



UNIVERSITÄT HAMBURG

TECHNISCHER REPORT

Strichextraktion und -analyse handschriftlicher chinesischer Schriftzeichen

Arved SOLTH

Bernd NEUMANN

Peer STELLDINGER

{solth|neumann|stelldinger}@informatik.uni-hamburg.de

November 2009

Zusammenfassung

Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, anhand dessen handschriftliche chinesische Schriftzeichen aus antiken Manuskripten in ihre einzelnen Striche segmentiert und anschließend bezüglich bestimmter Stricheigenschaften wie unter anderem Strichbreite, -länge und -richtung untersucht werden können. Diese Informationen können als Grundlage für eine Handschriftanalyse mit dem Ziel der Schreiberidentifikation dienen, indem Schreiber anhand von individuellen Merkmalen der Striche ihres Schriftbildes klassifiziert werden.

Das Vorgehen umfasst dabei mehrere Schritte: zuerst wird die Kontur eines in einem einzelnen digitalen Bild gespeicherten Schriftzeichens extrahiert. Die so gewonnenen Konturpunkte werden anschließend mit Hilfe der Eingeschränkten Delaunay-Triangulierung zu Dreiecken verbunden. Aus der Triangulierung der Form des Schriftzeichens werden im Folgenden Strichkreuzungen und partielle Striche extrahiert. Für diese Striche werden anschließend die oben beschriebenen Eigenschaften wie Länge und Schreibrichtung berechnet. Im abschließenden Schritt werden die so extrahierten Striche abhängig von den für sie berechneten Eigenschaften in mehrere Klassen chinesischer Grundstriche eingeteilt. Aufbauend auf diese Arbeit kann eine Klassifizierung von Schreibern entwickelt werden, welche auf den berechneten Stricheigenschaften der extrahierten Grundstriche arbeitet.

Abstract

The aim of this work is to segment handwritten Chinese characters in antique manuscripts into their individual strokes for a subsequent analysis of the extracted strokes. A comparison of the computed stroke features with those of known scribes may then be used for manuscript analysis and scribe identification.

The procedure includes the following steps: First, the contour of a Chinese character is detected using a simple contour detection method. Using the Constrained Delaunay Triangulation, individual contour points are then connected to triangles. These triangles are used to identify stroke parts and intersections and to compute features such as length and writing direction. Partial strokes are merged into more complex strokes depending on their individual features. In the final step, the extracted strokes are classified into several classes of Chinese strokes. Based on this work, scribe identification using the extracted stroke features will be developed.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Chinesische Schrift	3
2.1.1	Schriftzeichen	5
2.1.2	Radikale	6
2.1.3	Striche	8
2.2	Manuskriptanalyse	10
2.3	Bisherige Arbeiten zur Schriftanalyse	11
2.3.1	Holistische und radikalbasierte Verfahren	12
2.3.2	Strichbasierte Verfahren	14
3	Motivation und Vorverarbeitung	21
3.1	Motivation	21
3.2	Größen- und Grauwertnormalisierung	21
3.3	Glättung	22
3.4	Konturextraktion und Triangulierung	24
3.4.1	Kantendetektion mit Marching Squares	25
3.4.2	Delaunay-Triangulierung	27
3.4.3	Chordal Axis Transform	31
3.4.4	Delaunay-Triangulierung als Basis der Grundstrichextraktion	32
4	Strichextraktion	34
4.1	Junction-Extraktion	34
4.2	Teilstrichextraktion und -klassifizierung	36
4.2.1	Teilstrichextraktion	38
4.2.2	Berechnung der Stricheigenschaften	41
4.3	Junctionklassifizierung, -Reduktion und -Verschmelzung	47
4.3.1	Junction-Klassifizierung	48
4.3.2	Junction-Reduktion	49
4.3.3	Junction-Verschmelzung	51
4.4	Teilstrichverschmelzung zu Grundstrichen der chinesischen Schrift	52
4.4.1	Verschmelzung von Verbindungs- und Endstrichen	58
4.4.2	Verarbeitung von Störungen	59
4.5	Grundstrichklassifizierung	60
5	Ergebnisse und Ausblick	65

