls\ I^G(x, y) \geq \theta(x, y) \\ 0, falls\ I^G(x, y) < \theta(x, y) \end{cases}$$

Die Verteilung der Schwellenwerte $\theta(x, y)$ kann je nach Verfahren unterschiedlich ausfallen. Im einfachsten Fall wird ein globaler Wert für das gesamte Bild verwendet. Solche *globalen Schwellenwertverfahren* sind nur dann sinnvoll, wenn die Helligkeitsunterschiede im Bild gering sind und ein global optimaler Wert existiert. Demgegenüber stehen *lokale Schwellenwertverfahren*, die mehrere lokal gültige Werte festlegen und dadurch weniger anfällig für Helligkeitsschwankungen sind [26]. Die größere Robustheit der lokalen Verfahren geht jedoch häufig mit einem höheren Rechenaufwand einher. Im Extremfall muss an jeder Bildposition eine separate Auswertung der Schwellenwerte vorgenommen werden.

Eine Binarisierung kann auch als Clustering-Problem für die Pixelintensitätswerte aufgefasst werden. Das Verfahren von Otsu [28] gehört zu den bekanntesten und ältesten globalen Schwellenwertverfahren. Die Berechnung der Schwelle basiert auf einer iterativen Minimierung der Streuung innerhalb der Cluster bei ihrer gleichzeitigen Maximierung zwischen den Clustern. Andere Autoren [29][30]

setzen an der Stelle Entropie-basierte Ähnlichkeitsmaße ein. In [31] erfolgt die Ähnlichkeitsbewertung über die Abweichung der Pixelintensitätswerte von der Normalverteilung.

Bei lokalen Schwellenwertverfahren muss aus Effizienzgründen häufig auf iterative Optimierungsmethoden verzichtet werden. Stattdessen können parameterabhängige Näherungen berechnet werden, wie bspw. in der Methode von Niblack [32]:

$$\theta(x,y) = \tilde{\mu}(x,y) - \gamma\tilde{\sigma}(x,y)$$

Hier sind $\tilde{\mu}(x,y)$ der geschätzte Mittelwert und $\tilde{\sigma}(x,y)$ – die Standardabweichung der Pixelwerte in einer kleinen Nachbarschaft um die Stelle $(x,y)$. Der Parameter $\gamma$ muss dabei in Abhängigkeit von der Größe der Vordergrundbereiche und der Nachbarschaftsgröße im Voraus festgelegt werden.

Die Skalierungsproblematik ist typisch für lokale Schwellenwertverfahren [33], die deswegen häufig mit globalen Segmentierungsverfahren kombiniert werden [34][35]. Ein solches Hybridverfahren stellt auch Li et al. in seiner Arbeit [36] vor. Im ersten Schritt werden die Grenzbereiche, in denen Vor- und Hintergrundpixel nah beieinander liegen, durch eine Kantenextraktion identifiziert. Im direkten Vergleich können die lokal gültigen Schwellenwerte häufig zuverlässiger bestimmt werden als wenn der räumliche Bezug vernachlässigt wird, wie das z.B. bei histogrammbasierten Methoden der Fall ist.

Die LCS (engl. *Local Contrast Segmentation*)-Methode von Block et al. [37] basiert ebenfalls auf einer Kantenextraktion. Die Kanten werden in diesem Fall anhand hoher Übergangsenergie (engl. *transition energy*) der zugehörigen Pixel identifiziert, wobei die Berechnung des Energiebildes $E$ mit Hilfe des DOG-Filters (engl. *Difference of Gaussian (DOG)*) erfolgt [37]:

$$E(I,H,v) = v \cdot (I - I * H)$$

Dabei wird das mit einem Gaußkern $H$ geglättetes Bild vom Originalbild $I$ abgezogen und mit einer Variablen $v$ skaliert, die in Abhängigkeit von den zu extrahierenden Merkmalen gewählt wird. Pixel mit einer hohen positiven Energie werden

als Vordergrund klassifiziert, die mit einer hohen negativen als Hintergrund und die Pixel mit einem Wert nahe 0 werden vorerst keiner Klasse zugeordnet [37]:

$$E^{+}(I,H,v,\theta_{\varepsilon+}) = max(E(I,H,v),\theta_{\varepsilon+}), \qquad mit\ v > 0,$$

$$E^{-}(I,H,v,\theta_{\varepsilon-}) = max(E(I,H,v),\theta_{\varepsilon-}), \qquad mit\ v < 0,$$

Hier bezeichnet $E^{+}$ das Vordergrundbild, $E^{-}$ das Hintergrundbild und $\theta_{\varepsilon+}, \theta_{\varepsilon-}$ skalierungsabhängige Schwellenwerte. Durch eine im Anschluss stattfindende Dilatation werden die Lücken im Hintergrundbild geschlossen, die infolge der unvollständigen Klassifizierung im vorangegangenen Schritt entstanden sind. Danach sind die Vordergrundpixel im Ergebnisbild $R(I)$ komplett von zusammenhängenden Hintergrundbereichen umschlossen und können als Zusammenhangskomponenten extrahiert werden (s. Abb. 2.7.1).

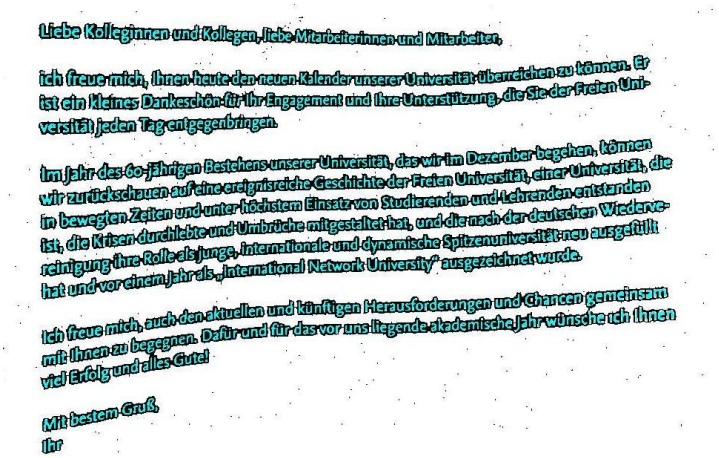


*Abb. 2.7.1: Vordergrundpixel (cyan) umgeben von den Hintergrundpixeln (schwarz) im Ergebnisbild $R(I)$.*

Die LCS-Binarisierungsmethode bietet eine Möglichkeit zur schnellen und zuverlässigen Zeichenextraktion, ohne dass rechenaufwändige Qualitätsverbesserungsmaßnahmen im Vorfeld notwendig sind. Genau aus diesem Grund wird sie in die Verarbeitungskette der hier präsentierten Arbeit eingegliedert. Die Approximation des Laplace-Filters durch DOG bringt wesentliche Laufzeitvorteile gegenüber vielen anderen Kantendetektionsmethoden [38]. Die Kosten für die rechenaufwändige Zusammenhangskomponenten-Analyse lassen sich an der Stelle ausklammern, da diese ohnehin im späteren Verlauf der Verarbeitung – bei der

Zeilenverfolgung (s. Abschnitt 6.5) und Bildentzerrung (s. Abschnitt 7.3) – unvermeidbar sind. Schließlich ist die Performance des Algorithmus im Umgang mit verschiedenen Textträgern bereits unter Beweis gestellt worden [37]. Problematisch ist hingegen die Abhängigkeit der LCS-Methode von mehreren im Voraus festzulegenden Parametern: der Gauß-Kernelgröße, der beiden te $\theta_{\varepsilon+}$, $\theta_{\varepsilon-}$ sowie der Größe des morphologischen Dilatation-Operators. Um ein optimales Binarisierungsergebnis zu erreichen, sollte eine skalierungsabhängige Parametrisierung des Algorithmus vor seiner Anwendung vorgenommen werden.

## 2.8 Fourier-Transformation

Neben der im Abschnitt 2.1 beschriebenen Möglichkeit der räumlichen Bilddarstellung ist die Darstellung der Bilder im Frequenzraum im Rahmen dieser Arbeit von großem Interesse. Dabei dienen periodische Muster als Basisbilder des Vektorraums, der zur Bildrepräsentation genutzt wird. Eine Transformation aus dem Ortsraum erfolgt i. d. R. mit Hilfe von globalen Bildoperationen. Das klassische Beispiel dafür ist die (diskrete) Fourier-Transformation, bei der ein N-fach abgetastetes Signal $I(n)$, wo $n \in [0, N-1]$, mit Hilfe von trigonometrischen Funktionen dargestellt wird [23]:

$$I(n) = \frac{1}{N}\sum_{k=0}^{N-1} \hat{I}(k) \cdot \omega_N^{nk},$$

Hier ist $\omega_N^{nk} = cos\left(\frac{2\pi}{N}nk\right) + i * sin\left(\frac{2\pi}{N}nk\right) = e^{\frac{2\pi i}{N}nk}$ und $\hat{I}$ bezeichnet die Fourier-Transformierte von $I$ . Die Transformation in den Frequenzraum findet gemäß

$$\hat{I}(n) = \sum_{k=0}^{N-1} I(k) \cdot \bar{\omega}_N^{nk}$$

statt, wo $\bar{\omega}_N^{nk}$ – das konjugiert Komplexe von $\omega_N^{nk}$ ist. Bildlich gesprochen wird ein Signal in seine von der Frequenz abhängigen Bestandteile, auch Frequenzspektrum genannt, zerlegt. Der Betrag von $\hat{I}(k)$ gibt an, wie stark eine bestimmte Frequenz $k/N$ im Frequenzspektrum vertreten ist. Im Frequenzraum lassen sich viele globale Eigenschaften eines Bildes einfacher analysieren als im Ortsraum: So kann die Orientierung der Textzeilen auf einer Dokumentaufnahme im einfachsten Fall direkt ablesen werden, indem der dominante Wellenvektor identifiziert wird [39].

Ein wichtiger Anwendungsbereich der Fourier-Transformation ist die effiziente Berechnung von Faltungsoperationen (s. Abschnitt 2.3). Die Grundlage dafür bietet das Faltungstheorem, welches besagt, dass eine Faltung im Ortsraum einer komponentenweise Multiplikation im Frequenzraum entspricht und umgekehrt:

$$\widehat{I * F} = \hat{I} \cdot \hat{F}$$

Dabei sind $I$ – Bild, $F$ – Filter, $*$ – Faltungsoperator, $\cdot$ – punktweise Multiplikation und $\hat{F}$ – die Fourier-Transformierte von $F$. Ist die Größe des Faltungskerns vergleichbar mit der Bildgröße $N$, dann sinkt die Berechnungszeit für die Faltung von $O(N^2)$ im Ortsraum auf $O(N)$. Ist die Fourier-Transformierte der gefilterten Aufnahme unbekannt, so kann diese mit Hilfe des *FFT*-Algorithmus (engl. *Fast Fourier Transform*) innerhalb von $O(N * log(N))$ Zeit bestimmt werden, sodass die Gesamtlaufzeit der Filterung im Bereich von $O(N * log(N))$ liegt [40].

## 2.9 Wavelet-Transformation

Wie bereits im Abschnitt 2.8 angesprochen, liefert das Frequenzspektrum eines Bildes wertvolle Informationen über seine globalen Eigenschaften, allerding fehlt dabei jeglicher Ortsbezug. Auf der anderen Seite lassen sich globale Merkmale nicht ohne weiteres anhand von einzelnen Pixelwerten feststellen, sodass oftmals eine Darstellung verwendet wird, die gleichzeitig eine bestimmte Auflösung im Orts- und im Frequenzraum bietet. Es ist insbesondere zu beachten, dass es prinzipiell nicht möglich ist, in beiden Räumen eine beliebig hohe Genauigkeit gleichzeitig zu erreichen[*].

Mit Hilfe der *Wavelet-Transformation* lassen sich die Lokalitätseigenschaften von extrahierten Merkmalen flexibel kontrollieren, wobei die Basisfunktionen eines solchen Vektorraums als Wavelets bezeichnet werden. Wavelets besitzen eine gewisse Lokalität im Frequenzspektrum und gleichzeitig einen kompakten Träger im Ortsbereich. Die Erzeugung von Wavelets erfolgt mittels Verschiebung und Skalierung einer Funktion, die oft als Mutter-Wavelet $\Psi$ bezeichnet wird [41]:

[*] Diese Aussage ist die signalanalytische Entsprechung der Heisenbergschen Unschärferelation und kann analog dazu als $\Delta\omega\Delta x \leq const$ formuliert werden, wo $\Delta\omega, \Delta x$ - die Auflösung im Frequenz bzw. Ortsraum sind [41].

$$\Psi_{j,k}(x) = 2^{j/2}\Psi\left(2^j x - k\right)$$

Hier sind $k, j \in Z$ – Verschiebungs- bzw. Skalierungsfaktoren. Von der als Mutter-Wavelet agierenden Funktion werden häufig bestimmte Eigenschaften wie die absolute und quadratische Integrierbarkeit und die Kompaktheit des Trägers [85] verlangt. Darüber hinaus ist die Glattheit der Funktion sowie Orthogonalität zu ganzzahligen Selbstverschiebungen aus Effizienzgründen und Gründen der numerischen Stabilität für Wavelets von Vorteil. Eine Funktion $f(x) \in L^2(R)$ kann im Wavelet-Raum wie folgt dargestellt werden

$$f(x) = \sum_{j,k \in Z} \alpha_j(k)\Psi_{j,k}(x),$$

wobei $\{\Psi_{j,k} | k, j \in Z\}$ – die Wavelet-Basis und $\alpha_j(k)$ - die Koeffizienten der Wavelet-Transformierten

$$\alpha_j(k) = \int_R f(x)\Psi_{j,k}(x)\, dx$$

sind. Da eine Auswertung des Integrals rechenintensiv sein kann [42], wird in der Praxis häufig schnelle Wavelet-Transformation eingesetzt, die auf der Theorie der Multiskalenanalyse basiert. Dabei wird der $L^2(R)$-Funktionsraum als eine Folge geschachtelter Unterräume mit gewissen Eigenschaften dargestellt [42]:

$$\{0\} \ldots \subset V_0 \subset V_1 \subset \cdots \subset V_n \subset V_{n+1} \subset \cdots \subset L^2(R)$$

Jeder der Unterräume $V_n$ enthält eine Approximation des Signals, wobei das Genauigkeitspotenzial mit höherem $n$ ansteigt. Die Basis der Unterräume wird analog zur Wavelet-Basis durch eine Skalierung und Verschiebung einer bestimmten Funktion, s. g. Vater-Wavelets $\varphi(x) = \varphi_{0,0}(x)$, produziert. Wird nun das Mutter-Wavelet $\Psi$ so gewählt, dass der von ihm aufgespannte Unterraum $W_n = span\{\Psi_{n,k}(x) | k \in Z\}$ das orthogonale Komplement von $V_n = span\{\varphi_{n,k}(x) | k \in Z\}$ in $V_{n+1} = span\{\varphi_{n+1,k}(x) | k \in Z\}$ ist:

$$V_{n+1} = V_n \oplus W_n$$

dann kann eine (n+1)-fach abgetastete Funktion $f^{n+1}(x) \in V_{n+1}$ wie folgt dargestellt werden [42]:

$$f^{n+1}(x) = \sum_k \beta_n(k)\varphi_{n,k}(x) + \sum_k \alpha_n(k)\Psi_{n,k}(x)$$

Die Koeffizienten $\beta_n$ aus dem ersten Teil des Ausdrucks werden Approximationskoeffizienten (engl. *approximation coefficients*) genannt, da die Linearkombination $\sum_k \beta_n(k)\varphi_{n,k}(x) \in V_n$ eine tiefpassgefilterte Version von $f^{n+1}$ ergibt. Die

Koeffizienten $\alpha_n$ aus dem zweiten Teil des Ausdrucks werden hingegen als Wavelet- oder auch Detailkoeffizienten (engl. *detail coefficients*) bezeichnet, da sie die im ersten Teil fehlende Detailinformation beschreiben, die das Ergebnis einer Bandpassfilterung des ursprünglichen Signals sind.

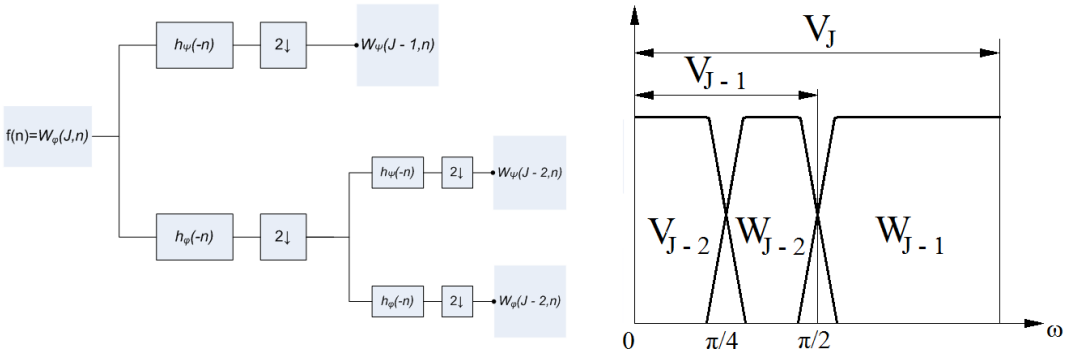


*Abb. 2.9.1: (links) Zwei Stufen der FWT aus [41]. (rechts) rekursive Frequenzbandzerlegung während FWT aus [41].*

Die Unterteilung des Signals in Frequenzbände wird durch die rekursive Zerlegung des Tiefpasskanals fortgesetzt, bis die notwendige Skalierungsstufe erreicht wird. Das Ergebnis einer FWT kann in Form eines Binärbaums mit den Wavelet-Koeffizienten verschiedener Skalierungsstufen als Blättern und den Approximationskoeffizienten als inneren Knoten dargestellt werden (s. Abb. 2.9.1 links). Die Approximation der letzten Stufe ist dabei ebenfalls ein Blatt und damit ein Bestandteil der Zerlegung dieser Stufe [42]:

$$V_{n+1} = V_{n-k} \oplus W_{n-k} \oplus \ldots \oplus W_{n-1} \oplus W_n$$

Im Laufe der Berechnung sinkt die Anzahl der Pixel und damit auch die Berechnungszeit von Stufe zu Stufe exponentiell (s. Abb. 2.9.1 links), sodass die Gesamtlaufzeit einer FWT im Bereich von $O(n)$ in der Anzahl der Abtastwerte liegt [41].

Die *Wavelet-Packet-Transformation* ist eine Erweiterung der FWT, bei der nicht nur die jeweiligen Approximationsanteile $W_i$ immer weiter rekursiv zerlegt werden, sondern auch die Bandpasskanäle, sodass am Ende ein vollständiger binärer Baum mit den Koeffizienten entsteht. Die Knoten des Baums in der $l$-ten Ebene stellt eine *äquidistante* Zerlegung (anstelle der logarithmischen der FWT) des Frequenzspektrums in $2^l$ Intervalle. Die Laufzeit der Wavelet-Packet-

Transformation steigt jedoch gegenüber von FWT auf $O(n * logn)$ in der Anzahl der Messwerte [43].

Eine 2D-Wavelet-Transformation wird in der Praxis häufig auf eine Folge von 1D-Transformationen reduziert. Es gibt vier Möglichkeiten die Tief- und Bandpasskanäle aus den beiden Dimensionen zu kombinieren:

- horizontale Merkmale $W_\Psi^H$ werden durch die Tiefpassfilterung der Spalten und die Bandpassfilterung der Reihen extrahiert
- vertikale Merkmale $W_\Psi^V$ werden durch die Bandpassfilterung der Spalten und die Tiefpassfilterung der Reihen extrahiert
- werden sowohl Reihen als auch Spalten mit dem Bandpassfilter gefaltet, so werden diagonal gerichtete Merkmale $W_\Psi^D$ hervorgehoben
- durch die Anwendung des Tiefpassfilters auf die Reihen und Spalten ergeben sich die Approximationskoeffizienten $W_\varphi$

Aufgrund ihrer Flexibilität und Effizienz spielt die Wavelet-Transformation und speziell die FWT im Rahmen dieser Arbeit eine wichtige Rolle. Insbesondere in der Anfangsphase der Verarbeitung kann die im Wesentlichen parameterlose Wavelet-basierte Frequenzbandzerlegung einen ersten Überblick über die Bildstruktur geben.

# 3. Kapitel

# Anforderungsanalyse

Die praktische Ausrichtung dieser Promotionsarbeit erfordert ein vertieftes Verständnis der alltäglichen Probleme sehbehinderter Menschen sowie Kenntnisse über die Funktionsweise, Vor- und Nachteile bereits verfügbarer Lösungen, sodass bereits in dem Planungsstadium des Projekts Kontakt zu Betroffenen gesucht wurde. In diesem Kapitel wird beschrieben, wie die Anforderungserhebung an das zu entwickelnde System ablief und wie auf der Grundlage der erhobenen Daten das zum Schluss präsentierte Pflichtenheft erstellt wurde.

## 3.1 InformA*-Projekt

Dank dem InformA-Projekt†, das in den Jahren 2008 bis 2011 durchgeführt wurde, bot sich für mich frühzeitig eine Gelegenheit an, die Zielgruppe des zu entwickelnden Vorlesegerätes besser kennenzulernen. Das Ziel des InformA-Projekts [22] war es, ein neuartiges Gerät zu entwickeln, das den blinden und sehbehinderten Menschen Interaktionen mit dem Internet erleichtern würde. Zu meinen Aufgaben im Rahmen des InformA-Projekts gehörte in erster Linie die Implementierung von der Vorlesefunktionalität des mit einer Videokamera ausgestatteten Geräts (s. Abb. 3.1.1). Um die Ergebnisse der Analyse zentral auswerten zu können, wurden alle Dokumentaufnahmen zu einem dedizierten OCR-Server übertragen, wo ihre Verarbeitung stattfand. Dabei konnte ich Erfahrungen im Umgang mit existierenden OCR-Lösungen sammeln und Ursachen für eine fehlerhafte

---

* InformA steht für Information Appliance (engl. *Informationsgerät*).

† Durchgeführt mit der freundlichen Unterstützung des Bundesministeriums für Wirtschaft und Technologie (EXIST), des Blinden- und Sehbehindertenvereins in Berlin (ABSV), Deutsche Telekom Laboratories und IBM Deutschland.

OCR-Verarbeitung identifizieren. Neben der Vorlesefunktion, ermöglichte das Gerät einen Zugriff auf Online-Zeitungen, Online-Radiosender, Audiobücher und war zudem in der Lage Emails zu empfangen und zu beantworten. Nicht zuletzt bestand für die Tester eine Möglichkeit bei Fragen und Anregungen über das Gerät Kontakt mit den Entwicklern aufzunehmen, wobei diese auch regelmäßig in Anspruch genommen wurde.

Im Jahr 2009/2010 wurde ein Feldversuch mit vierzig Betroffenen im Alter zwischen 18 und 88 Jahren durchgeführt. Dank der Callcenter-Funktionalität wurde mir ein direkter Zugang zu den Betroffenen ermöglicht, sodass ich eine Anforderungserhebung auf der Grundlage der Befragungen vornehmen und den Prototypen unter realistischen Bedingungen testen konnte. Eine Auswertung der Umfrageergebnisse ist im Anhang A zu finden. Auf zahlreichen Messen, auf denen das InformA-Gerät präsentiert wurde (bspw. SightCity 2010, ABSV Hilfsmittelausstellungen 2009/2010, Venture.Med 2011), konnte ich darüber hinaus neueste Entwicklungen auf dem Markt der Blindenhilfsmittel aus erster Hand beobachten. Ein Überblick über die existierenden Lösungsansätze wird im weiteren Verlauf dieses Kapitels gegeben.

Im Laufe der Studie wurden mit dem ausdrücklichen Einverständnis der Tester über 1000 Aufnahmen von realen Dokumenten auf dem Server gespeichert. Die Datenbank wurde u. a. dazu verwendet, verschiedene kommerzielle und nichtkommerzielle OCR-Programme miteinander zu vergleichen, um einen Überblick über den Stand der Technik zu bekommen und mögliche Problemfälle zu identifizieren.

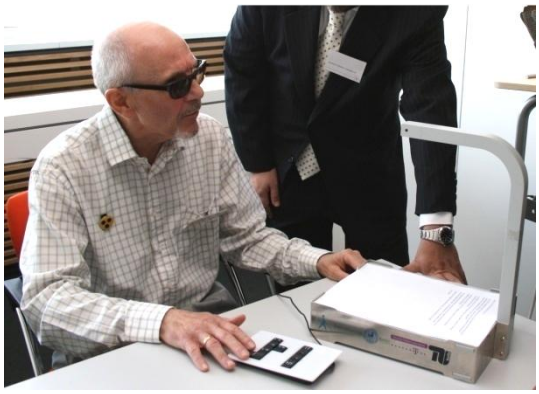

*Abb. 3.1.1: Ein Tester mit dem InformA-Gerät*

Im Anschluss an den Testversuch wurde eine Befragung der Teilnehmer durchgeführt. Die Tester wurden aufgefordert ihre Bewertungen, Ideen und Anregungen bzgl. des InformA-Geräts im Allgemeinen und speziell zum Thema Vorlesegeräte mitzuteilen, wobei interessante Einsichten in ihre Bedürfnisse und Erwartungen gewonnen wurden. So betonten die Befragten, dass ihnen äußere Unauffälligkeit des Geräts wichtig sei. Ein anderer Betroffener wies uns darauf hin, dass akustisches Feedback bei einer mobilen Anwendung nicht unproblematisch für blinde Menschen ist, da es den für sie wichtigsten Informationskanal zusätzlich beansprucht. Eine weitere interessante Erkenntnis aus den direkten Gesprächen war die Feststellung, dass insbesondere jüngere von den Betroffenen über die bemerkungswerte Fähigkeit verfügen, sprachliche Informationen in weitaus höheren Raten (bis zu 150% schneller) aufzunehmen, als Menschen ohne die Behinderung.

Eine Analyse der an den Server übermittelten Dokumente zeigte, dass viele der Tester keine Vorstellung über die Leistungsgrenzen des Geräts hatten, wobei manche Dokumentaufnahmen in völliger Dunkelheit gemacht wurden. Mehrere Versuchsteilnehmer beanstandeten zudem, dass keine handschriftliche Notizen vorgelesen werden konnten. Neben gedruckten Dokumenten wurden gelegentlich auch Produktverpackungen, Dosen und Flaschen unter die Kamera gehalten, wobei Falten bzw. Wölbungen der Oberfläche ein fehlerfreies Vorlesen unmöglich machten. Auch die z. T. lange Verarbeitungsdauer und die Qualität der Ausgabe, insbesondere was das Vorlesen von Tabellen und Formularen angeht, wurden von einigen Testern bemängelt (s. Anhang A).

Die Versuchsteilnehmer wurden auch nach den Hilfsmitteln gefragt, die sie in ihrem Alltagsleben verwenden. Über 50% der Befragten gaben an, ein scannerbasiertes Vorlesegerät zu besitzen, während lediglich 5% zudem ein lesefähiges Smartphone benutzten. Insbesondere ältere Versuchsteilnehmer begründeten das mit den Bedienungsschwierigkeiten, die sie bei der Handhabung der neuartigen Geräte hatten. Diese Aussagen decken sich mit den Erkenntnissen aus der Befragung zur Benutzungsfreundlichkeit des InformA-Geräts: Trotz des einfachen baumartigen Aufbaus vom Sprachmenü sowie der ausführlichen Einführungshilfen wurde die Bedienung auf der Skala von „5“ bis „1“ mit der Durchschnittsnote „2,25“ bewertet (s. Anhang A).

Beim Formulieren der Verbesserungsvorschläge orientierten sich die Probanden

an der Funktionalität der ihnen bekannten stationären Geräte. Es wurde bspw. mehrmals der Wunsch nach einer Möglichkeit zum Abspeichern und Wiedergabe der erkannten Texte geäußert, die in vielen scannerbasierten Produkten implementiert ist. Die Möglichkeit einer automatischen Echtzeit-Textdetektion war vielen Testern hingegen unbekannt.

## 3.2 Stationäre Vorlesegeräte

Bei vielen aktuell auf dem Markt verfügbaren Vorlesegeräten findet die Dokumentaufnahme mit Hilfe eines Flachbettscanner statt, der entweder an einen Rechner angeschlossenen ist oder einen Rechner eingebaut hat. In jedem Fall ist die Mobilität dieser Geräte durch die Größe des Scanners stark eingeschränkt [44][45]. Darüber hinaus muss ein Dokument physisch zugänglich sein und auf den Scanner passen, um aufgenommen zu werden. Dadurch ist das Vorlesen von Schildern und Aushängen einerseits und Beschriftungen auf nicht-flachen Textträgern wie Flaschen oder Dosen andererseits problematisch bzw. nicht möglich. Auch in Hinblick auf die Aufnahmegeschwindigkeit haben scannerbasierte Vorlesegeräte einen Nachteil gegenüber kamerabasierten Lösungen, da die Abtasteinheit mechanisch über das ganze Dokument bewegt werden muss. Die geringere Aufnahmegeschwindigkeit muss jedoch nicht zwangsläufig in einer größeren Gesamtverarbeitungszeit resultieren, da die Aufnahme eines Dokuments in einem Flachbettscanner unter nahezu idealen Bedingungen stattfindet, wodurch die Notwendigkeit einer zeitintensiven qualitätsverbessernden Bildverarbeitung entfällt.

## 3.3 Mobile Vorlesegeräte

Die Befragung der InformA-Teilnehmer hat gezeigt, dass die meisten mobilen Vorlesesysteme momentan noch (Stand 2011) Smartphone-basiert sind, was nicht zuletzt mit der Leistungssteigerung der Geräte in den letzten Jahren zusammenhängt. Problematisch ist hingegen die Geschlossenheit der Systeme, die Anpassungen von Hardware und auch Software erheblich erschwert oder unmöglich macht. Ein Problem stellt bspw. die Bedienung der Telefone dar, die meistens über einen Touchscreen stattfindet. Für diese Problematik existieren mittlerweile verschiedene Lösungsansätze: von Sprachsteuerung [18] bis

*Abb. 3.3.1: Bildschirmaufkleber aus [14]*

zu speziell angefertigten Bildschirmaufklebern (s. Abb. 3.3.1). Auch die vom Hersteller angebotene Programmierschnittstelle kann u. U. ein Hindernis für den Einsatz eines Smartphones als geschlossene Hardware-Plattform für die Entwicklung eines Vorlesegeräts darstellen. Erfüllt die vom Werk aus eingebaute Kamera bestimmte Anforderungen nicht oder sind bestimmte Kameraeinstellungen nicht ansprechbar, so wird das ganze System unbrauchbar, da ein Kameratausch häufig mit einem großen Aufwand verbunden ist.

Gegenwärtig reicht die Erkennungsqualität von mobilen Vorlesegeräten nicht immer an die der stationären Geräte heran. Die Gründe dafür liegen in der häufig mangelnden Qualität von Kameraaufnahmen und den damit verbundenen zusätzlichen Bildoptimierungsschritten. Die Tabelle 3.3.1 gibt einen Überblick über die spezifischen Probleme, die mit der Mobilität des Systems zusammenhängen.

*Tabelle 3.3.1: Vergleich: Kameraaufnahmen-Scanneraufnahmen*

| **KAMERAAUFNAHMEN** | **SCANNERAUFNAHMEN** |
|---|---|
| Dokument ist in einer komplexen natürlichen Umgebung eingebettet und muss erst lokalisiert werden | Die Position des Dokuments innerhalb des Bildes ist bekannt |
| Inhomogene, nicht optimale Beleuchtung, eventuell Sonnenlicht, Belichtungszeit nicht bekannt | Optimale Beleuchtung, Belichtungszeit entsprechend angepasst, keine Schatten |
| Abstand zum Dokument ist unbekannt | Abstand zum Dokument ist fest |
| Starke nicht-lineare und perspektivische Verzerrungen | Das Dokument liegt flach auf dem Scanner (abgesehen von der Bücher-Problematik) |
| Bewegungsunschärfe | Keine Bewegungsunschärfe |

Die Ausrichtungsproblematik und geeignete Kadrierung von Textobjekten mit Hilfe einer eingebauten Smartphone-Kamera stellt häufig eine besondere Herausforderung für die Betroffenen dar. In mehreren Smartphone-basierten Vorlesesystemen [7][9] wird daher folgende manuelle Methode zur Lokalisierung von Textabschnitten und Vermeidung von perspektivischen Verzerrungen angewandt:

- Das Dokument wird auf einer Oberfläche platziert
- Das Smartphone wird flach auf das Dokument gelegt
- Das Smartphone wird langsam angehoben, sodass die Sensorebene möglichst parallel zur Dokumentebene bleibt

Diese Manipulationen können trotz der softwareseitigen Unterstützung manchen Blinden und sehbehinderten Menschen große Schwierigkeiten bereiten, insbesondere wenn ihre Bewegungssicherheit durch bspw. Zitterkrankheiten beeinträchtigt ist. Des Weiteren ist die beschriebene Vorgehensweise nur für Dokumente geeignet, die physisch zugänglich sind. Dabei sind gerade im Außenbereich viele Textträger wie Aushängeschilder, Straßenzeichen und Poster nicht ohne Weiteres zu erreichen. Es ist daher eine Kameraausrichtungsmethode notwendig, die keinen physischen Kontakt mit dem Dokument erfordert. Für den Fall, dass die optimale Orientierung auf den zu verarbeitenden Textbereich nicht gewährleistet werden konnte, muss die evtl. vorhandene Restverzerrung softwareseitig korrigiert werden.

Digitale Verarbeitung hochaufgelöster Bilder kann sehr rechenaufwändig sein. Bereits einfachste Punktoperationen (s. Abschnitt 2.3) auf einer mehrere Megapixel großen Aufnahme, bei denen auf jedes Pixel genau einmal zugegriffen wird, erfordert Millionen von Rechenschritten. Auf der anderen Seite zeigte der Testlauf im Rahmen des InformA-Projekts deutlich, dass die Anwender häufig nicht bereit sind lange auf das Ergebnis der Verarbeitung zu warten. Trotz der durchschnittlichen Verarbeitungszeit von 30 s, wurde die Wartezeit von einigen Testern als zu hoch bewertet (s. Anhang A). Die Laufzeitanforderungen lassen sich unterdessen nur schwer mit den Mobilitätsanforderungen vereinbaren, sodass an vielen Stellen Kompromisslösungen notwendig sind. Die 30-Sekunden-Grenze dient dabei als Richtwert für die maximale Reaktionszeit des Systems.

## 3.4 Semi-mobile Vorlesegeräte

Als semi-mobile Systeme werden im Rahmen dieser Arbeit tragbare Geräte bezeichnet, die sich durch ihre vergleichsweise kompakten Ausmaße, geringeres Gewicht und eine eingebaute Batterie von stationären Geräten unterscheiden. Die Kompaktheit wird unterdessen häufig dadurch erreicht, dass herausragende Teile klappbar gemacht werden, sodass diese Geräte vor ihrer Verwendung durch Blinde aufgebaut werden müssen [10][46]. Im Gegensatz zu stationären Geräten werden semi-stationäre häufig mit Kameras ausgerüstet, was sie empfindlich gegenüber komplizierten Lichtverhältnissen macht. Dank des stabilen Gerüsts lässt sich jedoch i. d. R. eine höhere Qualität der Aufnahmen erreichen als das bei den mobilen Systemen der Fall ist. Somit stellen semi-mobile Systeme eine Kompromisslösung dar, die Vor- und Nachteile der mobilen und stationären Systeme in sich vereint.

## 3.5 OCR-Systeme – eine Vergleichsstudie

*Tabelle 3.5.1: OCR-Systeme, Testauswertung*

| | **OmniPage 16/SDK** | | **FineReader Engine 9** | | **Ocropus 0.4** | | **Ocrad 0.19** | |
|---|---|---|---|---|---|---|---|---|
| **Aufnahmen aus der Datenbank (wenig Verzerrung) Erkennungsrate/ Gesamtlaufzeit** | 96% | 958s | 96% | 846s | 88% | 1677s | 82% | 750s |
| **Aufnahmen mit dem Brillenprototyp (stark verzerrt) Erkennungsrate/ Gesamtlaufzeit** | 85% | 351s | 81% | 305s | 73% | 598s | 65% | 283s |

Bevor Miniaturkameras allgegenwärtig wurden, waren kommerzielle OCR-Programme in erster Linie für Scanneraufnahmen ausgelegt. Mittlerweile bieten führende Hersteller von OCR-Systemen [47][48] auch Lösungen für mobile An-

wendungen an, was nicht zuletzt mit der rasanten Verbreitung von Smartphones zusammenhängt. Die Entwicklung einer leistungsfähigen OCR-Engine ist aufwändig, sodass viele der auf dem Markt verfügbaren Vorlesegeräte den Herstellerangaben zufolge intern auf kommerzielle OCR-Lösungen von Drittanbietern zurückgreifen. Es existiert weltweit eine Handvoll von proprietären OCR-Systemen, die über eine API angesprochen und gesteuert werden können. Darüber hinaus sind einige freie OCR-Module verfügbar, die unter GPL-/BSD-/Apache-Lizenz stehen. Vier OCR-Bibliotheken wurden ausgewählt und evaluiert, um den Stand der Technik zu ermitteln und zu entscheiden, welche von ihnen am besten zu den Anforderungen des Projekts passen:

1. *OmniPage* von *Nuance Communications [47]* (proprietär)
2. *FineReader* von *ABBYY [48]* (proprietär)
3. *OCRopus [49]* (GPL)
4. *Ocrad [50]* (Apache-Lizenz)

Nuance Communications mit der Produktlinie OmniPage und ABBYY mit FineReader gehören zu den führenden Herstellern von OCR-Lösungen [51]. Bei der Auswertung wurden die 16. Version von OmniPage SDK sowie ABBY Mobile OCR Engine 4.0 verwendet, die von den Herstellern zu Evaluierungszwecken zur Verfügung gestellt wurden. Als mögliche Open-Source-Kandidaten wurden OpenOCR [52], GOCR [53], Ocrad, OCRFeeder [54] (mit Tesseract [55] als OCR-Engine), OCRopus (mit Tesseract als OCR-Engine) und Puma.NET [56] in Betracht gezogen. Die Auswahl fiel schließlich auf die beiden genannten Lösungen OCRopus und Ocrad, da sie einerseits eine C++ Programmierschnittstelle zur Verfügung stellen und andererseits über Funktionen für eine erweiterte Layoutanalyse verfügen. Es ist anzumerken, dass bei der Evaluierung von OCRopus Tesseract als Zeichenerkennungsmodul eingestellt wurde, da dieser weit verbreitet ist und in vielen anderen modular aufgebauten OCR-Produkten intern eingesetzt wird [55].

Im Mittelpunkt der Evaluierung stand die Leistung der OCR-Module bei der Texterkennung von Kameraaufnahmen. Für die Tests wurden zum einen Dokumentaufnahmen aus der InformA-Datenbank und zum anderen einige Aufnahmen, die speziell mit Hilfe des ersten Prototypen des Vorlesegeräts angefertigt wurden, verwendet. Für alle speziell angefertigten Aufnahmen wurde eine Ground-Truth-Ausgabe manuell editiert, während für die tausend Dokumente aus

der Datenbank nur eine Auswertung unter Verwendung von Wörterbüchern erfolgte. Das Ergebnis des Testlaufs ist in der Tabelle 3.5.1 zusammengefasst aufgeführt.

Beim Abgleichen der Ausgabetexte stellt die Problematik der mehrdeutigen Vorlesereihenfolge eine Herausforderung dar, da die für eine Dokument-Archivierung konzepierten kommerziellen Lösungen in erster Linie das Layout des Dokuments möglichst wahrheitsgetreu zu wiedergeben versuchen, wobei die lineare Ausgabereihenfolge z. T. willkürlich festgelegt wird. Um trotz Unterschiede in der Anordnung der Textblöcke einen Abgleich durchführen zu können, wurde der Smith-Waterman-Algorithmus [57] verwendet, um das optimale, lokale Alignment zweier Sequenzen von Textregionen zu bestimmen. Insgesamt zeigten die kommerziellen Systeme eine bessere Leistung was die Verarbeitungsgeschwindigkeit und die Genauigkeit der Ausgabe betrifft. Die Verarbeitungsdauer der mobilen Version von FineReader war generell etwas kürzer als die von OmniPage, dafür lieferte OmniPage höhere Erkennungsraten im Falle von stark verzerrten Dokumentaufnahmen. Die durchschnittliche Verarbeitungsdauer nach dem Abzug der Vorverarbeitungszeit betrug im Falle von OmniPage 9.4 s und im Falle von FineReader 8.6 s für je 1000 Zeichen.

Nach der Auswertung der Ergebnisse wurde entschieden, die beiden kommerziellen OCR-Module weiteren Tests zu unterziehen, wobei sie in zwei unterschiedlichen, parallel entwickelten Systemen zur Verwendung kamen. Die jeweiligen Schwächen der Lösungen wurden dabei durch die Konfiguration der Hardware ausgeglichen: Während OmniPage als primäres Zeichenerkennungsmodul in der mobilen Version des Geräts eingesetzt wird, wurde FineReader im Rahmen des InformA-Projekts in ein etwas leistungsschwächeres semi-stationäres System integriert.

Eine Analyse der OCR-Ergebnisse zeigte, dass die Erkennungsrate bei einer OCR-Verarbeitung von vielen Faktoren beeinflusst werden kann. Dazu gehören einerseits die in der Tabelle 3.3.1 aufgeführten äußeren Umstände und andererseits die Beschaffenheit der zu verarbeitenden Textabschnitte. Die Folgen der ungünstigen Aufnahmebedingungen lassen sich in bestimmten Fällen mittels qualitätsverbessernder Vorverarbeitungsmaßnahmen abgeschwächt werden. Alle vier evaluierten Systeme präsentieren sehr ähnliche Abfolgen von vordefinierten Verarbeitungsschritten, die mit teilweise optional zuschaltbaren Funktionen die ange-

sprochenen Fehlerquellen zu beseitigen versuchen:

1. Nach dem Einlesen einer Aufnahme werden qualitätsverbessernde Vorverarbeitungsschritte wie Kontrasterhöhung (optional) oder Rauschunterdrückung (optional) durchgeführt, die eine einfachere Binarisierung (s. Abschnitt 2.7) der Textbereiche zum Ziel haben.
2. Textblöcke werden lokalisiert, Textlinien und einzelne Symbole werden in Form von Zusammenhangskomponenten extrahiert. Dieser Schritt wird oft auch als geometrische Layoutanalyse bezeichnet [101].
3. Eine Entzerrung des Dokuments wird vorgenommen (optional).
4. Nun findet die eigentliche Zeichenerkennung statt, wobei für jede extrahierte Zusammenhangskomponente mehrere Buchstabenhypothesen generiert werden.
5. Die Wörter werden unter Zuhilfenahme eines Sprachmodells aus den erkannten Zeichen zusammengesetzt. Manche OCR-Systeme revidieren anschließend die Zeichenhypothesen in einem zweiten Durchlauf unter Einbeziehung der konstruierten Worthypothesen (optional).
6. Schließlich wird die Ausgabe generiert, typischerweise ein Text-, XML- oder PDF-Dokument, die die erkannten Textblöcke und ihre jeweiligen Eigenschaften enthält.

Alle evaluierten OCR-Systeme verfügen über ein Layouterkennungsmodul, d. h. sie sind in der Lage einzelne Textblöcke und -spalten zu identifizieren. OmniPage und FineReader können darüber hinaus Bilder, Diagramme, Separatoren, Bildunterschriften lokalisieren. Die Layouterkennung von OCRopus fiel unterdessen dadurch positiv auf, dass die Vorlesereihenfolge für die identifizierten Textblöcke systematisch [101] festgelegt wurde. Die Anordnung der Textblöcke, die von OmniPage und FineReader produziert wurde, war hingegen z. T. willkürlich.