5.1	Experimente	65
5.1.1	Statistische Auswertung	65
5.1.2	Aufgetretene Probleme	66
5.2	Fazit	71
	Literaturverzeichnis	73

1 Einleitung

Die Analyse antiker chinesischer Manuskripte stellt sowohl für die Sinologie als auch für die Informatik und insbesondere die Bildverarbeitung eine interessante Herausforderung dar. Da solche Schriftstücke mehr Informationen als den reinen Inhalt beziehungsweise die Bedeutung des Textes enthalten, ist die Untersuchung von Material, Orthographie und nicht zuletzt der Handschrift bei der Analyse der Manuskripte von ebenso großer Bedeutung wie ihre literarische Interpretation. Besonders die automatisierte Analyse handschriftlicher Zeichen stellt neben der Texterkennung (OCR - Optical Character Recognition) von Handschrift eine wichtige, wenn auch ungleich weniger erforschte Disziplin in der Bildverarbeitung dar. Da handschriftliche chinesische Schriftzeichen in antiken Manuskripten vor allem mit Hilfe von Pinseln oder ähnlichen Schreibwerkzeugen erzeugt wurden, unterscheidet sich ihr Erscheinungsbild von dem mit modernen Mitteln wie Stiften geschriebenen oder gar gedruckten Zeichen erheblich. Die dadurch entstehende Varianz im Erscheinungsbild eines Zeichens kann das Erkennen von und die Segmentierung in enthaltene Teilstrukturen wie Striche wesentlich erschweren. In dieser Arbeit soll ein Verfahren vorgestellt werden, mit dessen Hilfe es möglich ist, handschriftliche chinesische Schriftzeichen in ihre einzelnen Striche aufzuteilen und diese anschließend bezüglich vorher definierter Merkmale zu untersuchen.

Das vorgestellte Verfahren stellt ein konturbasiertes Vorgehen dar, dessen Vorteil gegenüber anderen in dieser Arbeit erwähnten Ansätzen darin liegt, lokale Krümmungseigenschaften der Zeichenkontur durch die Verwendung der Delaunay-Triangulierung in Zusammenhang mit der Breite von Strichen zu analysieren und dadurch die Segmentierung eines handschriftlichen Zeichens in seine einzelnen Striche verbessert.

Die Arbeit ist in folgende Kapitel unterteilt:

Kapitel 2 beschreibt die allgemeine dieser Arbeit zugrunde liegende Problemstellung. Als erstes wird in 2.1 eine Einführung in die chinesische Schrift und ihre Entwicklung gegeben, um ein Grundverständnis für die Struktur asiatischer Schriftzeichen zu vermitteln. Dabei wird ein kurzer Überblick über die Entwicklung chinesischer Schriftzeichen gegeben, sowie der Aufbau der Zeichen aus Strichen und sogenannten Radikalen erklärt. Die Grundlagen der Manuskriptanalyse werden im folgenden Abschnitt 2.2 erläutert, bevor der abschließende Teil 2.3 des Kapitels einige Arbeiten beschreibt, die sich mit der Verarbeitung chinesischer Schriftzeichen befassen.

Die nächsten beiden Kapitel 3 und 4 beschreiben das in dieser Arbeit entwickelte Verfahren zur Strichextraktion genauer. Kapitel 3 erklärt die nötige Vorverarbeitung von digitalen Bildern chinesischer Schriftzeichen. Im ersten Schritt wird dabei eine Grauwertnormalisierung der Bilder vorgenommen, bevor sie auf eine Standardgröße vergrößert

ßert anschließend mit Hilfe eines Gauss-Filters geglättet werden. Der zweite Teil der Vorverarbeitung beschreibt die Detektion der Kanten eines Eingabezeichens durch den Marching Squares-Algorithmus und die darauffolgende Triangulierung der Konturpunkte durch die Delaunay-Triangulierung. In Kapitel 4 wird zunächst auf die Extraktion von Teilstrichen und Junctions eingegangen, welche die Grundelemente des Verfahrens darstellen. Nach diesem Schritt folgt die Verschmelzung dieser Teilstriche zu komplexeren Strichen sowie deren Klassifikation in eine der Grundstrichklassen chinesischer Schriftzeichen.

Im abschließenden Kapitel 5 wird das Verfahren auf eine Testmenge chinesischer Schriftzeichen aus einem Manuskriptausschnitt angewendet. Die Ergebnisse dieser Experimente werden analysiert und aufgetretene Schwierigkeiten anhand von Beispielen erläutert. Abschließend werden mögliche Weiterentwicklungen und Lösungsvorschläge für die aufgetretenen Probleme angesprochen.

2 Grundlagen

In diesem Kapitel werden zuerst einige Grundlagen der chinesischen Schrift erklärt, wobei darauf verzichtet wird, auf die komplexen sprachlichen Zusammenhänge „oberhalb“ der visuellen Ebene der einzelnen Zeichen einzugehen, da diese nicht relevant für die Problemstellung dieser Arbeit sind.