Als ein besonderes Merkmal präsentieren die neuesten Versionen der beiden kommerziellen OCR-Systeme einen speziellen Modus für Digitalkamerabilder mit einer erweiterten Funktionalität für die Bildoptimierung, einschließlich einer Schärfungsfunktion, einer (Pseudo-) 3D-Korrektur der Verzerrungseffekte sowie Rauschunterdruckung. Deren Leistung war jedoch nicht immer ausreichend, um die Erkennungsraten der Scanneraufnahmen zu erzielen wie aus der Tabelle 3.5.1 entnommen werden kann. Der Modus bietet sich jedoch als Referenzsystem für

die Evaluierung der im Rahmen dieser Arbeit entwickelten Algorithmen an.

Textspezifische Eigenschaften wie Schriftgröße, -art, -stil haben neben der Aufnahmequalität ebenfalls einen Einfluss auf die Erkennungsraten. Die evaluierten Lösungen definieren die empfohlene Minimalhöhe der Zeichen als 10-11 Punkte bei 200-300 dpi Auflösung (OmniPage), 10 Punkte bei 300 dpi, 10 Punkte bei 300 dpi (Tesseract) bzw. „mindestens 20 px“ (Ocrad). Um eine solche Punktdichte in einer DIN A4 (8,2 × 11,6 Zoll) Dokumentaufnahme zu erreichen, muss diese eine Auflösung von mindestens 5 Mpx haben, wobei ein 4:3-Bildformat vorausgesetzt wird.

Bei allen vier getesteten Anwendungen handelt es sich um omni-font Lösungen, allerdings bietet keine von ihnen den eigenen Angaben nach eine Unterstützung für handschriftliche Dokumente an. Ein großer Vorteil von Open-Source-Systemen besteht darin, dass der innere Aufbau und Funktionsweise sämtlicher Verarbeitungsprozeduren bekannt ist. Insbesondere ist ein Zugriff auf Funktionen der niedrigen Ebene wie bspw. Erkennung von einzelnen Zusammenhangkomponenten möglich. Die OCR-Bibliothek Tesseract enthält darüber hinaus Funktionen zum Trainieren des Zeichenklassifikators anhand von eigenen Datensätzen.

## 3.6 Steuerungskonzept und Anwendungsfälle

Aufgrund der angestrebten Mobilität des Geräts muss sein möglicher Einsatz auf offener Straße und im Freien berücksichtigt werden. Dies hat weitreichende Folgen für die Anforderungen an die Hardware und Software. Insbesondere ist die Wetterfestigkeit und Robustheit der Systemkomponenten gegenüber Erschütterungen zu bedenken. Richtungsgebend sind dabei die in der Industrie geltende Standards darunter der kommerzielle Temperaturbereich (0°C bis +70°C) für Halbleiterbauelemente [58].

Die zu erwartende Vielfalt an lesbaren Textobjekten beim Einsatz im Außenbereich ist groß, wobei insbesondere solche Textträger wie Aushängeschilder und Straßenzeichen eine wichtige Rolle spielen. Eine weitere große Herausforderung stellen komplizierte Lichtverhältnisse dar, die beim Einsatz unter direktem Sonnenlicht zu erwarten sind. Dazu zählen vor allem Überbelichtung und scharfe Schattengrenzen. Auch eine unauffällige Erscheinung und Verhaltensweise des Geräts ist wegen eines möglichen Einsatzes auf offener Straße von Bedeutung.

Das Hauptszenario für den Einsatz des Geräts durch den Anwender zum Vorlesen von Texten sieht folgenderweise aus (s. auch Abb. 3.6.1):

- Das Dokument wird ggf. vor die Kamera gehalten
- Sofort nach dem Hochfahren fängt das System an, die Umgebung nach lesbaren Texten zu durchsuchen und informiert ggf. den Anwender per Sprachausgabe. Der Benutzer hat an der Stelle eine Möglichkeit die Aufnahme und Verarbeitung zu erzwingen, auch wenn die automatische Textdetektion ergebnislos verlief.
- Während der Benutzer entscheidet, ob er den gefundenen Text vorgelesen bekommen möchte, verfolgt das System die gefundenen Textstellen, benachrichtigt den Benutzer falls der Text nicht mehr komplett im Bild ist und gibt ihm ggf. Anweisungen wie er die Ausrichtung korrigieren kann.
- Während der Wiedergabe besteht eine Navigationsmöglichkeit innerhalb der erkannten Textabschnitte.
- Nach der Wiedergabe hat der Benutzer eine Möglichkeit den vorgelesenen Text zu speichern oder gleich in den Initialzustand zurück zu kehren.

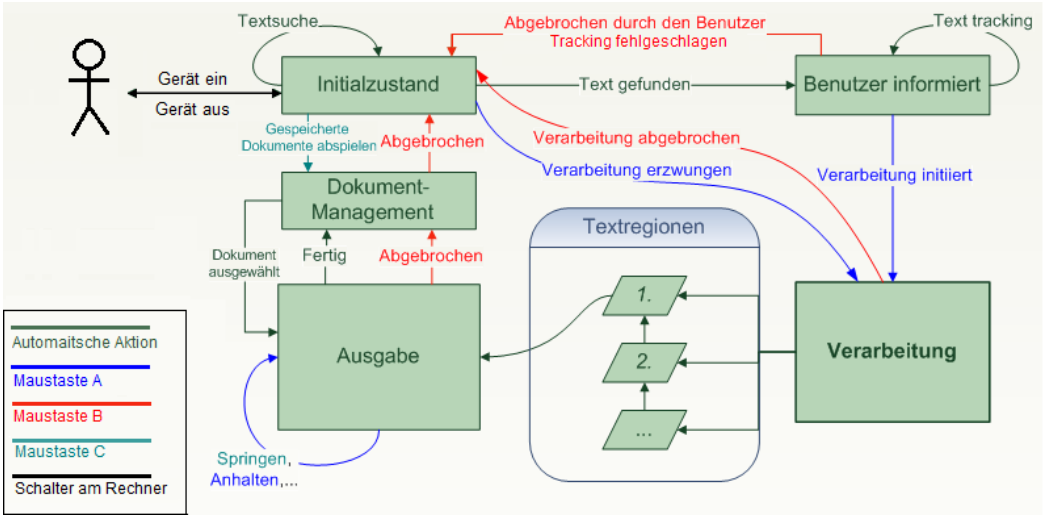


*Abb. 3.6.1:Anwendungsfall, Hauptszenario*

Alternativ kann der Anwender aus dem Initialzustand heraus zum Dokument-Management-Dialog wechseln und sich die gespeicherten Aufnahmen abspielen lassen. Im Fehlerfall wird der Benutzer per Sprachausgabe benachrichtigt und mit notwendigen Korrekturvorschlägen versorgt, woraufhin das Gerät wieder in den

Initialzustand wechselt.

In dem Initialzustand des Systems wird eine Echtzeitdetektion von lesbaren Textregionen in der Umgebung durchgeführt. Angesichts der zu erwartenden Leistungseinschränkungen des mobilen Rechners werden an der Stelle s. g. weiche Echtzeitanforderungen definiert, d. h. die Verarbeitungszeit je Zyklus kann u. U. variieren und ein Überscheiten der spezifizierten Zeitgrenzen wird nicht als Fehler gewertet [59]. Die Zeitvorgaben sind in dem Fall als Richtwerte zu betrachten, wobei hier der Wert von 1/12 s festgelegt wird, der von einigen Forschern als menschliche „Bildrate“ angesehen wird [60]. Zu beachten ist, dass selbst dieser für viele CV- und CG-Anwendungen eine eher locker gewählte Zeitvorgabe vollkommen ausreichend sein sollte, da die Ausrichtung der Kamera einschließlich ihrer Stabilisierung Sekunden dauern kann. Daher wird der Toleranzbereich für die Reaktionszeit nach der Positionierung des Dokuments vor die Kamera auf 1 s festgelegt.

Wie das InformA-Projekt deutlich zeigte, spielt eine intuitive und einfache Bedienung des Geräts für die Zielgruppe des Projekts eine überaus wichtige Rolle. *Sprachsteuerung* wurde von einigen der befragten Tester als die bevorzugte Kommunikationsart für das InformA-Gerät angegeben (Anhang A), weshalb diese Möglichkeit als erste in Betracht gezogen wurde. Sprachbasierte Steuerungssysteme gibt es schon seit geraumer Zeit und sie werden u. a. auch zur Bedienung von etwaigen Blindenhilfsmitteln eingesetzt [20][61]. Dafür wird typischerweise eine spezielle Grammatik mit einer begrenzten Auswahl an Wörtern definiert, die als Befehle oder Befehlsparameter interpretiert werden.

Für die Machbarkeitsstudie wurde eine kommerzielle Spracherkennungssoftware [62] installiert und getestet. Es zeigte sich, dass während die Erkennungsqualität in leisen Umgebungen zufriedenstellend war, sank diese in Umgebungen mit lauten Hintergrundgeräuschen dramatisch. Oftmals mussten die Befehle mehrmals laut wiederholt werden, bevor die gewünschte Wirkung erreicht wurde. Insgesamt scheint der Einsatz von „stillen“ Bedienelementen in der Öffentlichkeit angemessener zu sein, da sich unbeteiligte Menschen sonst angesprochen oder belästigt fühlen könnten. Zudem belastet der ständig im Hintergrund laufende Spracherkennungsprozess den Prozessor zusätzlich. Demnach ist die Sprachsteuerung vor allem als ein alternatives Bedienkonzept für den Einsatz im Heimbereich geeignet, allerdings steht die endgültige Bewertung der Benutzerfreundlichkeit dieser

Methode noch aus. Davon unabhängig sollte es eine Möglichkeit zur lautlosen

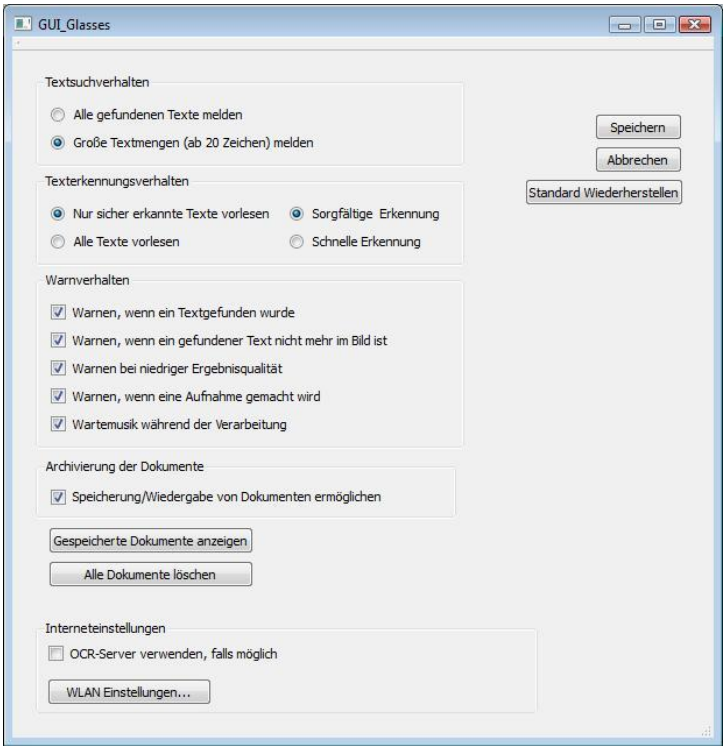


*Abb. 3.6.2:Erweiterte Einstellung des Geräts mittels einer grafischen Schnittstelle.*

Steuerung mit herkömmlichen Bedienelementen wie bspw. einer Präsentations-maus geben.

Um die Einfachheit der Bedienung zu gewährleisten, wurde ein Steuerungskonzept entwickelt, welches lediglich vier Knöpfe (bzw. Sprachbefehlen) vorsieht. Die größte Schwierigkeit dabei besteht darin, die Erlernbarkeit der Bedienung trotz der Überladung der Knöpfe zu erhalten. Der in Abb. 3.6.1 vorgestellte Steuerungsentwurf sieht deswegen vor, dass die Grundfunktionalität des Geräts mit einem einzigen Knopf (blau-grüner Pfad) genutzt werden kann, während die restlichen Tasten für die Navigation innerhalb des Textes sowie für die Archivierung und Abruf von Dokumenten notwendig sind. Einige erweiterten Funktionen wie bspw. spezielle Navigationsmodi sind für erfahrene Anwender gedacht und lassen sich über eine Mehrfachbetätigung der Tasten abrufen. Auf jede Anweisung des Benutzers reagiert das System mit einer Sprachmeldung, die jedoch auf Wunsch einzeln abgestellt werden können.

Um auch jüngeren und technisch versierten Anwendern gerecht zu werden, sollte das System eine Möglichkeit zur flexiblen Anpassung des Geräts an die jeweiligen Bedürfnisse bieten können (s. Anhang A). Damit die Bedienung für alle anderen Benutzer nicht unnötig kompliziert wird, werden die erweiterten Einstellungen des Gerät mittels einer grafischen Schnittstelle vorgenommen (s. Abb. 3.6.2), mit deren Hilfe das Verhalten des Systems flexibel gesteuert werden kann. Für den einfachen Betrieb ist die GUI allerdings nicht zwingend notwendig. Die Einstellung kann durch den Betroffenen selbst unter Verwendung eines Bildschirmleseprogramms bzw. von einem sehenden Helfer vorgenommen werden.

## 3.7 Zusammenfassung: Systemspezifikation

Das folgende Pflichtenheft wurde auf der Grundlage von der zuvor vorgestellten Analyse erstellt:

1. **Zielbestimmung**
   a. **Musskriterien:** Automatisches Vorlesen von Texten auf unterschiedlichen Textträgern wie Dokumenten, Büchern, Aushängeschildern.
   b. **Sollkriterien:** Navigation innerhalb des erkannten Textes: satzweise, blockweise, dokumentweise Springen, Anhalten, Abbrechen. Speicherung und erneuter Wiederabruf von Ergebnissen. Erweiterte Einstellungen des Geräts: variable Sprechrate, Kontrolle über das Benachrichtigungsverhalten für einen sicheren Betrieb im Straßenverkehr
   c. **Abgrenzungskriterien:** Keine Handschrifterkennung, kein Betrieb bei Niederschlägen
2. **Systemeinsatz**
   a. **Zielgruppe:** Blinde und sehbehinderte Menschen, in erster Linie Senioren, u. a. technisch wenig versierte Anwender
   b. **Betriebsbedingungen:** Vielfältige Einsatzbereiche: Zuhause wie im Freien. Das Gerät muss möglichst pflegeleicht sein und die in der elektronischen Industrie üblichen Toleranzvorgaben erfüllen [58].
3. **Bedienung**
   Eine akustisch-haptische Benutzerschnittstelle für die Verwendung durch die Betroffenen, eine grafische Benutzeroberfläche für die erweiterte Einstellungsverwaltung, die jedoch optional sein muss.
4. **Funktionale Anforderungen**
   Das System sucht selbstständig nach textueller Information in der Umgebung

und benachrichtigt den Benutzer im Falle einer Entdeckung. Als Richtwert für die Reaktionszeit des Textdetektionsalgorithmus wird 1 s angenommen. Der Benutzer hat die Kontrolle über das weitere Vorgehen. Das Gerät ist in der Lage, die Aufnahmebedingungen zu bewerten und nimmt diesen entsprechend Kameraeinstellungen vor. Bei Bedarf bittet das System den Anwender um seine Mitwirkung. Das System ist in der Lage verschiedene Dokumente innerhalb einer Aufnahme auseinander zu halten und die Vorlesereihenfolge zu bestimmen. Während des Vorlesens ist die Navigation innerhalb des erkannten Textes möglich, nach dem Vorlesen besteht darüber hinaus eine Archivierungsmöglichkeit.

5. **Nicht-funktionale Anforderungen**
   a. **(Funktions-) Zuverlässigkeit:** Das System muss mit diversen Schriftarten, Schriftgrößen und Schriftfarben, Texthintergründen, Textorientierungen, Layouts, Lichtverhältnissen zurechtkommen. Das System kann Situationen erkennen, wenn Textteile verdeckt oder abgeschnitten sind und eine entsprechende Warnung ausgeben. Das System nimmt bei Bedarf eine Entzerrung des Dokuments vor, um das Ergebnis der Zeichenerkennung zu verbessern.
   b. **Fehlertoleranz:** Das System betreibt eine permanente Überwachung des Hauptprogramms und initiiert bei schwerwiegenden Problemen einen Neustart. Das System gibt außerdem eine Warnmeldung bei niedrigem Akkustand oder beim Ausfall von Systemkomponenten und bietet eine Anleitung zum Beheben des Problems in sprachlicher Form an.
   c. **Aussehen und Handhabung:** Das System muss so unauffällig wie möglich aussehen. Es wird am Körper getragen und darf die Bewegungsfreiheit des Benutzers nicht einschränken. Es ist ferner darauf zu achten, dass die Sicherheit des Blinden im Straßenverkehr durch die Anwendung des Geräts nicht beeinträchtigt wird.
   d. **Benutzbarkeit:** Die Steuerung des Systems muss intuitiv und einfach sein, sodass auch technisch weniger versierte Menschen mit Behinderung damit klar kommen. Das bedeutet insbesondere, dass das System in der Lage sein muss, die Aufnahmebedingungen, Aufnahmequalität sowie das Resultat zu bewerten. Bei grundlegenden Entscheidungen wendet sich das Gerät per Sprachausgabe an den Anwender.
   e. **Leistung und Effizienz:** Die Benachrichtigung des Benutzers bei Textentdeckung sollte innerhalb von einer Sekunde erfolgen, sodass die Ausrichtung der Kamera auf die entdeckte Textregion, schnell reproduziert werden kann. Die Latenzzeit zwischen dem Einleiten der Verarbeitung durch den Benutzer

und dem Beginn des Vorlesevorgangs sollte den Richtwert von 30 s nicht überschreiten. Die resultierenden Erkennungsraten sollten mit denen der modernen kommerziellen OCR-Lösungen wie OmniPage für dieselben Dokumentaufnahmen stets vergleichbar sein.

f. **Wartbarkeit und Robustheit:** Das Gerät muss weitestgehend wartungsfrei sein. Auf die Anweisung des Geräts hin werden Batterien geladen/ausgetauscht. Auch auf die mechanische Robustheit ist aufgrund der Mobilität des Systems zu achten.

**6. Ergänzungen**

Das System wird aus preisgünstigen Standardkomponenten gebaut. Für die Zeichenerkennung wird ein kommerzielles Produkt verwendet.

# 4. Kapitel

# Systementwurf

Um die z. T. konkurrierenden Anforderungen an das System miteinander vereinen zu können, bedarf es einer sorgfältigen Planung und frühzeitigen Prioritätensetzung. In diesem Kapitel wird ein Grundkonzept für den Aufbau des Geräts diskutiert und ein Systementwurf auf der Grundlage der Anforderungsanalyse präsentiert.

## 4.1 Gesamtkonzept

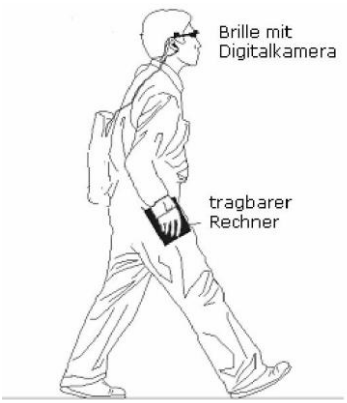


*Abb. 4.1.1: Erste Systementwürfe*

Ein wesentlicher Mangel der existierenden mobilen Vorlesesysteme besteht darin, dass der Benutzer selbst für eine geeignete Ausrichtung der Kamera auf das entsprechende Dokument sorgen muss. Das Konzept des *aktiven Sehens* liefert allerdings eine mögliche Lösungsstrategie für diese Problematik. Unter dem Konzept des aktiven Sehens wird die Fähigkeit eines visuellen Systems verstanden, seine Parameter wie Position, Orientierung, Fokus, Zoom, Blende den Anforderungen der aktuellen Aufgabe entsprechend aktiv zu regeln [63]. Doch während nahezu alle modernen Kameras standardmäßig mit Autofokus, einer automatischen Belichtungssteuerung und Weißabgleich ausgestattet werden, sind Systeme mit Blicksteuerung, die eine situationsabhängige Anpassung der Ausrichtung der Kamera vornehmen, nicht ohne Weiteres zu erwerben. Mobile Systeme sind in dieser Hinsicht besonders problematisch, da sich die dafür notwendigen elektromechanischen Elemente

nicht beliebig miniaturisieren lassen. Servomotoren steigern die Komplexität des Gesamtsystems erheblich und belasten zudem die Akkumulatoren in einem ebensolchen Maße, sodass die Robustheit und Mobilität des Gesamtsystems negativ beeinträchtigt wird. Schließlich ist es schwierig die Unauffälligkeit eines solchen Systems zu gewährleiten.

Alternativ zum Einsatz einer aktiven Kamera kann die Ausrichtungsproblematik durch eine Vergrößerung des Sichtfeldes mittels eines Kamera-Arrays angegangen werden. Dadurch entfallen die mechanischen Komponenten des Systems, während die Aufnahmezeit und Komplexität der Datenfusion reduziert werden. Diese Überlegung war ausschlaggebend für die Entscheidung zugunsten der Kamera-Array-Lösung. Zwar wurden in dem ersten Prototypen aufgrund der Bandbreitenbeschränkungen des Anschlusses lediglich zwei Kameras verbaut, allerdings wird an der Stelle davon ausgegangen, dass die im Rahmen dieser Arbeit entwickelten Algorithmen sich ohne Weiteres auf größere Arrays anwenden lassen. Durch die Verwendung von Kamera-Arrays rücken Stereo-Vision-Methoden in den Vordergrund der Betrachtung. Die zusätzlichen Distanzinformationen werden im Rahmen dieser Arbeit an mehreren Stellen zur Verbesserung der Bildqualität und Entscheidungsfindung eingesetzt.

Auch wenn die Restriktion bzgl. der Anzahl der Kameras sich negativ auf die Benutzbarkeit des Systems auswirkt, ist diese unvermeidbar, da sich kaum mehr als zwei Kameras unauffällig und gleichzeitig gewinnbringend für das Stereovision in einem Brillengestell unterbringen lassen. Der Grund dafür ist der notwendige Abstand zwischen den Kameras, von dem der Öffnungswinkel und die Genauigkeit der Distanzmessung abhängen (s. Abschnitt 2.6). Der Kamera-In-Der-Brille-Ansatz (s. Abb. 4.1.1) bringt indes gleich einige Vorteile mit sich: Zum einen ist es für späterblindete Menschen – und das ist die mit Abstand größte Gruppe [2] – vollkommen natürlich den Kopf nach dem auszurichten, was sie gerade interessiert, selbst wenn sie überhaupt nicht mehr sehen können und erst recht wenn noch ein Sehrest vorhanden ist. Zum anderen kann sogar bei vielen Geburtsblinden davon ausgegangen werden, dass sie ihren Kopf gerade halten [64], was die Textlokalisierung und -segmentierung erleichtert.

## 4.2 Hardwarekonzept

Die wichtigsten zu berücksichtigenden Anforderungen bei der Festlegung der Hardware-Komponenten sind *Mobilität, Aussehen und Handhabung* sowie *Robustheit*. Eine passende Videokamera zu finden war sicherlich die größte Herausforderung bei der Hardware-Beschaffung. Sie musste ausreichend klein sein, damit zwei davon in ein Brillengestell eingebaut werden konnten und gleichzeitig Bilder mit einer Auflösung von mindestens 5 Mpx (s. Abschnitt 3.5) liefern. Hohe Sensorempfindlichkeit ist eine weitere wichtige Kameraanforderung, denn lange Belichtungszeiten begünstigen Bewegungsartefakte, während eine große Blendenöffnung die Schärfentiefe reduziert [124]. Insbesondere ist der bei vielen modernen CMOS-Kameramodellen auftretende Rolling-Shutter-Effekt* problematisch. Außerdem muss die Miniaturkamera über ein elektrisch steuerbares Fokusmodul verfügen. Das MCB1172 Kameramodul von SONY erfüllt sämtliche genannten Anforderungen (s. Abb. 4.2.1) sowie die spezifizierten Anforderungen bzgl. der Betriebsbedingungen [65]. Das Modul liefert Bilder mit bis zu 8,3 Mpx und ist mit einem DSP- Chip mit zahlreichen integrierten Funktionen wie automa-

*Abb. 4.2.1: fit PC2 Rechner mit der Präsentationsmaus, der Brillenprototyp und das MCB1172 Kameramodul mit der USB Schnittstelle.*

tischem Weißabgleich, automatischer Belichtungszeit sowie Bildstabilisierung ausgestattet.

* Unter Rolling-Shutter-Effekt wird die Verzerrung einer Aufnahme aufgrund der zeilen- oder spaltenweise Belichtung/Auslesen des Bildsensors verstanden.

Es wurde zunächst ein Versuch unternommen, die Kamera direkt über die DSP-Schnittstelle einer Embedded-System-Platine* zu steuern und auszulesen. Dabei stellte sich jedoch heraus, dass ab einer Kabellänge von etwa 30 cm (gemessen von der Kamera bis zum DSP-Anschluss) kein zuverlässiger Bildabruf mehr möglich ist. Um die Übertragungsweite zu vergrößern, müssen die Frames zwischengespeichert und anschließend über ein Standard-Bussystem an den Rechner weitergeleitet werden, was infolge der beschränkten Übertragungskapazitäten zur Begrenzung hinsichtlich der Kameraanzahl führte. Es ist jedoch zu erwarten, dass mit Bussystemen der nächsten Generation wie bspw. USB 3.0 größere Kamera-Arrays realisierbar werden.

Zwei unterschiedliche Plattformen wurden bei der Auswahl der Systemkomponenten in Betracht gezogen: *igep Embedded Board* [66] auf Basis der ARM-Architektur und *fitPC2* [67], der mit einem Intel Atom Prozessor ausgestattet ist. Die Leistung der beiden Alternativen bei der Bewältigung von OCR-spezifischen Aufgaben wurde untersucht und verglichen. Das ARM-basierte System hatte erwartungsgemäß immense Vorteile bezüglich der Mobilität und zwar sowohl was die Leistungsaufnahme (8W gegen 3.5W bei voller Leistung) und Wärmeentwicklung als auch was die Abmessungen der Leiterplatten angeht. Das Atom-Board benötigte hingegen nur halb (56%) so viel Zeit bei der Bewältigung der OCR-Aufgaben. Beide Systeme weisen eine hohe Robustheit gegenüber Erschütterungen und einen hinreichend breiten zulässigen Betriebstemperaturbereich (s. Abschnitt 3.6) auf. Der Markt der eingebetteten Systeme befindet sich zurzeit in einer dynamischen Phase und so wurde beschlossen zwei Systeme parallel zu entwickeln und die Entwicklungen auf dem Markt zu beobachten. Auf dem Intel-System ist Windows XP und das *OmniPage*-Zeichenerkennungsmodul installiert, während auf dem ARM-Rechensystem eine Linux-Version von *FineReader* verwendet wird. Ergänzt wurden die beiden Systeme durch Bedienelemente, Headsets sowie Akkupacks für die Stromversorgung der Rechner.

## 4.3 Verarbeitungskonzept

Beim Entwurf des Verarbeitungskonzepts ist vor allem auf die Gewährleistung der Anforderungen *Leistung und Effizienz* sowie *Fehlertoleranz* zu achten. Ange-

* Es wurde das ARM basierte ISEE IGEP Modul verwendet.

strebt wird u. a. eine Echtzeit-Lokalisierung von leserlichen Textstellen im Bild mit Latenzzeiten von unter 1 s (s. Abschnitt 3.6). Da alleine schon die Übertragung von vollaufgelösten 8 Mpx Bilder zum Rechner fast 1 s dauert, findet die einleitende Textdetektion auf gebinnten* Aufnahmen statt. Das Binning gehört zu den integrierten Funktionen des gewählten Kameramoduls [65] und hat den positiven Nebeneffekt, dass das Signal-Rausch-Verhältnis einer Aufnahmen dadurch erhöht wird [68]. Die Zuverlässigkeit der Textdetektion und der *Leserlichkeitsbewertung* wird jedoch durch die Herunterskalierung verringert, da kleine Bildkomponenten wie bspw. Buchstaben dabei strak verzerrt oder komplett unkenntlich gemacht werden.

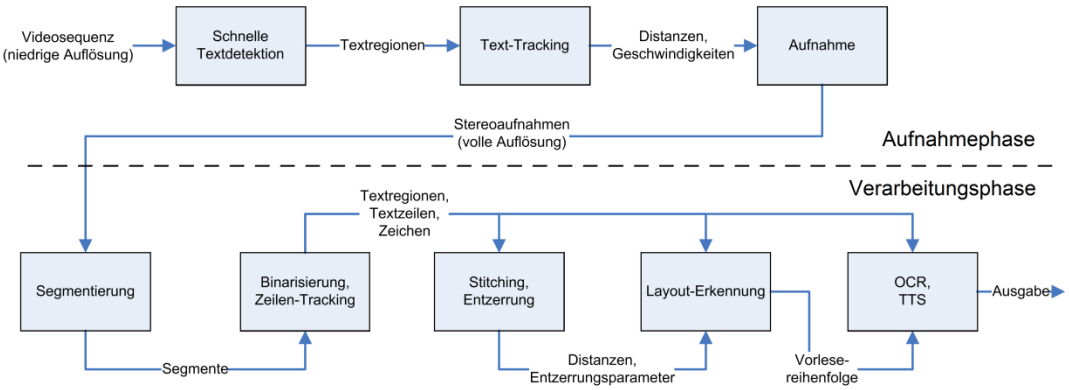


*Abb. 4.3.1:Verarbeitungkonzept (Flussdiagramm)*

Die Dokumentverarbeitungskette ist in Abb. 4.3.1 als Flussdiagramm dargestellt. Sobald eine Textregion entdeckt und als gut leserlich eingeschätzt wurde, erfolgt die Benachrichtigung an den Benutzer, währenddessen die Positionsänderungen der entdeckten Regionen überwacht werden. Die Verfolgung von Textstellen hat mehrere Ziele:

- Wiederherstellung der ursprünglichen Ausrichtung der Kamera
- Bestimmung der optimalen Kameraeinstellungen durch eine Messung der Lichtverhältnisse und Distanz zum Dokument

* Binning - Zusammenfassen mehrerer Pixel zu einem einzigen im Bildsensor.

- Intelligenter Auslöser mittels Bewegungsdetektion

Um Bewegungsartefakte zu vermeiden, muss die Kamera vor und während der Aufnahme stabilisiert werden. Findet sich innerhalb eines festgelegten Zeitintervalls kein günstiger Auslösezeitpunkt, so wird der Benutzer darauf hingewiesen und aufgefordert den Kopf stillzuhalten. Zum Schluss wird das Binning ausgeschaltet und die Aufnahme zweier 8 Megapixel-Bilder initiiert.

Die Verarbeitungsphase beginnt mit einer sorgfältigen Segmentierung der Aufnahmen, wobei gleichzeitig eine grobe Abschätzung der Schriftgrößen, Zeilenabstände und Orientierungen der Textzeilen in den Regionen stattfindet. Diese Merkmale werden anschließend für die Parametrisierung der Algorithmen zur Binarisierung (s. Abschnitt 2.7) und Zeilenextraktion verwendet. Das Ergebnis der Zeilenextraktion ist eine Liste von nach Zeilen sortierten Zeichen, die ihrerseits als Konturen von Zusammenhangskomponenten vorliegen. Mit Hilfe der extrahierten Merkmale wird die physische Struktur des Dokuments modelliert und ein 3D-Modell der Dokumentoberfläche berechnet, welches für die Entzerrung der Zeichenkonturen eingesetzt wird. Gleichzeitig findet die Korrektur des physischen Layoutmodells und ggf. das zeilenweise Stitching der aus den beiden Aufnahmen stammenden Textteile statt. Schließlich wird anhand des physischen Layoutmodells die logische Dokumentstruktur abgeleitet und die Vorlesereihenfolge bestimmt, sodass die Zeichenerkennung blockweise ausgeführt werden kann. Während die ersten Blöcke bereits ausgegeben werden, findet die Verarbeitung der restlichen Textstellen im Hintergrund statt. Um die Antwortzeit des Systems zusätzlich zu reduzieren kann die OCR und die TTS-Verarbeitung auf einen dedizierten Server ausgelagert werden. Diese Option ist in erster Linie für die Verwendung im Heimbereich oder einem sonstigen vorkonfigurierten WLAN-Netz gedacht. Zur erstmaligen Einrichtung einer WLAN Verbindung kann die graphische Benutzeroberfläche verwendet werden.

# 5. Kapitel

# Aufnahmephase

Eine intelligente geräteseitige Unterstützung des Benutzers bei der Entdeckung und Aufnahme textueller Informationen soll eine Verbesserung der Benutzbarkeit und Robustheit des Systems bewirken – schließlich hängt das Ergebnis der Verarbeitung von der Qualität der Eingabebilder ab. Die Unterstützungsfunktionalität in der Aufnahmephase umfasst folgende Dienste:

- Automatische Textentdeckung (Anschnitt 5.1)
- Unterstützung bei der Bestimmung der Kameraausrichtung und Kadrierung (Abschnitt 5.2)
- Automatische Einstellung des Fokus (Abschnitt 5.3)
- Intelligenter Auslöser zur Vermeidung von Bewegungsartifakten

Die vier oben genannten Aufgaben werden in diesem Kapitel in der Reihenfolge ihres Auftretens abgehandelt.

## 5.1 Schnelle Textlokalisierung

Dieser Abschnitt behandelt die Problematik der schnellen Textdetektion sowie der heuristischen Leserlichkeitsbewertung der entdeckten Textbereiche. Ein schneller Algorithmus für die Entdeckung von Textinformation in grob aufgelösten Aufnahmen wird präsentiert, der die im Abschnitt 3.7 Punkt 4 formulierten Anforderungen erfüllt.

### 5.1.1 Problemstellung

Im Rahmen dieser Arbeit werden zwei verschiedene Verfahren zur Textlokalisierung vorgestellt. Sorgfältige Textlokalisierung in hoch aufgelösten Aufnahmen für

die Layouterkennung wird im Kapitel 6 diskutiert. Das Thema dieses Abschnitts ist eine echtzeitfähige Methode zur heuristischen Detektion von voraussichtlich leserlichen Textstellen, die im Initialzustand des Systems (s. Abschnitt 3.6) angewendet wird.

Wie die im Abschnitt 3.5 präsentierte Vergleichsstudie ergab, setzen alle analysierten OCR-Systeme Algorithmen zur heuristischen Textlokalisierung im Vorfeld der eigentlichen Zeichenerkennung ein. Häufig kann dabei eine erhebliche Ausfallrate* des verwendeten Klassifikators in Kauf genommen werden, da es lediglich darum geht, mögliche Textkandidaten zu lokalisieren, ohne dass Textstellen dabei übersehen werden. Die aktuelle Problemstellung unterscheidet sich grundlegend von dem beschriebenen, klassischen Fall. Das präsentierte Steuerungskonzept sieht vor, dass jede Textdetektion im Initialzustand des Systems eine Benachrichtigung des Benutzers auslöst, sodass eine hohe Ausfallrate des Klassifikators die Benutzbarkeit des Systems stark beeinträchtigen würde. Gelegentlich produzierte falsch-negative Ergebnisse sind für die Textdetektion hingegen unproblematisch, da die Suche mehrmals pro Sekunde wiederholt wird.

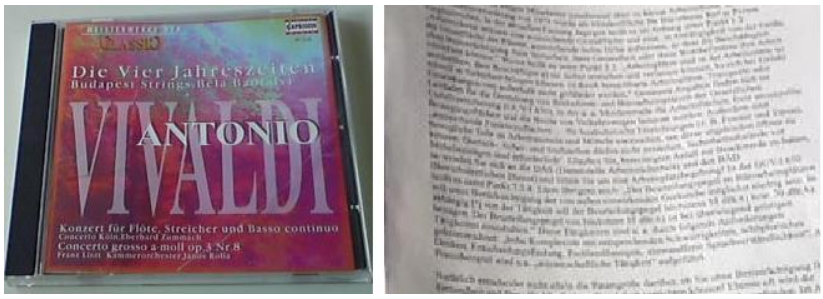


*Abb. 5.1.1: Probleme bei der Textlokalisierung. Auf der rechten Abbildung ist dargestellt, wie die einzelnen Buchstaben in grob aufgelösten Aufnahmen zu verschwommenen Streifen verschmelzen.*

Die wichtigsten Merkmale für die Bewertung der Leserlichkeit einer Textregion sind die geschätzten Schriftgrößen sowie die Kontrastqualität. Dabei werden die Schranken für die Schriftgröße in Anlehnung an die spezifizierten Werte (s. Abschnitt 3.5) festgelegt, während das Kontrastqualitätsmaß heuristisch festgelegt wird. Das liegt vor allem daran, dass die Bestimmung der Lesbarkeit von vollauf-

* Als Ausfallrate wird hier der Anteil der falsch-positiven Klassifikationsergebnisse bei der Textdetektion bezeichnet.

gelösten Bildern aufgrund der herunterskalierten Aufnahmen nur approximativ erfolgen kann (s. Abb. 5.1.1).

Die weichen Echtzeitanforderungen stellen angesichts der knappen verfügbaren Rechenleistung (s. Abschnitt 4.2) eine Herausforderung dar. Dabei geht es nicht nur darum den spezifizierten Richtwert für die Reaktionszeit von unter 1 s zu gewährleisten, sondern auch um die Minimierung der CPU-Last, die infolge der permanenten Suche nach Textstellen entsteht. Die Auslastung des Prozessors kann sowohl die Haltedauer der Batterie als auch die Wärmeentwicklung des Rechnergehäuses in einem erheblichen Maße beeinflussen[*].

Eine weitere wichtige Anforderung ist die hohe Robustheit der Textdetektion, da die Textstellen, je nach Anwendungsfall, verschiedene Schrifteigenschaften (Art, Größe, Farbe) und Layout-Eigenschaften (Textmenge, Textorientierung, Beschaffenheit des Textträgers und des Hintergrundes) besitzen können. Weil der Text in einer komplexen und dynamischen Umgebung eingebettet sein kann, muss des Weiteren eine mögliche teilweise Verdeckung oder Abschneidung der Textregionen sowie die u. U. inhomogene Beleuchtung bedacht werden.


*Abb. 5.1.2: FU-Logo*

Die Grenze zwischen dem Textuellen und dem Grafischen ist fließend, wie bspw. in Abb. 5.1.2 demonstriert wird. Aus diesem Grund wird an der Stelle eine Reihe von Kriterien an die Beschaffenheit der Dokumente festgelegt, die eine Identifikation von Textbereichen erleichtern sollen:

1. Bimodalität der Pixelwert-Dichtefunktion
2. Starke Maxima im Gradientenfeld an den Buchstabenkanten.
3. Homogenität der:
    a. Zeichenhöhen $h_s$ in einem Textblocks $R$:
    $2 * \min_{h_s \in R} h_s \geq \max_{h_s \in R} h_s$
    b. Abstände zwischen den Zeichen[†] einer Zeile $\Delta_s$:

[*] Bspw. beträgt die Spanne der Leistungsaufnahme von fitPC2 5-8 Watt [67].

[†] Der Abstand wird auch als Laufweite bezeichnet.

$2 * \min_{\Delta_s \in R} \Delta_s \geq \max_{\Delta_s \in R} \Delta_s$

c. Abstände zwischen den Zeilen eines Textblocks $\Delta_l$:
$2 * \min_{\Delta_l \in R} \Delta_l \geq \max_{\Delta_l \in R} \Delta_l$

4. Abstände zwischen den Zeichen $\bar{\Delta}_s$ sind:
   a. im Schnitt kleiner als Abstände zwischen den Zeilen:
   $4 * \bar{\Delta}_s < \bar{\Delta}_l$
   b. kleiner als die Höhen $h_s^1, h_s^2$ der benachbarten Zeichen:
   $4 * \Delta_s < min(h_s^1, h_s^2)$
5. Der Abstand zwischen zwei Textregionen ist größer als der größte Zeichenabstand in diesen Regionen:
   $\Delta_{R1,R2} > 4 * max\left(max_{\Delta_s \in R1} \Delta_s, max_{\Delta_s \in R2} \Delta_s\right)$.

Wie bereits angesprochen, sind Farbinformationen für eine Textlokalisierung unerheblich und können zudem, angesichts der u. U. komplizierten Lichtverhältnisse, nur unter einem erheblichen Aufwand extrahiert werden. Dementsprechend bezieht sich die 1. Bedingung auf die Intensitätswerte der Pixel. Aufgrund der Beleuchtungsinhomogenität kann bei einer Aufnahme von Textregionen nur von einer lokalen Bimodalität ausgegangen werden.

Die 2. Bedingung erlaubt den Einsatz von Kantenmerkmalen für die Textdetektion. An der Stelle muss zwischen dem sorgfältigen Ansatz auf den 8-Megapixel-Bildern und dem schnellen Ansatz auf den Videostrom-Frames unterschieden werden, da die Kanten infolge der Binning-Operation verschwimmen bzw. gänzlich verschwinden können.

Die im Punkt 3b aufgeführte Regelmäßigkeitsbedingung kann infolge der Binning-Operation u. U. verletzt werden, da die Abstände zwischen den einzelnen Zeichen $\Delta_s \leq 4$ Pixel dabei überbrückt werden (s. Abb. 5.1.1). Auf der vollaufgelösten Aufnahme derselben Szene kann die Textregion unterdessen noch lesbar sein.

Infolge einer Verdeckung von Textregionen kann es stellenweise zu Verletzungen der 5. Bedingung kommen.

### 5.1.2 Vorarbeiten

Die Entwicklung von Textdetektionsalgorithmen ging mit dem wachsenden Bedürfnis nach schnelleren Verarbeitungsverfahren für große Mengen von Dokumentaufnahmen einher. Seit den ersten Publikationen in den 80er Jahren wurde eine große Vielfalt von wissenschaftlichen Arbeiten zu dem Thema veröffentlicht, in denen verschiedene Kombinationen von Segmentierungs- und Klassifikationsalgorithmen für die Textlokalisierung vorgeschlagen wurden. Um einen besseren Überblick zu gewährleisten, werden die Methoden gemäß ihrem Segmentierungsverhalten in drei Gruppen unterteilt [110]:

- *Bottom-Up-Verfahren*, die eine Strategie der Vereinigung von einzelnen Pixeln zu immer größeren Pixelmengen verfolgen, wobei gleichzeitig für jede Menge statistische Daten erhoben und anschließend zur Klassifizierung der Segmente verwendet werden.
- *Top-Down-Verfahren* teilen die Aufnahmen rekursiv in immer kleinere Regionen auf. Die Klassifizierungsmerkmale werden ggf. erst am Ende eines Rekursionsschritts extrahiert.
- *Hybride Methoden*, die die Bottom-Up- und Top-Down-Strategien kombinieren.

Darüber hinaus lassen sich die Methoden danach gliedern, welche Merkmale sie für die Klassifikation einsetzen (vgl. [110]):

- *Pixelbasierte Merkmale*
- *Kantenbasierte Merkmale*
- *Texturbasierte Merkmale*
    - *Frequenzspektrum-basierte Merkmale*
    - *Maschinell gelernte Merkmale*

Die von K.Y. Wong et al. [69] beschriebene pixelbasierte Top-Down-Methode gehört zu klassischen Textlokalisierungsalgorithmen. Für die Bildsegmentierung wird der *RLS-Algorithmus (*engl. *run length smoothing)* [70] verwendet, der ein binarisiertes Bild horizontal und vertikal scannt, wobei die Lücken zwischen den Vordergrundpixeln geschlossen werden, die kleiner als ein vordefinierter Schwellenwert sind. In den so entstehenden zusammenhängenden Vordergrundbereichen werden bestimmte Merkmale, wie die Dichte der Vordergrundpixel, die mittlere

Länge der Blöcke im horizontal/vertikal geglätteten Bild und das Seitenverhältnis des jeweiligen Blocks extrahiert, mit deren Hilfe schließlich die Klassifikation stattfindet. Der RLS-Algorithmus ist der Vorläufer einer ganzen Klasse von Algorithmen, die die Bildsegmentierung mit Hilfe von *morphologischen Operatoren* (s. Abschnitt 3.3) durchführen [71][72][73][74][75][76][77]. Tatsächlich entspricht die RLSA-Glättung mit dem Lückenparameter $\alpha$ einer eindimensionalen *Schließen-Operation* [76] unter Verwendung einer Strukturmaske $\mathcal{M}_\alpha$, deren Größe in Abhängigkeit von $\alpha$ festgelegt wird:

$$RSLA_\alpha(I^B) = I^B \bullet \mathcal{M}_\alpha$$

Der große Vorteil dieser Algorithmen ist die hohe Zeiteffizienz, allerdings ist ihre Einfachheit mit einer ganzen Reihe von Schwachstellen verbunden:

- die Notwendigkeit einer frühzeitigen Binarisierung des Bildes
- Empfindlichkeit gegenüber der Zeilenorientierung und -krümmung
- Empfindlichkeit gegenüber der Schriftgröße

Kantenbasierte Methoden stellen eine Möglichkeit dar, die Binarisierungsproblematik zu umgehen. Neben den klassischen Kantenextraktionsoperatoren (s. Abschnitt 2.3) [78][79] werden Methoden der Grauwert-Morphologie [71][72] für eine Transformation in den Kantenraum eingesetzt, in dem bestimmte textspezifische Klassifizierungsmerkmale einfacher extrahiert werden können. Ein weiterer Vorteil der Transformation ist die dabei stattfindende Hochpass-Filterung der Aufnahme, wodurch die negativen Effekte einer inhomogenen Beleuchtung gemildert werden.

Auch für die Orientierungsproblematik wurden bereits zahlreiche Lösungsansätze vorgeschlagen. In [76] wird eine Weiterentwicklung des RLS-Algorithmus vorgestellt, wobei anstelle eines achsensymmetrischen strukturierenden Elements eine Familie von unterschiedlich ausgerichteten Filtern für die Schließen-Operation verwendet werden. Eine weitere Möglichkeit zur Feststellung der Textflussrichtung bieten *Projektionsprofile*. Dabei werden Pixelwerte entlang einer Kurve akkumuliert und die entstehende Projektionsmuster anschließend ausgewertet [80][81]. Mit Hilfe der so produzierten Profilhistogramme lässt sich u. U. nicht nur die Zeilenorientierung, sondern auch die Schriftgröße eines Textblocks grob

abschätzen. Zu den großen Schwächen der Projektionsprofile zählt ihre Anfälligkeit (s. Abb. 5.1.3) gegenüber Dokumentverzerrungen [80].

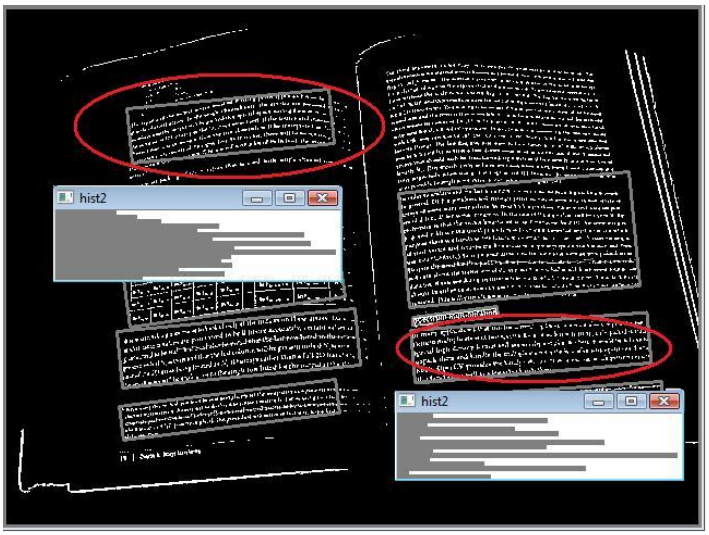


*Abb. 5.1.3: Projektionsprofile im Vergleich: links eine stark verzerrte Region, rechts gerade Linien*

Die Wahl einer passenden, schriftgrößenabhängigen Betrachtungsskala im Vorfeld einer Dokumentanalyse stellt eine große Herausforderung dar. Eine zuverlässige Merkmalsextraktion ist auf die Kenntnis der erwarteten Schriftgröße angewiesen, während für die Schätzung der Schriftgröße aussagekräftige textspezifische Merkmale benötigt werden. Viele der schnellen Methoden [71][72][73][75] werden daher durch die Wahl geeigneter Parameter auf einen bestimmten anwendungsabhängigen Schriftgrößenbereich zugeschnitten. Für den Fall, dass eine robustere Lösung benötigt wird, können texturbasierte Methoden häufig eine Abhilfe schaffen. Eine besondere Gruppe innerhalb der texturbasierten Algorithmen bilden Frequenzspektrum-basierte Verfahren (s. Abschnitt 2.8) und Verfahren, die auf der Multiskalenanalyse (s. Abschnitt 2.9) basieren. Eine Reihe von Autoren [82][83][84] schlagen die Verwendung von Gabor-Filtern für die Extraktion der Ortsfrequenzmerkmale vor, welche angesichts der charakteristischen repetitiven Muster innerhalb von Textregionen eine spezifische Verteilung aufweisen. Die für die Merkmalsextraktion notwendige Mehrfach-Filterung der gesamten Aufnahme mit unterschiedlich parametrisierten Gabor-Filtern (engl. *filter bank*) ist allerdings sehr rechenaufwändig, sodass die FWT [85][86][87] in vielen Fällen eine bessere

Alternative darstellt. Auch Bildpyramiden kommen zum Einsatz [77], wobei Merkmale aus verschiedenen Pyramidenebenen zu multiskalaren Textur-Deskriptoren kombiniert werden. Schließlich bieten sich DCT-Koeffizienten für die Textdetektion in komprimierten Videoströmen [88][89] an.

Neben der Segmentierung gehört die Klassifizierung zu den wichtigsten Teilaufgaben einer Textlokalisierung. Die Unterscheidung zwischen Text- und Nicht-Text-Stellen kann sowohl nach als auch während der Segmentierung unter Verwendung von maschinell gelernten textspezifischen Texturmerkmalen (*Modellbasierte Segmentierung*) [90][91][92][93][94] stattfinden, wobei diverse allgemeine Klassifikatoren eingesetzt werden: Nearest-Neighbor-Klassifikator [73][95], SVM (engl. *support vector machines [96]*)[92][97], neuronale Netze [90][94] [98], Bayes'sche Netze [93], AdaBoost (*adaptive boosting [99]*) [78]. Die meisten texturbasierten Verfahren sind geeignet für Dokumente mit verschiedenen Schriftgrößen und Zeilenorientierungen und robust gegenüber Dokumentverzerrungen. Der in der Arbeit [100] vorgestellte Klassifikator auf der Basis von Entscheidungsbäumen stellt ein Spezialfall der texturbasierten Bottom-Up-Verfahren dar, wobei Textstellen aus den einzelnen Zeichen zusammengestzt werden. Die Verwendung von gelernten Merkmalen bringt i. d. R. gewisse Laufzeitvorteile mit sich, birgt jedoch die Gefahr von Falschklassifikationen infolge einer Überanpassung des Textmodells.

Vollständigkeitshalber seien hier noch Top-Down-Segmentierungsalgorithmen erwähnt, die nicht nach Textblöcken suchen, sondern nach rechteckigen Zwischenräumen innerhalb eines Dokuments [101][102]. Diese Algorithmen wurden speziell für eingescannte Dokumentaufnahmen entwickelt und setzen eine geringe Oberflächenverzerrung voraus. Auch Verfahren, die auf Farbinformationen basieren, seien hier aufgrund von erwarteten komplizierten Lichtverhältnissen nur am Rande erwähnt [103][104].

### 5.1.3 Gesamtkonzept

Das Diagramm in Abb. 5.1.4 zeigt den Ablauf der schnellen Textdetektion, die im Initialzustand des Systems durchgeführt wird. Der Algorithmus besitzt eine Pipeline-Struktur zur schnellen Abweisung von Bildbereichen, die keine Textinformationen enthalten, mit einer stufenweisen Erhöhung des Untersuchungsaufwandes

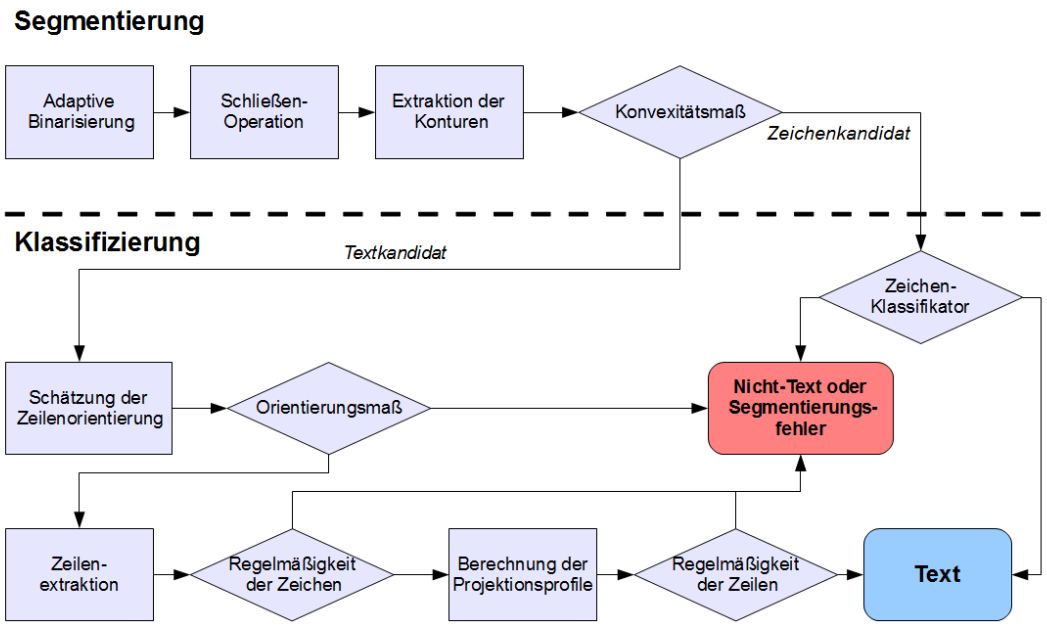


*Abb. 5.1.4: Schnelle Textdetektion, Flussdiagramm.*

für Textkandidaten. Ausschlaggebend für die Textdetektion sind die im Abschnitt 5.1.1 definierten Klassifizierungskriterien für Textstellen.

Da die Genauigkeit der Binarisierung für die Textdetektion im Initialzustand des Systems nicht kritisch ist, kann die Binarisierung des Bildes gleich zu Beginn der Verarbeitung vorgenommen werden. Angesichts der angenommenen lokalen Bimodalität der Pixelverteilung wurde ein einfaches lokales Schwellenwertverfahren (s. Abschnitt 2.7) für die grobkörnige Binarisierung der Aufnahme ausgewählt. Die Fenstergröße $n$ wird dabei in Abhängigkeit von der kleinsten zu detektierenden Schriftgröße $h_s^{min}$(s. Abschnitt 3.5) und dem Binning-Faktor $k_{binning}$ gesetzt:

$$n = 2 * (h_s^{min} / k_{binning})$$

Damit soll erreicht werden, dass vor allem kleine Elemente korrekt binarisiert werden, selbst wenn große homogene Vordergrundflächen $> n \times n$ dabei ausgehöhlt werden (s. Abb. 5.1.6 unten links).

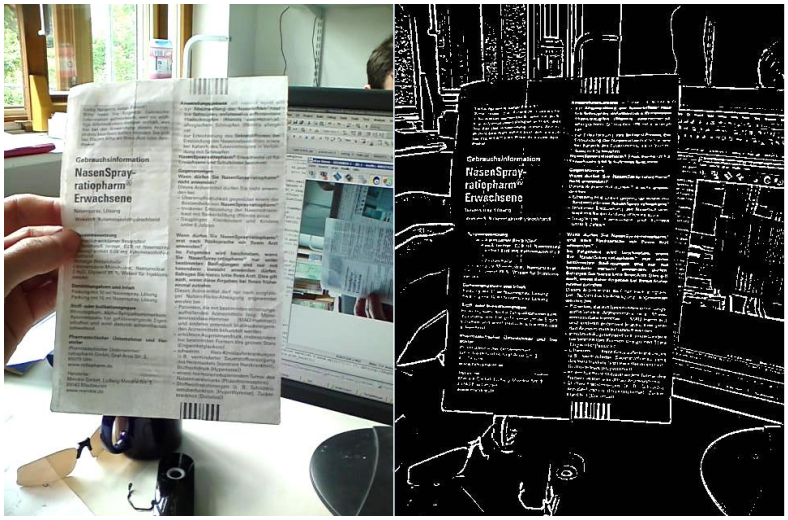


*Abb. 5.1.5: Das Binarisierungsergebnis. Zu beachten ist insbesondere das „Verkleben" der Buchstaben als Folge der Binning-Operation.*

Als Folge der Herunterskalierung werden Lücken $\Delta_s < k_{binning}$ zwischen den Buchstaben im Laufe der Binarisierung überbrückt. Das führt dazu, dass einzelne Wörter oder gar Zeilen zu durchgehenden Streifen verschmelzen (s. Abb. 5.1.5) und die Bedingung 3b für Textstellen (s. Abschnitt 5.1.1) nicht mehr erfüllt ist. Größere Zeichen sind unterdessen davon nicht betroffen und können u. U. trotz der schlechteren Qualität richtig klassifiziert werden. Um die beiden Fälle abdecken zu können, wurde gegen eine einheitliche Multiskalenmethode zugunsten eines kombinierten Ansatzes nach dem Vorbild von [13] mit drei unterschiedlichen Strategien für große ($25px < h_s < 40px$), mittelgroße ($15px < h_s < 25px$) und kleine Zeichen ($11px < h_s < 15px$) entschieden. Eine Textregion liegt vor, falls es sich dabei um:

- ein einzelnes Zeichen
- eine Textzeile mit einer regelmäßigen Zeichenstruktur
- einen Textabschnitt mit einer regelmäßigen Zeilenstruktur

handelt. Die Identifikaion von Einzelzeichen erfolgt mit Hilfe des in der Arbeit [100] vorgestellten Klassifikators auf Basis von Entscheidungsbäumen, während die Erkennung von Textzeilen/-abschnitten unter Bewertung der Regelmäßigkeit von Zeichen bzw. Zeilen stattfindet. Damit die Klassifikation richtig funktioniert, muss jedes Bildsegment höchstens eine Textregion mit homogenen Eigenschaften beinhalten. Das Segmentierungsverhalten des Algorithmus wird über die Größe des strukturierenden Elementes gesteuert, das für die Schließen-Operation eingesetzt wird. Als Richtwerte dient an dieser Stelle die minimale Laufweite der großen Zeichen (1/4 der Zeichenhöhe, s. Abschnitt 5.1.1) sowie der maximale Zeilenabstand in Textbereichen mit kleinen Schriften.

Die Schließen-Operation bewirkt, dass Bildsegmente, die je nach Schriftgröße aus einzelnen Zeichen, Zeilen oder Textabschnitten bestehen, als Zusammenhangkomponenten extrahiert werden können (s. Abb. 5.1.6). Die Extraktion erfolgt mittels einer Konturverfolgung [105], welche die Grenzen der Segmente in Form eines Polynoms liefert. Ein Konvexitätsmaß wird verwendet, um die extrahierten Segmente als Text-Kandidaten, Zeichen-Kandidaten und Nicht-Text-Bereiche zu klassifizieren

$$Convexity(S) = A(S)/A\big(H(S)\big),$$

wobei $A$ – die Flächenfunktion und $H$ – die konvexe Hülle eines Segments $S$ sind. Dahinter steht die Beobachtung, dass eine ausgeprägte Nicht-Konvexität des umschließenden Polygons auf Segmentierungsfehler hindeutet, wobei zwei Segmente durch eine wenige Pixel breite Verbindung überbrückt werden. Die Regionen mit einer kleinen Konvexitätsbewertung werden in Abhängigkeit von der Fläche entweder mit dem Zeichen-Klassifikator weiter untersucht oder gleich verworfen. Die notwendige Mindestgröße von Zusammenhangskomponenten begrenzt implizit die Anzahl von Zeichenkandidaten in einer Aufnahme und somit die Laufzeit der Zeichendetektion. Etwas aufwändiger ist die Klassifikation von Text-Kandidaten, die die Konvexitätsbedingung erfüllen, wobei es um die Bewertung der Regelmäßigkeit von Mustern in den Regionen geht.


*Abb. 5.1.6: Bildsegmentierung durch morphologische Schließen-Operation. Aushöhlung der großen Flächen infolge der festen Kernelgröße ist unten rechts zu sehen.*

### 5.1.4 Klassifikation der Segmente

Wie bereits erwähnt, ist die Zeichenregelmäßigkeit in lesbaren Textregionen infolge der Herunterskalierung der Aufnahme nicht immer gegeben. Alleinstehende Zeilen können nur dann identifiziert werden, wenn die Zeichenregelmäßigkeit feststellbar ist, während Textregionen mit mehreren Zeilen auch über die Zeilenregelmäßigkeit zu erkennen sind. Die Klassifikation der Text-Kandidaten erfolgt daher in drei Schritten:

1. Extraktion einer Textzeile und Schätzung ihrer Schriftgröße
2. Bewertung der Regelmäßigkeit der Zeichen in der extrahierten Zeile
3. Bewertung der Regelmäßigkeit der Zeilen

Eine Region kann entweder aus wenigen Zeilen mit einer hohen Bewertung der Zeichenregelmäßigkeit oder aus mehreren Zeilen bestehen, die ein regelmäßiges Muster erzeugen. Die Schwierigkeit der Aufgabe besteht darin, die Robustheit der Bewertung trotz einer eventuell vorhandenen Krümmung der Zeilen zu gewährleisten.

Damit die Extraktion der Textzeilen funktioniert, wird eine grobe Schätzung der Zeilenorientierung im Block benötigt. Die Untersuchung des Zeilenverlaufs findet unter Verwendung der MURs von den Regionskonturen statt (s. Abb. 5.1.7) und wird im Anschluss an diesen Abschnitt vorgestellt. Der Ablauf des Zeilen-Tracking-Algorithmus wird im Pseudocode 5.1.1 präsentiert. Angefangen mit einem Startpunkt $p(r, \alpha)$ an der Seite des extrahierten MURs der Region wird der

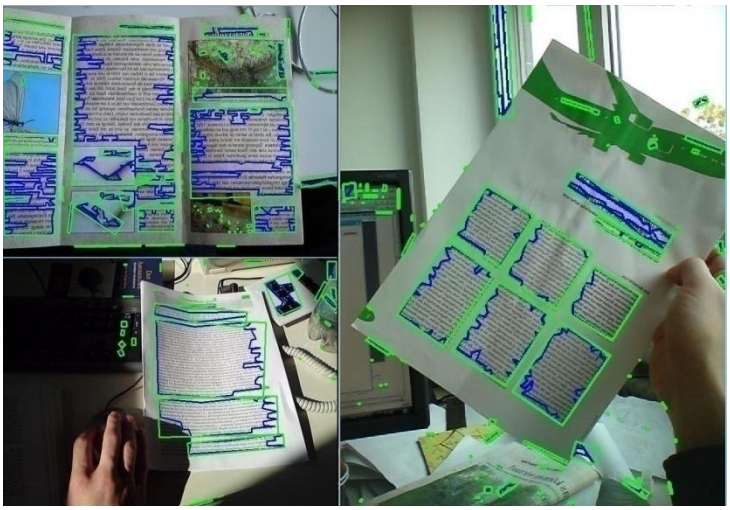

*Abb. 5.1.7: Ergebnis der Minimum-Umfang-Rechteck-Methode zur schnellen Bestimmung der Textorientierung.*

Bereich in die ermittelte Richtung der Textzeilen $\alpha$ durchlaufen, wobei die Zeichen in Form von Zusammenhangskomponenten extrahiert werden.

```
FUNCTION extractLine(startPoint, α , searchWindowSize)
// schriftgrößenabhängige Fenstergröße
        point =startPoint
        WHILE point ∈ region
                // durchsuche die Nachbarschaft nach Vordergrundpixeln
                contourPoint ≔ search(startPoint, α, searchWindowSize)
                // extrahiere die Zusammenhangskomponente, berechne Merkmale
                [x_k, σ_k^α, σ_k^(α+90°)] ≔ extractSym(contourPoint)
                // aktualisiere Suchfenstergröße
                searchWindowSize ≔ rect(2 ∗ σ_k^α, 2 ∗ σ_k^(α+90°))
                // berechne eine Prädiktion für die nächste Iteration
                point ≔ ( x_k+2 ∗ σ_k^α, y_k(x_k+σ_k^α) )
        END
END
```

*Pseudocode 5.1.1:Zeilen-Tracing-Algorithmus*

Die Auswahl des Initialwertes für die Suchfenstergröße erfolgt anhand der ersten gefundenen Zusammenhangskomponente in der Umgebung von $p(r, \alpha)$. Nach jeder extrahierten Komponente wird die Größe des Suchfensters $dx$ korrigiert und eine Voraussage für die Koordinaten des nächsten Zeichens $(x_n, y_n)$ gemacht. Die Prädiktion der Zeichenposition kann durch eine Extrapolation der Linienpunkte mit Hilfe von Lagrange Polynomen beliebiger Ordnung erfolgen. Aus Zeitgründen wird in der aktuellen Implementierung jedoch eine Approximation erster verwendet

$$y_n = y_{n-2} + \frac{(x_{n-1} + \Delta x) - x_{n-2}}{x_{n-1} - x_{n-2}} * (y_{n-1} - y_{n-2}),$$

zumal das Überspringen auf eine der benachbarten Zeile für die Bewertung der Regelmäßigkeit i. d. R. unproblematisch ist. Die Schrittweite $\Delta x$ wird dabei in Abhängigkeit von der gemittelten Zeichengröße aus den $k$ vorangegangenen Iterationen gewählt: $\sim 1/k \sum_{k=n-1}^{n-k} 2 * \sigma_k^{\alpha}$.

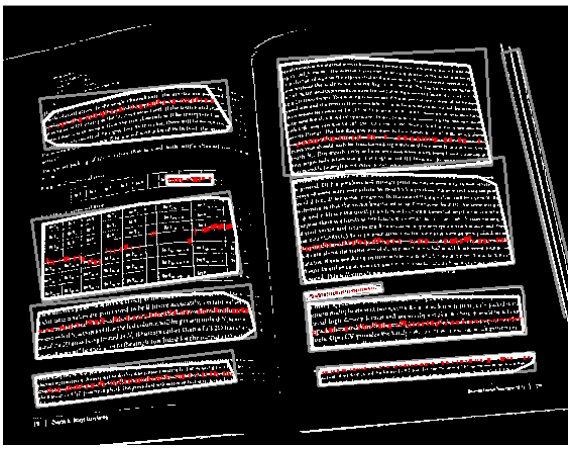

*Abb. 5.1.8: Linienextraktion. Die durchsuchten Bereiche sind rot markiert.*

Bei der Extraktion der einzelnen Zusammenhangskomponenten mittels des Konturverfolgungsalgorithmus [105], werden die Mittelpunkte und die Größen der vermeintlichen Zeichen als Streuung der Koordinaten $(x_{k,n}, y_{k,n})$ von den Konturpunkten in die Richtung des lokalen Zeilenverlaufs $\alpha$ sowie $\alpha + 90°$ geschätzt:

$$\sigma_k^{\alpha} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \tilde{x}_{k,n}{}^2}; \qquad \sigma_k^{\alpha+90°} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \tilde{y}_{k,n}{}^2}$$

Die Abweichungen in die Richtungen $\alpha$ und $\alpha + 90°$ sind als

$$\tilde{x}_{k,n} = (x_{k,n} - \bar{x}_k) * cos(\alpha) - (y_{k,n} - \bar{y}_k) * sin(\alpha)$$

und analog dazu

$$\tilde{y}_{k,n} = (x_{k,n} - \bar{x}_k) * sin(\alpha) + (y_{k,n} - \bar{y}_k) * cos(\alpha),$$

wobei

$$\bar{x}_k = \frac{1}{N}\sum_{n=1}^{N} x_{k,n}, \quad \bar{y}_k = \frac{1}{N}\sum_{n=1}^{N} y_{k,n}$$

zu berechnen. Während der Extraktion werden die Konturpunkte markiert, um eine Wiederentdeckung der Zusammenhangskomponente zu vermeiden. Das Tracking der Zeile wird abgebrochen sobald die gegenüberliegende Seite des MURs erreicht wird (s. Abb. 5.1.8).

Die bei der Zeilenverfolgung berechneten Merkmale werden zur Beurteilung der

Zeichenregelmäßigkeit in der Zeile verwendet. Anhand der ermittelten Höhen $h_s(k) = \sigma_k^{\alpha+90^\circ}$ der vermeintlichen Buchstaben wird als erstes die Schriftgröße $E(h_s)$ eingeschätzt und die normierte Varianz der Größen berechnet:

$$VarKoef(h_s) = \sqrt{Var(h_s)} \Big/ E(h_s)$$

In Übereinstimmung mit dem Klassifizierungskriterium 3a (s. Abschnitt 5.1.1) muss der Variationskoeffizient der Zeichenhöhen in einer Textzeile < 2 sein. Analog wird die Bedingung 3b anhand der Zeichenabstände

$$\Delta_s(k) = \|\bar{x}_k - \bar{x}_{k-1}, \bar{y}_k - \bar{y}_{k-1}\| - (\sigma_k^\alpha + \sigma_{k-1}^\alpha)/2$$

überprüft.

Falls aufgrund der Ausmaße der Region nicht davon ausgegangen werden kann, dass das untersuchte Segment drei weitere Zeilen enthält, dann wird die Gesamtregion direkt als Text- bzw. Nicht-Text-Segment klassifiziert, andernfalls wird die Entscheidung von der Bewertung der Zeilenregelmäßigkeit abhängig gemacht, die auf Projektionsprofilen basiert. Die Projektionsrichtung wird dabei von der extrahierten Zeile vorgegeben, sodass das Verfahren auch im Falle einer ausgeprägten Krümmung der Dokumentoberfläche zuverlässig funktioniert. Das resultierende Histogramm beschreibt die Verteilung der Vordergrundpixel in Abhängigkeit von dem Abstand zur Referenzzeile, wobei regelmäßig organisierte Textzeilen wellenförmige Profil-Muster ergeben (s. Abb. 5.1.9). Unter der Annahme, dass die Periodizität der Maxima/Minima der Histogrammwerte mit den Zeilenabständen in der Region korrespondiert, lässt sich die Zeilenregelmäßigkeit über das Signal-Rausch-Verhältnis (engl. *signal-to-noise-ratio (SNR)*) des Profils $P$ definieren. Die SNR-Berechnung wird mit Hilfe der Autokorrelationsfunktion $Corr_P(j)$ bestimmt, wo

$$Corr_P(j) = \frac{1}{N-j} \sum_{i=1}^{N-j} P(i)P(i+j)\,,$$

sodass

$$SNR(P) = \frac{\max_j(Corr_P(j)) - \min_j(Corr_P(j))}{\max_j(Corr_P(j)) + \min_j(Corr_P(j))},$$

wobei $P(i), 1 \leq i \leq N -$ Histogrammwerte und $j$ – Zeitverschiebung, die abhängig von der ermittelten Schriftgröße $h_s$ gewählt wird: $j \in [h_s, 4 * h_s]$. Der Koeffizient SNR gibt an, wie stark der regelmäßige Anteil des vermeintlichen Zeilenmusters in dem Histogramm ausgeprägt ist und wird für die endgültige Klassifizierung der Region eingesetzt.

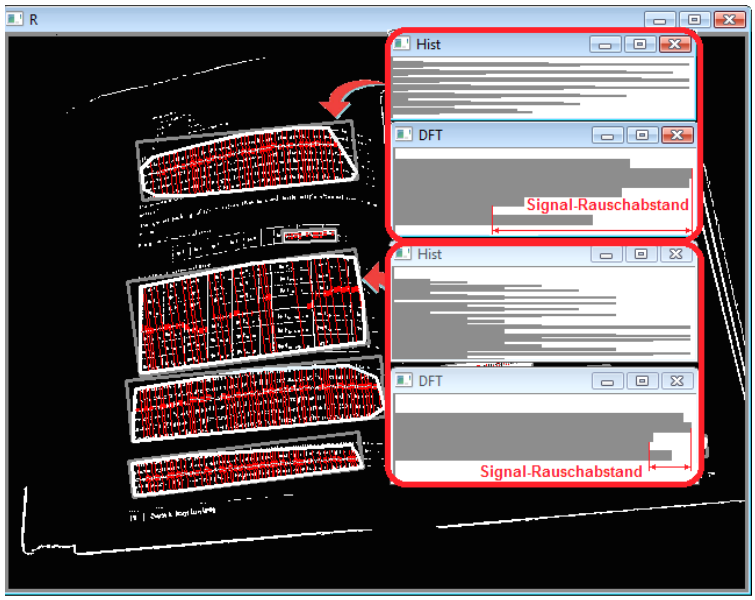


*Abb. 5.1.9:Messung der Zeilenregelmäßigkeit einer Region anhand der Integralprojektionen entlang der extrahierten Zeile. Rot markiert sind die extrahierten Zeilen sowie die Projektionslinien.*

### 5.1.5 Schätzung der Zeilenorientierung

Eine grobe Schätzung der Zeilenorientierung in dem untersuchten Textblock ist notwendig für ein zuverlässige Zeilenverfolgung als Teil der schnellen Textdetektion (s. Abschnitt 5.1.4) und der sorgfältigen Textlokalisierung. In den beiden Fällen wird eine Methode eingesetzt, die sich die geometrischen Eigenschaften von Textblöcken zunutze macht:

1. Bestimmung des Rotationswinkels des Textblocks mit Hilfe des Rotating-Calipers-Algorithmus [106], wobei angenommen wird, dass die Zeilen entlang eines der beiden Seitenpaare des MURs minimalen Umfangs verlaufen

2. Eindeutige Festlegung der Zeilenorientierung anhand von Projektionsprofilen

Die Schwierigkeit der Aufgabe besteht hier darin, dass trotz einer möglichen Krümmung der Zeilen ein einheitlicher Orientierungswert festgelegt werden muss.

Um zu demonstrieren, wie die Orientierung des minimalen MURs mit der Orientierung des Textblocks zusammenhängt, ist folgende Überlegung hilfreich. Sei $P_{ges}$ – der Umfang des MURs. Die Kontur des Rechtecks wird durch die Berührungspunkte mit der konvexen Kontur der Region in Segmente aufgeteilt (s. Abb. 5.1.10):

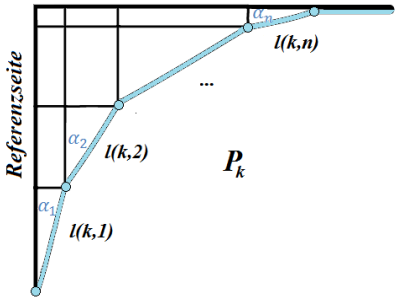


*Abb. 5.1.10: Ecke k des umschließenden Polygons*

$$P_{ges} = \sum_{i=1}^{m} P_k$$

Wird eines dieser Segmente $k$ zusammen mit den zugehörigen Polygonkanten $l_i, i \in C_k$ und Winkeln $\alpha_i$ zwischen den Kanten und einer Referenzseite des MURs betrachtet, so ergibt sich folgender Zusammenhang*:

$$P_k = \sum_{i \in C_k} l_i * (\sin\alpha_i + \cos\alpha_i) = \sum_{i \in C_k} l_i * \left(\sin\alpha_i + \sin\left(\alpha_i + \frac{\pi}{2}\right)\right)$$
$$= \sum_{i \in C_k} 2 * \sin\frac{2\alpha_i + \frac{\pi}{2}}{2} * \cos\frac{\pi}{4} * l_i$$
$$= \sum_{i \in C_k} \sqrt[2]{2} * \sin\left(\alpha_i + \frac{\pi}{4}\right) * l_i$$

Der Gesamtumfang ist somit

* Es wird o. b. d. A. angenommen, dass $\alpha \leq \pi/2$ ist. Da die Summe der Außenwinkel eines konvexen Polygons = 360° ist, kann es maximal drei $\alpha > \pi/2$ im gesamten Polygon geben und die entsprechenden Ecken würden auf den Seiten des minimalen Rechtecks liegen.

$$P_{ges} = \sum_{i \in C_{ges}} \sqrt[2]{2} * \sin\left(\alpha'_i + \frac{\pi}{4}\right) * l_i \quad \text{, wo } \alpha'_i = \begin{cases} |\alpha_i|, & |\alpha_i| \leq 90° \\ 180° - |\alpha_i|, & |\alpha_i| > 90° \end{cases}.$$

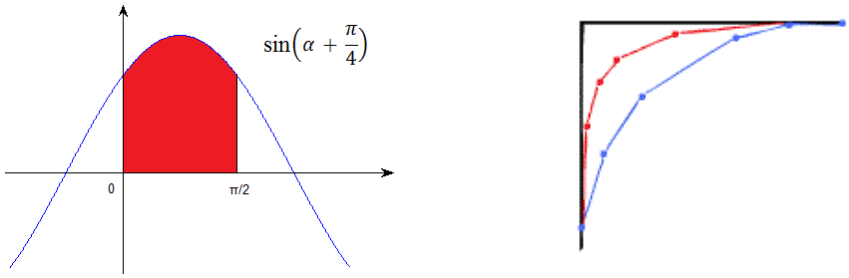


*Abb. 5.1.11: (links) Funktion $sin(\alpha + \frac{\pi}{4})$ hat innerhalb von [0, $\frac{\pi}{2}$ ]ihre Minima auf dem Rand. (rechts) Das Ausfüllen der Ecken des umschließenden Rechtecks mit kurzen Polygonkanten.*

Laut der zuvor aufgeführten Formel ist der Umfang des umschließenden Rechtecks genau dann minimal, wenn das akkumulierte Produkt $\sin(\alpha'_i + \pi/4) * l_i$ am kleinsten ist. Während $l_i$ unabhängig von der Orientierung des umschließenden Rechtecks ist, hat $\sin(\alpha'_i + \pi/4)$ sein Minimum entweder bei 0 oder bei $\pi/2$ (s. Abb. 5.1.11 links). Das impliziert, dass lange Polygonkanten entlang der Seiten des minimalen MUR verlaufen, während kurze Kanten aus dem Eckenbereich eines Textblocks, die Ecken des Rechtecks ausfüllen (s. Abb. 5.1.11 (rechts)). Je mehr die Form der Kontur einem Rechteck ähnelt, desto zuverlässiger funktioniert die Methode. Die Orientierung von Blöcken, die einzelne oder wenige Textzeilen enthalten, wird i. d. R. ebenfalls sicher ermittelt, selbst wenn diese verzerrt sind (s. Testaufnahmen im Anhang B). Als Ausrichtung des Blocks gilt in dem Fall die Orientierung der Geraden, die die beiden Endpunkte der Textzeile verbindet.

Auch wenn der Rotationswinkel $\alpha$ des untersuchten Textblocks erfolgreich bestimmt wurde, bleibt die Orientierung der Zeilen immer noch zweideutig: Diese kann dem Rotationswinkel $\alpha$ (horizontale Ausrichtung) oder $\alpha + 90°$ (vertikale Ausrichtung) entsprechen. Sei $I(i,j)$ ein Bildausschnitt, dessen Koordinatenachsen mit den Seiten des Rechtecks der Größe $N \times M$ übereinstimmen. Um die Orientierung der Zeilen eindeutig bestimmen zu können werden die horizontale und die vertikale Projektionsprofile der Region berechnet als:

$$P_h(j) = \sum_{i=1}^{N} I(i,j) \qquad P_v(i) = \sum_{j=1}^{M} I(i,j)$$

Die Orientierungsentscheidung wird anhand der Ableitungen der beiden Histogramme getroffen

$$score(P_h) = \sum_{j=1}^{M-1} \frac{|(\bar{P}_h(j))'|}{M} = \sum_{j=1}^{M-1} \frac{|\bar{P}_h(j+1) - \bar{P}_h(j)|}{M}$$

und

$$score(P_v) = \sum_{i=1}^{N-1} \frac{|(\bar{P}_v(i))'|}{N} = \sum_{i=1}^{N-1} \frac{|\bar{P}_v(i) - \bar{P}_v(i+1)|}{N}.$$

Hier bezeichnet $\bar{P}$ eine geglättete Versionen von $P$. Anhand eines Verhältnismaßes der beiden Bewertungen

$$orientation(I) = {score(P_h)}\Big/{score(P_v)}$$

lässt sich schließlich die Orientierung der Zeilen ableiten. Die Überlegung dahinter ist die, dass die Projektion in die Richtung der Textzeilen bei mehreren Zeilen größere Gefälle aufweist als die quer über die Zeilen laufende Projektion, was darauf zuführen ist, dass die Lücken zwischen den Zeichen verschiedener Zeilen meistens nicht überlappend sind (s. Abb. 5.1.12). Die Zuverlässigkeit der Unterscheidung wird in der Praxis von einer ganzen Reihe von Faktoren beeinflusst,


*Abb. 5.1.12: Projektionsprofile in Richtung der Seiten des Rechtecks zur Bestimmung der Textorientierung.*

darunter der Fehler bei der Einschätzung der Orientierung des Textblocks, die Dokumentverzerrung, die Form des Textblocks sowie die Dichte der Vordergrundpixel. Die Auswirkungen der Faktoren auf die Robustheit der Unterscheidung können mit Hilfe der Radon-Transformation visualisiert werden, die das

Projektionskonzept in einer anschaulichen Form darstellt. Bei der Radon-Transformation $R$ werden die Werte, ähnlich wie bei der Berechnung die Projektionsprofile, entlang einer Geraden $L = \{(x,y)|\, x * cos(\alpha) + y * sin(\alpha) - r = 0\}$ integriert:

$$RI(r,\alpha) = \int I\big(r * cos(\alpha) - s * sin(\alpha), r * sin(\alpha) + s * cos(\alpha)\big)\, ds$$

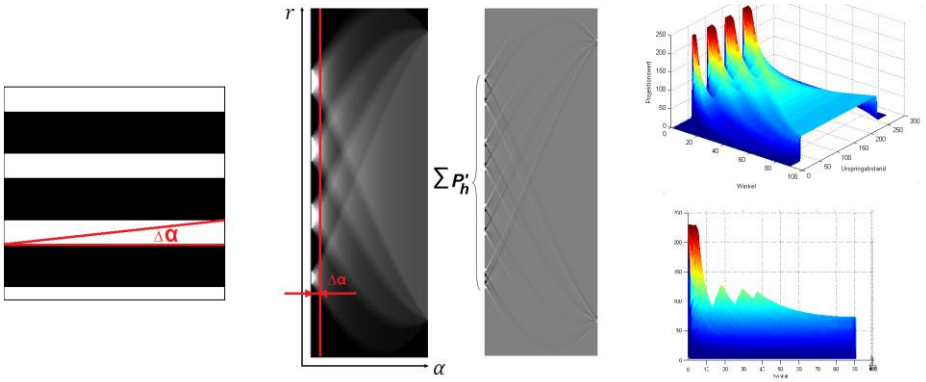


*Abb. 5.1.13: (links)Textblockmodell, Zeile-Leerraum-Abfolge, (mittig) Radon-Transformation des Textblockmodells und das Kantenbild der Radon-Transformation (rechts) 3D-Darstellung der Radon-Transformierten..*

In Abb. 5.1.13 sind ein einfaches Textblockmodell, die zugehörige Radon-Transformation ([0°, 90°]-Ausschnitt) sowie ihr Kantenbild dargestellt. Auffällig sind die sinusförmigen Linien in dem Kantenbild, wobei es sich um die Transformierten der Geraden $L'$ vom Rand der modellierten Textlinien handelt:

$$L'(x,y) = \delta(r' - x * \cos(\alpha') - y * \sin(\alpha'))$$

Hier bezeichnet $\delta$ die Delta-Funktion. Für eine Steigung $\alpha$ der Projektionsgeraden werden die Punkte der Linie $L'(x,y)$ wie folgt transformiert [107]*:

* Die Untersuchung der diskreten Radon-Transformation ist aufgrund von vielen zusätzlichen Parametern wie Diskretisierungsschrittweiten, Interpolationsmethode, Grenzen der Eingabefunktion wesentlich aufwändiger [107].

$$RI(r,\alpha) = \int \delta\Big(r' - \big(r * cos(\alpha) - s * sin(\alpha)\big) * cos(\alpha') - \big(r * sin(\alpha) + s * cos(\alpha)\big) * sin(\alpha')\Big)\, ds$$

$$= \int \frac{1}{|sin(\alpha - \alpha')|} * \delta\left(\frac{r' - r \cdot cos(\alpha - \alpha')}{sin(\alpha - \alpha')} + s\right) ds$$

$$= \begin{cases} \dfrac{1}{|sin(\alpha - \alpha')|}\ , & wenn\ \ sin(\alpha - \alpha') \neq 0 \\ \displaystyle\int \delta(r' - r)ds\ , & wenn\ \ sin(\alpha - \alpha') = 0 \end{cases}$$

Die Kantenstärke erreicht das Maximum wenn die Projektionsrichtung $\alpha = \alpha'$ ist, und fällt $\sim \frac{1}{|sin(\alpha - \alpha')|}$ ab, wobei $\alpha - \alpha'$ als Orientierungsfehler anzusehen ist. Somit ist der Orientierungskoeffizient:

$$orientation(I) \sim \frac{\frac{1}{|sin(\alpha - \alpha')|}}{\frac{1}{|sin(\alpha - \alpha' + 90°)|}} = cot(\alpha - \alpha')$$

Wie aus Abb. 5.1.14 ersichtlich ist, unterscheiden sich die Radon-Transformierten

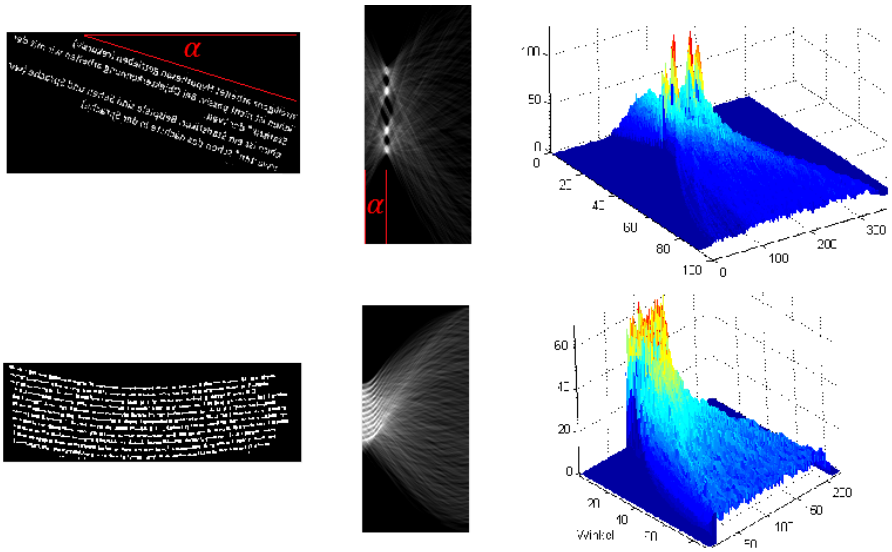


*Abb. 5.1.14:Radon-Transformation, Beispiele von realen Textaufnahmen.*

realer Textausschnitte nicht wesentlich von der des Modells, insbesondere wenn

die Projektionsprofile vor der Auswertung geglättet werden. Bei einer nichtlinearen Verzerrung der Zeilen wird ein Transformationsbild produziert, wie es bei einer Überlagerung von unterschiedlich ausgerichteten geraden Textlinien entstehen würde.

Der berechnete Koeffizient $orientation(I)$ wird nicht nur zur Feststellung der Zeilenorientierung im Block verwendet, sondern dient auch als Indikator von Nicht-Text-Regionen und Segmentierungsfehlern. Ist der Wert von $|orientation(I) - 1|$ kleiner als ein Schwellenwert, dann lassen sich keine zuverlässigen Aussagen bzgl. der Zeilenausrichtung im Block machen und die Region wird verworfen.