Anschließend werden die grundsätzliche Problemstellung sowie die verschiedenen Herangehensweisen an das Thema der Schriftverarbeitung und Manuskriptanalyse vorgestellt. Dabei wird zuerst auf die Ausgangssituation eingegangen und im Folgenden eine Auflistung verschiedener Ansätze und Methoden zur Aufgabenlösung gegeben.

2.1 Chinesische Schrift

Oft wird der Anfang einer Kultur mit der Entstehung einer entsprechenden Schrift gleichgesetzt, da diese die erste überprüfbare Möglichkeit bietet, kulturelle Errungenschaften zu fixieren und für die Nachwelt zu konservieren. Die chinesische Kultur gilt als eine der ältesten der Welt und entsprechend lang ist die Geschichte ihrer Sprache und Schrift. [Faz91] beschreibt in seinem Buch einige wichtige Aspekte der Entstehung der chinesischen Schrift. Die frühesten Funde von Vorläufern chinesischer Schriftzeichen datieren demnach auf das sechste Jahrtausend v. Chr. und stammen von Tonscheiben aus einem zentralchinesischen Dorf namens Banpo in der Nähe der chinesischen Großstadt Xi'an. Diese ersten Bilderschriften wurden noch mit weit primitiveren Schreibwerkzeugen auf entsprechenden Unterlagen verfasst als den zu späteren Zeiten üblichen Pinseln und Papierunterlagen. Erste Schreibutensilien bestanden häufig aus angespitzten Stöcken oder Steinen, mit welchen Bilder in Knochen oder Keramiken geritzt wurden.

Mit der Entwicklung erster Pinsel, welche im Gegensatz zu heute üblichen Haarpinseln Spitzen aus Fasern oder Lederstreifen besaßen, änderten sich auch die Unterlagen, auf denen geschrieben wurde. So wurden bald Holz, Bambus und Seide zum Verfassen von Texten verwendet, bevor die Erfindung von Papier einen weiteren Fortschritt in der Geschichte der Schrifterzeugung darstellte. Papier erlaubte es Schreibern durch das Aufsaugen der Tinte, große Variationen wie verschiedene Schattierungen im Erscheinungsbild von Strichen und Zeichen zu generieren. Später entwickelte Pinsel mit Spitzen aus Tierhaaren erhöhten die Kontrolle weiter, die Schreiber über ihre Schrift hatten und lieferten zusätzliche Möglichkeiten, rundere und individuellere Formen beim Schreiben zu erzeugen. Folgende wichtige Epochen nennt Fazzioli für die Entstehung der chinesi-

schen Schrift¹. Die erwähnten Schriftzeichentypen wie Piktogramme und Ideogramme werden im nächsten Abschnitt 2.1.1 erklärt.

1. Die *primitive Periode* von ca. 8000 - 3000 v. Chr. stellt die früheste Zeit dar, in der Vorläufer chinesischer Schriftzeichen verwendet wurden. Diese hatten noch sehr direkten graphischen Charakter und stellten Objekte relativ schwach abstrahiert und stattdessen sehr bildhaft dar. Auch frühe Formen von Piktogrammen wurden in dieser Epoche bereits benutzt.
2. In der darauffolgenden sogenannten *archaischen Periode* zwischen ca. 3000 und 1600 v. Chr. verbreitete sich die Verwendung von Piktogrammen stark und es wurde außerdem erstmals möglich, abstraktere Konzepte darzustellen, da diese Zeit die Einführung von Ideogrammen markierte.
3. Als *geschichtliche Periode* bezeichnet man den Zeitabschnitt der ersten großen Dynastien Chinas von ca. 1600. v. Chr., der Zeit der Yin-Dynastie, bis zur Han-Dynastie um 220 v. Chr. Damals wurden erstmals phonetische Zeichen in der chinesischen Schrift verwendet.
4. Den letzten bedeutenden Schritt in der Entwicklung der chinesischen Schrift nennt der Autor die *gegenwärtige Periode* von der Gründung der Volksrepublik China 1949 bis heute. In der zweiten Hälfte des 20. Jahrhunderts wurden aufgrund der hohen Analphabetenrate in China viele Zeichen vereinfacht, wodurch die heutige Unterscheidung zwischen traditionellen und vereinfachten Schriftzeichen entstanden ist.

Die Entwicklung der ersten chinesischen Schriftzeichen und Piktogramme wird grundsätzlich in mehrere aufeinander aufbauende Schreibstile eingeteilt, welche anhand einiger in Tabelle 2.1 vorkommender Schriftzeichen die Entwicklung zwischen den Stilen verdeutlichen soll.