### 5.1.6 Auswertung und Zusammenfassung

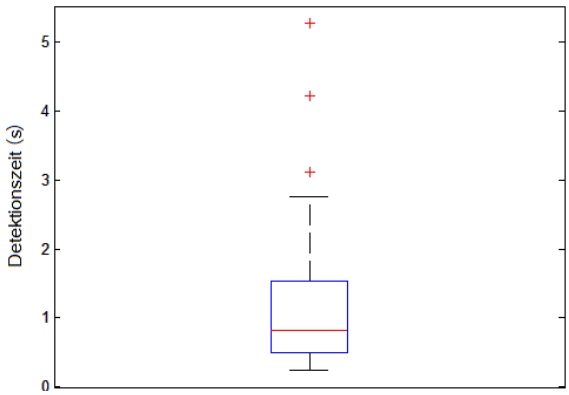


*Abb. 5.1.15: Reaktionszeiten des Systems als Boxplot-Diagramm.*

Das vorgestellte Textdetektionsverfahren weist eine ausgeprägte Pipeline-Struktur (s. Abb. 5.1.4) auf, wobei eine immer kleiner werdende Gruppe von Text-Kandidaten einer immer aufwändigeren Analyse unterzogen wird. Abhängig davon, wie viele Segmente bereits in frühen Stadien der Verarbeitung verworfen werden, kann die Laufzeit der Prozedur stark variieren. Um den spezifizierten Richtwert für die Verarbeitungsrate von 12 fps einhalten zu können, wird in der Praxis eine probabilistische Modifikation des Algorithmus eingesetzt: Anstatt alle Zeichenkandidaten zu klassifizieren, wird eine zufällig ausgewählte Teilmenge der Regionen aus verschiedenen Bildbereichen stichprobenartig analysieret. Auch die Vorgehensweise bei der Berechnung der Projektionsprofile hat probabilistische Züge (s. Abschnitt 5.1.4). Dieser randomisierter Ansatz ist an der Stelle dadurch gerechtfertigt, dass die Vollständigkeit des Detektionsergebnisses für die gegebene Aufgabenstellung unerheblich ist, da bereits eine einzige sicher detektierte Textregion ausreicht, um die Benachrichtigung des Benutzers auszulösen. Um die Robustheit des Algorithmus gegenüber falsch-positiven Detektionsergebnissen zu erhöhen, können Funde aus mehreren aufeinander folgenden Se-

quenzbildern kombiniert werden. Die Reaktionszeit des Systems steigt dabei proportional zu der Anzahl der zu berücksichtigenden Iterationen. Die Empfindlichkeit der Textdetektion kann mit Hilfe der graphischen Benutzeroberfläche (s. Abb. 3.6.2) angepasst werden.

Die Auswertung des Algorithmus (s. Anhang B) fand unter realitätsnahen Bedingungen im Freien statt. Es wurde dabei die Reaktionszeit des Systems gemessen, nachdem ein Textobjekt vor der Kamera positioniert und die Textdetektion eingeleitet wurde. Bei allen 69 Durchläufen mit 23 unterschiedlichen Aufnahmeszenen verlief die Detektion erfolgreich. Der Anteil der falsch-positiven Ergebnisse in den ausgewerteten Aufnahmen (genau 100 Regionen) lag bei 3%. Die Latenzzeiten werden in Abb. 5.1.15 als Boxplot-Diagramm dargestellt, wobei die Detektionsentscheidung auf zwei Iterationen basierte. Die Median-Zeit lag mit 0.8 s unterhalb des spezifizierten Richtwerts von 1s.

## 5.2 Verfolgung von Textstellen

Nach einer erfolgten Textdetektion und der anschließenden Benachrichtigung des Benutzers verfolgt das System die entdeckten Textregionen, während es auf die Rückmeldung wartet. Die dabei gesammelten Informationen dienen dazu, die Einstellung der Kamera vorzunehmen und die notwendige Bildqualität zu gewährleisten. In diesem Abschnitt wird ein Algorithmus beschrieben, der eine robuste Verfolgung von Textstellen unter Verwendung von Distanzdaten ermöglicht.

### 5.2.1 Problemstellung

Die Verfolgung von bewegten Objekten gehört zu den klassischen Aufgaben der Videoverarbeitung. Das Ziel dabei ist es, Positionsänderungen eines Objekts zu registrieren, indem folgende zwei Teilaufgaben gelöst werden:

- *Messung*: Lokalisierung der Objekte im aktuellen Bild der Videosequenz
- *Filterung* und *Datenfusion*: Minimierung des Messfehlers unter Verwendung der Messungen aus den vorangegangenen Iterationen

Typischerweise werden die beiden Schritte abwechselnd ausgeführt, sodass am Ende eines jeden Durchlaufs die geschätzten Positionen der verfolgten Objekte

zur Verfügung stehen. Es bietet sich an, den im Abschnitt 5.1 präsentierten Textlokalisierungsalgorithmus, dessen Echtzeitfähigkeit bereits unter Beweis gestellt wurde, in der Messphase zu verwenden. Im Gegensatz zur Textdetektion reicht es an dieser Stelle nicht mehr aus, einige wenige der in einem Bild vorhandenen Textstellen zu identifizieren, sodass eine Analyse sämtlicher Textkandidaten notwendig ist. Ein besonderes Problem stellt angesichts der geringen Unterscheidbarkeit von Textstellen der Assoziationsschritt dar, wobei eine (teilweise) Verdeckungen von Regionen, perspektivische Effekte und Klassifizierungsfehler die Wiedererkennung behindern können. Tiefendaten können an der Stelle als zusätzliche Merkmale für die Unterscheidung der Textobjekte genutzt werden.

### 5.2.2 Vorarbeiten

Mehrere Autoren beschäftigen sich speziell mit dem Thema der Textverfolgung in Videosequenzen [15][108][109][110], wobei sie die Verwendung von allgemeinen Zustandsschätzern wie Kalman- oder Partikel-Filtern in Kombination mit speziell entwickelten Textdetektionsalgorithmen vorschlagen. Im Mittelpunkt der Forschung steht u. a. die Assoziation von Textobjekten, die für eine gleichzeitige Verfolgung von mehreren Textstellen notwendig ist. Ein möglicher Lösungsansatz besteht darin, transformationsinvariante Merkmale (bspw. SIFT-Merkmale engl. *scale-invariant feature transform*) [111] zu verwenden, um eine einfache Nächste-Nachbarn-Klassifikation [109] für die Wiedererkennung und Unterscheidung der Regionen durchzuführen. Aus Effizienzgründen ist jedoch eine Wiederverwendung von Klassifizierungsmerkmalen aus der Lokalisierungsphase vorzuziehen, da sie ohne erheblichen Mehraufwand eingesetzt werden können. Textspezifische Merkmale sind nicht zwingenderweise transformationsinvariant, sodass bewegungsbedingte Veränderungen der Merkmalswerte unter Verwendung des Bewegungsmodells simuliert werden müssen [108][112]. Für das Tracking im 3D-Zustandsraum bieten sich vor allem homographiebasierte (s. Abschnitt 2.5) Beobachtungsmodelle an [113][114][115], wobei angenommen wird, dass alle Merkmale des verfolgten Objekts einer Ebene angehören, die verschoben und rotiert wird. Nichtlineare Modelle sind mit einem erheblichen Mehraufwand verbunden [115] und werden nur selten eingesetzt.

Der Kalman-Filter [116] kommt dank seiner Effizienz sehr häufig in verschiedenen Echtzeit-Anwendungen zum Einsatz [117]. Der klassische Kalman-Filter ist

ein optimaler* Zustandsschätzer, der nach dem *Prädiktion-Korrektur*-Prinzip funktioniert. Eine geeignete Modellierung des Zustandsraums sowie der System-Dynamik ist entscheidend für die zuverlässige Objektverfolgung mit Kalman-Filtern.

### 5.2.3 Modellierung des Zustandsraums

Vor dem Hintergrund der möglichen stereovisionbasierten Distanzmessungen ist die Verwendung eines 3D-Zustandsraums bei der Verfolgung von Dokumenten in einer Videosequenz naheliegend. Demensprechend wird der Zustand einer Textregion $i$ zum Zeitpunkt $t$ durch einen Vektor $x_i^t = (\vec{p}_i^t, \vec{n}_i^t, \vec{v}_i^t, \vec{\omega}_i^t)$ repräsentiert, wo $\vec{p}_i^t$ – 3D-Kamerakoordinaten der Kontur-Schwerpunkte, $\vec{n}_i^t$ – Normalenvektor der Region-Ebene, $\vec{v}_i^t$–Geschwindigkeitsvektor der Translationsbewegung und $\vec{\omega}_i^t$ – 3D-Geschwindigkeitsvektor der Rotationsbewegung sind. Gegeben der Zustand $x_i^{t-1}$ einer Textregion $i$ zum Zeitpunkt $t-1$, wird eine Voraussage $\tilde{x}_i^t$ für den Zeitpunkt $t$ berechnet als

$$\tilde{x}_i^t = A_i^{t-1} x_i^{t-1} + N_{12}(0, \Sigma_i^{t-1}),$$

wobei $A_i^{t-1}$– die *Zustandsübergangsmatrix*, mit deren Hilfe die Dynamik des Systems modelliert wird, und $N(0, \Sigma_i^{t-1})$ – der modellierte Fehler sind. Aus Effizienzgründen wird bei der implementierten Tracking-Methode auf lineare Bewegungsmodelle der Form

$$\tilde{x}_i^t = (\tilde{p}_i^t, \tilde{n}_i^t, \tilde{v}_i^t, \tilde{\omega}_i^t),$$

wo

$$\tilde{p}_i^t = \vec{p}_i^{t-1} + \vec{v}_i^{t-1}, \quad \tilde{n}_i^t = \vec{n}_i^{t-1} + \vec{\omega}_i^{t-1}, \quad \tilde{v}_i^t = \vec{v}_i^{t-1} \; und \; \tilde{\omega}_i^t = \vec{\omega}_i^{t-1} \; ,$$

zurückgegriffen, wobei $\tilde{p}_i^t$ – Prädiktion der Position und $\tilde{v}_i^t, \tilde{\omega}_i^t$ – Prädiktion der Translations- und Rotationsgeschwindigkeiten. Die Modellierung des Messvorgangs erfolgt mit Hilfe des *Beobachtungsmodells* $H$:

---

* Ein Schätzer heißt optimal, wenn er die Eigenschaften *Erwartungstreue*, *Konsistenz* und *minimale Varianz* hat.

$$\tilde{m}_i^t = H\tilde{x}_i^t + N(0, \Sigma_i^{t-1})$$

Da die Position und Orientierung einer Textregion aus dem Überlappungsbereich der Stereoaufnahmen direkt gemessen werden kann, wird $\tilde{m}_i^t = (\tilde{p}_i^t, \tilde{n}_i^t)$ gesetzt. Die Zustandsschätzung $x_i^t$ erfolgt schließlich durch die Fusion der vorhergesagten $\tilde{m}_i^t$ und gemessenen Werte $m_i^t$

$$x_i^t = \tilde{x}_i^t + K_i^t(m_i^t - \tilde{m}_i^t),$$

wobei $K_i^t$ – die zugehörige Kalman-Matrix [116] ist.

### 5.2.4 Messung und Assoziation

Die Messung der gegenwärtigen Positionen und Orientierungen von Textregion-Ebenen erfolgt unter Verwendung der Ergebnisse des Textlokalisierungsalgorithmus, insbesondere der extrahierten Konturen von den erkannten Textregionen. Mindestens drei Korrespondenzpaare von Konturpunkten aus den beiden Bildern der Stereoaufnahme sind erforderlich, um die 3D-Pose einer Textebene zu bestimmen. Die Voraussetzung für eine erfolgreiche Messung der Pose einer Region ist, dass die Punkte weit genug voneinander entfernt sind, um den notwendigen Genauigkeitsgrad zu gewährleisten (s. Abschnitt 2.6).

Die Bestimmung der Stereokorrespondenzen erfolgt über die Ausrichtung der Konturen von Regionen mittels eines konventionellen Sequenzalignment-Algorithmus (engl. *sequence alignment*) [118]. Da die quadratische Laufzeit des Algorithmus, der auf der dynamischen Programmierung basiert, bei langen Kontursequenzen die Laufzeit des Verfahrens negativ beeinträchtigen könnte, findet die Ausrichtung in zwei Schritten (s. Abb. 5.2.1) statt:

a) Lokale Ausrichtung einiger Sequenzsegmente einer Aufnahme mit der korrespondierenden Sequenz aus dem zweiten Bild
b) Kombination der Ergebnisse zu einem globalen Alignment

Vor dem Alignment werden die Kontursequenzen mit Hilfe des Teh-Chin-Verfahrens [119] komprimiert. Der Algorithmus extrahiert dominante Punkte einer geschlossenen Kurve, indem er die Relevanz der Sequenzelemente in einem speziell ausgewählten Kontext (engl. *region of support (ROS)*) bewertet. Im End-

effekt wird dabei eine skalierungsabhängige Glättung der Kurve vollzogen, damit die anschließende Alignment-Operation schneller und zuverlässiger funktioniert. Dafür werden die rektifizierten Koordinaten (s. Abschnitt 2.6) der dominanten Sequenzpunkte berechnet, sodass die Ähnlichkeit zweier Sequenzelemente anhand des Höhenunterschieds $\left|\vec{p}_i.y - \vec{p}_j.y\right|$ zwischen den zugehörigen Epipolarlinien effizient bewertet werden kann. Als weiteres Ähnlichkeitsmerkmal werden die Winkel zwischen den adjazenten Kanten in die Berechnung einbezogen:

$$\alpha(\vec{p}_i) = arccos\frac{\langle \vec{p}_{i-1} - \vec{p}_i, \vec{p}_{i+1} - \vec{p}_i \rangle}{\|\vec{p}_{i-1} - \vec{p}_i\| \cdot \|\vec{p}_{i+1} - \vec{p}_i\|}$$

Die Kostenfunktion setzt sich aus dem Winkelmaß $\alpha(\vec{p}_i)$ und den Höhendifferenzwerten wie folgt zusammen

$$\delta(\vec{p}_i, \vec{p}_j) = \begin{cases} \left|\alpha(\vec{p}_i) - \alpha(\vec{p}_j)\right|, & wenn\ \left|\vec{p}_i.y - \vec{p}_j.y\right| < \varepsilon \\ 2\pi, & sonst \end{cases},$$

wobei $\varepsilon$ – ein schriftgrößenabhängiges Schwellenwert ist. Die Gap-Kosten werden gemäß folgender Rechenvorschrift basierend auf den Außenwinkeln $\hat{\alpha}(\vec{p}_j) = 180 - \alpha(\vec{p}_i)$ der adjazenten Kanten ermittelt:

$$\delta(\vec{p}_i, gap_{j1,j2}) = \begin{cases} \left|\left(\sum_{j=j1}^{j2} \alpha(\vec{p}_j)\right) - \alpha(\vec{p}_i)\right|, & wenn\ \left|\frac{\vec{p}_{j1}.y + \vec{p}_{j2}.y}{2} - \vec{p}_i.y\right| < \varepsilon \\ 2\pi, & sonst \end{cases}$$

Unter allen Konturpunktpaaren, die in den beiden Aufnahmen auf gleicher Höhe liegen, werden damit diejenigen gut bewertet, bei denen die Winkelwerte am ähnlichsten sind.

Die im ersten Schritt berechneten Alignment-Kosten für die einzelnen Kontursegmente (s. Abb. 5.2.1 b) bilden die Ähnlichkeitsmatrix des zweiten Schrittes. Das globale Alignment der Konturen findet unter Berücksichtigung der Abstände $d_{12}, d_{23}$ zwischen den gewählten Kontursegmenten statt (s. Abb. 5.2.1 c).

Durch die Ausrichtung der Konturen lassen sich mehrere Aufgaben gleichzeitig lösen: Zum einen liefert das berechnete globale Alignment die für die Bestim-

mung der Pose einer Textregion-Ebene im 3D-Raum notwendigen Stereokorrespondenzpaare und zum anderen werden die normalisierten Alignment-Kosten als Ähnlichkeitsmaß für die Assoziation der Regionen verwendet. Die Suche nach korrespondierenden Konturpolygonen findet stufenweise statt. Als erstes werden die erwarteten Positionen des Mittelpunkts in das Bildkoordinatensystem projiziert, um den Kreis der Kandidaten einzugrenzen. Das Kontur-Alignment wird für diejenigen Regionen durchgeführt, die von der Lage und von der Größe her zu den erwarteten Werten passen. Bei der Validierung des Assoziationsergebnisses werden sowohl die Alignment-Kosten als auch die 3D-Pose der Regionen berücksichtigt.

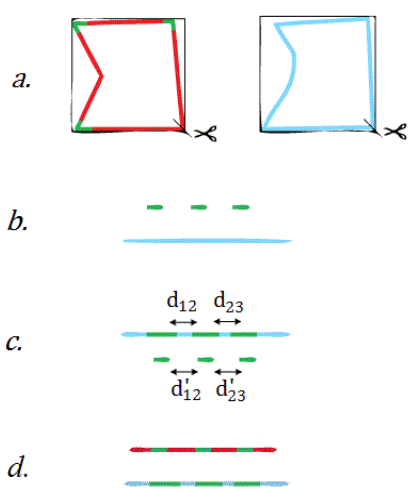


*Abb. 5.2.1: Zwei-Schritt-Alignment der Kontursequenzen.*

Für den Einsatz des 3D-Zustandsmodells spricht nicht zuletzt die Tatsache, dass eine Unterscheidung zwischen Textstellen, die zu verschiedenen Dokumenten gehören, aufgrund der höheren Dimensionalität des Zustandsraums zuverlässiger funktionieren könnte als mit reinen 2D-Bilddaten. Im Falle von Regionen aus den nicht-überlappenden Bereichen der Stereoaufnahmen sind jedoch keine Tiefenmessungen möglich, sodass für diese ein alternatives Messverfahren angewandt wird, welches auf der Bewegungsanalyse (engl. *Structure from Motion (SfM)*) basiert. Anstelle von Stereobildern werden markante Stellen zweier assoziierten Kontursequenzen aus aufeinander folgenden Videoframes verwendet. Mit Hilfe von vier korrespondierenden Punktpaaren lässt sich die homographische Abbildung zwischen den als Referenz gewählten bekannten 3D-Koordinaten einer Kontursequenz und den 2D-Bildkoordinaten aus der aktuellen Messung berechnen und damit die aktualisierte Pose der Regionsebene schätzen. Eine homographische Transformation $H$ beschreibt eine Komposition beliebig vieler Rotationen, Translationen und Projektionen und lässt sich mit $n \geq 4$ Messungen ermitteln (s. Abschnitt 2.5):

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = (H_1 \quad H_2 \quad H_3)$$

Da die Bewegung einer Textregion als Translation und Rotation einer Ebene im 3D-Raum modelliert wird, lassen sich die beiden Bewegungskomponenten über eine Zerlegung der homographischen Abbildung $H$ ermitteln. Angenommen, dass die Punkte der Referenz-Kontur in einer Ebene liegen, der Ursprung des Koordinatensystems mit dem Mittelpunkt der Region übereinstimmt und die z-Achse in die Richtung der Normalen der Ebene zeigt. Sei ferner $X, Y$ – die Bildkoordinaten eines Konturpunktes in der aktuellen Messung und $\lambda C^{-1}$ – die mit einem Faktor $\lambda$ skalierte Kameramatrix. Dann besteht folgender Zusammenhang zwischen $(X, Y)$ und dem korrespondierenden Punkt in der Referenzkontur $\vec{p}_i^{\,ref} = (x, y, 0)$

$$\lambda C^{-1} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = (R_1 \quad R_2 \quad R_3 \quad T) \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = (R_1 \quad R_2 \quad T) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = (H_1 \quad H_2 \quad H_3) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix},$$

wo $R_1, R_2, R_3$-Spalten der Rotationsmatrix, $T$ –Translationsvektor und $H_1, H_2, H_3$ – Spalten der Homographie-Matrix sind. Wegen der angenommenen Orthogonalität der Rotationsmatrix lassen sich die beiden gesuchten Bewegungskomponenten wie folgt berechnen [115]:

$$R_1 = {H_1}/{\|H_1\|}, \qquad R_2 = {H_2}/{\|H_2\|}, \qquad R_3 = R_1 \times R_2$$

$$T = {2H_3}/{(\|H_1\| + \|H_2\|)}$$

Damit kann die Messung $m_i^t = (\vec{p}_i^{\,t}, \vec{n}_i^{\,t})$ zu einem Zeitpunkt $t$ geschätzt werden als:

$$\vec{p}_i^{\,t} = \vec{p}_i^{\,ref} + T, \qquad \vec{n}_i^{\,t} = (R_1 \quad R_2 \quad R_3) \cdot \vec{n}_i^{\,ref}$$

Analog zu $\vec{p}_i^{\,ref}$ wird mit $\vec{n}_i^{\,ref}$ der Normalenvektor von der Referenzebene notiert. Nach jeder erfolgreich durchgeführten Messung der 3D-Pose einer Ebene werden die Referenzwerte aktualisiert und außerdem die zugehörige Kontursequenz zu-

sammen mit den Korrespondenzinformationen gespeichert, sodass die Assoziation über mehrere Iterationen hinweg funktionieren kann.

### 5.2.5 Zusammenfassung

Der vorgestellte Algorithmus ermöglicht das Tracking von Textregionen anhand von Stereoaufnahmen. Dank dem eingesetzten 3D-Zustandsmodell kann der Benutzer Orientierungshinweise erhalten, die auf der Grundlage von Tiefenmessungen basieren, sodass eine geeignete Ausrichtung der Kamera auf das Dokument erzielt werden kann, bei der die perspektivische Verzerrung möglichst gering ist. Eine Fehlermeldung wird ausgegeben falls:

- eine Kollision mit dem Bildrand festgestellt wird
- die Distanz zum Dokument zu groß für die geschätzte Schriftgröße ist
- der Winkel zwischen der Dokumentebene und der Sensorebenen zu groß wird

Zusätzlich zu den Warnungen bietet das System Korrekturvorschläge für die entstandene Situation an.

Angesichts der probabilistischen Natur des zugrundeliegenden Textlokalisierungsalgorithmus ist es wichtig, dass die Verfolgung auch dann fortgesetzt wird, wenn die Texterkennung in einem der beiden Bilder fehlschlägt und keine aktuellen Tiefenmessungen vorliegen. Durch die zweifache Aufnahme einer Szene aus zwei unterschiedlichen Kamerapositionen sinkt die Wahrscheinlichkeit, dass ein verfolgtes Text-Objekt aufgrund von Verdeckungen oder Erkennungsfehlern nicht mehr gefunden oder identifiziert werden kann.

Der zusätzliche Rechenaufwand, der für die Assoziation und Filterung der Lokalisierungsergebnisse notwendig ist, schlägt sich spürbar auf die Framerate nieder, die bei voller Prozessorauslastung auf 5 fps zurückgeht. Schnelle Kopfbewegungen stellen somit ein Problem dar und sollten vermieden werden. Kann das System über mehrere Iterationen hinweg keine der verfolgten Textstellen mehr entdecken, dann fordert es den Benutzer auf, die ursprüngliche Kopforientierung einzunehmen und geht in den Initialzustand über. Eine ähnliche Benachrichtigung

wird ausgegeben, wenn die gemessene Translationsgeschwindigkeit $\tilde{v}_i^t$ einen heuristisch ermittelten Schwellenwert überschreitet. Nachdem der Benutzer die Verarbeitung initiiert hat, wird anhand der ermittelten Bewegungsgeschwindigkeit $\tilde{v}_i^t$ ein günstiger Auslösezeitpunkt bestimmt.

## 5.3 Fokuseinstellung

Eine sorgfältige Einstellung des Kamerafokus ist von entscheidender Bedeutung für das Ergebnis der automatischen Zeichenerkennung. In einer dynamischen und komplexen Umgebung ist es wichtig, dass eine Fokussierung auf ausgewählte Interessenbereiche (engl. *Region of Interest(ROI)*) schnell und zuverlässig funktioniert, was sich mit Hilfe von stereovisionbasierten Methoden ohne großen Aufwand realisieren lässt.

### 5.3.1 Problemstellung

Die klassische Vorgehensweise zur automatischen Fokuseinstellung aufgrund von Einzelbildern beruht auf der Optimierung einer Bewertung von Kontrasteigenschaften [121] des Bildes in Abhängigkeit vom gesetzten Fokuswert. Die Kontrastbewertung funktioniert am besten entlang von langen und scharfen Kanten, die in Dokumentaufnahmen fehlen können. Darüber hinaus können bereits kleinste Eigenbewegungen der Kamera Bewegungsartifakte bewirken, wodurch Kanten verschwimmen, sodass die Kontrastbewertung erschwert wird. Ein großes Problem stellt dementsprechend die für die Messung notwendige Wartezeit dar, in der der Benutzer den Kopf stillhalten muss. Der Zeitbedarf für den Autofokus beträgt bei der aktuellen Hardware-Konfiguration in etwa 90% der Gesamtaufnahmezeit, was insbesondere bei langen Belichtungszeiten kritisch ist.

Die eingebaute Stereokamera ermöglicht unterdessen eine direkte Messung der Abstände, auf deren Grundlage der optimale Fokuswert ermittelt werden kann. Angesichts der zu erwartenden perspektivischen Effekte und Krümmungen der Dokumentoberfläche kann eine explizite Berechnung der Fokuseinstellungen unter Berücksichtigung der Parameter des optischen Systems von Vorteil sein. Da die Distanzmessung anhand der Videosequenzbilder aus der Tracking-Phase vorgenommen wird, kann die Gesamtaufnahmezeit und damit auch die Wahrscheinlichkeit von Bewegungsartefakten erheblich reduziert werden.

### 5.3.2 Korrespondenzfindung

Die Ermittlung von Distanzwerten findet mit Hilfe der *Triangulation* statt. Voraussetzung dafür ist eine im Vorfeld durchgeführte Kalibrierung (s. Abschnitt

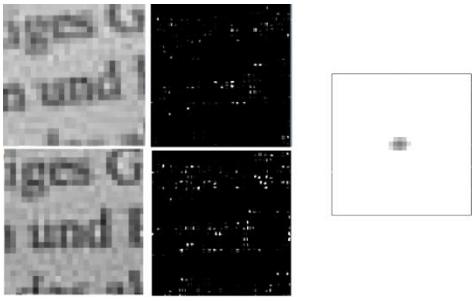

*Abb. 5.3.1: Block-Matching-Verfahren – (links) Patch, (mittig) Harris-Ecken, (rechts) Verteilung der Ähnlichkeitsbewertung .*

2.6) der Stereokamera, bei der die *intrinsischen* und *extrinsischen* Parameter der beiden Kameras berechnet werden (s. Abschnitt 2.6). Die Bestimmung der Stereokorrespondenzen in Dokumentaufnahmen ist aufgrund der geringen Unterscheidbarkeit von textspezifischen Mustern sehr fehleranfällig. Erschwerend kommt hinzu, dass angesichts der knappen Zeitvorgaben keine vollständige Rektifizierung der beiden Aufnahmen vorgenommen werden kann, sodass perspektivische Verzerrungseffekte die Korrespondenzsuche zusätzlich behindern. Die bereits verfügbaren Distanzinformationen aus der Tracking-Phase werden durch weitere Abstandsmessungen unter Verwendung eines Block-Matching-Algorithmus [26] ergänzt, der Korrespondenzpunkte anhand einer Ähnlichkeitsbewertung größerer Bildausschnitte ermittelt (s. Abb. 5.3.1). Obwohl die Block-Matching-Methode langsamer ist als der präsentierte Kontur-Alignment-Algorithmus, lässt sie sich auch für Bildbereiche einsetzen, die nicht am Rande einer Textregion liegen.

Um die Robustheit des Abgleichs zu erhöhen werden die Harris-Ecken [122] extrahiert und rektifiziert. Die Fenstergröße wird adaptiv gewählt, sodass eine gewisse Unterscheidbarkeit der Blöcke garantiert ist. Die Ähnlichkeitsbewertung für zwei Punkte $p_l = (x_l, y_l)$ und $p_r = (x_r, y_r)$ findet über die Berechnung der normalisierten quadratischen Abweichung statt, und weist somit eine hohe Robustheit [123] gegenüber möglichen Intensitätsunterschieden in den beiden Bildern $I_l, I_r$ auf:

$$E(x_l, y_l, x_r, y_r) = \frac{\sum_{x,y}(I_l(x_l + x, y_l + y) - I_r(x_r + x, y_r + y))^2}{\sqrt{\sum_{x,y} I_l(x_l + x, y_l + y)^2 \cdot \sum_{x,y} I_r(x_r + x, y_r + y)^2}}$$

Das Ähnlichkeitsmaß ist absolut, sodass auch eine Abwesenheit von korrespondierenden Punkten festgestellt werden kann.

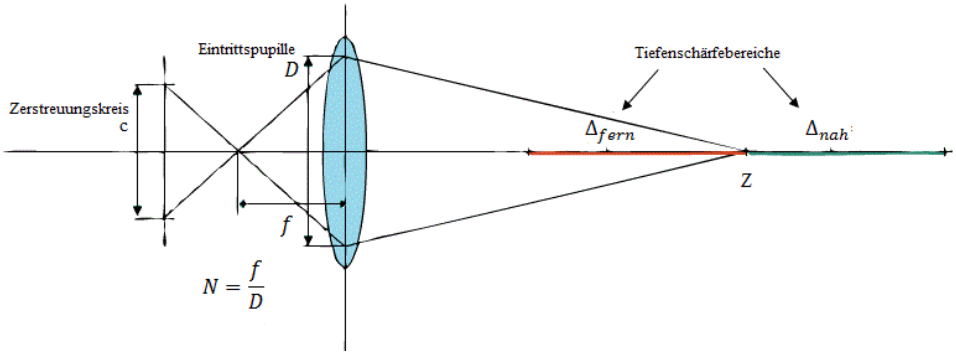


*Abb. 5.3.2:Bestimmung der Fokuseinstellungen, Linsenmodell*

Maßgeblich für die Fokuseinstellung ist die flächenmäßig größte der entdeckten Textregionen aus dem mittleren Bildbereich. Damit sich die ausgewählte Textstelle anschließend so gut wie möglich im Fokus befindet, wird deren Mittelpunkt $Z$ als Referenzpunkt für die weiteren Berechnungen festgesetzt. Der Fokuswert wird nun so gewählt, dass neben der zentralen Region möglichst viele weitere Textstellen innerhalb des Tiefenschärfe-Bereichs liegen. Dabei kommen nur diejenigen Textregionen in Frage, deren Mittelpunkte den Abstand $z \in [\Delta_{fern}, \Delta_{nah}]$ haben (s. Abb. 5.3.2):

$$\Delta_{fern} = Z\frac{f^2 + Ncf}{f^2 + ZNc} \qquad \Delta_{nah} = Z\frac{f^2 - Ncf}{f^2 - ZNc}$$

da sonst nicht garantiert werden kann, dass $Z$ in den Fokus-Bereich fällt.

Sei $z_{min}$ – der minimale der gemessenen Abstände aus $[\Delta_{fern}, \Delta_{nah}]$ und $z_{max}$ – der maximale. Der Abstand zur Fokusebene $z_{fokus}$, der eine optimale Abdeckung des Bereichs$[z_{min}, z_{max}]$ bei einer festen Blendenzahl $N$ (s. Abb. 5.3.2) garantiert, wird berechnet als [124]:

$$z_{fokus} = \frac{2z_{min}z_{max}}{z_{min} + z_{max}}$$

Die Ziel-Motorwerte für die Fokussteuerung der beiden Kameras werden schließlich anhand von $z_{fokus}$ interpoliert.

# 6. Kapitel

# Layoutanalyse

In diesem Kapitel geht es um die Analyse des physischen Layouts von Dokumenten, darunter Bildsegmentierung, Identifikation von Textblocken, Lokalisierung von Textzeilen und Zeichen sowie ihrer Eigenschaften.

## 6.1 Problemstellung

Das Ziel der Layouterkennung besteht darin, inhaltliche Zusammenhänge zwischen Komponenten eines Dokuments aufzudecken, die zum Verstehen des Textes notwendig sein können. Die Prozedur lässt sich in zwei Phasen unterteilen:

1. *Layoutanalyse* dient der Identifikation von Textregionen, Textzeilen und Zeichen sowie deren Eigenschaften. Nach einer erfolgten Layoutanalyse wird ein Dokument als hierarchische Komposition von den identifizierten Elementen repräsentiert.
2. Als *Dokumentverstehen* (engl. *document understanding*) wird eine Modellierung der logischen Dokumentstruktur unter Verwendung der Ergebnisse der Layoutanalyse bezeichnet (Kapitel 8). Dabei werden Informationen extrahiert, die der Autor über die Formatierung seines Dokuments vermitteln wollte (s. Abb. 6.1.1 aus [125]). Das verwendete Modell der logischen Dokumentstruktur kann je nach Dokumentart stark variieren und beinhaltet typischerweise solche Komponenten wie Abschnitte, Absätze, Überschriften, Fußnoten usw.

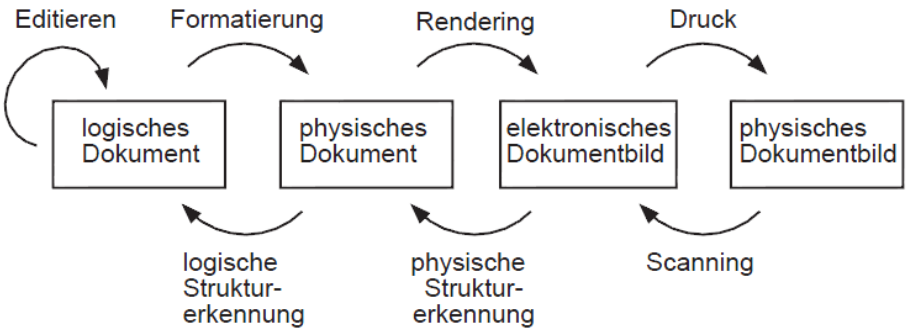


*Abb. 6.1.1: Produktion und Erkennung von Dokumenten aus [125]*

Im Rahmen dieses Kapitels wird die erste Phase der Layouterkennung behandelt, wobei die Ergebnisse der Layoutanalyse sowohl für die Festlegung der Vorlesereihenfolge als auch bei der Entzerrung der Dokumentoberfläche benötigt werden. Der Schritt umfasst eine Segmentierung der Aufnahme mit anschließender Klassifizierung der Segmente in Text- und Nicht-Text-Objekte. Im Gegensatz zur Textdetektionsphase (s. Abschnitt 5.1) ist die Genauigkeit der Segmentierungsoperation an dieser Stelle von entscheidender Bedeutung. Das Paradigma im Umgang mit unsicheren Ergebnissen ändert sich dabei grundlegend: Während der schnelle Ansatz auf die Reduktion von falsch positiven Ergebnissen abzielt, ist es beim sorgfältigen Ansatz vielmehr darauf achten, dass keine lesbaren Textregionen aussortiert werden. Die im Anschluss stattfindende Klassifizierung hat das Ziel Textregionen zu identifizieren und die zugehörigen geometrischen sowie textspezifischen Attribute zu bestimmen. Weil die Bildsegmentierung der erste Schritt der Verarbeitungsphase ist, liegen a-priori keinerlei Informationen über die Dokumenteigenschaften vor. Eine wichtige Aufgabe des Segmentierungsschrittes ist die Extraktion von Informationen, die zur Parametrisierung der später angewendeten Algorithmen zur Binarisierung und Zeilenextraktion benötigt werden. Die Laufzeit der Verarbeitung spielt angesichts der Größe der Aufnahmen eine entscheidende Rolle, sodass die Wiederverwendbarkeit der Merkmale eine wichtige Nebenbedingung darstellt.

## 6.2 Vorarbeiten

Ein Überblick der Vorarbeiten zum Thema Dokumentsegmentierung und Textdetektion wurde im Abschnitt 5.1.2 gegeben. Dort wurde festgestellt, dass texturba-

sierte Segmentierungsalgorithmen, die auf der Multiskalenanalyse basieren, sich durch ihre besondere Robustheit auszeichnen. Das Ziel der Segmentierung kann dabei als Unterteilung einer Aufnahme in Regionen mit verschiedenen Textureigenschaften definiert werden. Texturbasierte Segmentierung von Aufnahmen unter Verwendung der Wavelet-Transformation (s. Abschnitt 2.9) wurde bereits in zahlreichen Arbeiten [126][127] thematisiert, wobei mehrere Klassifizierungsmerkmale auf Basis der Wavelet-Koeffizienten vorgeschlagen wurden. In [128] werden Wavelet-basierte Segmentierungsmethoden hinsichtlich ihrer Leistung bei der Texturklassifizierung und -diskriminierung untersucht. Unter anderem werden energiebasierte Merkmale

$$Energy = \sum_{k=1}^{N} |\alpha(k)|^2$$

und entropiebasierten Merkmale

$$Entropy = \sum_{k=1}^{N} \frac{|\alpha(k)|}{N} \cdot log \frac{|\alpha(k)|}{N},$$

wobei $N$ – Anzahl der Pixel und $\alpha(k)$ – die Wavelet-Koeffizienten bezeichnen, miteinander verglichen. Die Untersuchung zeigte (s. Abb. 6.2.1), dass das einfache energiebasierte Ähnlichkeitsmaß in über 99% Prozent der Fälle eine zuverlässige Unterscheidung der Texturen erlaubt. Es konnte des Weiteren erwartungsgemäß festgestellt werden, dass die Verwendung von längeren Transformationsfiltern sowie eine Erhöhung der Dimensionalität des Merkmalsraums mittels Wavelet-Packet-Transformation zur Steigerung der Klassifizie-

CLASSIFICATION RESULTS COMPARING THE PERFORMANCE OF TWO SIGNATUREMETRICS, COMPLETE AND OVERCOMPLETE WAVELET PACKET REPRESENTATIONS AND TWO ANALYZING FUNCTIONS

| Sig. | Selected Wavelet Packets | N | Analy. Func. | Num. of Err. | % Correct |
|---|---|---|---|---|---|
| E | Comp. | 341 | $D_{20}$ | 0 | 100 |
| | | | $D_6$ | 0 | 100 |
| | Stand. | 17 | $D_{20}$ | 0 | 100 |
| | | | $D_6$ | 4 | 99.3 |
| H | Comp. | 341 | $D_{20}$ | 0 | 100 |
| | | | $D_6$ | 0 | 100 |
| | Stand. | 17 | $D_{20}$ | 1 | 99.8 |
| | | | $D_6$ | 5 | 99.1 |

E: Energy, H: Entropy, N: Number of Features

*Abb. 6.2.1 Auswertung verschiedener Wavelet-basierten Klassifizierungsmethoden aus [128].*

rungszuverlässigkeit führt. Mit dem speziellen Fall einer waveletbasierten Segmentierung von Dokumentaufnahmen beschäftigt sich Li et al. in seiner Arbeit [129] und stellt dabei fest, dass die Detail-Koeffizienten von transformierten Textaufnahmen annähernd Laplace-verteilt sind. Daraufhin wird ein Texturmodell von Textstellen entwickelt, das auf dieser Eigenschaft basiert.

Einige waveletbasierte Segmentierungsalgorithmen verwenden einen Ansatz, bei dem ein Grundgerüst aus großen und sicher klassifizierten Segmenten schrittweise durch angrenzende kleinere Regionen erweitert wird [86][129][130][131]. Der Grund dafür ist die abnehmende Zuverlässigkeit der Klassifizierung mit der steigenden räumlichen Auflösung der Untersuchung, die aus der Einschränkung der gleichzeitigen Auflösung im Orts- und Frequenzraum entsprechend der Heisenbergschen Unschärferelation (s. Abschnitt 2.9) folgt. Die von Lee et al. präsentierte Methode [131] auf der Grundlage von Quadtrees [132] ist insofern besonders interessant, als die sukzessive Zerlegung einer Aufnahme in quadratische Regionen im Einklang mit der rekursiven Vorgehensweise im Laufe der FWT steht.

## 6.3 Modell der physischen Dokumentstruktur

Abbildung Abb. 6.3.1 zeigt das entwickelte Modell der physischen Dokumentstruktur. Eine Aufnahme kann eine beliebige Anzahl von Text- und Nicht-Text-Regionen enthalten, wobei in der ersten Phase der Layouterkennung keine Unterscheidung zwischen verschiedenen Dokumenten und Seiten stattfindet. Jede Region wird durch ihre Grenzkontursequenz samt dem zugehörigen MUR charakterisiert. Eine Textregion besteht aus mindestens einer Zeile, die ihrerseits wenigstens ein Zeichen enthält. Kleine Punktzeichen werden gesondert behandelt, da die Verwechslungsgefahr mit Rauschpunkten in dieser Phase der Verarbeitung groß ist. Die endgültige Klassifizierung wird in die zweite Phase ausgelagert als die Orientierung des

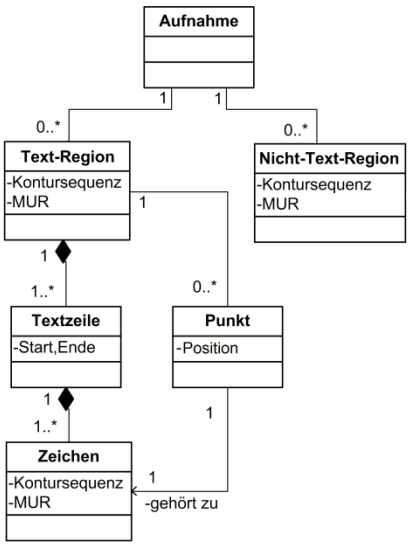


*Abb. 6.3.1:Modell der physischen Dokumentstruktur*

Dokuments bekannt ist und die Punkte den einzelnen Zeichen bzw. Zeilen zuordnen werden können.

## 6.4 Bildsegmentierung

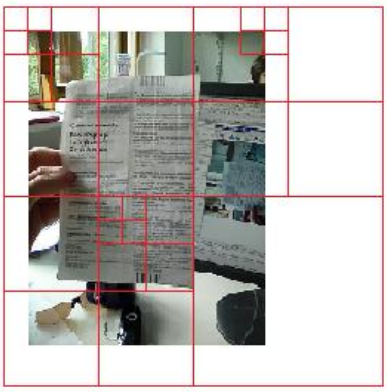

*Abb. 6.4.1:Quad-Tree-Zerlegung einer nicht quadratischen Aufnahme.*

Einige Autoren [129][131] schlagen eine Detektion von Textstellen auf Basis der Wavelet-Koeffizienten vor. Experimentelle Untersuchungen ergaben, dass die vorgeschlagenen Modelle nicht geeignet sind, um eine zuverlässige Klassifizierung von stark verrauschten oder verzerrten Textstellen gewährleisten zu können. Ein wesentlich stabileres Ergebnis konnte indes bei der Identifikation von homogenen Hintergrundbereichen erzielt werden, die anschließend als Abgrenzungen zwischen den Segmenten dienen können (s. Abb. 6.4.2). Einen möglichen Ansatz liefern dabei Verfahren zur Rauschunterdrückung in digitalen Audio- und Videoaufnahmen [133][134], die die Verteilung von Wavelet-Koeffizienten untersuchen, um irrelevante Signalkomponenten zu erkennen. Insbesondere werden hochfrequente Bestandteile mit kleinen Amplituden als Rauschen angesehen und unter Verwendung eines skalierungsabhängigen Schwellenwerts (engl. *universal threshold* aus [133])

$$\lambda_u = \sqrt{2lnN}\hat{\sigma},$$

wobei $N$ – die Anzahl der Koeffizienten und $\hat{\sigma}$ – der Median der absoluten Abweichung vom Median (engl. *Median Absolute Deviation (MAD)*), herausgefiltert. Der Schwellenwert lässt sich auch für die Identifikation von textlosen Hintergrundflächen verwenden, welche komplett aus Rauschanteilen bestehen müssen. Aus Effizienzgründen wird anstelle des Median-Maßes die Standardabweichung $\sigma_j$ der Koeffizienten-Verteilung in der jeweiligen Skalierungsstufe $j$ zur Bestimmung des Referenzwerts

$$\lambda^j = \gamma * \sqrt{2 * lnN} * 0.67 * \sigma_j$$

eingesetzt, wobei $\gamma$ – eine Justierungskonstante des Rauschpegels ist, die je nach der eingestellten Detektionsempfindlichkeit (s. Abb. 3.6.2) zwischen 0.1 und 1 liegt. Der Faktor 0.67 dient der Angleichung der Standardabweichung an den tat-

sächlichen MAD-Wert [135].

Aufgrund der rekursiven Natur von FWT bietet es sich an, die Quadtree-Datenstruktur als Basis für die Segmentierung einzusetzen. Die Konstruktion einer Quadtree-Repräsentation eines Bildes findet statt, indem das Bild in immer

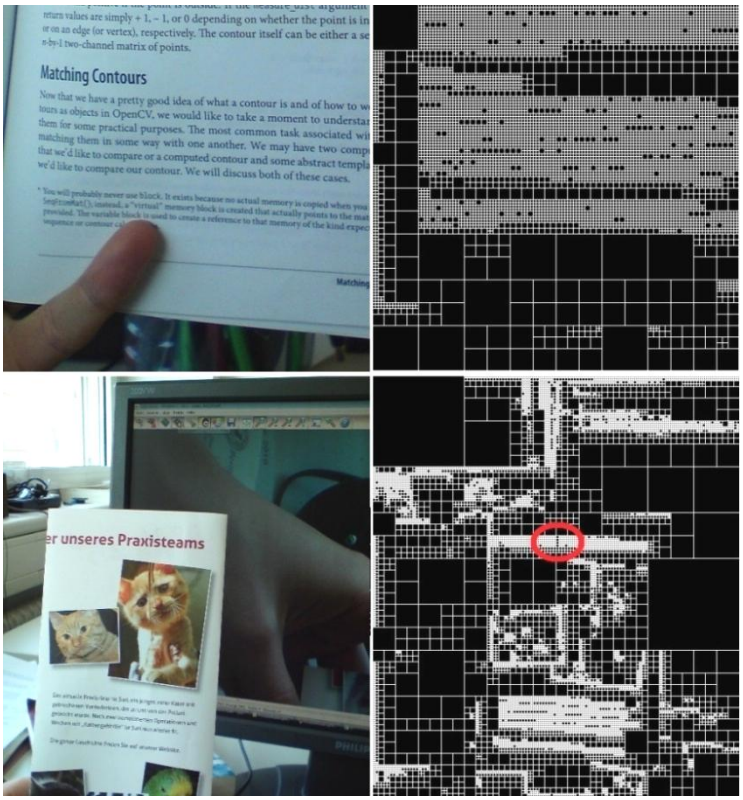


*Abb. 6.4.2: Identifizierung von Textzwischenräumen mittels Quadtree-Zerlegung und FWT-Koeffiziente, (unten) Übersegmentierungsproblematik.*

kleinere quadratische Bereiche zerlegt wird (s. Abb. 6.4.1), die anschließend in einer baumartigen Struktur organisiert werden. Alle inneren Baumknoten haben dabei genau vier Kinder, von denen jedes einen quadratischen Bildbereich repräsentiert, deren Fläche einen Viertel des von dem Vaterknoten repräsentierten Bereiches ausmacht. Auf diese Weise besteht die Möglichkeit den Betrachtungsmaßstab in jede FWT-Zerlegungsstufe entsprechend der zur Verfügung stehenden Auflösung der Frequenzmerkmale zu wählen.

In Übereinstimmung mit dem 5. Kriterium für Textstellen (s. Abschnitt 5.1.1) sind

die trennenden Leerräume zwischen den Textregionen schriftgrößenabhängig, wobei sie größer sein müssen als die Laufweiten innerhalb der angrenzenden Textbereiche. Die Suche nach homogenen Hintergrundbereichen erfolgt in einem Bottom-Up-Verfahren parallel zur Berechnung der Energieverteilung. Ein Quadtree-Knoten der Tiefe $j$ wird als Hintergrund markiert, falls

1. das Rauschen in der entsprechenden Bildregion unterhalb des globalen Schwellenwerts $\lambda^j$ liegt
2. alle seine Kinder (falls vorhanden) als Hintergrund markiert sind
3. alle Nachbarknoten zwei oder mehr Kinder haben, die als Hintergrund markiert sind.

Knoten, deren Kinder sowohl dem Vordergrund als auch dem Hintergrund angehören werden als potenzielle Grenzregionen angesehen. Trotz der stabilen Ergebnisse bei der Identifikation der Hintergrundbereiche kann es bei stark verrauschten Aufnahmen dazu kommen, dass kleine Leerräume zwischen den einzelnen Textbereichen übersehen werden. In einem solchen Fall kann eine nachträgliche Korrektur der Segmentgrenzen mittels einer Untersuchung der Textureigenschaften Abhilfe schaffen. Die Untersuchung basiert auf der Beobachtung, dass die Verteilung der Wavelet-Koeffizienten wichtige textspezifische Eigenschaften der Regionen widerspiegeln. Die Abb. 6.4.3 zeigt prozentuale Verteilungen der Energien aus unterschiedlichen Textbereichen auf die Skalierungsstufen der FWT-

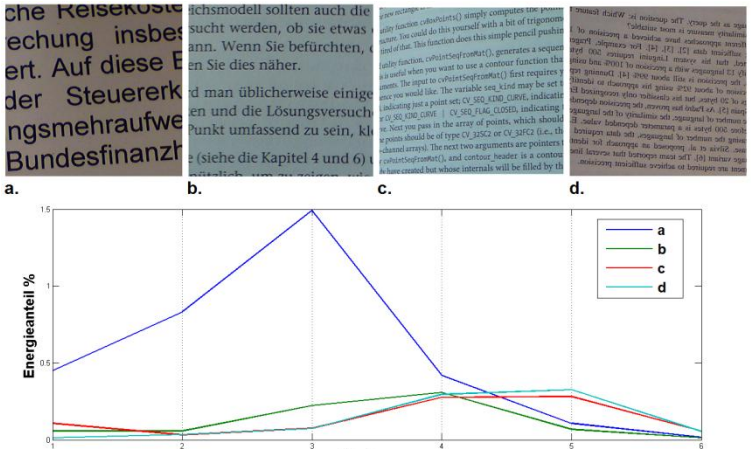


*Abb. 6.4.3: Skalierungsabhängige Energieverteilung von Textausschnitten mit verschiedenen Schriftgrößen. Zu beachten ist insbesondere wie sich die Energieverteilungen der Bilder c. und d. ähneln, die Textausschnitte mit einer vergleichbaren Schriftgröße enthalten.*

Frequenzbände. Wie deutlich zu erkennen ist, ähneln sich die Verteilungsmuster bei Textregionen mit vergleichbaren Schriftarten und -größen (s. Abb. 6.4.3). Auffällig ist die Abhängigkeit der Energieverteilung, insbesondere der Position des Maximalwertes, von der Schriftgröße im Textblock, die sich wie folgt formal beschreiben lässt: Gegeben seien zwei unterschiedlich skalierte Signale $f(x)$ und $g(x) = f(2^t x)$ sowie ihre Wavelet-Koeffizienten $\alpha_j, \beta_j$ der $j$-ten Stufe

$$\alpha_j(k) = \int_R f(x)\Psi_{j,k}(x)\,dx \text{ (s. Abschnitt 2.9)}$$

und

$$\beta_j(k) = \int_R g(x)\Psi_{j,k}(x)\,dx = \int_R f(2^t x)\Psi_{j,k}(x)\,dx.$$

Entsprechend der Skalierungseigenschaft [85] von Wavelet-Transformation sind

$$\beta_j(k) = 2^{-t/2}\alpha_{j-t}(k)$$

für alle sinnvollen $j, k$, sodass für die Energien der $j$-ten Zerlegungsstufe der Zusammenhang

$$Energy_j^{\beta} = \sum_k \left(\beta_j(k)\right)^2 = \sum_k 2^{-t}\left(\alpha_{j-t}(k)\right)^2 = 2^{-t} Energy_{j-t}^{\alpha}$$

gilt. Das Skalierungsverhältnis von $f(x)$ und $g(x)$ repräsentiert an der Stelle die unterschiedliche Schriftgrößen zweier Regionen. Das Energiespektrum des Signals $g(x)$ stellt indes eine verschobene und skalierte Version des Energiespektrums von $f(x)$. Ist die Energieverteilung innerhalb eines Segments gegeben, so lassen sich nicht nur Aussagen über den Inhalt des Segments machen, sondern auch Rückschlüsse auf die textspezifischen Eigenschaften des vermeintlichen Textblocks ziehen. Auch eine grobe Einschätzung der Zeilenorientierung in dem untersuchten Segment kann anhand der Verteilung der Energie auf die le $W_\Psi^H, W_\Psi^V, W_\Psi^D$ (s. Abschnitt 2.9) abgeleitet werden [131].

Als erster Segmentierungsschritt wird die FWT der Aufnahme durchgeführt. Die sinnvolle Tiefe des FWT-Zerlegungsbaumes hängt i. Allg. von den Abmessungen des Dokuments und von den verwendeten Wavelet-Funktionen ab. Im gegebenen Fall wird darüber hinaus die Größe der minimalen erkennbaren Zwischenräume berücksichtigt, die als obere Grenze für die Genauigkeit der Segmentierung dient. In Übereinstimmung mit den festgelegten Textkriterien (s. Abschnitt 5.1.1) kann der minimale Abstand zwischen zwei Regionen mit $4 * \Delta_s < 4 * \frac{h_{min}}{4} = h_{min} = 20px$ (s. Abschnitt 3.5) nach unten abgeschätzt werden.

Für alle Knoten $R$ der $j$-ten Ebene der Quadtree-Baumstruktur werden die Energien der Detail-Koeffizienten berechnet

$$Energy_j^{\alpha}(R) = \sum_{k \in R} \left(\alpha_j(k)\right)^2,$$

wobei in jedem der drei Kanäle $W_{\Psi}^{H}, W_{\Psi}^{V}, W_{\Psi}^{D}$ die Berechnung unabhängig stattfindet. Die Varianzen $\sigma^2(\alpha_j)$, die zur Festlegung der Schwellenwerte $\lambda^j$ benötigt werden, lassen sich anhand der berechneten Energie-Werte effizient bestimmen:

$$\sigma^2(\alpha_j) = E(\alpha_j^2) - \left(E(\alpha_j)\right)^2 = {}^1/_{N_j} \sum_{R \in \mathcal{R}_j} Energy_j^{\alpha}(R) - \left(E(\alpha_j)\right)^2$$

Hier bezeichnen $E$ – den Erwartungswert, $\alpha_j$– die Detailskoeffizienten, $R \in \mathcal{R}_j$ –

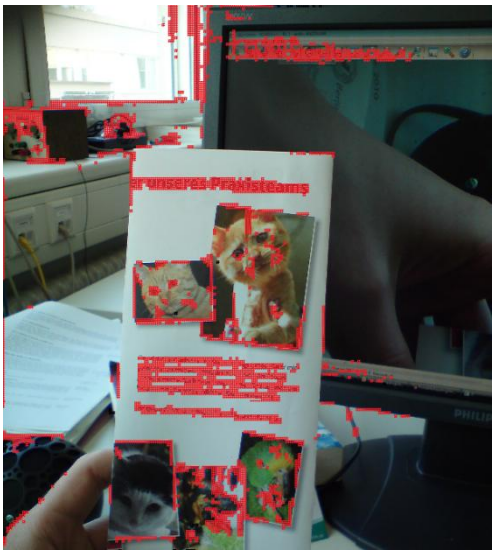

*Abb. 6.4.4: Ergebnis der Textsegmentierung. Text-Kandidaten sind mit grau markiert.*

die quadratischen Regionen der Quadtree-Struktur der Größe $N_j$. Damit können die Leerräume zwischen den benachbarten Segmenten mit ähnlichen Energieverteilungsmustern unter Berücksichtigung der ermittelten Schriftgrößen identifiziert und ggf. überbrückt (s. Abb. 6.4.2 unten) werden. Anschließend werden die durch die Zwischenräume voneinander isolierten Bereiche im Rahmen eines einfachen Region-Growing-basierten Segmentierungsverfahren [41] extrahiert, wobei die berechneten Energie-Verteilungsmuster zur Berechnung des Ähnlichkeitsmaßes herangezogen werden. Gleichzeitig werden die Grenzen der Segmente markiert

und die textspezifischen Merkmale der Blöcke *Schriftgröße* und *Zeilenorientierung* anhand der Energie-Maxima geschätzt.

Bereits in dem Segmentierungsstadium können die ersten Text-Klassifizierungsschritte vorgenommen werden, indem Segmente ohne ausgeprägte Maxima der Energieverteilung als Nicht-Text-Regionen markiert werden (s. Abb. 6.4.4). Es wird dabei in Übereinstimmung mit den festgelegten Klassifikationskriterien vorausgesetzt, dass alle Textstellen eine dominante Orientierungskomponente aufweisen. Die groben Schätzungen der *Schriftgröße* und *Zeilenorientierung* einer Region anhand der Verteilungen sind für die Parametrisierung der im Anschluss stattfindenden Extraktion der Textzeilen notwendig.

## 6.5 Extraktion der Textzeilen

Das Ziel der Zeilenextraktion ist es, die Position und mögliche Zeilenzugehörigkeit der einzelnen in einem Segment enthaltenen Zeichen zu bestimmen. Mit Hilfe der bereits vorgestellten Maße zur Regelmäßigkeitsbewertung von Vordergrundelementen (s. Abschnitt 5.1.4) lässt sich danach eine Klassifizierung der Segmente in Text- und Nicht-Text-Bereiche vornehmen. Da die gewählte LCS-Methode zur Binarisierung der Aufnahme (s. Abschnitt 2.7) auf einer Zusammenhangskomponentenanalyse basiert, bietet es sich an, die beiden Operationen in einem Algorithmus zu kombinieren. Im Gegensatz zur schnellen Zeilenextraktion der Textdetektionsphase ist eine probabilistische Vorgehensweise für die sorgfältige Textlokalisierung nicht sinnvoll, da alle übersehenen Zusammenhangskomponenten auf der binarisierten Version von dem Dokument fehlen würden. Auch eine Feststellung des exakten Verlaufs von den Zeilen ist kritisch, da die Richtigkeit der Vorlesereihenfolge maßgeblich von der Anordnung der Zeichen abhängt. Hilfreich ist hier, dass das Verschmieren der Konturen und die daraus resultierende Verschmelzung der Zeichen an dieser Stelle kein Problem mehr darstellen und dass die Zeitvorgaben nicht so eng sind wie bei der Textdetektionsphase im Initialzustand des Systems.

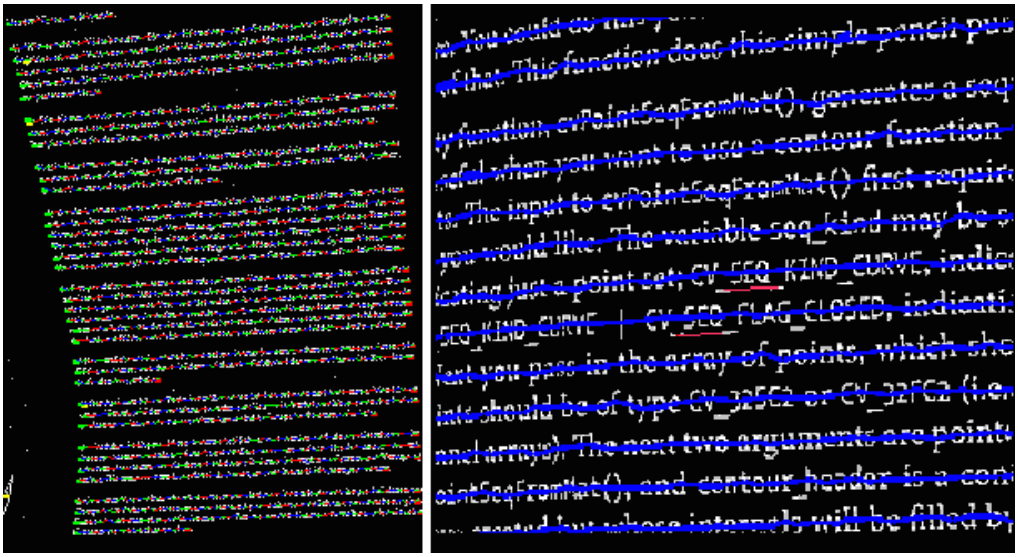

*Abb. 6.5.1: Ergebnis der Textzeilenextraktion.*

Als Ausgangspunkt für die Zeilenverfolgung (s. Abb. 6.5.1) dienen die im Abschnitt 6.4 extrahierten globalen Merkmale der Segmente: Maxima der Verteilung der Signalenergie auf die einzelnen Kanäle, welche mit der Schriftgröße und der Zeilenausrichtung (vertikal, horizontal oder diagonal) einer Textregion korrespondieren. Die Hauptorientierung des Zeilenverlaufs innerhalb der Text-Kandidaten kann indes mit Hilfe der im Abschnitt 5.1.5 diskutierten Methode der minimalen umgebenden Rechtecke genauer bestimmt werden. Für die Zeilenextraktion wird das Bild der positiven Übergangsenergien $E^+$ verwendet, welches im Rahmen der Binarisierung berechnet wurde (s. Abschnitt 2.7). Die Extraktion der Zeichenkonturen bei Entdeckung eines Zeichens findet auf dem kombinierten Bild $R$ statt, wo der Konturverlauf genauer eingezeichnet ist [37].

Um die Zeilenzugehörigkeit der Zeichen zuverlässig bestimmen zu können, wird der Verlauf der Zeilen in den extrahierten Blöcken mit Hilfe von lokalen Projektionsprofilen (vgl. [80]) untersucht. Dabei wird das gesamte Segment in rechteckige Unterregionen aufgeteilt, deren Abmessungen $w$, $h$ von der geschätzten Schriftgröße abhängen. Für jede dieser Unterregionen werden mehrere Projektionsprofile $P_\alpha^{x,y}$ mit jeweils unterschiedlichen Projektionsrichtungen berechnet

$$\alpha_{max}^{x-1,y} - \Delta_{font} < \alpha < \alpha_{max}^{x-1,y} + \Delta_{font},$$

wobei $\Delta_{font}$ - eine schriftgrößenabhängige feste Schranke und $\alpha_{max}^{x-1,y}$ die Orien-

```
FUNCTION scanSegment(Hull, localOrientationArray)
  bb := boundingBox(Hull)
  p_tl := bb.tl; p_br := bb.br   // Eckpunkte des MURs von dem Segment
  // β – Orientierung des MURS
  lines := [] // Zusammenhangskomponenten nach Linien sortiert
  // starte bei der oberen Seite bzw. linken Seite des umgebenden Rechtecks
  FORALL Point p_start ∈ L(p_tl, β)
      // gehe senkrecht zur angenommenen Zeilenrichtung vor
      // beachte die eingezeichnete Grenze der Region
      FORALL Point p ∈ L(p_start, β + 90°) ∧ p ∈ Hull
          IF I(p) = 1 ∧ not(marked(p))     // ein Vordergrundpixel gefunden
              // extrahiere die Zusammenhangsk. und markiere ihren Rand
              [massCenter, boundingBox] := extractComponent(p)
              // lese die lokale Orientierung aus der Tabelle
              α := localOrientationsArray[(p.x – p_tl.x)/w][ (p.y – p_tl.y)/h]
              // finde die zugehörige Linie
              line := findLine(lines, boundingBox, α)
              // falls gefunden
              IF line = NOT_FOUND       // neue Textlinie hinzufügen
                  addNewLine(lines, boundingBox, α)
              ELSE  // sortiere die Zusammenhangskomponente ein
                  addToLine(lines(index), boundingBox, α)
              END
              mark(p)
          ELSE IF marked(p)          // bereits extrahiert und markiert
              p := skip(p, β + 90°)
          END
      END
  END
END
```

*Pseudocode 6.5.1: Extraktion der Textzeilen und Zeichen.*

tierung der Nachbarregion sind. Das Projektionsprofil mit der größten Bewertung

$$score(P_\alpha^{x,y}) = \sum_{i=1}^{N-1} \frac{|P_\alpha^{x,y}(i) - P_\alpha^{x,y}(i+1)|}{N}$$ (vgl. Abschnitt 5.1.5)

wird ermittelt und die Projektionsrichtung $\alpha_{max}^{x,y}$ als lokale Ausrichtung dieser Unterregion in einer Orientierungstabelle vermerkt.

Nach der Ermittlung der lokalen Orientierungen der Zeilensegmente können die vermeintlichen Textzeilen eines Blocks mit einem einfachen Algorithmus extrahiert werden. Der Ablauf des Algorithmus wird im Pseudocode 6.5.1 schematisch dargestellt. Der Bereich innerhalb der Grenzen des untersuchten Bildsegments wird senkrecht zur geschätzten Hauptrichtung der Zeilen abgescannt. Bei der Entdeckung eines Vordergrundpixels, wird von diesem Punkt ausgehend die korrespondierende Zusammenhangskomponente aus dem kombinierten Bild $R$ extrahiert, wobei gleichzeitig die extrahierten Punkte in $E^+$ markiert werden, um eine Wiederentdeckung der Bildelemente erkennen zu können. Für jede extrahierte Kontursequenz werden ihr Schwerpunkt und die zugehörigen MURs berechnet, um mit deren Hilfe die Einreihung der Komponenten in eine Textzeile stattfindet (s. Pseudocode 6.5.2). Maßgeblich für die Bestimmung der passenden Textzeile ist der Abstand des vermeintlichen Zeichens zu dem zuletzt hinzugefügten Element dieser Zeile, der unter Berücksichtigung der lokalen Orientierung $\alpha_{max}^{x,y}$ berechnet wird, sowie das Größenverhältnis der Komponenten, das anhand ihrer MURs eingeschätzt wird. Laut der in 5.1.1 präsentierten Textkriterien ist das Verhältnis der Höhen von Zeichen einer Zeile nach oben begrenzt, sodass an der Stelle die ersten Nicht-Text-Elemente herausgefiltert werden können. Dank der systematischen Scanweise ist es sichergestellt, dass sämtliche Zusammenhangskomponenten im Bildsegment untersucht werden. In Anbetracht der maximal zulässigen Höhendifferenz zweier benachbarten Zeichen (s. Abschnitt 5.1.1) kann man desweiteren davon ausgehen, dass die Zeichen einer Zeile reihenfolgegetreu entdeckt werden, solange der Orientierungsfehler kleiner als $arctan(\Delta w/\Delta h)$ ist (s. Abb. 6.5.2).

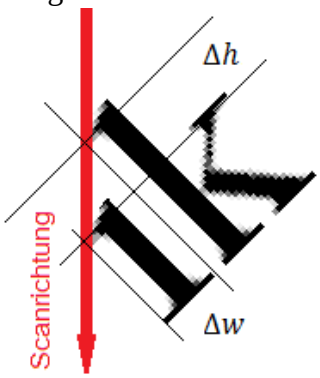


*Abb. 6.5.2: Reihenfolge der Entdeckung von Dokumentelementen.*

```
FUNKTION findLine(lines, massCur, boxCur, α)
        FORALL line IN lines        // durchsuche alle bislang extrahierten Segmente
                (massEnd, boxEnd) := line.getLastElement()
                IF belongsToLine(massEnd, boxEnd, massCur, boxCur, α)=FOUND
                RETURN line
        END
        RETURN NOT_FOUND
END
// finde  Textzeile unter Berücksichtigung der lokalen Zeilenorientierung α
FUNKTION belongsToLine(massEnd, boxEnd, massCur, boxCur, α)
        IF boxEnd.height/boxCur.height > 2  // das Regelmäßigkeitskriterium
                RETURN NOT_FOUND
        END
        // berechne den Gradientenvektor zwischen den Schwerpunkten
        gradient := massCur – massEnd
        angle := arctan(gradient.y) / |gradient.x|
        IF (|α – angle| < 30° ∧ (gradient.x)² + (gradient.y)² / boxSizeCur * boxSizeEnd < 3)
                RETURN FOUND     // gehört zur Textzeile
        ELSE
                RETURN NOT_FOUND
        END
END
```

*Pseudocode 6.5.2: Sortierung der extrahierten Zusammenhangskomponenten nach Zeilen.*

## 6.6 Klassifizierung der Regionen

Die Klassifizierung der Segmente findet über eine stufenweise Abweisung von Nicht-Text-Regionen in den beiden Phasen der oben vorgestellten Layoutanalyse statt. Während die Gruppe der Textkandidaten im Laufe der Analyse immer kleiner wird, kann der Aufwand der Untersuchungen für die dann übrig bleibenden Regionen erhöht werden. Die Tabelle 6.6.1 gibt einen Überblick über die Klassifizierungskriterien, die in verschiedenen Phasen der Layoutanalyse zum Einsatz kommen. Als Grundlage für die Klassifikation dienen analog zum schnellen Ansatz Regelmäßigkeitsbewertungen für die Muster innerhalb der Segmente.

*Tabelle 6.6.1: Klassifikationskriterien verschiedener Phasen der Layoutanalyse*

| ***Analysephase*** | ***Ausschlusskriterien für Nicht-Text-Regionen*** | ***Verwendete Merkmale*** |
|---|---|---|
| *Segmentierung* | Breite/Höhe eines Blocks passt nicht zur geschätzten Schriftgröße und Ausrichtung der Zeilen | • MUR<br>• Schriftgrößenabhängige Schwellenwerte $\sigma_{font}$ |
| *Bestimmung lokaler Orientierungswinkel der Zeilen* | Sprunghafte Veränderung der ermittelten Orientierungswerte | • $Var(\lvert\alpha_{max}^{x,y} - \alpha_{max}^{x,y-1}\rvert)$ |
| *Zeilenverfolgung* | Größe der Zusammenhangskomponenten passt nicht zu der geschätzten Schriftgröße | • $VarKoef(h_s)$ (s. Abschnitt 5.1.4)<br>• $VarKoef(\Delta_s)$ (s. Abschnitt 5.1.4 )<br>• Anzahl der Zeichen pro Zeile |

Nach der Zeilenextraktionsphase werden alle nicht-aussortierten Segmente als Text-Regionen betrachtet. Die Struktur des Dokuments wird in Übereinstimmung mit dem in Abb. 6.3.1 dargestellten Modell hierarchisch aufgebaut. Auf jeder Ebene der Hierarchie sind folgende Eigenschaften für die entsprechenden Elemente bekannt:

- Koordinaten der Grenzpunkte
- Durchschnittliche Schriftgröße
- Anzahl der Kinder-Elemente

Darüber hinaus werden die Kontursequenzen der einzelnen Zeichen gespeichert.

## 6.7 Auswertung und Zusammenfassung

Die Auswertung der Methoden zur Modellierung der physischen Layoutstruktur erfolgte unter Verwendung von ausgewählten Dokumenten aus der Datenbank MediaTeam der Universität Oulu [136]. Die Datenbank beinhaltet eingescannte Dokumente verschiedener Typen einschließlich der *Ground-Truth-Daten* bzgl. der Position und der Klasse von den enthaltenen Bildsegmenten (s. Abb. 6.7.1).

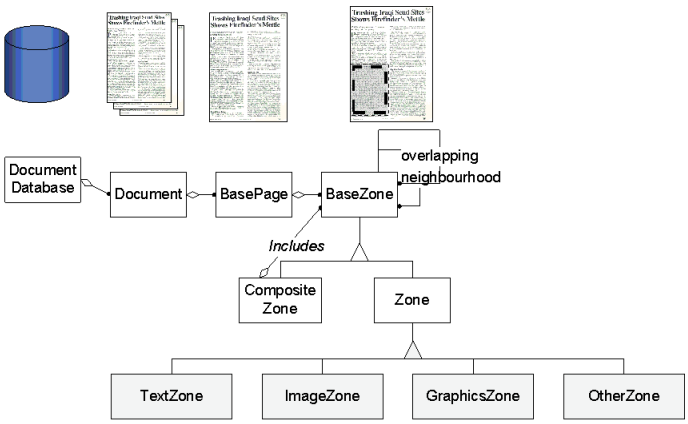


*Abb. 6.7.1:Physische Dokumentstruktur der MediaTeam Oulu Datenbank aus [136].*

Da die physische Dokumentstruktur dabei lediglich bis zur Regionen-Ebene vorgegeben wird, lässt sich das Ergebnis der Zeilenverfolgung an der Stelle nicht untersuchen. Eine ausführliche Auswertung des gesamten Algorithmus zur Layout-Erkennung wird im Anhang D vorgestellt. In erster Linie wurde die Fähigkeit des Algorithmus evaluiert, Textregionen zu identifizieren. Ein Klassifizierungsfehler liegt dementsprechend dann vor, wenn zu einer vorgegebenen Textregion (engl. *Zone*) kein Textsegment identifiziert werden konnte, dessen Mittelpunkt in der Nähe der Ground-Truth-Koordinaten liegt.

*Tabelle 6.7.1: Auswertungsergebnis für den Algorithmus zur Layoutanalyse.*

| Dokument-DB | Erkennungsrate |
|---|---|
| *MediaTeam (191 Regionen)* | 69% |
| *Zusätzliche Aufnahmen (128 Regionen)* | 84% |

Der tolerierte Abweichungsbereich wurde in Abhängigkeit von den Ausmaßen der jeweiligen Region festgelegt, wobei Regionen mit einer geringen Größe aus der Bewertung herausgenommen wurden. Ähnlich wie bei den bereits beschriebenen Vergleichsanalyse der OCR-Module (s. Abschnitt 3.5) wurde die Testmenge um einige speziell angefertigte und per Hand ausgewertete Aufnahmen erweitert, sodass auch die Robustheit des Algorithmus bei Kamera-spezifischen Herausforderungen getestet werden konnte. Die Tests haben gezeigt, dass die Klassifizierungsmethode praktisch keine falsch positiven Ergebnisse produziert (s. Anhang D). Wie aus der Tabelle 6.7.1 zu entnehmen ist, war die Erfolgsrate bei den speziell angefertigten Dokumenten trotz der z. T. erheblichen Verzerrungen um einiges höher als bei den Exemplaren aus der "MediaTeam" Datenbank. Auf den im Anhang D präsentierten Dokumenten ist es zu erkennen, dass viele Fehler darauf zurückzuführen sind, dass die vorgegebene Segmentierungsstruktur über eine rein physische Strukturierung der Dokumente hinausgeht (s. auch Abb. 6.7.2). Die dadurch entstehenden Abweichungen können bei der nachfolgenden Analyse des


*Abb. 6.7.2: Erkannte und vorgegebene Segmentierungen, zwei Problemfälle.*

logischen Layouts u. U. beseitigt werden, sodass eine qualitative Bewertung der Layouterkennung erst nach der zweiten Phase sinnvoll ist. Als Konsequenz aus den vorgestellten Evaluierungsergebnissen findet die Untersegmentierungsproblematik während der Analyse der logischen Dokumentstruktur eine besondere Beachtung.

Während die Laufzeit des Segmentierungsalgorithmus weitestgehend unabhängig vom Bildinhalt ist, hängt die für die Zeilenextraktion notwendige Rechenzeit maßgeblich von der Anzahl der Zusammenhangskomponenten und damit von der in der Aufnahme vorhandenen Textmenge ab. Dank der Effizienz des FWT-Zerlegungsalgorithmus nimmt die Segmentierung der Aufnahmen etwa 5 s in Anspruch, während die Zeilenverfolgung und die Binarisierung im Schnitt bis zu 3 ms pro Zeichen dauern können.

# 7. Kapitel

# Entzerrung und Dokument-Stitching

Das Thema dieses Kapitels ist die Korrektur von Verzerrungsartefakten auf der Grundlage der Ergebnisse der Layoutanalyse (s. Kapitel 6) und unter Verwendung der stereovisionbasierten Distanzmessung (s. Abschnitt 2.6).

## 7.1 Vorarbeiten

Eine Entzerrung von Dokumentaufnahmen im Vorfeld der Zeichenerkennung kann eine signifikante Verbesserung des OCR-Gesamtergebnisses bewirken. Es werden je nach Verzerrungsgrad Steigerungen der Erkennungsraten von 3,5% [137] bis über 10% [138] berichtet. Die angesprochene Problematik stellt seit geraumer Zeit ein aktives Forschungsfeld dar. Allerdings konzentrierten sich viele Arbeiten [139][140][141] lange Zeit auf die Flachscanner-spezifischen Verzerrungsartifakte, wie sie bspw. entstehen, wenn ein aufgeschlagenes Buch auf der Scanneroberfläche liegt (s. Abb. 7.1.1). Auch eine Ermittlung der Textflussrichtung wurde von mehreren Autoren thematisiert [80][142]. Mit der rasanten Entwicklung auf dem Smartphone-Markt wurden Videokameras immer häufiger zur Dokumentdigitalisierung eingesetzt, wodurch der Bedarf nach robusteren Entzerrungsalgorithmen entstand.

So wurden im Laufe der Zeit zunehmend komplexere Oberflächenmodelle der

Textträger in Betracht gezogen (s. Abb. 7.1.1). Einige Autoren [143][144] schlagen an der Stelle globale homographiebasierte Modelle, deren Einsatzgebiet sich jedoch i. d. R. auf die Korrektur von perspektivischen Effekten beschränkt. Wie bereits im Abschnitt 2.6 diskutiert wurde, besitzt eine Homographie-Matrix acht Freiheitsgrade, sodass die Positionen von mindestens vier Dokumentpunkten auf einer sensorparallelen Ebene nötig sind, um ein solches Modell zu parametrisieren. Die dafür erforderlichen Korrespondenz-Beziehungen zwischen Punkten der verzerrten und der flachen Versionen der Dokumentoberfläche kann bspw. unter Ausnutzung von a priori bekannten Dokumenteigenschaften erfolgen. So werden in [144][145] die Form der Dokumentgrenzen bzw. Textblöcke, die Parallelität der Textzeilen [146] oder auch die Ausrichtung der Linienzüge (engl. *Vertical Stroke Boundary (VSB)*) der einzelnen Buchstaben [147] analysiert. Da die 3D-Rekonstruktion einer Szene anhand von einer einzigen Ansicht prinzipiell nicht möglich ist, kann in einem solchen Fall keine Korrektur ohne zusätzliche Annahmen über die Dokumenteigenschaften wie bspw. seine Layout- Struktur [138][150] oder die Orientierung der Zeichen [147] vorgenommen werden. Im Grunde werden die fehlenden Tiefenwerte dabei durch Kontextinformationen ersetzt, was u. U. eine globale Vorgehensweise bei der Schätzung der Verzerrungsparameter notwendig macht.

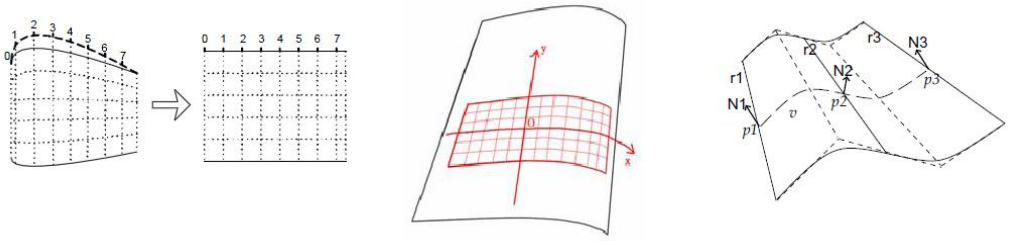


*Abb. 7.1.1: Entwicklung des Verzerrungsmodells: (links) einfache Wölbung einer Buchseite [141], (mittig) Wölbung kombiniert mit einer perspektivischen Verzerrung [148], (rechts) abwickelbare Flächen [150].*

Neben der perspektivischen Verzerrung kann eine eventuell vorhandene Krümmung der Dokumentoberfläche ein Problem für die Zeichenerkennung darstellen. Anders als bei den perspektivischen Verzerrungseffekten sind an dieser Stelle nicht-lineare Oberflächenmodelle erforderlich, wobei zwischen globalen und lokalen Modellierungsmethoden unterschieden werden kann. Im Falle einer lokalen Vorgehensweise wird die Dokumentoberfläche aus lokal abwickelbaren Teilflä-

chen zusammensetzt, während bei einem globalen Ansatz die globale Abwickelbarkeit der gesamten Oberfläche zu gewährleisten ist, wobei es sich um ein schwieriges Problem handelt. So kann bspw. eine monotone Seitenkrümmung eines aufgeschlagenen Buches durch einen Zylindermantel modelliert werden. Müssen darüber hinaus auch mögliche perspektivische Verzerrungseffekte berücksichtigt werden, dann ist die Verwendung eines kegelförmigen Oberflächenmodells angemessener. Die Methode stößt allerdings schnell auf ihre Grenzen, wenn die Seite mehrfach gefaltet ist [148][149].

Falls keine Informationen über das globale Krümmungsverhalten der zu entzerrenden Dokumentoberfläche vorliegen, sind in den meisten Fällen lokale Methoden vorzuziehen. Liang et al. schlagen in ihrer Arbeit [150] einen Algorithmus vor, der eine Dokumentoberfläche entlang der Erzeugenden in mehrere Streifen zerlegt, diese linear approximiert und mittels Rotations- und Translationstransformationen auf eine gemeinsame Fläche abbildet. Ein großes Problem der lokalen Methoden stellen die Unstetigkeitsstellen an den Grenzbereichen der Teilflächen dar. Durch die Zerlegung der Oberfläche entlang der Erzeugenden werden insbesondere inneren Formverzerrungen (Stauchung, Dehnung) an den Schnittstellen vermieden, wobei eine solche Zerlegung für Regelflächen immer möglich sein muss [150]. Die Triangulierung stellt eine weitere Möglichkeit dar, eine komplex gekrümmte Fläche stückweise linear zu approximieren. Pilu präsentiert in seiner Arbeit [151] eine Methode, mit der die Abwickelbarkeit eines Triangulierungsnetzes mittels Kantenrelaxation erzielen werden kann. Auf diese Weise wird die Dokumentoberfläche durch Tangentialebenen approximiert und kann abgewickelt werden. In [152] wird das Konzept mit Stereovision-Verfahren kombiniert und für die Dokumententzerrung eingesetzt.

Die beschriebenen lokalen Entzerrungsmethoden sind häufig rechenaufwändiger als globalen Verfahren. Der höhere Zeitbedarf ist vor allem darauf zurückzuführen, dass die Anzahl der benötigten Messwerte direkt mit der Anzahl der Teilflächen zusammenhängt, die für eine Modellierung verwendet werden. Die Bestimmung von Messwerten ist indes häufig spekulativ und rechenintensiv, sodass viele Autoren auf Interpolationsmethoden zurückgreifen, um die Messungsgenauigkeit zu erhöhen. In den Arbeiten [153][137][154] kommen an der Stelle Glättungssplines (engl. *smoothing splines*) zum Einsatz, die jedoch i. Allg. keine globale Abwickelbarkeit der Oberfläche garantieren [155].

## 7.2 Problemstellung

Die verzerrungsbedingte Verfälschung der Buchstabenkonturen ist kritisch für das Ergebnis einer Zeichenerkennung. Da Erkennungsfehler stark konzentriert in verzerrten Bereichen auftreten, bringt auch die orthographische Fehlerkorrektur im Anschluss an die OCR-Prozedur u. U. keine Verbesserung der Erkennungsraten. Dies führt dann dazu, dass die Verständlichkeit von ganzen Textabschnitten durch die Verzerrung gefährdet ist. Darüber hinaus kann eine verzerrungsbedingte Krümmung von Textzeilen die Anordnung von Zeichen durcheinander bringen und die Ermittlung der Vorlesereihenfolge erschweren.

Das Ziel an dieser Stelle ist die Entwicklung eines Algorithmus, der mit Hilfe von Stereovision-Distanzdaten auch dann zufriedenstellende Ergebnisse produziert, wenn Einzelbild-Verfahren an ihre Grenzen stoßen. Im Mittelpunkt der Betrachtung stehen vor allem perspektivische Verzerrungen, deren Korrektur aufgrund von Schwierigkeiten bei der Schätzung der "vertikalen" Fluchtpunkte für Einzelbild-Verfahren ein Problem darstellen kann. Eine entscheidende Herausforderung besteht hier darin, konventionelle Modellierungsmethoden und Stereovision-Verfahren in einem Algorithmus zu kombinieren, damit auch Dokumentbereiche, für die keine zuverlässigen Distanzdaten verfügbar sind, entzerrt werden können. Das betrifft insbesondere Dokumentteile aus den nicht-überlappenden Bildbereichen, in denen keine stereovisionbasierte Messungen möglich sind. Es bietet sich an der Stelle an, die Zusammensetzung der beiden Teile der Stereoaufnahme zu einem Dokument in den Entzerrungsalgorithmus zu integrieren. Die Genauigkeit der Entzerrung stellt angesichts der Empfindlichkeit der Zeichenerkennung gegenüber kleinsten Rundungsfehlern eine große Herausforderung dar (s. Abb. 7.2.1).

*Abb. 7.2.1: Geringe Fehlertoleranz bei OCR-Aufgaben. "e" kann schnell mit "c" verwechselt werden.*

## 7.3 Modellierung der Dokumentoberfläche

Als erster Schritt der Entzerrung wird eine Modellierung der Dokumentoberfläche mit Hilfe von Distanzdaten vorgenommen. Eine Methode zur Distanzmessung anhand von Stereoaufnahmen sowie die spezifische Problematik der Entdeckung von Korrespondenzpunkten in Dokumentaufnahmen wurden bereits in den Abschnitten 2.6 und 5.3.2 diskutiert. Bei der Abstandsmessung zur Bestimmung des

optimalen Fokuswertes konnte das Problem der geringen Unterscheidbarkeit einzelner Hintergrundpixel gelöst werden, indem ihre lokale Nachbarschaft analysiert wurde. In der Nachbarschaft eines Zeichens befinden sich die markantesten Punkte häufig an den Kanten der umgebenden Buchstaben, sodass es naheliegend ist, die während der Layoutanalyse (s. Abschnitt 6.5) extrahierten Zeichenkonturen für die Stereokorrespondenzsuche zu verwerten.

Die Bestimmung der Korrespondenzpunkte im Überlappungsbereich der beiden Aufnahmen erfolgt hierarchisch in mehreren Stufen: Erst werden korrespondierende Textblöcke identifiziert, danach werden korrespondierende Textzeilen innerhalb der Blöcke gefunden und schließlich erfolgt die Bestimmung von korrespondierenden Wörtern und Zeichen in den Zeilen. Mit den Konturseqenzpunkten der Zeichen kann darüber hinaus eine noch höhere Dichte des Vermessungsnetzes erreicht werden.

Für eine robuste Bestimmung der Korrespondenzen in der jeweiligen Hierarchieebene werden die zu Sequenzen angeordneten Elemente aus den beiden Aufnahmen mit Hilfe des Needleman-Wunsch-Algorithmus [118] aufeinander ausgerichtet. Je nach Ebene werden verschiedene Merkmale (s. Tabelle 7.3.1) als Basis für die Ähnlichkeitsbewertung verwendet, wobei im Vorfeld eine Rektifizierung der Koordinaten stattfindet. Der Ablauf des klassischen Alignment-Algorithmus und die Bestimmung der Alignment-Kosten wurde im Rahmen dieser Arbeit bereits ausführlich diskutiert (s. Abschnitt 5.2.4). Die einzige Modifikation des Verfahrens, die an dieser Stelle vorgenommen wird, besteht in einer Ergänzung der als Metrik eingesetzten Levenshtein-Distanz[*] um die Verschmelzung-Operation, die eine Kombination aus Ersetz- und Einfüge- bzw. Lösche-Operationen darstellt.

[*] Levenshtein-Distanz (auch als edit-distance bekannt) zwischen zwei Zeichensequenzen ist die minimale Anzahl von Einfüge-, Lösche- und Ersetz-Operationen, die notwendig ist, um die beiden Sequenzen anzugleichen.