In der Literatur werden sechs grundsätzliche Schreibstile der Entwicklung der chinesischer Schriftzeichen beschrieben, welche auch als die Hauptkategorien der chinesischen Kalligraphie bekannt sind. In verschiedenen Werken werden dabei zum Teil ähnliche, aber nicht identische Bezeichnungen verwendet. Die folgende Unterteilung wurde von [Sch90] vorgenommen.

1. Große Siegelschrift (Zhou-Dynastie, 800 v.Chr.–220 n.Chr.)
2. Kleine Siegelschrift (Qin-Dynastie, 221–207 v.Chr.)
3. Kanzleischrift (Ende Qin-Dynastie–Han-Dynastie, 206 v.Chr.–220 n.Chr.)
4. Vollkursivschrift, auch Gras- oder Konzeptschrift (Han-Dynastie, 206 v.Chr.–220 n.Chr.)

¹in den 2000 Jahren zwischen der geschichtlichen und der gegenwärtigen Epoche wurde die chinesische Schrift nicht signifikant weiterentwickelt. Stattdessen wurde sie in dieser Zeit größtenteils in der Form verwendet, wie sie sich bis zur Han-Dynastie entwickelt hatte. [Faz91]

























Bedeutung	Sonne	Regen	Vogel	Berg
Orakelknochen				
Siegelschrift				
Kanzleischrift				
Kursivschrift				
Grasschrift				
Regelschrift				

Tabelle 2.1: Historische Schreibstile der chinesischen Schrift

5. Normalschrift, auch Regelschrift (spätes 10. Jahrhundert)
6. Halbkursivschrift, auch Kursiv- oder moderne Schrift (frühes 3. Jahrhundert)

Die große und kleine Siegelschrift werden häufig zur Siegelschrift zusammengefasst. Wie aus den Beispielen in Tabelle 2.1 ersichtlich wird, unterscheidet sich das Erscheinungsbild dieser verschiedenen Schreibstile teilweise erheblich. Da sich neben den verwendeten Schreibwerkzeugen auch die dargestellten Schriftzeichen stark verändert haben und Informationen über die Teilstrukturen der ältesten Schreibstile der Orakelknochen und der Siegelschrift weit weniger verfügbar sind als über die späteren Schreibstile, wird die vorliegende Arbeit nicht auf die Verarbeitung von Zeichen dieser frühesten Epochen eingehen.

2.1.1 Schriftzeichen

Die meisten chinesischen Schriftzeichen sind *Logogramme*. Ein Logogramm steht für ein Wort oder ein Morphem, also eine Bedeutungseinheit einer Sprache. Dies steht im Gegensatz zu *Phonogrammen*, welche Phoneme wie die Buchstaben unseres Alphabets beschreiben und eine Aussprache von Sprachelementen definieren. Chinesische Schriftzeichen lassen sich jedoch noch feiner in verschiedene Unterklassen von Logogrammen einordnen. Dabei ist die Definition der einzelnen Typen chinesischer Schriftzeichen größtenteils, aber nicht vollständig vereinheitlicht. Die folgende Liste gibt eine Übersicht über die verschiedenen Typen nach der Definition von [Han93], welche jedoch auch bei den meisten anderen Unterscheidungen so vorkommen. Tabelle 2.2 zeigt Beispiele der einzelnen Schriftzeichentypen.



Abbildung 2.1: Piktogramm des Zeichens „Berg“

1. **Piktogramme** besitzen eine Bedeutung, nicht jedoch eine Aussprache. Sie stellen oft konkrete Objekte, in natürlichen Sprachen beispielsweise Substantive dar.
2. **Ideogramme** repräsentieren häufig abstraktere Konzepte wie Adjektive, Adverbien und Verben in natürlichen Sprachen. Festgelegte Aussprachen beziehungsweise Laute werden auch Ideogrammen nicht zugeordnet.
3. **Kombinierte Ideogramme** sind Zeichen, die aus mehreren einfacheren Zeichen bestehen, von denen jedes einzelne ein Ideogramm ist.
4. **Phonetikum-Signifikum-Kombinationen** bestehen aus mindestens einer lautgebenden Komponente, dem Phonetikum, sowie mindestens einem bedeutunggebenden Teil, dem Signifikum. Diese Teilkomponenten werden auch als Radikale bezeichnet. Schriftzeichen mit sowohl laut- als auch mit bedeutungsgebender Komponente kommen in der chinesischen Schrift häufig vor.
5. **Rebusträger** sind Zeichen, die ausschließlich lautgebenden Charakter haben, jedoch keine eigene Bedeutung tragen. Dieser Schriftzeichentyp ist also eher der Kategorie der Phonogramme statt der der Logogramme zuzuordnen.

Einige Autoren erwähnen weitere Schriftzeichentypen. [Sch90] beschreibt zu den oben genannten Typen beispielsweise zusätzlich die Klasse der *antithetischen Ideogramme*. Dies sind Ideogramme, deren Bedeutung die Umkehrung der Bedeutung eines anderen Ideogrammes darstellen. Der Autor schreibt jedoch, dass diese Klasse von Zeichen verhältnismäßig selten sei. Nach der Klassifizierung von [Faz91] existieren zusätzlich sogenannte *Ableitungen*. Das sind sinnverwandte Schriftzeichen, welche ähnliches Aussehen sowie ähnliche Bedeutungen, jedoch unterschiedliche Aussprachen haben.

Heutzutage bestehen die meisten chinesischen Wörter aus zwei oder mehr Schriftzeichen, einzelne Zeichen stellen selten komplette Wörter dar. Dies bedeutet, dass Zeichen unterschiedlicher optischer Gestalt identische Phonetik besitzen können.

2.1.2 Radikale

Als Radikale bezeichnet man die einzelnen bedeutungstragenden Bestandteile komplexer chinesischer Schriftzeichen. Zeichen der Kategorie der im letzten Abschnitt beschriebenen Phonetikum-Signifikum-Kombinationen, zu der die meisten chinesischen Schriftzeichen gezählt werden, enthalten mindestens ein solches Radikal als Signifikum sowie eine variable Anzahl von Radikalen als Phonetika. Über die Bezeichnung der Phonetika als

Zeichentyp	Bedeutung	Beispielzeichen
Piktogramm	Berg	山
Ideogramm	drei	三
Kombiniertes Ideogramm	hell (aus den Zeichen für Mond und Sonne zusammengesetzt)	明
Phonetikum-Signifikum-Kombination	sie (weibliches Personalpronomen)	她
Rebusträger	kommen	來

Tabelle 2.2: Beispiele der Schriftzeichentypen

女 夕 大 女 子 鼻 齊

(a) Radikale mit 3 Strichen

(b) Radikale mit 14 Strichen

Abbildung 2.2: Radikale verschiedener Komplexität

Radikale gibt es unterschiedliche Auffassungen. Während einige Autoren die lautgebenden Anteile eines Zeichens auch als Radikale bezeichnen ([Sch90]), werden in anderen Arbeiten nur die bedeutungstragenden Teile eines Zeichens Radikal genannt. Strukturell unterscheiden sich Signifikum- und Phonetikumanteile nicht, wodurch Elemente identischer Struktur in manchen Zeichen als lautgebender und in anderen als bedeutungstragender Teil auftreten können. Zur Zeit des als erstes chinesisches Zeichenlexikon geltenden *Shou wen jie zi*-Wörterbuches aus dem Jahr 121 n. Chr. existierten 540 Radikale. Seit dieser Zeit werden chinesische Schriftzeichen in Zeichenlexika nach ihren Radikalen geordnet und in entsprechende Kategorien sortiert. Die Radikale selber werden über die Anzahl ihrer Einzelstriche sortiert, welche bei den heute gebräuchlichen Radikalen von 1 bis 17 reicht. Abbildung 2.2 zeigt einige Beispiele für einfache Radikale mit 3 Strichen sowie komplexere Radikale mit 14 Strichen. Im Jahr 1717 wurde die Anzahl der Radikale zur Vereinfachung der Schrift von 540 auf 214 reduziert, welche bis heute gebräuchlich sind und zur Zeicheneinteilung verwendet werden.

2.1.3 Striche

Striche stellen die elementarsten Bestandteile jedes chinesischen Zeichens dar. Als handschriftliche Striche gelten die graphischen Elemente, die in einem Zug geschrieben werden, ohne dass das Schreibwerkzeug, wie Pinsel oder Stift, auf dem Weg zwischendurch von der Schreibunterlage abgehoben wird.