*Tabelle 7.3.1:Zusammensetzung der Ähnlichkeitsbewertung in verschiedenen Stadien der Korrespondenzsuche.*

| **Ebene** | **Merkmale** |
|---|---|
| **Blockebene:**<br><br>$\boldsymbol{D = \{b_1, b_2, ..\}}$ | • Koordinaten der Mittelpunkte $b_i.middle$<br>• Größe der MUR $b_i.width, b_i.height$<br>• Anzahl der Zeilen im Block |
| **Zeilenebene:**<br><br>$\boldsymbol{line = \{c_1, \dots, c_{end}\}}$ | • Mittlere Schriftgröße einer Zeile<br>• Y-Koordinaten des ersten/letzten Zeichens der Zeile $(c_1.y, c_{end}.y)$. |
| **Wortebene:**<br><br>$\boldsymbol{line = \{w_1, \dots, w_{end}\}}$<br>$\boldsymbol{w_k = \{c_{I(k)}, \dots, c_{I(k+1)}\}}$ | • Wortabstände bzw. Zeichenabstände:<br>$\Delta_{i,i+1}$<br>$= \|(c_{i+1}.box.left, c_{i+1}.box.top)$<br>$- (c_i.box.right, c_i.box.top)\|$<br>die mehr als doppelt so groß sind, wie der Median der Zeichenabstände in der Zeile:<br>$I = \{ i \mid \Delta_{i,i+1} > 2 * median(\Delta_{j,j+1}), i, j \in [1, end - 1]\}$<br>• Angrenzende Zeichen: $(c_{I(k)}, c_{I(k)+1})$<br>• Wortlängen: $I(k+1) - I(k)$ |
| **Zeichenebene:**<br><br>$\boldsymbol{w_k = \{c_{I(k)}, \dots, c_{I(k+1)}\}}$ | • Koordinaten der Mittelpunkte $c_{I(k)}.middle$<br>• Größe der MUR $c_{I(k)}.width, c_{I(k)}.height$<br>• Länge der Kontursequenz |

Im Ergebnis der Alignierungsoperation wird eine Disparitätskarte produziert, wobei die Auflösung der Karte mit jeder neuen Ebene erhöht wird. Anhand der berechneten Disparitätswerte $d$ können schließlich die 3D-Kamerakoordinaten $(X, Y, Z)$ der entsprechenden Punkte $(x, y)$ berechnet werden [25]

$$Q_{rect}\begin{pmatrix} x \\ y \\ d \\ 1 \end{pmatrix} = \begin{pmatrix} X' \\ Y' \\ Z' \\ W \end{pmatrix},$$

wobei

$$(X\ Y\ Z) = {}^{1}/_{W}\ (X'\ Y'\ Z').$$

Die modellierte Oberfläche eines Dokuments wird auf der Abb. 7.3.1 dargestellt. Dank der hierarchiebasierten Teile-und-Herrsche-Methode kann eine Zerlegung des Korrespondenzfindungsproblems in kleinere Teilprobleme vorgenommen werden, wodurch die Laufzeit des dynamischen Alignment-Algorithmus reduziert wird.

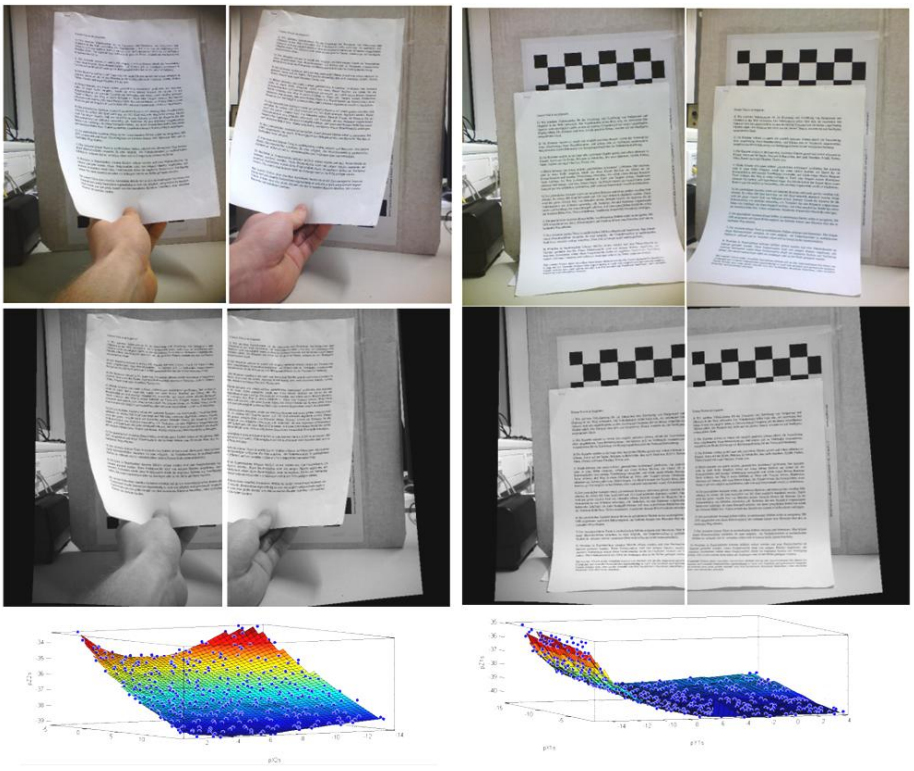

*Abb. 7.3.1: Erstellung eines 3D-Oberflächen-Modells (1): (oben) das Original, (mittig) nach der Rektifizierung, (unten) das 3D-Modell*

## 7.4 Modellierung der Verzerrung

Als Verzerrungsmodell wird im Rahmen dieser Arbeit ein mathematisches Modell zur Beschreibung solcher geometrischen Eigenschaften der Dokumentoberfläche bezeichnet, die sich negativ auf das Ergebnis der OCR-Verarbeitung auswirken könnten. Die Wahl des Verzerrungsmodells beeinflusst maßgeblich den Ablauf der Korrekturtransformation.

Im Gegensatz zu Einzelbildverfahren sind stereovisionbasierte Dewarping-Methoden nicht zwingenderweise auf zusätzliche Annahmen über die Layout-Eigenschaften angewiesen, um ein 3D-Modell der geometrischen Verzerrung erstellen zu können. Da eine unabhängige Tiefenmessung an sämtlichen OCR-relevanten Bildpunkten bei einem hochaufgelösten Vermessungsnetz aufwändig und aufgrund der Verwechslungsproblematik fehleranfällig ist, wird ein Großteil der Tiefendaten durch eine Interpolation bestimmt. Eine Gewährleistung der globalen Abwickelbarkeit der interpolierten Fläche ist grundsätzlich nur unter erheblichem Aufwand möglich [155], sodass an der Stelle darauf verzichtet wird. Infolge der fehlenden Abwickelbarkeit können innere Verformungen der geglätteten Oberfläche nicht mehr ausgeschlossen werden, es lässt sich jedoch gewährleisten, dass keine kritischen Bereiche davon betroffen sind. An der Stelle bietet es sich an, das bereits berechnete Modell des physischen Layouts (s. Abschnitt 6.3) zu verwenden, um die Positionen der Zeichen sowie der Leerräume dazwischen zu ermitteln und die Korrektur auf die Bildbereiche zu beschränken, die für die Texterkennung relevant sind. Auf diese Weise kann die Dokumentoberfläche derart stückweise approximiert werden, dass die Unstetigkeitsstellen kein Problem darstellen. Aus Effizienzgründen wird in der aktuellen Implementierung eine Approximation der Teilflächen innerhalb der Zeichenbereiche durch Tangentenflächen vorgenommen. Es sei an der Stelle darauf hingewiesen, dass auch andere abwickelbare Flächenmodelle dafür in Frage kommen, ohne dass die globale Abwickelbarkeit gewährleistet werden muss.


*Abb. 7.4.1: Perspektivische Verzerrung einzelner Zeichenebenen im Verzerrungsmodell.*

Das verwendete Verzerrungsmodell gibt an, wie die angepassten Ebenen der Zeichen relativ zur Sensorebene orientiert sind (s. Abb. 7.4.1) und stellt die u. U.

komplizierte Krümmung der Dokumentoberfläche durch eine Komposition von lokal approximierten perspektivischen Verzerrungen dar. Um die Zeichenebenen zu parametrisieren, werden drei nicht-kollineare Punkte aus der jeweiligen Zeichenregion benötigt. Bei größeren Zeichen lassen sich Punkte der Kontursequenz für diese Zwecke verwenden, in den meisten Fällen ist die erzielbare Auflösung der stereovisionbasierten Distanzmessung (s. Abschnitt 2.6) für eine zuverlässige Parametrisierung der Teilebenen jedoch nicht ausreichend, sodass die Messwerte interpoliert werden müssen.

Für die Interpolation der Tiefendaten wird das Thin-Plate-Splines (TPS)-Modell [156] verwendet. Die TPS gehören zur Familie der Smoothing-Splines, die über glättende Eigenschaften verfügen und deswegen häufig bei der Approximation von Dokumentoberflächen eingesetzt werden. Die entsprechenden Verfahren minimieren dabei eine Energiefunktion, die aus einem Fehlermaß (Fehlerquadratsumme) und einem Glattheitsmaß (Quadrate der zweiten räumlichen Ableitungen) zusammengesetzt ist [156]:

$$E = \sum_{i=1}^{N} \|f(x_i) - y_i\|^2 - \lambda \iint \left[ \left( \frac{\delta^2 f}{\delta x^2} \right)^2 + 2 \left( \frac{\delta^2 f}{\delta xy} \right)^2 + \left( \frac{\delta^2 f}{\delta y^2} \right)^2 \right] dxdy$$

Der Regularisierungsparameter $\lambda$ ermöglicht eine Einstellung der Glättungseigenschaften der Interpolation und wird in Abhängigkeit von der Schriftgröße gewählt. Die Lösung der Minimierungsaufgabe hat eine geschlossene Form, beinhaltet jedoch eine Invertierung von Matrizen der Größe $3 * (3 + N + 1)$ in der Anzahl $N$ der Stützstellen, was zu einer kubischen Laufzeit des Algorithmus führt [157]. Um den Rechenaufwand zu reduzieren, wird auch bei der Interpolation der Tiefenwerte auf ein globales Modell verzichtet. Stattdessen wird zeilenweise vorgegangen, wobei lediglich die Messwerte der Zeile selbst sowie der jeweils benachbarten Textzeilen berücksichtigt werden. Auf diese Weise muss in jedem Schritt nur eine Teilmenge der Stützstellen in die Berechnung miteinbezogen werden, was zu einer Verbesserung der Laufzeit (s. Abb. 7.4.2 links) und der Kondition der Interpolation führt [157]. Die bei dieser lokalen Vorgehensweise entstehenden Unstetigkeitsstellen landen in den Zwischenräumen der Zeilen und sind somit unproblematisch. Es ist jedoch stets darauf zu achten, dass genügend Stützstellen für eine Interpolation zur Verfügung stehen. Insbesondere ist eine Verteilung der Gitterknoten, bei der alle verwendeten Stützpunkte aus einer einzi-

gen (geraden) Zeile stammen zu vermeiden. Solche Situationen lassen sich anhand der schlechten Kondition der Koeffizientenmatrix erkennen* und durch das Hinzufügen weiterer Messwerte beheben. Die Genauigkeit der Anpassung ist insbesondere bei der Extrapolation von Tiefenwerten für Zeilensegmente kritisch, die außerhalb des Überlappungsbereichs liegen.

Sind drei interpolierte Punkte $(p_1, p_2, p_3)$ aus der Umgebung eines Zeichens gegeben, so wird die Normale der zugehörigen Ebene als das Kreuzprodukt $\vec{n} = (p_2 - p_1) \times (p_3 - p_1)$ berechnet und in dem Verzerrungsmodell gespeichert.

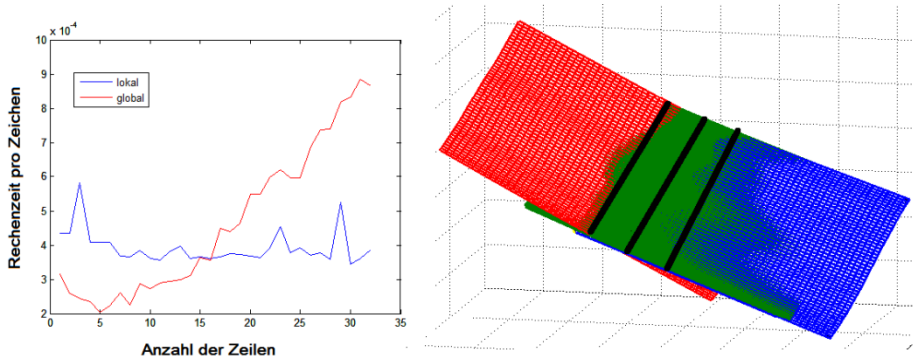


*Abb. 7.4.2: (links) durchschnittliche Rechenzeit pro Zeichen in Abhängigkeit von der Anzahl der Zeilen mit durchschnittlich 25 Messwerten pro Zeile, (rechts) das Ergebnis der "stückweisen" Spline-Interpolation. Die berücksichtigten Nachbarlinien sind schwarz eingezeichnet. Drei benachbarten, approximierten Ebenen sind blau, rot und grün markiert.*

Die Operation wird für alle Zeichen eines Bereichs durchgeführt, für den eine Entzerrung benötigt wird.

## 7.5 Berechnung der Korrekturtransformationen

Die Korrektur der Verzerrung zur Verbesserung der Zeichenerkennung wird unter Verwendung des Verzerrungsmodells vorgenommen und umfasst folgende Schritte:

- Korrektur der individuellen perspektivischen Verzerrungen der Zeichen

* Die beinahe Kollinearität der Messwerte bewirkt, dass einige Eigenwerte der Koeffizientenmatrix sehr klein werden.

- Korrektur der Anordnung der Zeichen in der Zeile
- Zusammensetzung einzelner Zeilen zu einer Region

Die perspektivische Entzerrung der Zeichen erfolgt über die Berechnung von Homographie-Abbildungen, mit deren Hilfe die Frontalansichten auf die Zeichenebenen simuliert werden. Die Korrekturtransformation besteht aus einer Rotation der approximierten Ebenen im Kamerakoordinatensystem gefolgt von einer Projektion der rotierten Punkte auf die Sensorebene. Als erstes werden der Winkel $\alpha_i^p$ zwischen der Normalen einer Zeichenebene $\vec{n}_i$ und der Z-Achse des Kamerakoordinatensystems berechnet, der als Maß der vorhandenen Verzerrung dient. Anschließend wird die Rotationsmatrix $R_i^p$ für die Drehung der Ebene um den Vektor $\vec{n}_i \times [0,0,1]^T$ und den Winkel $-\alpha_i^p$ berechnet. Da die Rotationsachse in der Zeichenebene liegt, bleibt eine mögliche Neigung der Zeichen infolge der nicht waagerechten Orientierung der Zeilen unverändert. Diese muss dann durch eine weitere Rotation $R_i^r$ um die rotierte Normale der Zeichenebene, die nun in die Richtung der Z-Achse zeigt, beseitigt werden. Der dafür notwendige Rollwinkel $\alpha_i^r$ wird aus der Tabelle der lokalen Zeilenorientierungen $\alpha_{max}^{x,y}$ entnommen, die während der Zeilenextraktion berechnet wurde (s. Abschnitt 6.5). Die angestrebte Ausrichtung der Zeichenebenen wird durch die Kombination der beiden Rotationstransformationen $R_i = R_i^r R_i^p$ erreicht.

Mit Hilfe des vorgestellten Verzerrungsmodells lassen sich auch komplexe nichtlineare Verzerrungen darstellen und unter Anwendung von linearen Transformationen reduzieren. Allerdings hat diese Flexibilität ihren Preis, da für jedes einzelne Zeichen eine individuelle Korrekturtransformation berechnet werden muss. Aus diesem Grund ist die Verwendung einer Look-Up-Tabelle für die Rotationsmatrizen sinnvoll, mit deren Hilfe die Mehrfachauswertung der Rotationstransformationen vermieden werden kann. Dadurch lassen sich u. U. Laufzeitvorteile erzielen, insbesondere wenn die nicht-lineare Verzerrungskomponente klein ist und die Ausrichtungen der Teilebenen ähnlich sind.

Nach der erfolgten Rotationstransformation kann die angestrebte Koplanarität der Zeichenebenen durch eine Translation der Zeichenpunkte auf eine gemeinsame *Dokumentebene* erreicht werden. Unter allen in Frage kommenden frontalen Ebenen wird diejenige ausgewählt, bei der die kleinsten Korrekturänderungen infolge der nötigen Translationen zu erwarten sind. Zum einen wird dadurch eine Minimierung der notwendigen Verschiebungstransformationen angestrebt, zum ande-

ren wird die notwendige Gesamtverschiebung so klein wie möglich gehalten. Die Dokumentebene verläuft parallel zur Sensorebene und ist $d_{doc}$ von dieser entfernt:

$$d_{doc} = \sum_i d_i \cdot \tilde{w}_i$$

Die Abstände $d_i$ zu den Zeichenebenen werden in Abhängigkeit von dem Orientierungswinkel $\alpha_i^p$ gewichtet, sodass die ursprünglich frontal-parallel ausgerichteten Dokumentbereiche einen größeren Einfluss auf die Entscheidung über die Lage der gemeinsamen Dokumentebene haben:

$$\tilde{w}_i = \frac{w_i}{\sum_i w_i}, mit\ \ w_i = \frac{1}{(1+|\alpha_i^p|)}$$

Bei einer rein perspektivischen Verzerrung der Oberfläche sind die Gewichte gleichverteilt und die Ebene verläuft durch die Mitte des Dokuments.

Die Translation der Zeichen auf die Dokumentebene findet zeilenweise unter Einhaltung der Abstände und einer Begradigung der Textzeilen statt. Außerdem werden die Zeilen horizontal ausgerichtet, was für eine erfolgreiche Zeichenerkennung in den Textblöcken notwendig ist. Eine der Zeilen aus dem mittleren Bereich des Dokuments dient dabei als Referenz und wird so verschoben, dass ihre Mitte am Schnittpunkt der optischen Achse mit der Dokumentebene $(0{,}0, d_{doc})$ liegt. Durch die Festlegung einer Ebene, eines Orientierungswinkels und eines Punktes in der Ebene ist eine Gerade im 3D-Raum eindeutig definiert.

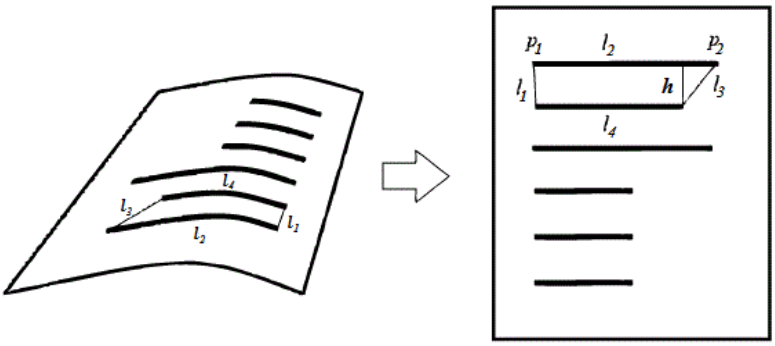


*Abb. 7.5.1:Bestimmung der Distanz zwischen zwei benachbarten Zeilen.*

Angefangen mit der Referenzzeile wird der Abstand zur jeweils nachfolgenden Textzeile unter Verwendung des Trapezmodells berechnet (s. Abb. 7.5.1), bis die Positionen aller Zeilen eindeutig festgelegt sind. Die Abstände zwischen zwei Punkten auf einer gekrümmten Ebene werden als Weglängen im 3D-Raum definiert. Wie die Abschätzung der Längen vorgenommen wird, hängt von der Anzahl der zur Verfügung stehenden Messwerte ab: Die Bogenlängen der Zeilenkurven $l_2, l_4$ können als Summe der Distanzen zwischen den Mittelpunkten der Zeichen angenähert werden, während die Berechnung von Distanzen zwischen zwei benachbarten Textzeilen $l_1, l_3$ eine Interpolation von zusätzlichen Werten erfordert. Seien $p_i, p_j$ Positionen zweier Zeichen aus verschiedenen Zeilen und $(X_i, Y_i), (X_j, Y_j)$ – ihre Bildkoordinaten, dann wird die Distanz $dist(p_i, p_j)$ im 3D-Kamerakoordinatensystem wie folgt berechnet:

$Interpolation$:

$$(X_{i,j}^k, Y_{i,j}^k) = (X_i, Y_i) + \frac{(X_j, Y_j) - (X_i, Y_i)}{\|(X_j, Y_j) - (X_i, Y_i)\|} \cdot k, \qquad k \in N.$$

$Projektion$:

$$\vec{p}_{i,j}^k = (x_{i,j}^k, y_{i,j}^k, z_{i,j}^k)^T = TPS(X_{i,j}^k, Y_{i,j}^k) \cdot \left( \begin{pmatrix} 1/f & 0 & -c_x \\ 0 & 1/f & -c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_{i,j}^k \\ Y_{i,j}^k \\ 1 \end{pmatrix} \right)$$

Die Berechnung des Abstandes erfolgt schließlich mit

$$dist(p_i, p_j) = \sum_k \|\vec{p}_{i,j}^k - \vec{p}_{i,j}^{k-1}\|,$$

wobei $TPS(\cdot)$ – die TPS-Interpolation, $f$ – die Brennweite, $(c_x, c_y)$ – die Koordinaten der Bildmittelpunke (s. Abschnitt 2.6) sind. Die im Laufe der Berechnung der Zeilenlängen $l_2, l_4$ produzierten Zwischenergebnisse werden gespeichert und bei der Verschiebung der Zeichen wiederverwendet. Um eine Begradigung der Zeilen vorzunehmen, werden die extrahierten Zeichen entlang der berechneten Geraden unter Einhaltung der Abstände zwischen den Zeichen positioniert.

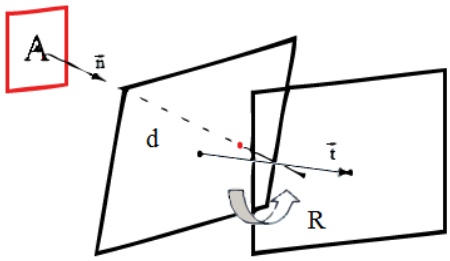


*Abb. 7.5.2: Entzerrungsoperation als Multiple-View-Aufgabe. Anstelle des Dokuments wird jedoch der Sensor rotiert und verschoben.*

Nachdem die Rotationsparameter und die Translationsvektoren der Zeichenebenen im 3D-Kamerakoordinatensystem berechnet wurden, lassen sich die Korrekturabbildungen für die Entzerrung der Aufnahme ohne Weiteres ermitteln. Bei der Aufgabe handelt es sich um eine Instanz des bereits diskutierten Multisicht-Problems (s. Abschnitt 2.6), allerdings mit dem kleinen Unterschied, dass anstelle der Kameraverschiebung die Verschiebung des Objekts (d. h. der Zeichenebenen) modelliert werden muss. Sei $\vec{p}_i^{old}$ – die ursprünglichen Koordinaten des Zeichenmittelpunktes im 3D-Raum und $\vec{p}_i^{new}$ – die korrigierten Koordinaten. Der Zusammenhang zwischen den beiden Kameraкoordinaten ist (s. Abb. 7.5.2) [25]:

$$\vec{p}_i^{new} = \left(R_i - \frac{\vec{t}_i \vec{n}_i^{\,T}}{d_i}\right) \vec{p}_i^{old}$$

Die notwendige Kameraverschiebung $\vec{t}_i$ kann indes wie folgt bestimmt werden:

$$\vec{n}_i^{\,T} \vec{p}_i^{old} = d_i \rightarrow \vec{p}_i^{new} = R_i \vec{p}_i^{old} - \vec{t}_i \rightarrow$$

$$\vec{t}_i = R_i \vec{p}_i^{old} - \vec{p}_i^{new}$$

Die Homographie $H_i$, die den Zusammenhang zwischen dem ursprünglichen und dem korrigierten Ansichten auf die Zeichenebene beschreibt, berechnet sich schließlich aus

$$H_i = C \cdot \left(R_i - \frac{\vec{t}_i \vec{n}_i^{\,T}}{d_i}\right) \cdot C^{-1},$$

wobei $C$ – die Kameramatrix ist (s. Abschnitt 2.6).

## 7.6 Entzerrung der Aufnahmen

Für die Berechnung der entzerrten Version eines Textblocks wird eine Rückwertstransformation (s. Abschnitt 2.5) der Bildpunkte mittels $H_i^{-1}$ durchgeführt, wobei lediglich die Pixel aus dem MUR-Bereich der Zeichen transformiert werden. Diese selektive Vorgehensweise hat den Vorteil, dass die Pixel aus den großen homogenen Zeilen-Zwischenräumen nicht berücksichtigt werden müssen. Die Höhe der MUR wird so angepasst, dass die eventuell vorhandenen abgetrennte Buchstabenteile wie bspw. i-Punkte über bzw. unter den gefundenen Zeichen erfasst werden.

Im Zuge der beschriebenen Entzerrungsmethode findet das Stitching des Dokuments beiläufig statt. Die Zeichenpixel aus den nicht-überlappenden Bildbereichen werden auf die gleiche Weise transformiert wie die restlichen Bildbereiche. Der einzige Unterschied besteht unterdessen darin, dass die erforderlichen Tiefenwerte durch die Spline-*Extrapolation* ermittelt werden müssen. Da die Entzerrung der Dokumente blockweise stattfindet, ist es darauf zu achten, dass ausreichend viele Stützpunkte für die Spline-Interpolation/Extrapolation zur Verfügung stehen.

## 7.7 Auswertung und Zusammenfassung

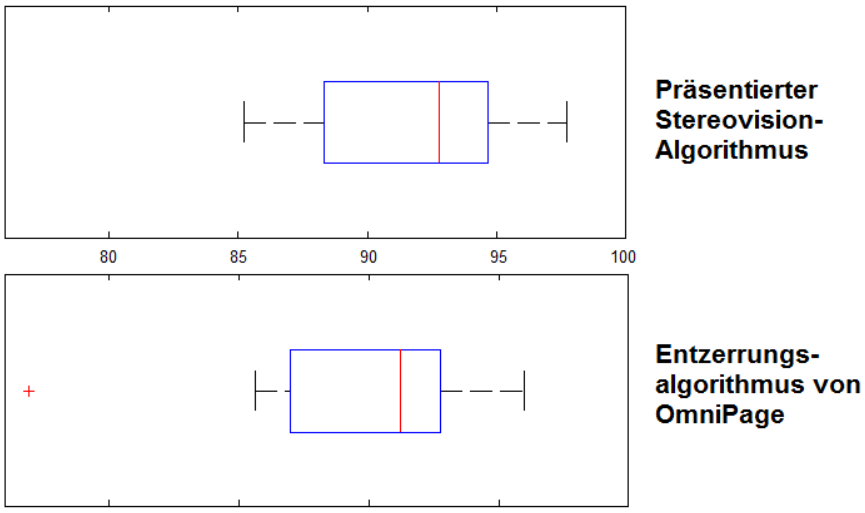


*Abb. 7.6.1: Evaluierungsergebnis des Entzerrungsalgorithmus.*

Die präsentierte Methode dient einer stereovisionbasierten Entzerrung von Dokumenten. Dank der hierarchischen Merkmalstruktur und der bedarfsorientierten Vorgehensweise ist dabei die Kontrolle über die Anzahl der berücksichtigten Distanzwerte und damit auch über den notwendigen Rechenaufwand gegeben, sodass die Laufzeit von unter 3 ms pro Zeichen erreicht werden kann. Es konnte eine Steigerung der Erkennungsraten nach der Korrektur festgestellt werden, allerdings nur bis zu einem bestimmten Grad der Verzerrung. Der Leistungsabfall bei außerordentlich starken Verzerrungen liegt darin begründet, dass die Verformung der Dokumentoberfläche nicht nur eine Deformierung der Zeichenkonturen zur Folge hat, sondern auch zu Fokussierungsproblemen und Helligkeitsschwankungen in den stark verzerrten Bereichen führt. Außerdem liegen die am stärksten verzerrten Regionen i. d. R. am Rande eines Dokuments und befinden sich somit häufig in einem der nicht-überlappenden Bildbereiche, wo die Distanzwerte extrapoliert werden und gerade bei starken Krümmungen mit erheblichen Approximationsfehlern gerechnet werden muss (s. Abb. 7.7.2).

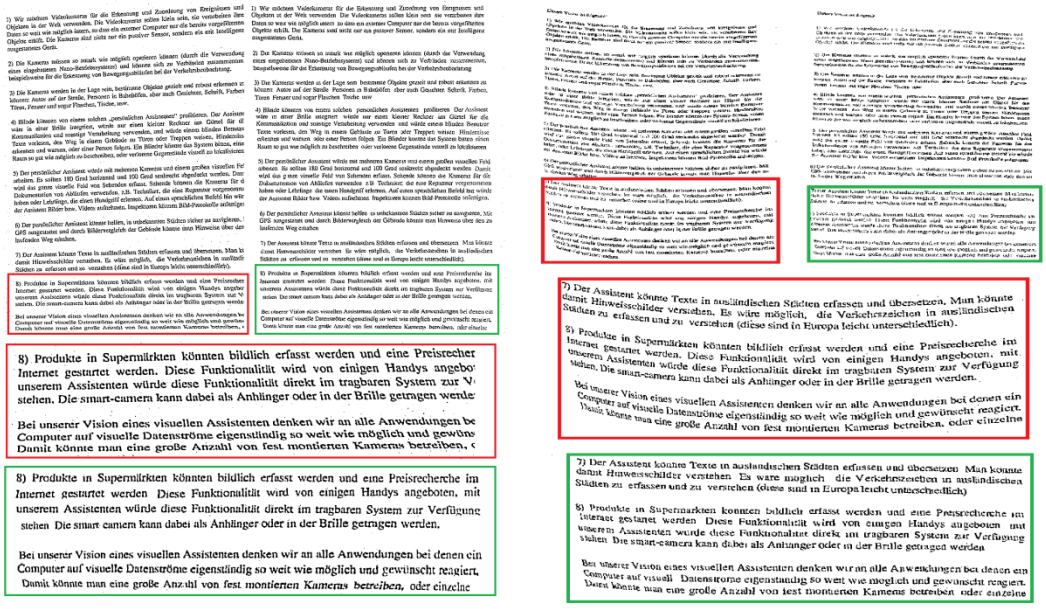

*Abb. 7.7.2: Ergebnis der Entzerrungsoperation. Rot umrandet sind Dokumentbereiche (und deren Vergrößerung) vor der Entzerrung, grün umrandet – dieselben Bereiche danach.*

Alle für die Tests verwendeten Aufnahmen wurden unter Verwendung des Proto-

typen des Geräts manuell angefertigt und ausgewertet. Das Ergebnis der Evaluierung wird in der Abb. 7.6.1 als Boxplot dargestellt. Eine detaillierte Beschreibung ist im Anhang C zu finden. Um die Leistung des Algorithmus zu bewerten, wurde eine Gegenüberstellung der Erkennungsraten jeweils nach der Entzerrung mit der präsentierten Methode und der entsprechenden OmniPage-Funktion angefertigt, wobei eine leichte Überlegenheit des vorgestellten Entzerrungsverfahrens gezeigt wurde.

# 8. Kapitel

# Bestimmung der Vorlesereihenfolge

Das Kapitel behandelt die Untersuchung der logischen Struktur von Dokumenten mit dem Ziel semantische Zusammenhänge zwischen Elementen des physischen Layouts zu identifizieren und darauf basierend die Ausgabereihenfolge zu bestimmen. Die Verarbeitung knüpft an die Ergebnisse der Layoutanalyse (Kapitel 6) sowie der geometrischen Analyse der Dokumentoberfläche (Kapitel 7) an.

## 8.1 Problemstellung

Obwohl es keine allgemeingültigen und allumfassenden Kriterien für die Richtigkeit einer Vorlesereihenfolge gibt, kann die Verständlichkeit der Ausgabe im Falle einer falschen Anordnung der Dokumentkomponenten in einem erhebliche Maße negativ beeinflusst werden. Während einzelne Artikel einer Zeitung i. d. R. in einer beliebigen Reihenfolge gelesen werden können, gehört die Anordnung von den Zeilen innerhalb eines Artikels zu den inhaltlichen Informationen und ist von kritischer Bedeutung. Weil semantische Zusammenhänge zwischen den einzelnen Layoutelementen sich nicht immer zuverlässig aus den geometrischen Eigenschaften ableiten lassen [158], werden an dieser Stelle textspezifische Merkmale als zusätzliche Informationsquelle in die Berechnung miteinbezogen. Die Bestimmung der Vorlesereihenfolge ist der letzte Schritt der Vorverarbeitung, sodass zahlreiche textspezifische Merkmale wie die Schriftgrößen und Zeilenorientierungen der Textblöcke stehen zu diesem Zeitpunkt bereits zur Verfügung.

Die Anordnung der einzelnen Zeichen und Zeilen in den Textregionen wurde bereits während der Zeilenextraktion als Teil der Layoutanalyse (s. Abschnitt 6.5) festgelegt. Gesucht wird ein Verfahren zur effizienten Identifikation von inhaltli-

chen Zusammenhängen zwischen den Textregionen, die im Laufe der Layoutanalyse erkannt wurden. Eine möglicherweise nicht-achsensymmetrische Orientierung des Dokuments in Verbindung mit einer möglichen nicht-linearen Verzerrung der Oberfläche kann die Messungsgenauigkeit bei der Extraktion der textspezifischen Merkmalen beeinträchtigen, was im weiteren Verlauf der Verarbeitung berücksichtigt werden muss. Es wird eine universelle Lösung benötigt, die für verschiedene Layouts ein sinnvolles Ergebnis produziert, wobei für die folgenden drei Dokumenttypen die Vorlesereihenfolge explizit spezifiziert wird (s. Tabelle 8.1.1).

*Tabelle 8.1.1: Dokumentspezifische Anforderungen an eine sinnvolle Vorlesereihenfolge.*

| **Dokumentart** | **Reihenfolge** |
|---|---|
| *Zeitung* | Artikelweise: Überschrift, Untertitel, Textspalten |
| *Buch* | Seitenweise, ggf. Spalten einer Seite erkennen |
| *Brief* | Absenderadresse, Empfängeradresse, Betreffzeile, Briefkörper, Kontaktdaten |
| *Sonstige* | Von oben nach unten: Überschrift, zugehörige Spalten (von links nach rechts) |

Sollten sich mehrere Dokumente innerhalb einer Aufnahme befinden, so werden diese getrennt voneinander in einer beliebigen Reihenfolge ausgegeben, wobei der Anfang eines neuen Dokuments sprachlich markiert werden soll. Aus Effizienzgründen wird auf die Identifikation von speziellen Layoutelementen wie Kopfzeilen, Fußzeilen, Fußnoten und Tabellen verzichtet, da die dafür notwendige Detektion von Separator-Linien und anderen Sondermarkierungen angesichts einer eventuell vorhandenen Verzerrung der Dokumentoberfläche sowie der Verwechslungsgefahr mit Objekten einer natürlichen Umgebung einen zusätzlichen und schwer abzuschätzenden Rechenaufwand bedeuten könnte.

## 8.2 Vorarbeiten

Die Ableitung der logischen Struktur eines Dokuments nur anhand seines physischen Layoutmodells ist prinzipiell spekulativ [158]. Einige der Autoren [159]

[160] schlagen deshalb vor, semantische und linguistische Informationen in die Analyse einzubeziehen, was allerdings eine im Vorfeld stattfindende Zeichenerkennung voraussetzen würde. Bei einem anderen, häufig verwendeten Ansatz werden typenspezifische Dokumentmodelle definiert, die für einen bestimmten Einsatzbereich zugeschnitten sind [161][162]. Neben solchen spezialisierten Verfahren gibt es auch Methoden, die eine automatische Typenklassifizierung durchführen und auf diese Weise a priori Informationen über die Dokumentstruktur bestimmen [161][163]. Schließlich existiert eine Reihe von universellen Ansätzen, die auf der Grundlage von Annahmen über die Formatierungskonventionen die hierarchischen Beziehungen zwischen den Komponenten eines generischen Dokuments ermitteln [164][165].

Je nach Zielsetzung variieren auch die eingesetzten Layout-Repräsentationen, wobei zwei davon eine besondere Bedeutung haben: mittels Bäumen [166] und über formale Sprachen [167]. Abhängig von der gewählten Darstellung werden entweder Regeln zur Baumtransformation oder formale Grammatiken definiert, mit deren Hilfe die Identifikation der Elemente des logischen Layouts oder die Feststellung von hierarchischen Beziehungen zwischen den Elementen oder beides stattfindet. Stochastische Modelle [163][168] bilden eine besondere Klasse von Layoutrepräsentationen und ermöglichen eine Klassifizierung des Dokumenttyps, die parallel zur Layouterkennung stattfindet.

Die überwiegende Mehrheit der erwähnten Verfahren wurde für den Einsatz im Bereich der Verwaltung von digitalen Dokumentarchiven entwickelt. Während die u. U. nicht-achsensymmetrischen Ausrichtung des Dokumentes von den Autoren berücksichtigt und sogar als Klassifikationsmerkmal eingesetzt wird [169], findet die Verzerrungsproblematik i. d. R. keine Beachtung und ist ein besonderes Merkmal der gegebenen Aufgabenstellung.

## 8.3 Universelles Modell der logischen Dokumentstruktur

Unter den drei explizit spezifizierten Dokumenttypen *Zeitungen, Briefe* und *Bücher* sind es die Zeitungen, die häufig eine besonders komplexe Layoutstruktur aufweisen, wobei die einzelnen Artikel einerseits und die Bild-Bildunterschrift-Paare andererseits jeweils untrennbare semantische Einheiten bilden. Zu den wesentlichen Bestandteilen eines Artikels gehören eine *Überschrift,* ein *Vorspann* und eine oder mehrere *Textspalten.* Um die spezifizierte Vorlesereihenfolge für

Zeitungen erzielen zu können, müssen diese vier Komponenten richtig identifiziert werden. Alle weiteren Layout-Elemente wie Kopf- und Fußzeilen können dabei als Textspalten ohne Überschrift und somit als eigenständige Artikel angesehen werden. Das beschriebene vereinfachte Modell eines Zeitungsartikels dient als Vorbild für das universelle Modell der logischen Dokumentstruktur (s. Abb. 8.3.1), das in einer verallgemeinerten Form zur Analyse der Inhaltsstruktur eines generischen Dokuments eingesetzt wird.

Eine Aufnahme kann mehrere Dokumente enthalten, wobei zwei Textblöcke genau dann zu einem Dokument gehören, wenn sie sich einen Textträger teilen. Eine Menge von Textregionen, zwischen denen ein semantischer Zusammenhang besteht und die sequenziell auszugeben sind, wird künftig als Abschnitt bezeichnet. Da eine Erkennung von Abschnitten, deren Teile auf zwei oder mehr Seiten verteilt sind, ohne Informationen über den Textinhalt problematisch ist, ist die Zuordnung von Artikeln zu Seiten eindeutig. Die richtige Vorlesereihenfolge ist in einem solchen Fall durch die Anordnung der Seiten zu gewährleisten. Ein Abschnitt enthält mindestens eine Textregion, die als "Textspalte/n" bezeichnet wird. Eine "Überschrift" hat nur wenige Zeilen und erstreckt sich über alle Spalten des Artikels, allerdings muss es sich dabei nicht um eine Überschrift im herkömmlichen Sinne handeln – so kann bspw. die Betreffzeile eines Briefes als Überschrift erkannt werden. Das Paar Bild-Bildunterschrift bildet ebenfalls eine semantische Einheit und wird als "Grafik" bezeichnet, wobei eine Bildunterschrift sich immer unterhalb einer Abbildung befindet. Bildunterschriften werden beim Vorlesen sprachlich gekennzeichnet.

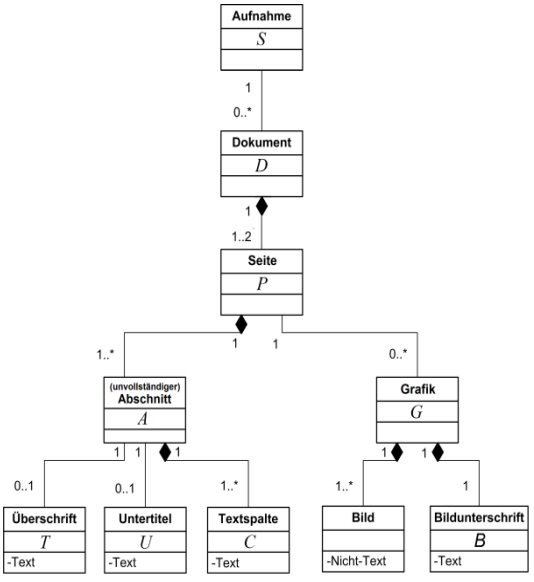


*Abb. 8.3.1: Das Universalmodell der logischen Layoutstruktur*

Die Einfachheit des präsentierten Modells ist eine Folge der Universalitäts-Anforderung – die wenigen allgemeingültigen Formatierungsannahmen, die für ein generisches Dokument gemacht werden können, lassen keinen Raum für eine auf-

wändigere Modellierung. Anstatt dass etwaige, dokumenttypspezifische Layout-Elemente als Sonderfälle behandelt werden, soll ein generischer Ansatz möglichst viele Spezialfälle abdecken und in jedem Fall ein zufriedenstellendes Ergebnis produzieren.

## 8.4 Merkmalsextraktion

Die Untersuchung der logischen Dokumentstruktur basiert auf den Ergebnissen der Analyse des physischen Layouts (s. Kapitel 6), d. h. der geometrischen und textspezifischen Merkmale von den extrahierten Textregionen. Allerdings können die Merkmale einen nicht zu vernachlässigenden Messfehler infolge der Krümmung der Dokumentoberfläche aufweisen, sodass eine Korrektur der Merkmalswerte unter Verwendung des im Kapitel 7 berechneten Oberflächenmodells durchgeführt werden muss. Mit den korrigierten Werten wird anschließend eine angepasste Repräsentation der physischen Layoutstruktur berechnet, die Informationen über die Nachbarschaftsbeziehungen zwischen den Textblöcken beschreibt.

### 8.4.1 Korrektur verzerrungsbedingter Messfehler

So wie viele regelbasierten Verfahren ist die im nächsten Abschnitt vorgestellte Klassifizierungsmethode empfindlich gegenüber Ungenauigkeiten des physischen Layoutmodells, auf dessen Grundlage die Bestimmung der logischen Dokumentstruktur stattfindet. Mit einer Kamera gemachte Dokumentaufnahmen sind indes besonders anfällig gegenüber der Verzerrungsproblematik. Folgende Korrekturschritte können u. U. nötig sein, um einige wichtige Fehlerquellen zu beseitigen:

- *Festlegung der globalen Zeilenorientierung.* Die Ausrichtung der Zeilen einer Region ist ein wichtiges Klassifizierungsmerkmal, dessen Extraktion infolge

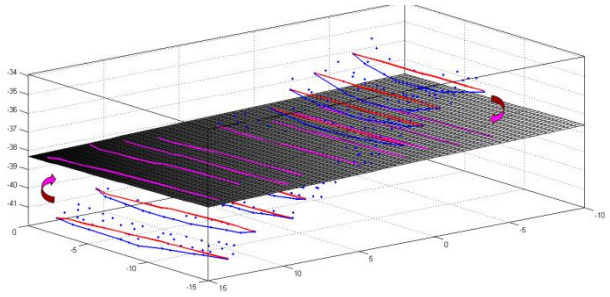

*Abb. 8.4.1: Extraktion der Zeilenorientierung. Die Zeichenpunkte (blau) werden auf eine gemeinsame Ebene rotiert (magenta).*

von Zeilenkrümmungen problematisch sein kann. Mit den ermittelten Normalvektoren $\vec{n}_i$ der Zeichenebenen im 3D-Raum (s. Abschnitt 7.4) lässt sich die Begradigung der Zeilen vornehmen, indem die im Dokument enthaltenen Zeichen auf eine gemeinsame frontal ausgerichtete Dokumentebene rotiert werden (s. Abb. 8.4.1). Die Rotation wird um die Schnittgerade der jeweiligen Zeichenebene mit der Dokumentebene ausgeführt. Zwei Endpunkte einer Zeile reichen aus, um eine grobe Abschätzung der Zeilenausrichtung einer Region vorzunehmen, sodass der Zeitaufwand dieses Korrekturschrittes vernachlässigbar klein ist.

- *Anpassung der Schriftgrößen.* Perspektivische Verzerrungseffekte verfälschen auch die MUR-Maße von den Buchstaben, sodass die Schriftgrößen in den Regionen nicht ohne weiteres als Ähnlichkeitsmerkmal verwendet werden können. Eine erneute Berechnung der Schriftgrößen anhand der entzerrten Versionen der Textblöcke kann der Problematik entgegen wirken.
- *Korrektur des geometrischen Layoutmodells.* Infolge der willkürlichen Orientierung der Regionen und der Verzerrung der Dokumentoberfläche ist die Messung der Manhattan-Distanzen zwischen den Textbereichen nicht ohne weiteres möglich. Bei der Untersuchung der logischen Dokumentstruktur (s. Abschnitt 7.3) spielen vor allem die Abstände zwischen den benachbarten Seiten zweier Textregionen sowie deren relativer Versatz und ihre Überlappung eine wichtige Rolle. Die Berechnung der Distanzen zwischen den benachbarten Textblöcken erfolgt mit einer Methode, die der Trapez-Methode zur Bestimmung der Distanzen zwischen den Zeilen eines Blocks im Entzerrungsschritt der Verarbeitung (s. Abb. 7.5.1) ähnlich ist. Dabei wird die Höhe eines Trapezes ermittelt, die aus den benachbarten MUR-Seiten sowie zwei Verbindungslinien zwischen den beiden besteht. Für eine sorgfältige Messung der Seitenlängen kann auch hier Spline-Interpolationsverfahren verwendet werden.
- *Eindeutige Bestimmung der Orientierung eines Textblocks.* Der Orientierungswinkel der Zeilen sagt nichts darüber aus, wo sich der obere Teil einer Textregion befindet. Die definitive Feststellung der Orientierung erfordert eine OCR-Analyse der Zeichen, allerdings sind eine heuristische Methoden ebenfalls möglich. So können Punktobjekte Hinweise über die Textrichtung einer Zeile liefern – einerseits als Satzzeichen und andererseits als Buchstaben-Komponenten. Bei der Zeilenextraktion (s. Abschnitt 6.5) werden die Punkte zwar extrahiert und einer Zeile zugeordnet, eine sichere Identifikation von

Punkten findet jedoch nicht statt. Satzpunkte werden daran erkannt, dass sie sich am Ende einer Zeile oder eines Wortes* befinden. Um die Orientierung der Regionen zu bestimmen, werden die Abstände zwischen den Punkten und Zeichen, denen sie zugeordnet wurden, analysiert. Das Intervall links von einem Satzzeichen sollte kleiner sein als rechts, während sich die Buchstaben-Punkte oberhalb der naheliegenden Zeichen befinden sollten. Ungünstige Verzerrungen der Dokumentoberfläche und Binarisierungsfehler können u. U. zu Klassifikationsfehlern führen. Bei widersprüchlichen Ergebnissen erfolgt die Entscheidung auf der Grundlage kleiner Ausschnitte aus der jeweiligen Region, für die probeweise eine Zeichenerkennung veranlasst wird.

### 8.4.2 Berechnung des Nachbarschaftsgraphen

Die Topologie der Layoutkomponenten bildet eine wichtige Grundlage für die Analyse der logischen Dokumentstruktur. Die im Kapitel 6 verwendete Repräsentation des physischen Layouts enthält lediglich Koordinaten von den Regionsgrenzen, sodass weitere Verarbeitungsschritte nötig sind, um daraus die Nachbarschaftsbeziehungen abzuleiten. Das Ergebnis wird in Form eines gerichteten Graphen $G = (V, E)$ mit folgenden Eigenschaften dargestellt:

- jeder Textblock wird von einem Knoten repräsentiert
- unmittelbare Nachbarschaften werden durch gerichtete Kanten gekennzeichnet
- inhaltlich zusammenhängende Regionen bilden einen zusammenhängenden Untergraphen

Jede Nachbarschaftsrelation wird charakterisiert durch:

- eine Markierung gemäß dem Vierer-Nachbarschaftskonzept: eine Kante zeigt dabei in die Richtung des rechten Nachbarn bzw. des unteren Nachbarn
- Überlappungsinformationen: Seitenabstände

Die benachbarten Regionen werden im späteren Verlauf der Verarbeitung paarweise untersucht. Maßgeblich für die Bewertung einer Nachbarschaftsbeziehung

* Die Identifikation der Wort-Grenzen ist ein Teil des Stereo-Matching-Algorithmus im Abschnitt 7.3.

ist der Minimalabstand zwischen den MURs der Regionen. Annähernd parallel ausgerichtete Seitenpaare mit dem kleinsten Abstand werden im Folgenden als *benachbarte Seiten* bezeichnet. Damit zwei Textregionen als Nachbarn erkannt werden, müssen folgenden Eigenschaften erfüllt sein:

1. die Ausrichtung der Textzeilen in den beiden Regionen ist vergleichbar
2. benachbarte Seiten müssen ähnlich ausgerichtet sein und überlappen
3. es befindet sich keine weitere Region zwischen zwei benachbarten Seiten
4. der Abstand zwischen den Seiten ist kleiner als ein schriftgrößenabhängiger Schwellenwert (s. Abschnitt 5.1.1)

Direkt nach der Initialisierung enthält der Graph $G$ keine Kanten und einen Knoten für jede Textregion. Für eine schnelle Abschätzung der Entfernung zwischen zwei Regionen werden die Mittelpunkte ihrer MURs mit einer Geraden verbunden. Das Segment der Verbindungslinie, welches sich außerhalb der beiden Regionen befindet, wird in zwei Komponenten $l_x, l_y$ gemäß dem Orientierungswinkel der Zeilen $\alpha_1$ zerlegt (s. Abb. 8.4.2):

$$L_{AB} = \vec{w}_1 \cdot \vec{L} \Big/ \|\vec{w}_1\|; \quad L_{CD} \approx \vec{w}_2 \cdot \vec{L} \Big/ \|\vec{w}_2\|$$

und damit

$$l_x = L_{BC} = |x_1 - x_2| \cdot \cos(\alpha_1) - L_{AB} - L_{CD};$$
$$l_y - analog$$

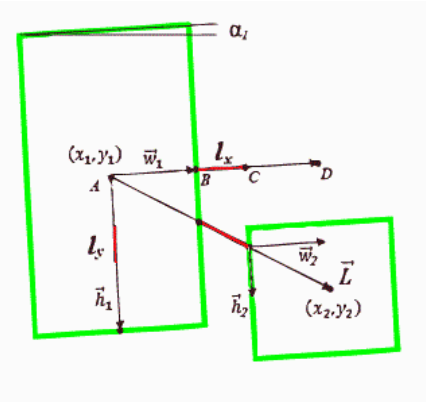


*Abb. 8.4.2: Maß für die Abstände zwischen Texblöcken Nachbarschaftsgraph.*

Sind die beiden Komponenten kleiner als ein schriftgrößenabhängiger Schwellenwert $\Delta_{i,j} = 0.5 \cdot (Font_i + Font_j)$, dann wird eine neue Kante in den Nachbarschaftsgraphen eingefügt, wobei die relative Lage der Regionen nach dem Konzept der Vierer-Nachbarschaft (linker/rechter bzw. oberer/unterer Nachbar) an der Kante vermerkt wird. Die jeweils am nächsten liegende Region wird ebenfalls ermittelt, und zwar unabhängig davon, ob es sich dabei um eine Nachbarn im Sinne der spezifizierten Eigenschaften handelt (in Abb. 8.4.3 blau markiert).

Der nach der oberen Berechnungsvorschrift identifizierten Nachbarschaften sind nicht immer eindeutig, da eine Region mehrere Nachbarn mit dem Abstand unter dem Schwellenwert $\Delta_{i,j}$ haben kann. Um die Anzahl der Kanten im Graphen zu reduzieren, werden zwei minimale Spannbäume $B_v = (V, E'_v)$, $B_h = (V, E'_h)$ berechnet – jeweils einen für die vertikalen und horizontalen Kanten gemessen an der Ausrichtung der Zeilen. Als Kantengewichte dienen dabei die Abstände zwischen den benachbarten Seiten, sodass unmittelbare Nachbarschaften in Übereinstimmung mit der 2. Eigenschaft von Nachbarschaftsbeziehungen favorisiert werden. Der resultierende Nachbarschaftsgraph entsteht durch die Vereinigung der Kantenmengen: $G = (V, E'_h \cup E'_v)$.

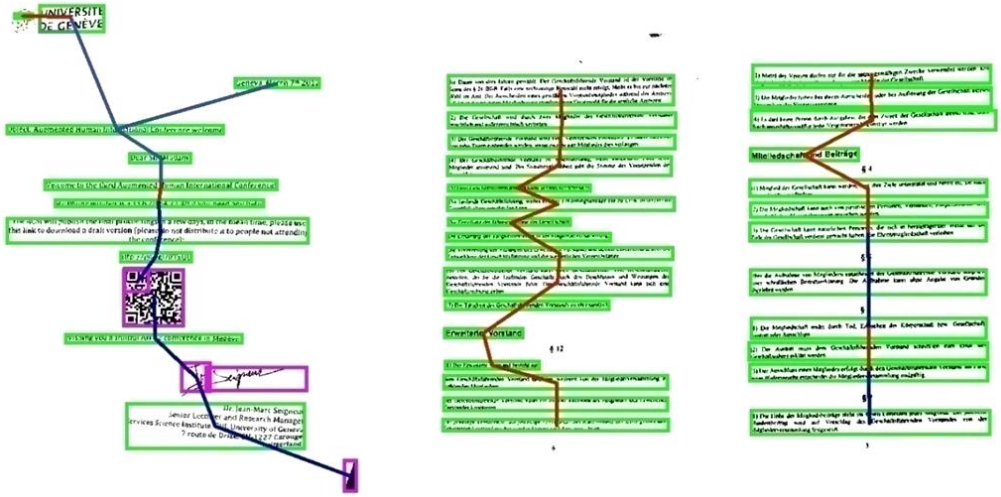

*Abb. 8.4.3: Visualisierung des Nachbarschaftsgraphen. Rote Linien markieren Nachbarschaftsbeziehungen von Textregionen. Mit Blau ist die jeweils am nächsten liegende Region gekennzeichnet.*

## 8.5 Regelbasierte Analyse

Das Thema dieses Abschnitts ist die Analyse der logischen Struktur von Dokumenten sowie die Bestimmung einer strikten Totalordnung für die Ausgabe der Textblöcke. Die Analyse erfolgt in drei Schritten:

1. *Abschnittserkennung:* Erkennung von inhaltlich abgeschlossenen Kompositionen von Textregionen
2. *Makroanalyse:* Sortierung der Abschnitte
3. *Mikroanalyse:* Sortierung der Regionen in den Abschnitten

Die gestaffelten Vorgehensweise bei der Sortierung der Regionen soll die Schwächen des vorgestellten Universalmodells (s. Abb. 8.3.1) kompensieren, die es für spezielle, komplexe Layoutstrukturen nur bedingt geeignet machen. Eine intelligente Anordnung der Abschnitte während der Makroanalyse ist nämlich immer dann von wichtiger Bedeutung, wenn die Mikroanalyse infolge einer Übersegmentierung der Abschnitte keine zufriedenstellende Vorlesereihenfolge produzieren kann.

### 8.5.1 Logische Segmentierung des Dokuments

Die Analyse beginnt mit einer hierarchischen Klassifizierung, bei der die einzelnen Textstellen als Elemente der logischen Layoutstruktur erkannt und basierend auf den semantischen Relationen zu Abschnitten, Dokumenten, Seiten zusammengefasst werden. Der hierarchische Aufbau des Dokuments wird gemäß dem Universalmodell (s. Abschnitt 8.3) als Baum dargestellt.

Als Erkennungsmodell für die Untersuchung des logischen Layouts wird eine kontextfreie Grammatik $Gr = \{N, \Sigma, P, S\}$ eingesetzt, sodass die Analyse als Parsing von Wörtern einer formalen Sprache aufgefasst werden kann, wobei das Layout eines Dokuments in einer kodierten Form darstellt wird. Die Eigenschaften der physischen Dokumentstruktur – die Klassifizierung als *Textblock* $(t)$ bzw. *Nicht-Text-Region* $(n)$ sowie die Nachbarschaftsbeziehungen *unterer Nachbar* $(v)$ bzw. *rechter Nachbar* $(h)$ – werden mit Hilfe der Terminalsymbole repräsentiert

$$\Sigma = \{t, n, v, h\}$$

Die Regionen $t_i$ werden außerdem durch die Schriftgröße $font(t_i)$, Anzahl der Textzeilen $lines(t_i)$, MUR-Abmessungen Breite $width(t_i)$ und Höhe $height(t_i)$ sowie Überlappungen $(t_i h t_j, t_i v t_j)$ charakterisiert, wobei diese zusätzliche Merkmale der Übersichtlichkeit halber als Verweis angegeben werden: $t^{\{a,b,c\}}, v^{\{a,b,c\}}$ oder $h^{\{a,b,c\}}$. Es sei an dieser Stelle darauf hingewiesen, dass aus formaler Sicht es sich dabei um unterschiedliche Terminalsymbole handelt, d. h. $t^a \neq t^b$.

Die Menge der Nichtterminalsymbole besteht aus den Elementen der logischen Dokumentstruktur (s. Abb. 8.3.1):

$$N = \{S, D, P, A, T, U, C, B\}$$

Gesucht wird eine Zerlegung des Layouts, die durch eine Ableitung bzgl. der Grammatik *Gr* repräsentiert ist. Der Nachbarschaftsgraph wird hierfür als Folge von Terminalsymbolen dargestellt und die Klassifikationsregeln werden dabei als Produktionsregeln von *Gr* definiert:

- **Klassifizierung nach Dokumenten:** $S \rightarrow \varepsilon$; $S \rightarrow D^*$; $D \rightarrow P \mid P h^b P$
  a. Eine Aufnahme kann beliebig viele Dokumente enthalten
  b. Ein Dokument besteht aus einer oder zwei horizontal ausgerichteten Seiten
- **Erkennung von Seiten:** $P \rightarrow A^* v^a G^* v^a A^*$
  a. Alle Textregionen haben vergleichbare Zeilenorientierungen (auch 180°-Rotationen werden erkannt (s. Abschnitt 8.4)) und Distanzwerte
- **Erkennung von Grafiken/Bildunterschriften:** $B \rightarrow t^{abc}$; $G \rightarrow n \mid n v B$
  a. Oberhalb einer Textregion befindet sich ein Nicht-Text-Element
  b. Enthält nur wenige Zeilen: $lines(R_i) < 4$
  c. Schriftgröße ist kleiner als die der unteren Nachbar oder kein unterer Nachbar mit einer vergleichbaren MUR-Breite existiert:
$$\forall j(R_i v R_j): font(R_i) < font(R_j) \vee$$
$$\nexists j(R_i v R_j): width(R_i) \approx width(R_j)$$
- **Erkennung von Abschnitten:** $A \rightarrow T v^{bc} A_{part} \mid T v U v^{bc} A_{part} \mid A_{part}$
  a. Ein Abschnitt besteht aus einer Überschrift, ggf. einem Untertitel und mehreren Textspalten.
  b. Die Schriftgrößen der Textspalten sind kleiner als die Schriftgrößen der Überschrift und des Untertitels, die sich über den Textspalten befinden:
$$\forall i, j(R_i v R_j): font(R_i) > font(R_j)$$
  c. Textspalten werden einem Abschnitt zugeordnet falls eine Überlappung mit derselben Überschrift oder Untertitel besteht
- **Erkennung von unvollständigen Abschnitten:** $A_{part} \rightarrow C \mid C h^a A_{part} \mid A_{part} h^a C$
  a. Textspalten werden einem Abschnitt zugeordnet falls ihre Schriftgrößen sowie MUR-Breiten übereinstimmen:
$$\forall i, j(R_i h R_j): font(R_i) \approx font(R_j) \wedge width(R_i) \approx width(R_j)$$
- **Erkennung von Überschriften:** $T \rightarrow t^a \mid t v^b T \mid T h^c T$
  a. Eine Überschrift hat nur wenige Zeilen und eine größere Schrift als der untere Nachbar:

$$\forall j\left(R_i v R_j\right): font(R_i) > font\left(R_j\right) \wedge lines(R_i) \leq 3$$

b. Enthält bis zu drei Zeilen, die eine ähnliche Schriftgröße haben:
$$\exists j\left(R_i v R_j\right): font(R_i) \approx font\left(R_j\right) \wedge lines(R_i) + lines\left(R_j\right) \leq 3$$

c. Besteht aus mehreren horizontal ausgerichteten Segmenten in Folge einer Übersegmentierung:
$$\exists j\left(R_i h R_j\right): font(R_i) \approx font\left(R_j\right) \wedge lines(R_i) = lines\left(R_j\right)$$

- **Erkennung von Untertiteln:** $\boldsymbol{U \rightarrow t^a \mid t v^b U}$
  a. Ein Untertitel hat nur wenige Zeilen und eine größere Schrift als der untere Nachbar:
  $$\forall j\left(R_i v R_j\right): font(R_i) > font\left(R_j\right) \wedge lines(R_i) \leq 10$$
  b. Enthält bis zu fünf Zeilen, die eine ähnliche Schriftgröße haben:
  $$\exists j\left(R_i v R_j\right): font(R_i) \approx font\left(R_j\right) \wedge lines(R_i) + lines\left(R_j\right) \leq 10$$
- **Erkennung von Textspalten:** $\boldsymbol{C \rightarrow t \mid t v^{ab} C \mid C v^{ab} t \mid C v^{ab} G v^{ab} C}$
  a. Eine Textspalte besteht aus mehreren Textregionen, die vertikal ausgerichtet sind (Paragraphen):
  $$\exists! j\left(R_i v R_j\right): left(R_i) \approx left\left(R_j\right) \wedge width(R_i) \approx width\left(R_j\right)$$
  b. Sind gekennzeichnet durch eine ähnliche Schriftgröße:
  $$\forall i, j\left(R_i v R_j\right): font(R_i) \approx font\left(R_j\right)$$
  c. Kann Grafiken enthalten

Die Terminalsymbole $\{h, v\}$ in den Zeichenfolgen repräsentieren Nachbarschaftsrelationen zwischen den Textregionen. Wird eine gültige Zeichensequenz $w = s_1 \ldots s_N \in L(Gr)$ so aufgetrennt, dass in sämtlichen Teilwörtern die Terminalzeichen $\{t, n\}$ und $\{h, v\}$ abwechselnd auftreten

$$Partition(w) = \{w_k\} = \left\{s_{I(k)+1} \ldots s_{I(k+1)} \middle| k \in 1..|I| - 1\right\}$$

$$mit\ \ I = \{0\} \cup \left\{\, i \mid s_i, s_{i+1} \in \{t, n\}\right\} \cup \{N\},$$

dann entspricht das einer Partitionierung des Nachbarschaftsgraphen in mehrere zusammenhängende Untergraphen. Durch die Gestaltung der Produktionsregeln ist sichergestellt, dass ein Wort, welches von dem Nichtterminalsymbol *A (Abschnitt)* ausgehend abgeleitet werden kann, eine solche alternierende Zeichenabfolge darstellt:

$$(A \Rightarrow_{Gr}^{*} w \wedge w \in \Sigma^{*}) \rightarrow Partition(w) = \{w\}$$

Es lässt sich einfach per Induktion über die Wortlänge zeigen, dass ausgehend von (*A, T, U, $A_{part}$, C, G*) nur alternierende Teilfolgen produziert werden können. Werden nun die Nachbarschaftsbeziehungen in den Untergraphen als inhaltliche Relationen interpretiert, dann kann die oben beschriebene Partitionierung als Zerlegung der Textregionen in inhaltlich zusammenhängende Klassen aufgefasst werden. Je mehr inhaltliche Relationen identifiziert werden konnten, desto länger werden die Teilsequenzen $w_k \in Partition(w)$, sodass ein Maß für die Komplexität der Layoutstruktur wie folgt definiert werden kann:

$$Score(w_{ges}) = \frac{1}{|w_{ges}|^2} \sum_{w_k \in Partition(w_{ges})} (|w_k|)^2$$

Als Vorlage für die Gliederung des Dokuments wird dabei das Wort bzw. eines der Wörter $w_{max}$ mit der höchsten Bewertung $max\left(Score(w_{ges})\right)$ ausgewählt. Die oben aufgelisteten Produktionsregeln sind nur eine formale Darstellung der implementierten Klassifizierungsmethode. Um den Suchraum möglicher Zeichenfolgen zu reduzieren, findet die Analyse stufenweise statt:

1. Die Klassifizierung der Regionen nach Dokumenten erfolgt bereits im Zuge der Erstellung des Nachbarschaftsgraphen (s. Abschnitt 8.4.2) und wird hier nur der Vollständigkeit halber aufgeführt
2. Identifikation von Überschriften, Untertiteln, Unterschriften, Textspalten
3. Identifikation von vollständigen Abschnitten *A*
4. Alle Textregionen eines Dokuments, die nicht einem *Abschnitt* oder einer *Grafik* zugeordnet werden konnten, werden vorerst als eigenständige Textabschnitte ohne Überschrift $A_{part}$ angesehen

Die Produktionsregeln sind derart konzipiert, dass der Ableitungsbaum eines Wortes *w* eine Struktur entsprechend dem Universalmodell aufweist, sodass wichtige Informationen über den logischen Aufbau eines Dokuments, darunter die Zugehörigkeit der Textregionen zu verschiedenen Abschnitten, daraus abgelesen werden können. Durch die Gruppierung der Knoten aus $G$ gemäß dem konstruierten Syntaxbaum wird ein Nachbarschaftsgraph der Abschnitte[*] $\mathcal{G} = \{\mathcal{A}, \mathcal{E}_h \cup \mathcal{E}_v\}$

[*] Der Einfachheit halber werden sowohl eine Partition der Regionen-Menge als auch die Knotenmenge im Nachbarschaftsgraphen der Abschnitte als $\mathcal{A}$ bezeichnet, wobei jeder Knoten ein Element der Partition repräsentiert.

produziert, in dem jeder Knoten $A \in \mathcal{A}$ einen ganzen Abschnitt des Dokuments repräsentiert. Analog zum Nachbarschaftsgraphen der Regionen $G$ besitzen alle Knoten $A \in \mathcal{A}$ aus $\mathcal{G}$ folgenden Eigenschaften:

- *MUR*($A$) des Abschnitts: MUR der Eckpunkte der zugehörigen Regionen
- Schriftgrößen in den Spalten sowie in der Überschrift:
  *columnFont(A), titelFont(A)*
- Anzahl der Spalten: $cols(A)$

Zwei Knoten des neuen Graphen sind unterdessen genau dann durch eine vertikale ($\mathcal{E}_v$) und/oder horizontale ($\mathcal{E}_h$) Kante verbunden, wenn wenigstens eine Nachbarschaftsbeziehung zwischen den Regionen der Abschnitte existiert.

### 8.5.2 Vervollständigung der Abschnitte

Würde es sich bei den ermittelten Abschnitten immer um vollkommen isolierte und semantisch unabhängige Komponenten eines Dokuments handeln, ließe sich die Ausgabereihenfolge für sie beliebig gestalten. Diese Annahme ist jedoch angesichts der großen Vielfalt der möglichen Layouttypen unrealistisch, da die im Abschnitt 8.5.1 präsentierte Partitionierungsmethode physische Nähe von inhaltlich zusammenhängenden Regionen voraussetzt. Dies ist jedoch nicht immer erfüllt. Infolge von Unregelmäßigkeiten in der Layoutstruktur kommt es gelegentlich zu einer Übersegmentierung der Dokumentstruktur, wobei Abschnittsteile als eigenständige *unvollständige Abschnitte* klassifiziert werden. Oftmals lassen sich die Lücken zwischen den Textregionen überbrücken, indem die Nachbarschaftsbeziehungen ($\mathcal{E}_h \cup \mathcal{E}_v$) zwischen den unvollständigen ($\mathcal{A}_{part}$) und vollständigen Abschnitten ($\mathcal{A}_{comp}$) in $\mathcal{G} = \{\mathcal{A}_{comp} \cup \mathcal{A}_{part}, \mathcal{E}_h \cup \mathcal{E}_v\}$ analysiert werden. Der Algorithmus *Pseudocode 8.5.1* sucht nach Kombinationsmöglichkeiten in $\mathcal{G}$ und wird solange wiederholt, bis schließlich keine Korrekturen mehr vorgenommen werden können.

```
FORALL A1, A2 ∈ 𝒜
    // vollständige Abschnitte werden nicht miteinander kombiniert
    IF A1 ∈ 𝒜comp ∧ A2 ∈ 𝒜comp
        CONTINUE
    END

    // vollständige Abschnitte erweitern
    IF A1 ∈ 𝒜comp ∧ A2 ∈ 𝒜part
        // die Überschrift erweitern
        IF titelFont(A1) = titelFont(A2)∧(A2, A1) ∈ ℰv
            Merge(A1, A2)
            Remove(𝒜part, A2)
        END
        // die Spaltenmenge erweitern
        IF columnFont(A1) = columnFont(A2)
            //eine Region ist umgeben von den Spalten des Abschnitts
            IF (A1, A2) ∈ ℰh∧(A2, A1) ∈ ℰh
            Merge(A1, A2)
                Remove(𝒜part, A2)
            END
            // eine Region liegt am Rande des Abschnitts
            IF ∃A3: ((A3, A1) ∈ ℰh ∧(A3, A2) ∈ ℰh∧(A2, A1) ∈ ℰh) ∨
                ((A1, A3) ∈ ℰh ∧(A2, A3) ∈ ℰh ∧(A1, A2) ∈ ℰh)
                Merge(A1, A2)
                Remove(𝒜part, A2)
            END
        END
    END

    // zwei unvollständige Abschnitte vereinen
    IF A1 ∈ 𝒜part ∧ A2 ∈ 𝒜part
        IF columnFont(A1) = columnFont(A2)
            IF (A1, A2) ∈ ℰh
                Merge(A1, A2)
                Remove(𝒜part, A2)
            ELSE IF (A2, A1) ∈ ℰh
                Merge(A2, A1)
                Remove(𝒜part, A1)
            END
        END
    END
END
```

*Pseudocode 8.5.1: Untersuchung der Kombinationsmöglichkeiten von Abschnitten.*

Die Methode *Merge()* hat die Aufgabe, Textregionen eines unvollständigen Ab-

schnittes in die Struktur eines anderen Abschnittes zu integrieren und den Nachbarschaftsgraphen $\mathcal{G}$ entsprechend zu aktualisieren.

### 8.5.3 Makrostruktur von Dokumenten

Trotz der zuvor beschriebenen Vervollständigungsprozedur kann nicht immer von einer semantischen Abgeschlossenheit der einzelnen Abschnitte ausgegangen werden, was insbesondere folgende Gründe haben kann:

1. Die logische Struktur des Dokuments ist nicht kompatibel mit dem Universalmodell.
2. Die vorausgesetzte physikalische Nähe zwischen den Regionen mit einem inhaltlichen Zusammenhang ist nicht gegeben.

Um die Auswirkungen der möglichen Segmentierungsfehler abzuschwächen, werden die Abschnitte nach dem gleichen Prinzip angeordnet wie die Textblöcke in den Abschnitten. Auf diese Weise lässt sich auch dann eine sinnvolle Vorlesereihenfolge erzielen, wenn inhaltlich zusammenhängende Regionen voneinander getrennt wurden.

Als Erstes werden die Abschnitte in Spalten unterteilt, die anschließend von links nach rechts sortiert werden. Dafür werden ausgehend von den Knoten, die keine oberen Nachbarn besitzen, unter Verwendung von dem Dijkstra-Algorithmus [170] die kürzesten Wege im Nachbarschaftsgraphen $\mathcal{G}$ ermittelt (s. Pseudocode 8.5.2):

$$\mathcal{A}_{vA} = \{A_1 \in \mathcal{A} | \forall A_2 \in \mathcal{A}: (A_1, A_2) \notin \mathcal{E}_v\}$$

Die Knoten aus $\mathcal{A}_{vA}$ repräsentieren Abschnitte, die ganz oben in den jeweiligen Spalten liegen könnten.

```
// bestimme Spalten-Abstände
FORALL A ∈ 𝒜_vA
        distH(A, 𝒜) := Dijkstra(𝒢, A)
END
FORALL A ∈ 𝒜_vA
        distH(A, A) := 0
END
```

*Pseudocode 8.5.2: Bestimmung der Abstände zwischen den Regionen in dem Nachbarschaftsgraphen der Abschnitte*

Kantengewichte in dem Graphen $\mathcal{G}$ werden dabei so gesetzt, dass die Distanzwerte $distH(\mathcal{A}, A)$ den Abstand zwischen zwei Regionen in Spalten angeben:

$$(A_1, A_2) \in \mathcal{E}_h \rightarrow w\big((A_1, A_2)\big) = cols(A_1)$$
$$(A_1, A_2) \in \mathcal{E}_v \rightarrow w\big((A_1, A_2)\big) = 0, w\big((A_2, A_1)\big) = 0$$

Knoten, die von einem Startknoten $A \in \mathcal{A}_{vA}$ ausgehend über vertikale Kanten erreichbar sind, haben den Abstand $distH(A, \mathcal{A}) = 0$ und gehören in die gleiche Spalte wie $A$. Es ist garantiert, dass jeder Knoten $A_2 \notin \mathcal{A}_{vA}$ von einem Startknoten $A_1 \in \mathcal{A}_{vA}$ erreicht werden kann (der Beweis kann durch vollständige Induktion geführt werden):

$$\forall A_1 \in \mathcal{A}\colon (A_1 \notin \mathcal{A}_{vA} \rightarrow \exists A_2 \in \mathcal{A}_{vA}\colon distH\big(A_{1,} A_2\big) = 0)$$

Startknoten ohne eingehende vertikale und horizontale Kanten sind dagegen offensichtlich unerreichbar:

$$\mathcal{A}_0 = \{A_2 \in \mathcal{A}_{vA} | \forall\, A_1 \in \mathcal{A}_{vA}\colon distH(A_1, A_2) = \infty\}$$

Um eine totale Ordnung auf der Menge der Abschnitte festzulegen, werden die Knoten aus $\mathcal{A}_0$ nun paarweise untersucht (Pseudocode 8.5.3).

```
// Relaxation der Kanten
FORALL A1, A2 ∈ A0
    FORALL A3 ∈ A
        d := distH(A1, A3) − distH(A2, A3)
        IF d > 0 ∧ d < distH(A1, A2)
            distH(A1, A2) := d
            w(A1, A2) := d      // neue Kante hinzufügen
            Remove(A0, A2)
            BREAK
        ELSE IF d = 0 ∧ Above(A1, A2)
            distH(A1, A2) := 0
            w(A1, A2) := 0      // neue Kante hinzufügen
            w(A2, A1) := 0
            Remove(A0, A2)
            BREAK
        END
    END
END
```

*Pseudocode 8.5.3: Sortierung von Artikeln nach Spalten ausgehend von dem in Spalten gemessenen Abstand.*

Zwei Abschnitte gehören demnach auch dann zur selben Spalte, wenn sie den gleichen Abstand zu einem dritten Abschnitt haben. Auf diese Weise werden Abschnittsspalten zusammengefügt, die zu weit voneinander entfernt sind, um als Nachbarn identifiziert zu werden (s. Abb. 8.5.1 b). Der untere der zwei Abschnitte (s. Pseudocode 8.5.4) wird als Nachfolger des Anderen markiert und aus $\mathcal{A}_0$ entfernt, sodass schließlich ein einziges Element $\mathcal{A}_0 = \{A_0\}$ übrig bleibt. Jeder Abschnitt aus $\mathcal{A}_{vA}$ ist direkt oder indirekt ein Nachfolger von $A_0$, da die verwendete Vergleichsoperation eine totale Ordnungsrelation [171] darstellt, die reflexiv, antisymmetrisch

$$\forall\, A_1, A_2, A_3 \in \mathcal{A}_{vA}\colon (distH(A_1, A_2) > 0) \rightarrow (distH(A_2, A_1) < 0) \wedge \\ Above(A_1, A_2) \rightarrow \neg\big(Above(A_2, A_1)\big)$$

und transitiv ist*

$$\forall\, A_1, A_2, A_3 \in \mathcal{A}_{vA}\colon (distH(A_1, A_2) \neq \infty) \wedge (distH(A_2, A_3) \neq \infty) \\ \rightarrow distH(A_1, A_3) \neq \infty.$$

Da $A_0$ das kleinste Element von $\mathcal{A}$ ist, muss dieses eindeutig bestimmt sein. Weil $A_0$ weder über eingehende noch über ausgehende horizontale Kanten erreichbar ist, ist es sichergestellt, dass der entsprechende Abschnitt keine oberen und linken Nachbarn hat und folglich in die linkeste Spalte einzuordnen ist. Eine lineare Anordnung der restlichen Abschnitte wird festgelegt, indem die kürzesten Pfade in $\mathcal{G}$ zwischen $A_0$ und allen anderen Knoten aus $\mathcal{A}$ berechnet werden, wobei die gewünschte Links-nach-rechts-Reihenfolge der Spalten erzielt wird.

Alle Knoten $A_2 \in \mathcal{A}$, die von einem Startknoten $A_1 \in \mathcal{A}_{vA}$ ausgehend über vertikale Kanten erreichbar sind, haben den gleichen Abstand zu $A_0$:

$$\forall\, A_2 \in \mathcal{A} \forall A_1 \in \mathcal{A}_{vA}\colon distH(A_1, A_2) = 0 \rightarrow distH(A_2, A_1) = 0 \rightarrow \\ distH(A_0, A_1) \leq distH(A_0, A_2) + 0 \wedge distH(A_0, A_2) \leq distH(A_0, A_1) \\ \rightarrow distH(A_0, A_1) = distH(A_0, A_2)$$

* Zyklische Abhängigkeiten werden durch das sukzessive Entfernen von erreichbaren Knoten aus $\mathcal{A}_0$ vermieden.

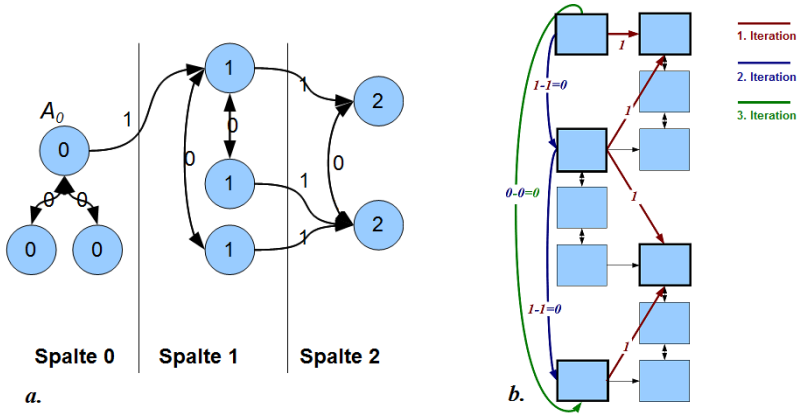


*Abb. 8.5.1:a. Konstruktion des Abschnittbaums. b. Relaxation der Kanten*

Die Einordnung der Abschnitte in Spalten auf der Basis der berechneten Distanzen ergibt eine Partitionierung, bei der jeder Abschnitt in dieselbe Spalte einsortiert wird wie alle seine unteren Nachbarn (s. Abb. 8.5.1 a). Auch Abschnitte, die einen gemeinsamen unteren Nachbarn haben, landen automatisch in einer Spalte.

Eine *strenge* Totalordnung auf der Abschnittsmenge wird schließlich über die Anordnung der Spalten in der aufsteigenden Reihenfolge der Abstände von $A_0$ festgelegt, wobei die Abschnitte in den einzelnen Spalten von oben nach unten sortiert werden. Analog zur „horizontalen“ Sortierung nach Spalten erfolgt die „vertikale“ Sortierung über die Bestimmung der kürzesten Pfade $distV$ im Nachbarschaftsgraphen der Abschnitte $\mathcal{G}$. Diesmal werden die Kanten neu gewichtet, sodass nur die vertikalen Übergänge gezählt werden:

$$(A_1, A_2) \in \mathcal{E}_v \rightarrow w\big((A_1, A_2)\big) = MUR(A_1).height$$

und

$$(A_1, A_2) \in \mathcal{E}_h \rightarrow w\big((A_1, A_2)\big) = 0, w\big((A_2, A_1)\big) = 0$$

Die Distanzen werden in jeder Spalte separat berechnet, wobei der jeweils oberste Abschnitt einer Spalte als Ausgangspunkt für die Pfadsuche dient und den Abstand $distV(A_i) = 0$ hat.

Der Distanzwert $distV$ repräsentiert die Höhe eines Knotens innerhalb seiner Spalte. Allerding weisen alle Knoten, die sich einen unteren oder oberen Nachbarn teilen, die gleichen Werte auf, sodass die Reihenfolge der Ausgabe in einem solchen Fall nicht eindeutig ist. Abschnitte einer Spalte, die sich auf der gleichen Höhe befinden, werden nach dem Links-nach-rechts-Prinzip angeordnet. Die relative Lage zweier Abschnitte wird mit dem Algorithmus (Pseudocode 8.5.4) ermittelt.

```
// bestimme den höheren von zwei Abschnitten
Above(A1, A2)
        IF distV(A1, A2) = ∞
                RETURN FALSE
        ELSE IF distV(A2, A1) = ∞
                RETURN TRUE
        ELSE IF distV(A2, A1) > distV(A1, A2)
                RETURN FALSE
        ELSE IF distV(A2, A1) < distV(A1, A2)
                RETURN TRUE
        ELSE
                // direkter Vergleich anhand der benachbarten MUR-Seiten
                RETURN ||MUR(A1).bottom − MUR(A2).top|| <
                        ||MUR(A2).bottom − MUR(A1).top||
        END
END


// bestimme den linkeren von zwei Abschnitten
Left(A1, A2)
        IF distH(A1, A2) = ∞
                RETURN FALSE
        ELSE IF distH(A2, A1) = ∞
                RETURN TRUE
        ELSE IF distH(A2, A1) > distH(A1, A2)
                RETURN FALSE
        ELSE IF distH(A2, A1) < distH(A1, A2)
                RETURN TRUE
        ELSE
                // direkter Vergleich anhand der benachbarten MUR-Seiten
                RETURN ||MUR(A1).right − MUR(A2).left|| <
                        ||MUR(A2).right − MUR(A1).left||
        END
END
```

*Pseudocode 8.5.4: Bestimmung der relativen Lage zweier Artikel*

### 8.5.4 Mikrostruktur von Dokumenten

Beim Anordnen von Textregionen innerhalb der Abschnitte wird analog zur Sortierung der Abschnitte innerhalb des gesamten Dokuments vorgegangen. Jeder Abschnitt $A$ wird in dem Nachbarschaftsgraphen der Regionen $G$ durch einen zusammenhängenden Untergraphen $G_A = \{V_A, E_A\}$ repräsentiert, der sich aus der Hierarchie des Syntaxbaums ergibt (s. Abschnitt 8.5.1). Während im Laufe der Makroanalyse alle Knoten des Nachbarschaftsgraphen $\mathcal{G}$ einheitlich behandelt werden, muss bei der Mikroanalyse zwischen den Überschrift-, Untertitel- und Textblock-Knoten unterschieden werden (s. Abb. 8.5.2):

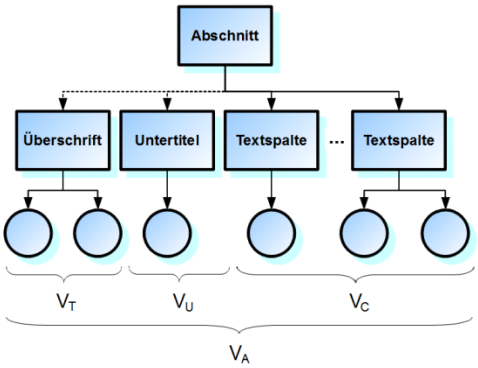


*Abb. 8.5.2:Syntaxbaum, Mikrostruktur. Blätter des Baums repräsentieren einzelne Textregionen.*

$$V_A = V_T \cup V_U \cup V_C$$

Der Grund dafür ist der, dass die Überschriften und Untertitel im Gegensatz zu den Textspalten nach dem linker-Nachbar-zuerst-Prinzip vorgelesen werden müssen. Dementsprechend unterschiedlich werden auch die Startknoten für die Pfadsuche gewählt:

Überschrift: $V_{hT} = \{v \in V_T | \forall u \in V_T : (u, v) \notin E_h\}$
Untertitel: $V_{hU} = \{v \in V_U | \forall u \in V_U : (u, v) \notin E_h\}$
Textblöcke: $V_{vC} = \{v \in V_C | \forall u \in V_C : (v, u) \notin E_v\}$

Die Vorlesereihenfolge wird für jede Teilmenge $V_T, V_U, V_C$ unabhängig bestimmt, worauf die drei sortierten Teilfolgen nach dem Schema *Überschrift→Untertitel→Textspalten* zusammengesetzt und entsprechend der ermittelten Reihenfolge der Abschnitte in die gemeinsame Ausgabeliste eingefügt werden.

## 8.6 Auswertung und Zusammenfassung

Die Auswertung der Analyse-Methoden für die Modellierung der physischen und logischen Layoutstrukturen erfolgte unter Verwendung von ausgewählten Dokumenten aus der Datenbank *MediaTeam*. Die *MediaTeam* Dokumenten-Datenbank [136] enthält eingescannte Dokumente verschiedener Typen, unter anderem eine Vielzahl von Zeitungsartikeln, einige Briefe und Bücher, einschließlich der Ground-Truth-Daten bzgl. der semantischen Abhängigkeiten zwischen den Textregionen (s. Abb. 8.6.1). Darüber hinaus wurden mehrere speziell angefertigte Aufnahmen mit einer ausgeprägten Verzerrung der Dokumentoberfläche (s. Anhang D) in die Evaluierung einbezogen, für die die jeweils angestrebte Vorlesereihenfolge manuell festgelegt wurde.

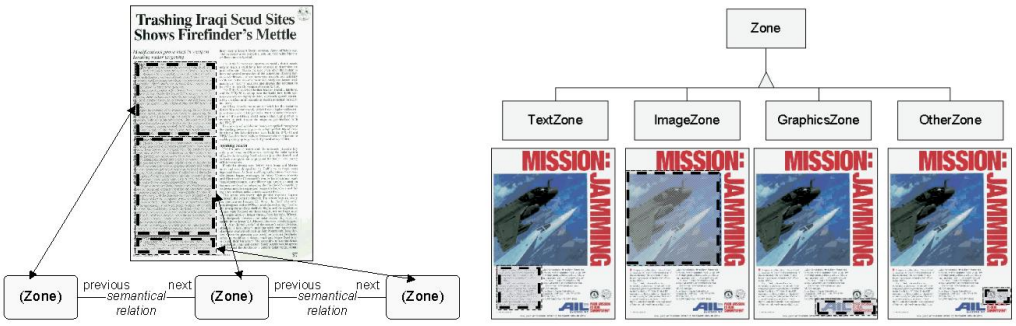


*Abb. 8.6.1: Angabe der semantischen Abhängigkeiten zwischen den Textregionen in der MediaTeam Datenbank aus [136].*

Das Ziel der Evaluierung war es, die vom Algorithmus identifizierten sequentiellen Nachbarschaftsbeziehungen der Regionen zu bewerten. Die Auswertung wurde dabei einerseits dadurch erschwert, dass für viele Dokumente der MediaTeam Datenbank keine Totalordnung vorgegeben ist, weshalb in bestimmten Fällen keine Aussagen über die Richtigkeit der erkannten Anordnung gemacht werden konnte. Andererseits führte die Untersegmentierungsproblematik (s. Abschnitt 6.7) gelegentlich dazu, dass für einige der Ground-Truth-Relationen keine Entsprechung in der berechneten Abfolge gefunden wurde. Infolge dieser Beobachtungen wurde die Auswertung wie folgt vorgenommen:

- Für jede der vorgegebenen Nachbarschaftsrelationen $relation(R_i, R_j)$, $i > j$ wurde in der berechneten Partitionierung nach den Entsprechungen für die beiden Regionen $R'_i, R'_j$ unter Verwendung der Mittelpunkt-Koordinaten gesucht

- Falls eine Regionen nicht gefunden werden konnte, dann wurde ein Versuch unternommen aufeinanderfolgende Ground-Truth-Regionen zu kombinieren und auf diese Weise die Untersegmentierungsproblematik zu lösen
- Es wurde kontrolliert, ob die transitive Hülle der berechneten Nachbarschaftsrelationen die Relation $relation(R'_i, R'_j)$ enthält
- Im Erfolgsfall wurde die Relation $relation(R_i, R_j)$ selbst sowie die u. U. bei der Verschmelzung der Regionen verwendeten Relationen als korrekt erkannt markiert
- Die Anzahl der korrekt erkannten Relationen wurde über die Gesamtanzahl der vorgegebenen Relationen normiert

Eine Zusammenfassung der Evaluierungsergebnisse ist aus der Tabelle 8.6.1 zu entnehmen. Eine detaillierte Analyse ist im Anhang D zu finden.