Striche können aufgeteilt werden in *Grundstriche* und *komplexe Striche* ([Ber97]). Erstere stellen horizontale, vertikale oder diagonale Elemente dar, die keine wesentliche Richtungsänderung enthalten, während komplexe Striche als Kombinationen von Grundstrichen konstruiert werden. Auf diese Weise können zusammengesetzte chinesische Striche plötzliche Richtungsänderungen enthalten und theoretisch beliebig komplex sein. Die Klassifizierung von Strichen in chinesischen Schriftzeichen ist jedoch nicht vollständig einheitlich definiert. Während die sechs Grundstriche ohne große Unterschiede meistens den in Tabelle 2.3 vorgestellten Strichen entsprechen, variiert die Anzahl und Einteilung der komplexen Striche von Autor zu Autor teilweise erheblich. Im Folgenden wird auf einige Einteilungen eingegangen. [Sch90] beschreibt ein System aus sechs Grundstrichen, die, erweitert um einige Variationen, in Abbildung 2.3 zusammen mit der Richtung, in der sie geschrieben werden, sowie mit einem Beispielzeichen dargestellt werden. Diese Liste kann durch Kombinationen der Grundstriche um dreizehn weitere komplexe Striche ergänzt werden. Alle chinesischen Schriftzeichen bestehen laut Schmidt aus Kombinationen dieser Striche. Auch [Ber97] definiert sechs Grundstriche, die die Basis für alle komplexen Striche und damit für alle chinesischen Schriftzeichen darstellen. Tabelle 2.3 zeigt diese sechs Grundstriche mit ihrer Schreibrichtung, der Übersetzung ihrer chinesischen Bezeichnung sowie einem entsprechenden Beispielzeichen. Durch Kombinationen der Grundstriche können wiederum komplexe Striche konstruiert werden. Der Autor erwähnt dabei, dass die Anzahl der komplexen Striche nicht standardisiert ist, wodurch komplexe Striche, welche in einigen Quellen als eigenständige Striche gelten, an anderer Stelle als Variationen gelistet werden. Berkenbusch legt dabei selber keine feste Anzahl komplexer Striche fest, erwähnt in dem Zusammenhang jedoch, dass beispielsweise der Haken („hook“) (siehe Tabelle 2.4) am Ende von Strichen von manchen Autoren als Variation eines Striches oder als komplexer Strich angesehen wird, während andere Autoren keine Unterscheidung zwischen Grundstrichen mit oder ohne Haken vornehmen.

Der Unicode-Zeichensatz ab Version fünf enthält insgesamt 36 sogenannte *CJK-Striche*², von denen acht Striche als Grundstriche und die restlichen 28 Striche als zusammengesetzte beziehungsweise komplexe Striche aufgeführt werden. Die unterschiedliche Anzahl der Grundstriche kommt daher, dass hier der bereits erwähnte Haken am En-

²Der Name CJK-Strich ist aus den Namen der Schriften **C**hinesisch, **J**apanisch und **K**oreanisch hergeleitet, deren Schriftzeichen in Unicode zu den sogenannten *CJK-Zeichen* zusammengefasst werden. Der Grund für die Zusammenfassung von drei Schriftsystemen macht sich die Tatsache zunutze, dass sowohl die japanische als auch die koreanische Schrift in relativ großem Umfang chinesische Schriftzeichen neben ihren eigenen Silbenalphabeten verwenden. Da diese gemeinsamen Schriftzeichen aus den erwähnten 36 CJK-Strichen aufgebaut sind, lassen sich so die grundlegenden Elemente mehrerer Schriften auf einmal darstellen. Des Weiteren existiert auch eine Erweiterung mit dem Namen *CJKV*, welche zusätzlich noch die vietnamesische Schrift miteinbezieht, da auch in Vietnam bis ins 20. Jahrhundert chinesische Schriftzeichen verwendet wurden.

Grundstrich	Schreibrichtung	Bezeichnung	Beispielzeichen
●	↘	„dot“	六
一	→	„horizontal“	大
㇇	↙	„press down“	人
㇈	↘	„throw away“	失
丨	↓	„vertical“	中
㇏	↗	„Rise“	水

Tabelle 2.3: Sechs Grundstriche chinesischer Schriftzeichen nach [Ber97]

Grundstrich	Schreibrichtung	Bezeichnung
㇏	↓	„bend“
㇏	-	„hook“

Tabelle 2.4: Zwei zusätzliche Typen chinesischer Grundstriche

de chinesischer Striche als eigenständiger Grundstrich geführt wird und die Klasse des herabfallenden, diagonalen Striches in zwei Unterklassen aufgeteilt wird. Die übrigen Grundstrichtypen entsprechen den von [Ber97] beschriebenen Klassen. Diese Klassifizierung richtet sich nach den Grundelementen der chinesischen Kalligrafie. Tabelle 2.4 zeigt diese beiden zusätzlichen Grundstrichtypen.

Die „Unicode CJK Unified Ideographs Extension“, aufgeteilt in die Teile A und B, listet alle im Unicode enthaltenen CJK-Striche und -Zeichen.

Die Schreibrichtung der Striche stellt besonders für die Strichanalyse und -extraktion von mit Pinseln oder ähnlichen Schreibutensilien erzeugten handschriftlichen Zeichen ein wichtiges Indiz dar. Prinzipiell werden Striche von oben nach unten und von links nach rechts geschrieben, es existieren jedoch ein paar Ausnahmen. Für alle Striche existieren außerdem sieben Grundregeln für die Reihenfolge, in der Striche eines Zeichens geschrieben werden:

1. Horizontale Striche werden vor vertikalen Strichen geschrieben.
2. Steigende diagonale Striche werden vor fallenden diagonalen Strichen geschrieben.
3. Vertikale aufeinanderfolgende Striche werden von oben nach unten geschrieben.
4. Horizontal aufeinanderfolgende Striche werden von links nach rechts geschrieben.

丶	点	(diǎn)	丿	不, 六
一	横	(héng)	→	不
丨	竖	(shù)	↓	不, 你
㇇	撇	(piě)	㇇	心
㇏	捺	(nà)	㇏	八, 大
㇀	提	(tí)	㇀	我
㇇→	横钩	(hénggōu)	㇇→	你, 好
丨↓	竖钩	(shùgōu)	丨↓	你, 好
㇇↘	斜钩	(xiégōu)	㇇↘	我
→	横折	(héngzhé)	→	口
丨↘	竖折	(shùzhé)	丨↘	七

Abbildung 2.3: Elf Grundstriche chinesischer Schriftzeichen nach [Sch90]

5. Umschließende Striche werden vor umschlossenen Strichen geschrieben.
6. Umschlossene Komponenten werden geschrieben, bevor die umschließenden Striche geschlossen werden.
7. Bei mehreren horizontal angeordneten vertikalen Grundstrichen wird der mittlere Grundstrich zuerst geschrieben.

Da sich mit der Entwicklung der Schriftstile nach Abschnitt 2.1 auch die Schriftzeichenbestandteile wie Striche und Radikale stark verändert haben, kann ein automatisierter strichbasierter Analyseansatz, wie ihn diese Arbeit bereitstellen möchte, nicht alle Stile gleichzeitig behandeln. Aus diesem Grund wird sich diese Arbeit auf die Analyse der in den meisten zu untersuchenden Manuskripten vorkommenden Zeichen der Stile Kanzlei-, Kursiv-, Regel- und Grasschrift beschränken, deren Striche eine relativ hohe Ähnlichkeit aufweisen und welche zum großen Teil den im modernen Chinesisch verwendeten Schriftzeichen entsprechen.

2.2 Manuskriptanalyse

Die Verarbeitung von antiken chinesischen Manuskripten ist sowohl für die Sinologie als auch für die Informatik aus verschiedenen Gründen eine interessante und wichtige Herausforderung. In der Sinologie beschäftigt man sich unter anderem mit der Aufgabe, durch die Untersuchung teilweise mehrere tausend Jahre alter Manuskripte nicht nur nach ihren inhaltlichen, sondern auch bezüglich ihrer gestalterischen Gesichtspunkte Hinweise auf die Herkunft, die historische Einordnung und eventuell die Zuordnung eines

Textes, eines Textabschnittes oder auch nur eines einzelnen Zeichens zu erlangen. Kombinationen von bestimmten Ausprägungen verschiedener Merkmale wie der Breite, Länge, Lage oder Neigung einzelner Striche können zu für bestimmte Autoren beziehungsweise Schreiber oder Schreiberschulen repräsentative Merkmalsvektoren zusammengefasst und für die geschichtliche Einordnung sowie zur besseren Interpretation von Texten genutzt werden. [Ric04] beschreibt, welche Bedeutung die Unterscheidung verschiedener Handschriften in Manuskripten für die Sinologie hat und welche Aspekte hierfür bei der Manuskriptanalyse wichtig sind. Durch die Zuordnung einzelner Schriften oder auch nur Textabschnitte zu bestimmten Schreibern oder Schreiberschulen können Texte zu zusammengehörenden Skripten gruppiert und klassifiziert werden. „Schreiber“ steht hierbei nicht gleichbedeutend für „Autor“, da die Beherrschung der chinesischen Schrift in der chinesischen Antike eine eigene Disziplin darstellte und Schreiber für das Aufschreiben von Texten, welche von anderen Personen verfasst wurden, angestellt wurden.

Des Weiteren können bei identifizierten Handschriften Rückschlüsse darüber gezogen werden, in welchem Zusammenhang und in welchem formalen Kontext Manuskripte verfasst wurden. Das Wissen über die Schreiber einzelner Texte führt zu einem besseren Verständnis eines kompletten Werkes und hilft bei dessen geschichtlicher Einordnung.

Für die Informatik ist die Entwicklung von Techniken, die die Manuskriptanalyse unterstützen und automatisieren ein interessantes und anspruchsvolles Forschungsgebiet. Dabei stellen sowohl die Beschaffenheit von mit Pinseln geschriebenen und mitunter nur unzureichend in seine enthaltenden Striche aufteilbaren Schriftzeichen, als auch durch das Alter und die Beschaffenheit des verwendeten Materials entstandene Fehler in den Schriftzeichen große Herausforderungen dar.