*Tabelle 8.6.1:Evaluierungsergebnis des Algorithmus zur Layoutanalyse.*

| | ***Erfolgsrate*** |
|---|---|
| ***MediaTeam Dokumente (119 Relationen)*** | 92.4% |
| ***Prototyp-Aufnahmen 105 Relationen*** | 89.5% |

Zwei zusätzliche Beispiele der berechneten Vorlesereihenfolgen werden des Weiteren in Abb. 8.6.2 demonstriert. Trotz der (quantitativ) schlechterer Resultate für die Dokumente der MediaTeam Datenbank nach der ersten Phase der Layouterkennung (s. Tabelle 6.7.1), fiel das produzierte Ergebnis der zweiten Phase insgesamt sogar etwas besser aus als das Ergebnis, das für die Prototyp-Aufnahmen erzielt wurde. Die Erfolgsrate liegt bei über 90%, wobei 3/4 der erkannten Ausgabefolgen fehlerfrei sind. Der Einfluss von Segmentierungsfehlern auf das Endergebnis ließ sich unterdessen dank der Sortierung während der Makroanalyse teilweise eindämmen. (s. Anhang D)

Die erzielte Erfolgsrate ist mit den berichteten Trefferquoten der anderen Autoren [172] vergleichbar. Aufgrund der i. d. R. geringen Anzahl der Regionen $\ll 50$, wird die Gesamtlaufzeit der Layouterkennung von der geometrischen Layoutana-

lyse dominiert. Für die sämtlichen Testdokumente lag die Rechenzeit des Algorithmus stets unterhalb von 1 s. Durch den Verzicht auf die semantische Analyse der Textblöcke lässt sich die Bestimmung der Vorlesereihenfolge vor der eigentlichen Zeichenerkennung durchführen, sodass die OCR-Verarbeitung und Ausgabe der Blöcke gemäß ihrer logischen Anordnung stattfinden kann.

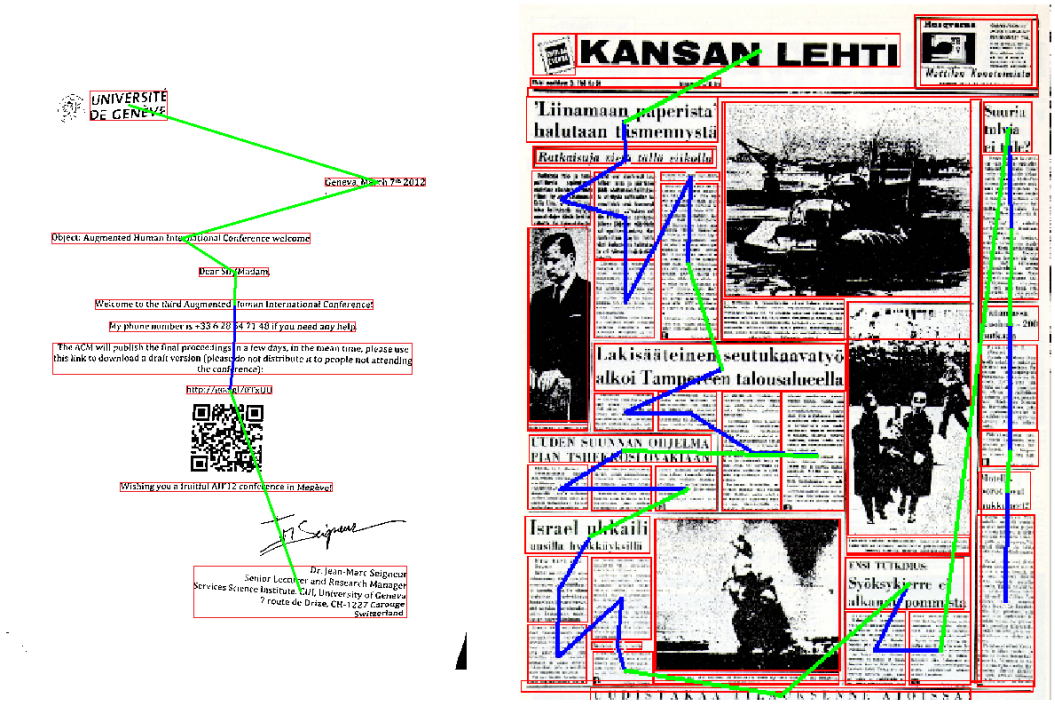


*Abb. 8.6.2: Resultierende Vorlesereihenfolge. Blaue Linien markieren die Vorlesereihenfolge innerhalb der Abschnitte, während die grünen Linien die Anordnung der Abschnitte kennzeichnen.*

# 9. Kapitel

# Ergebnis und Ausblick

## 9.1 Resultierendes System

Die Anfertigung eines funktionierenden Prototypen von einem mobilen Vorlesesystem gehörte zu den wichtigsten Aufgaben des Projekts. Im Verlauf der Entwicklung wurden zwei Geräte mit unterschiedlichen Hardware- und Software-Konfigurationen produziert (s. Abschnitt 4.2). Beim ersten Gerät handelt es sich

*Abb. 9.1.1: (links) Das mobile Vorlesesystem im Einsatz. (rechts) Das semi-mobile System.*

um eine Implementierung des in Kapitel 4 präsentierten Entwurfs, sodass für seine Auswertung das Pflichtenheft aus dem Abschnitt 3.7 maßgeblich ist (s. Abb. 9.1.1 links). Das alternative System (s. Abb. 9.1.1 rechts) ist in die Familie der semi-mobilen Vorlesegeräte (s. Abschnitt 3.4) einzuordnen und könnte aufgrund der reduzierten Komplexität schneller eine Marktreife erreichen.

Sowohl das mobile Vorlesesystem als Ganzes als auch die einzelnen Komponenten des Systems absolvierten bereits die ersten Testläufe unter realistischen Be-

dingungen. Dank dem Einsatz von intelligenten Assistenzalgorithmen kann das System die Entdeckung und Aufnahme textueller Information übernehmen, wodurch die Handhabung des Geräts für den Benutzer erleichtert werden soll. Die implementierte Textdetektionsmethode erfüllt die spezifizierten weichen Echtzeitanforderungen, wobei der gemessene Median für die Reaktionszeit des Systems unterhalb von 1 s liegt. Die Evaluierung des Algorithmus fand im Freien und unter Verwendung einer Vielzahl von verschiedenen Szenen aus der natürlichen Umgebung statt (s. Abschnitt 5.1.6). Das System ist in der Lage, den Benutzer in der Aufnahmephase durch sprachliche Anweisungen zu unterstützen. Dank der eingesetzten stereovisionbasierten Autofokusierungsmethode konnte die Aufnahmedauer im Vergleich zu der kameraeigenen Funktion um mehr als 1 s reduziert werden. Diese Verbesserung sowie die Methode zur Bestimmung des Aufnahmemoments (s. Abschnitt 5.2.5) sollen der Problematik von Bewegungsartefakten entgegenwirken.

Für die Reaktionszeit des Systems nach der Einleitung des Verarbeitungsvorgangs wurde ein Richtwert von 30 s spezifiziert. Dank der entwickelten Methode zur schnellen Layouterkennung können die Ausgabe des erkannten Textes und die Verarbeitung der restlichen Blöcke zeitgleich erfolgen. Der Entzerrungsalgorithmus (s. Abschnitt 7.7) ist derart konzipiert, dass der Zeitbedarf vom Grad der Verzerrung anhängig ist. Die Extraktion, Binarisierung und Entzerrung der Textkomponenten findet zeichenweise statt, sodass eine flexible Partitionierung des Texts im Vorfeld der eigentlichen Zeichenerkennung möglich ist. Der Gesamtzeitbedarf für die Vorverarbeitungsphase berechnet sich aus den Laufzeiten der Segmentierung (3 s), der Zeichenextraktion und Binarisierung (3 ms pro Zeichen), der Analyse des logischen Layouts (1 s) und der Entzerrung (3 ms pro Zeichen) (s. Abschnitte 6.7, 7.7, 8.6). Dazu kommt die Laufzeit der Zeichenerkennung, die mit 10 ms pro Zeichen geschätzt wird (s. Abschnitt 3.5). Ein DIN A4 Dokument mit 3000 Zeichen kann somit innerhalb von 54 s komplett verarbeitet werden, wobei die ersten 1000 Zeichen bereits nach $5\,s + 1\,s + 3000 * 0{,}003\,s + 1000 * (0.01\,s + 0.003\,s) = 28\,s$ zur Verfügung stehen. Bei einer Ausgaberate von 140 Wörter pro Minute [173] kann die Verarbeitung der restlichen 2000 Zeichen ($2000 * (0.01\,s + 0.003\,s) = 26\,s$), noch während der Ausgabe abgeschlossen werden. Auf diese Weise kann die spezifizierte Reaktionszeit im Normalfall eingehalten werden.

Die Verständlichkeit der Ausgabe kann je nach Aufnahmequalität stark variieren. Trotz der Vorbeugemaßnahmen, die in Aufnahmephase getroffen werden, kann es bei ungünstigen Lichtverhältnissen zu einer bewegungs- oder fokussierungsbedingten Unschärfe der Zeichenkonturen kommen*. Auch Reflektionen auf einer Glanzoberfläche eines Dokuments stellen ein kritisches Problem dar. Dank der stereovisionbasierten Entzerrungsmethode sind die resultierenden Erkennungsraten vergleichbar und u. U. sogar höher als die, die von OmniPage-Funktionen alleine produziert werden (s. Abschnitt 7.7). Bei der Evaluierung der Algorithmen zur Layouterkennung wurde darauf geachtet, dass ein Teil der verwendeten Aufnahmen unter realitätsnahen Bedingungen hergestellt wurde (Einbettung in eine natürlichen Umgebung, Sonnenlicht, hoher Grad der Verzerrung), um Aussagen über die Robustheit des Verfahrens machen zu können. Aufgrund der zufriedenstellenden Ergebnisse der Evaluierung kann geschlussfolgert werden, dass es sich beim Prototypen um ein voll einsatzfähiges System im Sinne der Spezifikation im Abschnitt 3.7 handelt, welches dank des einfachen Steuerungskonzepts (s. Abschnitt 3.6) bereits nach einer kurzen Lernphase bedient werden kann. Dank der Verwendung von preisgünstigen Standardkomponenten ist eine Massenproduktion des Geräts vorstellbar.

## 9.2 Beiträge der Arbeit

Beim Entwurf und der Implementierung des Systems galt das besondere Augenmerk der Laufzeit der Algorithmen. Aufgrund der Ressourcenbeschränkung der mobilen Plattform stellte die Einhaltung der spezifizierten Reaktionszeiten eine große Herausforderung dar. Ein wesentlicher Beitrag dieser Arbeit besteht in der Entwicklung eines einzigartigen Konzepts für die Dokumentverarbeitungskette, das auf einer gesamtheitlichen Sicht auf das Problem basiert und durch ein Zusammenspiel der Teilschritte sowie Wiederverwertung der Teilergebnisse den durchschnittlichen Gesamtzeitbedarf zu reduziert versucht. Außerdem wird eine Verwendung von Stereovision-Methoden im Bereich der digitalen Dokumentverarbeitung schwerpunktmäßig thematisiert und die daraus resultierenden Möglichkeiten aufgezeigt. Im Folgenden werden die einzelnen Beiträge zusammengefasst und kurz erläutert:

* Einerseits sinkt die Schärfentiefe mit einer Vergrößerung der Blendenöffnung [124], andererseits steigt mit einer Verkleinerung der Blendenöffnung die notwendige Belichtungszeit.

1. Ein probabilistischer Algorithmus zur schnellen Textlokalisierung in Videosequenzen, der eine Echtzeit-Textdetektion trotz der Leistungseinschränkungen eines mobilen Systems ermöglicht.
2. Ein Text-Tracking-Algorithmus, der textspezifische Merkmale für die Stereokorrespondenzsuche verwendet und die 3D-Pose von Dokumentregionen ermittelt.
3. Eine Methode zur Segmentierung von Dokumentaufnahmen, die mit einer Extraktion von textspezifischen Merkmalen und Binarisierung des Dokuments einhergeht.
4. Ein Verfahren zur Analyse der logischen Dokumentstruktur und Festlegung einer sinnvollen Vorlesereihenfolge, welches auch für Aufnahmen von verzerrten Dokumenten geeignet ist.
5. Eine stereovisionbasierte Entzerrungsmethode, die eine bessere Leistung bei der Korrektur von massiven perspektivischen und komplexen nichtlinearen Verzerrungen der Dokumentoberfläche erbringt als ein modernes, kommerzielles Einzelbild-Entzerrungsverfahren. Hier werden textspezifische Merkmale der vorausgegangenen Verarbeitungsschritte für die Korrespondenzsuche wiederverwendet.

Trotz der vielversprechenden Möglichkeiten, die die Stereovision-Technologie speziell beim Einsatz in mobilen Systemen bietet, ist kein anderes tragbares Vorlesegerät bekannt, das mit einer Stereokamera ausgerüstet ist und diese für die Zwecke der Dokumentverarbeitung nutzt.

## 9.3 Ausblick auf zukünftige Arbeiten

### 9.3.1 Ausführliche Testläufe und Erlangung der Produktionsreife


*Abb. 9.3.1: Computer-on-modul von gumstix aus [174].*

Abschließende Testläufe für das erste Gerät stehen noch aus, da für einen größeren Testlauf mehrere Prototypen produziert werden müssen. Insbesondere die implementierten Methoden zur Textdetektion und Layoutanalyse sollten angesichts der Vielzahl möglicher Anwendungsfälle unter Beteiligung von Betroffenen weiteren ausführlichen Tests unterzogen werden. Eine großflächige Abdeckung aller möglichen Anwendungsfälle und Fehlerquellen unter simulierten

Bedingungen ist unrealistisch, da ein Großteil der standardisierten und öffentlich zugänglichen Dokument-Datenbanken wichtige Voraussetzungen (Stereobilder, Verzerrung, Bewegungsartifakte) nicht erfüllt.

Das Design des Systems bedarf noch einiger Verbesserung, vor allen Dingen bezüglich der optischen Unauffälligkeit und der Wetter-/Wasserbeständigkeit. Eine besondere Herausforderung stellt die Kalibrierung der Stereokamera dar, die sich nur schwer vollständig automatisieren lässt. Dank der ständig voranschreitenden Miniaturisierung elektrischer Bauteile (s. Abb. 9.3.1) könnte eine Integration der Rechnerkomponente in das Brillengestell in naher Zukunft möglich werden, sodass sich das komplette System als Einzelmodul herstellen ließe. Aufgrund des kleineren Abstands zwischen der Kamera und dem Rechner wäre eine theoretisch höhere Bildübertragungsgeschwindigkeit erzielbar, was die Voraussetzung für ein größeres Array von Kameras schaffen würde.

### 9.3.2 Optimierung der Software

Während die komplette Vorverarbeitungsprozedur eine Eigenimplementierung ist, findet die eigentliche Zeichenerkennung unter Verwendung von kommerziellen OCR-Modulen statt. Dabei entsteht ein erheblicher Doppelaufwand, da die Textdetektion und die Extraktion der Zeichenkonturen als Teil der OCR-Analyse wiederholt werden. Durch eine Weiterentwicklung des Klassifizierungsmethode aus [100] könnte auf die Verwendung von Drittanbieter-Software verzichtet werden, infolgedessen u. U. enorme Laufzeitvorteile möglich wären.

### 9.3.3 Erweiterung der Funktionalität

Es bietet sich des Weiteren an, die Funktionalität des Geräts um einige zusätzliche Dienste zu erweitern. Erste Machbarkeitsuntersuchungen zur Gesichts- und Objekterkennung unter Einsatz von Softwarelösungen *FaceSDK* [175] bzw. *Sentisight SDK* [176] wurden bereits durchgeführt und einige schwerwiegende Probleme identifiziert. Die Vorgehensweise ist in beiden Fällen gleich: In der Lernphase wird ein Objekt vor die Kamera gehalten, um ein Modell zu erstellen, welches in der Identifikationsphase für die Klassifikation verwendet wird. Jedem gelernten Objekt wird eine akustische Bezeichnung zugewiesen, die in der Lernphase aufgenommen und im Falle der Entdeckung abgespielt wird. Während die Detektion und Segmentierung von Gesichtern i. d. R. einwandfrei funktioniert,

hängt die Leistung der Objekterkennungssoftware stark von den Objekteigenschaften ab. Was die Zuverlässigkeit der Klassifizierung anbetrifft, haben die beiden Lösungen große Schwächen offenbart, sodass an der Stelle noch ein Verbesserungspotential besteht.

## Eidesstattliche Versicherung:

Ich versichere, dass ich die vorliegende Promotionsarbeit selbstständig verfasst habe. Andere als die angegebenen Hilfsmittel und Quellen wurden nicht benutzt. Die Arbeit hat keiner anderen Prüfungsbehörde vorgelegen.

# Literaturverzeichnis


[1] "Global Data on Visual Impairments 2010", WHO, Webseite: http://www.who.int/blindness/publications/globaldata/

[2] "Altersbedingte Makula-Degeneration (AMD)", DBSV e.V. : Webseite: http://www.dbsv.org/infothek/broschueren-und-mehr/

[3] "Deutsches Ärzteblatt", Ausgabe Oktober 2012, S. 471, 2012

[4] "Popular science", 2, S. 125-127, 1949

[5] H. F. Schantz, "The History of OCR", Data processing magazine, 12, S. 46, 1970

[6] R. C. Kurzweil, A. Kleiner, "A Description of the Kurzweil Reading Machine and Status Report on its Testing and Dissemination", Bulletin of Prosthetics Research, 1977

[7] "User Guide, kReader Mobile, knfbReader Mobile", K-NFB Reading Technology Inc., 2008, Webseite: http://www.knfbreader.com/

[8] Maurizio Pilu, Stephen Pollard, "A light-weight text image processing method for handheld embedded cameras", British Machine Vision Conference, S. 547–556, 2002

[9] "TextScout Instruction Manual", Webseite: http://www.textscout.eu/

[10] "Intel Reader User Manual", Webseite mit dem Handbuch: http://www.careinnovations.com/Data/Downloads/Intel_Reader/SupportDocs/reader_user_manual_us_rev30.pdf

[11] N. G. Bourbakis, D. Kavraki, "Intelligent Assistants for Handicapped People's Independence: Case Study", International Journal of Semantic and Infrastructure Services, S. 337-244, 1996

[12] S. Panchanathan, J. Black, M. Rush, V. Iyer, "iCare – A User Centric Approach to the Development of Assistive Devices for the Blind and Visually Impaired", International Conference on Tools with Artificial Intelligence, S. 641-648, 2003

[13] N. Ezaki, M. Bulacu, L. Schomaker, "Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons", International Conference on Pattern Recognition, 2, S. 683-686, 2004

[14] R. Keefer, "A Wearable Document Reader for the Visually Impaired: Dewarping and Segmentation", International Journal on Artificial Intelligence Tools, 18(3), S. 467-486, 2009

[15] M. Tanaka, H. Goto, "Text-Tracking Wearable Camera System for Visually-Impaired People", International Conference on Pattern Recognition, 19, S. 1-4, 2008

[16] J. Zelek, R. Audette, J. Balthazaar, C. Dunk, "A Stereo-vision System for the Visually Impaired", Techical Report, University of Guelph, 2000

[17] P. Meijer, "Seeing with Sound: Wearable Computing for the Blind", Nordic Interactive Conference, 2001

[18] W. Fink, M. Tarbell, J. Weiland and M. Humayun, "DORA: Digital Object Recognition Audio-Assistant For The Visually Impaired", Investigative Ophtalmology and Visual Science, 45, S. 4201, 2004

[19] J. L. González-Mora, A. Rodríguez-Hernández, L. F. Rodríguez-Ramos, L. Díaz-Saco, N. Sosa, "Development of a new space perception system for blind people, based on the creation of a virtual acoustic space", International Work Conference on Artificial and Natural networks, 2, S. 321-330, 1999

[20] D. Dakopoulos, "Tyflos: A wearable navigation prototype for blind visually impaired; Design, modelling and experimental results", Dissertation, Write State University, 2009

[21] H. D. Benington, "Production of Large Computer Programs", IEEE Annals of the History of Computing, 5(4), S. 350–361, 1983

[22] InformA-Projekt, Webseite: http://www.symplektikon.de/

[23] B. Jähne, "Digitale Bildverarbeitung", Springer, ISBN 978-3-540-24999-3, 2005

[24] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images", International Conference on Computer Vision, S. 839-846, 1998

[25] R. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision", Cambridge, Cambridge University Press, 2006

[26] G. Bradski, A. Kaehler, "Learning OpenCV Computer Vision with the OpenCV Library", O'Reilly Media Inc., ISBN 978-0-596-51613-0, 2008

[27] D. C. Brown, "Decentering distortion of lenses", Photogrammetric Engineering **7**, 32, S. 444-462, 1966

[28] N. Otsu, "A threshold selection method from gray-level histograms", Automatica, 11, S. 23-27, 1975

[29] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram", Computer Vision, Graphics and Image Processing, 29, S. 273-285, 1985

[30] M. Portes de Albuquerque, I.A. Esquef, A.R. Gesualdi Mello, M. Portes de Albuquerque, "Image thresholding using Tsallis entropy", Pattern Recognition Letters, 25, S. 1059-1065, 2004

[31] J. Kittler and J. Illingworth, "Minimum error thresholding", Pattern Recognition, 19(1), S. 41-47, 1985

[32] W. Niblack, "An Introduction to Digital Image Processing", Prentice Hall, ISBN 978-0134806747, 1986

[33] J. Sauvola, M. Pietikäinen, "Adaptive document image binarization", Pattern Recognition, 33(2), S. 225-236, 2000

[34] S. Tabbone, L. Wendling, "Multi-scale binarization of images", Pattern Recognition Letters 24, S. 403-411, 2003

[35] E. Kavallieratou, "A Binarization Algorithm specialized on Document Images and Photos", International Conference on Document Analysis and Recognition, S. 463-467, 2005

[36] Y. Li, C. Suen, M. Cheriet, "A Threshlod Selection Method Based on Multiscale and Graylevel Co-occurrence Matrix Analysis", International Conference on Document Analysis and Recognition, S. 463-467, 2005

[37] M. Block, R. Rojas, "Local Contrast Segmentation to Binarize Images", International Conference on Digital Society, S. 294-299, 2009

[38] D. Marr, E. Hildreth, "Theory of Edge Detection", Proceedings of the Royal Society of London, S. 215-217, 1980

[39] Y. Ishitani, "Document skew detection based on local region complexity", International Conference on Document Analysis and Recognition, S. 49-52, 1993

[40] J. W. Cooley, J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. Computation, 19, 297-301, 1965

[41] R. C. González, R. E. Woods, "Digital image processing", Prentice Hall International, ISBN 978-0201180756, 2001

[42] S. G. Mallat, "A theory for multiresolution signal decomposition : the wavelet representation", IEEE Pattern Analysis and Machine Intelligence, 11, S. 674-693, 1989

[43] R. R. Coifman, M. V. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection", IEEE Information Theory, 38, 1992

[44] "PoetCompact 2", Info-Broschüre, Webseite mit der Broschüre: http://www.baum.de/cms/fileadmin/downloads/prospekte/en/Poet_Compact_2.pdf

[45] "Leselöwe", Info-Broschüre, Webseite mit der Broschüre: http://www.bhvd.de/produkte/flspri/llp/

[46] "beyo pr100" von beyo GmbH, Webseite: http://www.baum.de/de/news/20080509_portablesvorlesegeraet.html

[47] "OmniPage 16, User's Guide", Webseite: http://www.nuance.de

[48] "ABBYY FineReader Version 11, User's Guide", Webseite: http://www.abbyy.com/

[49] T. M. Breuel, "The OCRopus Open Source OCR System", SPIE-IS&T Electronic Imaging, 2008, Webseite von dem Projekt: code.google.com/p/ocropus/

[50] Project OCRad, Webseite von dem Projekt: http://www.gnu.org/software/ocrad/

[51] M. Block-Berlitz, "Verbesserung, Lokalisierung und Entzerrung von Textdokumentaufnahmen", Dissertation, Freie Universität Berlin, 2008

[52] Projekt OpenOCR/Cuneiform, Webseite von dem Projekt: http://en.openocr.org/

[53] Projekt GOCR, Webseite von dem Projekt: http://jocr.sourceforge.net/

[54] Projekt OCRFeeder, Webseite von dem Projekt: https://live.gnome.org/OCRFeeder

[55] Projekt Tesseract, Webseite von dem Projekt: http://code.google.com/p/tesseract-ocr/

[56] Projekt Puma.NET, Webseite von dem Projekt: http://pumanet.codeplex.com/

[57] T. F. Smith, M. S. Waterman, "Identification of Common Molecular Subsequences", Journal of Molecular Biology, 147, S. 195-197, 1981

[58] "Intel Xeon Processor — Thermal Management", Intel Corporation, 2010

[59] H. Wörn, U. Brinkschulte, "Echtzeitsysteme. Grundlagen, Funktionsweisen, Anwendungen.", Springer, ISBN 3-540-20588-8, S. 321, 2005

[60] P. Read, M.-P. Meyer, "Restoration of motion picture film.", Gamma Group, ISBN 0-7506-2793-X, 2000

[61] R. Keefer, S. Narayanan, N. Bourbakis, "Voice Commands for a Mobile Reading Device for the Visually Impaired", International Conference on Pervasive Technologies Related to Assistive Environments, 2010

[62] "Dragon Naturally Speaking, Version 11.5, User Guide", Webseite mit dem Handbuch: http://www.nuance.com/

[63] M. J. Swain, M. A. Stricker. "Promising directions in active vision", International Journal of Computer Vision, 11(2), S. 109-126, 1993

[64] Webseite von der Blindenschule Neuwied: http://www.blindenschule-neuwied.de/SchuleFruehfoerderung.htm

[65] "MCB1172, Info-Broschüre", Sony Corp. DIBG Imaging Products Dept., 2008

[66] "IGEP v2 Hardware Reference Manual", Webseite mit dem Handbuch: http://isee.biz/component/zoo/item/igepv2-hardware-reference-manual

[67] "fit-PC2 Specifications", Webseite mit dem Handbuch: http://www.fit-pc.com/web/fit-pc/fit-pc2-specifications/

[68] Zhimin Zhou, B. Pain, E. R. Fossum, "Frame-transfer CMOS active pixel sensor with pixel binning", IEEE Electron Devices, 44, S. 1764-1768, 1997

[69] K. Y. Wong, R. G. Casey, F. M. Wahl, "Document Analysis System", IBM Journal of Research and Development, 26, 1982

[70] F. M. Wahl, K. Y. Wong, R. G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", Computer Graphics and Image Processing, 20, S. 375-390, 1982

[71] J. C. Wu, J. W. Hsieh and Y. S. Chen, "Morphology-based Text Line Extraction", Machine Vision and Applications, 19, S. 1432-1769, 2008

[72] Y. M. Y. Hasan, L. J. Karam, "Morphological text extraction from images", IEEE Transactions on Image Processing, 9, S. 1978-1983, 2000

[73] T. Pratheeba, V. Kavitha, S. Raja Rajeswari, "Morphology Based Text Detection and Extraction from Complex Video Scene", International Journal of Engineering and Technology, 2, S. 200-206, 2010

[74] G. Rama Mohan Babu, P. Srimaiyee, A. Srikrishna, "Text Extraction from heterogenous images using mathematical morphology", Journal of Theoretical and Applied Information Technology, 16, S. 419-426

[75] A. K. Das, B. Chanda, "A fast algorithm for skew detection of document images using morphology", International Journal on Document Analysis and Recognition, S. 109-114, 2001

[76] L. Najman, "Using Mathematical Morphology for Document Skew Estimation", The International Society for Optical Engineering, S. 182-191, 2003

[77] D. S. Bloomberg, "Multiresolution Morphological Approach to Document Image Analysis, Visual Communications and Image Processing, S. 648-662, 1992

[78] Xiangrong Chen, A. L. Yuille, "Detecting and reading text in natural scenes", Computer Vision and Pattern Recognition, S. II-304, 2004

[79] T. Sato, T. Kanade, E. Hughes, and M. Smith, "Video OCR for Digital News Archives", Content-Based Access of Image and Video Databases, S. 52-60, 1998

[80] Changhua Wu, Gady Agam, "Document Image De-warping for Text/Graphics Recognition", Structural and Syntactic Pattern Recognition, S. 348-357, 2002

[81] D. J. Ittner and H. S. Baird, "Language-free layout analysis", International Conference on Document Analysis and Recognition, S. 336-340, 1993

[82] A. C. Bovik, M. Clark, W. Geisler, "Multichannel texture analysis using localized spatial filters", IEEE Pattern Analysis and Machine Intelligence, 12, S. 55-73, 1990

[83] Jain A.K., Farrokhnia F.: "Unsupervised texture segmentation using Gabor filters", Pattern Recognition, 24(12), S. 1167-1186, 1991

[84] V. Gaudissart, S. Ferreira, C. Thillou, B. Gosselin "Mobile Reading Assistant for Blind People", European Signal Processing Conference, 2005

[85] O. Rioul, M. Vetterli, "Wavelets and Signal Processing", Signal Processing Magazine, 8(4), S. 14-38, 1991

[86] P. Gupta, N. Vohra, S. Chaudhury, S. D. Joshi, "Wavelet Based Page Segmentation", Indian Conference on Computer Vision, Graphics and Image Processing, S. 20-22, 2002

[87] Acharyya, M., Kundu, M. K., "Document image segmentation using wavelet scale-space features", IEEE Circuits and Systems for Video Technology, 12(12), S. 1117-1127, 2002

[88] Y. Zhong, H. Zhang, A.K. Jain, "Automatic caption localization in compressed video", IEEE Pattern Analysis and Machine Intelligence, 22(4), S. 385-392, 2000

[89] Y. K. Lim, S. H. Choi, S. W. Lee, "Text extraction in MPEG compressed video for content-based indexing", International Conference on Pattern Recognition, S. 409-412, 2000

[90] C. Strouthopoulos, N. Papamarkos, "Text Identification for Document Image Analysis Using a Neural Network", Image and Vision Computing, 16(12-13), S. 879-896, 1998

[91] Xiaojun Li, Weiqiang Wang, Shuqiang Jiang, Qingming Huang, Wen Gao "Fast and Effective Text Detection", International Conference on Image Processing, S. 969-972, 2008

[92] Datong Chen, Hervé Bourlard, Jean-Philippe Thiran "Text Identification in Complex Background Using SVM", Computer Vision and Pattern Recognition, S. 621-626, 2001

[93] Jia-Lin Chenxe, "A simplified approach to the HMM based texture analysis and its application to document segmentation", Pattern Recognition Letters, 18(10), S. 993-1007, 1997

[94] K. Etemad, D. S. Doermann, R. Chellappa, "Multiscale Segmentation of Unstructured Document Pages Using Soft Decision Integration", IEEE Pattern Analysis and Machine Intelligence, 19(1), S. 92-96, 1997

[95] J. S. Payne, "Document segmentation using texture analysis", International Conference on Pattern Recognition, S. 380-382, 1994

[96] C. Cortes, V. Vapnik, "Support-Vector Networks", Machine Learning, 20, S. 273-297, 1995

[97] K. I. Kim, K. Jung, S. H. Park, H. J. Kim, "Support vector machine-based text detection in digital video", Pattern Recognition, 34(2), S. 527-529, 2001

[98] A. K. Jain, Y. Zhong, "Page segmentation using texture analysis", Pattern Recognition, 29, S. 743-770, 1996

[99] Y. Freund, Robert E. Schapire, "A Short Introduction to Boosting", Journal of Japanese Society for Artificial Intelligence, 14(5), S. 771-780, 1999

[100] M. Lindner, M. Block, R. Rojas, "Object Recognition Using Summed Features Classifier", International Conference on Artificial Intelligence and Soft Computing, S. 543-550, 2012

[101] T. M. Breuel, "High Performance Document Layout Analysis", Symposium on Document Image Understanding Technology, S. 209-218, 2003

[102] A. Antonacopoulos, R. T. Ritchings, "Representation and classification of complex-shaped printed regions using white tiles", International Conference on Document Analysis and Recognition, S. 1132, 1995

[103] Qingsheng Zhu, Yunfeng Li, Xiping He, "A neural network-based color document segmentation approach", Joint International Computer Conference, S. 925-928, 2005

[104] Yu Zhong, K. Karu, A. K. Jain, "Locating text in complex color images", International Conference on Document Analysis and Recognition, 1, S. 146, 1995

[105] Satoshi Suzuki, Keiichi Abe, "Topological structural analysis of digitized binary images by border following", Computer Vision, Graphics, and Image Processing, 30(1), S. 32-46, 1985

[106] Toussaint, G. T., "Solving geometric problems with the rotating calipers", MELECON, S. A10, 1983

[107] P. Toft, "The Radon Transform: Theory and Implementation", Dissertation, Dept. Mathematical Modeling, Technical University of Denmark, 1996

[108] H. Li, D. Doermann, O. Kia, "Automatic text detection and tracking in digital video", IEEE Image Processing, 9(1), S. 147-156, 2000

[109] C. Merino, M. Mirmehdi, "A framework towards realtime detection and tracking of text", 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, S. 10-17, 2007

[110] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey", International Journal on Document Analysis and Recognition, S. 84-104, 2005

[111] Lowe, David G., "Object recognition from local scale-invariant features", International Conference on Computer Vision, 2, S. 1150-1157, 1999

[112] A. Zandifar, R. Duraiswami, A. Chahine, L. Davis, "A Video Based Interface to Textual Information for the Visually Impaired", International Conference on Multimodal Interaction, S. 325-330, 2002

[113] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment", International Journal of Pattern Recognition and Artificial Intelligence, 2(3), S. 485-508, 1988

[114] J. J. Guerrero, R. Martinez-Cantin, C. Sagüés, "Visual map-less navigation based on homographies", Journal of Robotic Systems, 22(10), S. 569-581, 2005

[115] C. Xu, B. Kuipers, and A. Murarka, "3D pose estimation for planes", International Conference on Computer Vision, S. 673-680, 2009

[116] Kalman, R.E., "A new approach to linear filtering and prediction problems", Journal of Basic Engineering, 82(1), S. 35-45, 1960

[117] B. Ristic, S. Arulampalam, N. Gordon, "Beyond the Kalman Filter: Particle Filters for Tracking Applications", ISBN 978-1580536318, 2004

[118] S. B. Needleman, C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", Journal of Molecular Biology, 48, S. 443-453, 1970

[119] C. H. Teh, R. T. Chin, "On the detection of dominant points on digital curves", IEEE Pattern Analysis and Machine Intelligence, 11, S. 859-872, 1989

[120] R. Guilbourd, N. Yogev, R. Rojas, "Stereo camera based wearable reading device", Augmented Human International Conference, S. 10-16, 2012

[121] Wei Huang and Zhongliang Jing, "Evaluation of focus measures in multifocus image fusion", Pattern Recognition Letters, 28(4), S. 493-500, 2007

[122] J. Shi, C. Tomasi, "Good features to track", IEEE Computer Vision and Pattern Recognition, S. 593-600, 1994

[123] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient", American Statistician, 42, S. 59-66, 1988

[124] H. Haferkorn, "Optik. Physikalisch-technische Grundlagen und Anwendungen", ISBN 3-87144-570-3, 1981

[125] R. Brugger: "Eine statistische Methode zur Erkennung von Dokumentstrukturen", PhD thesis, University of Freiburg, 1998

[126] M. Unser, "Texture Classification and Segmentation Using Wavelet Frames", Image Processing, 4, S. 1549-1560, 1995

[127] T. Chang, C. C. J. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform", IEEE Image Processing, 2, S. 429-441, 1993

[128] A. Laine, J. Fan, "Texture Classification by Wavelet Packet Signatures", Pattern Analysis and Machine Intelligence, 15, S. 1186-1191, 1993

[129] J. Li and R. M. Gray, "Context-based multiscale classification of document images using wavelet coefficient distributions", IEEE Image Processing, 9, S. 1604-1616, 2000

[130] H. Choi and R.G. Baraniuk, "Multiscale Document Segmentation Using Wavelet-Domain Hidden Markov Models", IEEE Image Processing, S. 1309-1321, 2001

[131] Seong-Whan Lee, Dae-Seok Ryu, "Parameter-Free Geometric Document Layout Analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23, S. 1240-1256, 2001

[132] R. A. Finkel and J. L. Bentley, "Quad trees: a data structure for retrieval on composite keys", Acta Informatica, 4(1), S. 1-9, 1974

[133] Donoho, D. L. and Johnstone, I. M., "Ideal Spatial adaptation via wavelet shrinkage", Biometrika, 81, S. 425-455, 1994

[134] M. Lang, H. Guo, J.E. Odegard, C.S. Burrus, R.O. Wells, "Noise reduction using an undecimated discrete wavelet transform", Signal Processing Letters, 3(1), S. 10-12, 1996

[135] D. Ruppert, "Statistics and Data Analysis for Financial Engineering", Springer, ISBN 9781441977878, 2011

[136] "MediaTeam Document Database II. CD-ROM collection of document images", University of Oulu, Finland:
http://www.mediateam.oulu.fi/MTDB/index.htm

[137] A. Masalovitch, L. Mestetskiy, "Usage of continuous skeletal image representation for document images dewarping", Int. Workshop on Camera-Based Document Analysis and Recognition, 2007

[138] A. Ulges, C. H. Lampert, T. M. Breuel, "Document image dewarping using robust estimation of curled text lines", International Conference on Document Analysis and Recognition, S. 1001-1005, 2005

[139] T. Kanungo, R. M. Haralick, I. Phillips, "Global and local document degradation models", International Conference on Document Analysis and Recognition, S. 730-734, 1993

[140] H. Baird, "Calibration of document image defect models", Symposium on Document Analysis and Information Retrieval, S. 1-16, 1993

[141] H. Cao, X. Ding, C. Liu, "Rectifying the bound document image captured by the camera: A model based approach", International Conference on Document Analysis and Recognition, S. 71-75, 2003

[142] S. Messelodi, C. M. Modena, "Automatic identification and skew estimation of text lines in real scene images", Pattern Recognition, 32, S. 791-810

[143] P. Clark and M. Mirmehdi, "Rectifying perspective views of text in 3D scenes using vanishing points", Pattern Recognition, 36, S. 2673-2686, 2003

[144] L. Jagannathan, C.V. Jawahar, "Perspective correction methods for camera based document analysis", International Conference on Document Analysis and Recognition, S. 148-154, 2005

[145] P. Clark and M. Mirmhedi, "Location and recovery of text on oriented surfaces", SPIE Conf. on Electronic Imaging, S. 267-277, 2000

[146] P. Clark, M. Mirmehdi, "Estimating the orientation and recovery of text planes in a single image", British Machine Vision Conference, S. 421-430, 2001

[147] Lu S., Chen B.M., Ko C.C. "Perspective rectification of document images using fuzzy set and morphological operations", Image and Vision Computing, 23, S. 541-553, 2005

[148] B. Fu, M. Wu, R. Li, W. Li, and Z. Xu, "A model-based book dewarping method using text line detection", Camera Based Document Analysis and Recognition, S. 63-70, 2007

[149] N. Stamatopoulos, B. Gatos, I. Pratikakis, and S. J. Perantonis, "A two-step dewarping of camera document images", International Association for Pattern Recognition, S. 209-216, 2008

[150] J. Liang, D. F. DeMenthon, D. Doermann, "Flattening curved documents in images", Computer Vision and Pattern Recognition, S. 338-345, 2005.

[151] M. Pilu, "Undoing page curl distortion using applicable surfaces", Computer Vision and Pattern Recognition, S. 67-72, 2001

[152] A. Ulges, C. H. Lampert, T. M. Breuel, "Document capture using stereo vision", ACM Symposium on Document Engineering, S. 198-200, 2004

[153] H. Ezaki, S. Uchida, A. Asano, H. Sakoe, "Dewarping of document image by global optimization", International Conference on Document Analysis and Recognition, S. 500-506, 2005

[154] A.Yamashita, A. Kawarago, T. Kaneko, K. T. Miura, "Shape Reconstruction and Image Restoration for Non-Flat Surfaces of Documents with a Stereo Vision System", International Conference on Pattern Recognition, 1, S. 482-485, 2004

[155] H. Pottmann and J. Wallner, "Approximation Algorithms for Developable Surfaces", Computer Aided Geometric Design, S. 539-556, 1999

[156] G. Wahba, "Spline models for observational data", Society for Industrial and Applied Mathematics, ISBN 978-0898712445, 1990

[157] G. Donato, S. Belongie, "Approximate Thin Plate Spline Mappings", European Conference on Computer Vision, S. 21-31, 2002

[158] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: a literature survey", Document Recognition and Retrieval, S. 197-207, 2003

[159] C. C. Lin, Y. Niwa and S. Narita, "Logical structure analysis of book document images using contents information", International Conference on Document Analysis and Recognition, S. 1048-1054, 1997

[160] J. Kim, D. X. Le, and G. R. Thoma, "Automated labeling in document images", SPIE Conference on Document Recognition and Retrieval, S. 111-122, 2001

[161] D. Niyogi, S. N. Srihari, "Knowledge-based derivation of document logical structure", International Conference on Document Analysis and Recognition, S. 472-475, 1995

[162] A. Dengel, F. Dubiel, "Computer understanding of document structure", International Journal of Imaging Systems and Technology, 7, S. 271-278, 1996

[163] J. Hu, R. Kashi, G. Wilfong, "Document image layout comparison and classification", International Conference on Document Analysis and Recognition, S. 285-288, 1999

[164] H. Fujisawa, Y. Nakano, K. Kurino, "Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis", In Proceedings of IEEE, 80, S. 1079-1092, 1992

[165] K. Summers, "Near-wordless document structure classification", International Conference on Document Analysis and Recognition, S. 462-465, 1995

[166] S. Tsujimoto, H. Asada, "Understanding multi-articled documents", International Conference on Pattern Recognition, S. 551-556, 1990

[167] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals", IEEE Transactions on Pattern Analysis and Machine Intelligence, S. 737-747, 1993

[168] Y. Tateisi, N. Itoh, "Using stochastic syntactic analysis for extracting a logical structure from a document image", International Conference on Pattern Recognition, S. 391-394, 1994

[169] L. O'Gorman, "The Document Spectrum for Bottom-Up Page Layout Analysis", Advances in Structural und Syntactic Pattern Recognition, S. 270-279, 1992

[170] E. W. Dijkstra, "A note on two problems in connexion with graphs", Numerische Mathematik, S. 269-271, 1959

[171] R. Nederpelt, F. Kamareddine, "Logical Reasoning: A First Course", King's College Publications, ISBN 978-0954300678, 2004

[172] M. Aiello, C. Monz, L. Todoran, M. Worring, "Document understanding for a broad class of documents", International Conference on Document Analysis and Recognition, 5, S. 1-16, 2002

[173] "Realspeak, User's Guide for German", Scansoft Inc.

[174] Webseite: http://www.gumstix.com

[175] "Luxand Face SDK 4.0, Face Detection and Recognition Library Developer's Guide", Luxand Inc., Webseite: http://www.luxand.com/

[176] "SentiSight SDK, Object recognition for robotics and computer vision", Neurotechnology, Webseite: http:// neurotechnology.com/