2.3 Bisherige Arbeiten zur Schriftanalyse

[SYB07] unterscheiden zwischen drei prinzipiellen Herangehensweisen an die Verarbeitung von handschriftlichen oder gedruckten chinesischen Zeichen, welche zu den in 2.1.1 erläuterten strukturellen Komponenten eines Schriftzeichens korrespondieren. Es wird hier unterschieden zwischen der Analyse

- ganzer Zeichen (holistischer Ansatz),
- der Radikale chinesischer Schriftzeichen,
- der einzelnen Striche chinesischer Schriftzeichen.

Obwohl die meisten Arbeiten in dem Gebiet der Zeichenanalyse als Ziel die Schrifterkennung haben, sind ihre grundsätzlichen Herangehensweisen in großen Teilen auch für die Manuskriptanalyse anwendbar, da beide Disziplinen als Zwischenziel die Extraktion zusammenhängender Strukturen aus einem Schriftzeichen haben.

Der Unterschied liegt im weiteren Vorgehen der beiden Disziplinen. Während die Schrifterkennung anhand von Datenbanken und Strukturmodellen versucht, Zeichen zu

erkennen und von individuellen Ausprägungen einzelner Zeichen abstrahiert, ist es das Ziel der Handschriftanalyse, gerade diese von Schreiber zu Schreiber unterschiedlichen Merkmale zu extrahieren und zu analysieren. Bei der Handschriftanalyse ist damit weniger die Erkennung von Schriftzeichen das Ziel als die Extraktion bestimmter Merkmale, welche Rückschlüsse über den Schreiber, Schreibstil oder eine Schreiberschule geben.

[WH00] erläutert einige Gründe, welche die Erkennung und Analyse chinesischer Schriftzeichen besonders anspruchsvoll gestalten. Zum einen ist der der Verarbeitung zugrundeliegende Zeichensatz im Chinesischen ungleich größer als bei westlichen Schriften wie beispielsweise der lateinischen, für welche inzwischen bereits relativ verlässliche kommerzielle Systeme existieren. Die Gesamtzahl der chinesischen Zeichen liegt bei über 60000 sich teilweise strukturell nur leicht unterscheidender Schriftsymbole, wodurch die Gefahr der Falsch-Klassifizierung stark steigt und die Erfolgsquote eines OCR-Systems für chinesische Schrift deshalb weit unterhalb der eines auf unserem Alphabet basierenden Verfahrens liegt. Eine weitere Schwierigkeit bei der Verarbeitung chinesischer Schriftzeichen liegt in der hohen strukturellen Komplexität einzelner Zeichen. Wählt man als Maß für die Komplexität die Anzahl der das Zeichen konstruierenden Striche, wird der Unterschied zu unserem Alphabet deutlich. Während beispielsweise die durchschnittliche Strichzahl aller 26 Großbuchstaben des Alphabets bei 2,41 Strichen pro Buchstabe liegt, sind chinesische Schriftzeichen in der Regel aus einer weit höheren Anzahl einzelner Striche aufgebaut. Beispielsweise bestehen die Zeichen aus dem in Kapitel 5 beschriebenen Manuskriptausschnitt, mit welchem das in den folgenden Kapiteln vorgestellte Verfahren getestet wird, aus durchschnittlich 9 Grundstrichen. Diese Komplexität resultiert aus dem extrem umfangreichen Zeichensatz, da eine hohe Anzahl von Kombinationen einzelner Teilelemente nötig ist, damit alle Zeichen optisch voneinander unterscheidbar bleiben.

Trotz dieser Schwierigkeiten gibt es eine große Zahl an vielversprechenden Ansätzen, die sich der vorgestellten Problematik annehmen. In den folgenden Abschnitten wird auf die verschiedenen Ansätze eingegangen. Verfahren, welche Schriftzeichen als Ganzes oder als Radikalkomposition verarbeiten, werden dabei nur kurz angesprochen, während der Schwerpunkt auf der Erklärung einiger strichbasierter Algorithmen liegt, da das in dieser Arbeit entwickelte Verfahren in die gleiche Richtung geht.

2.3.1 Holistische und radikalbasierte Verfahren

Holistische Verfahren

Holistische Ansätze betrachten komplette Zeichen ohne weitere Zerlegung in Teilkomponenten, da das Finden dieser Teilkomponenten eine schwierige und in manchen Fällen unlösbare Aufgabe darstellt, die den Prozess der Schriftzeichenanalyse oder -erkennung erheblich erschweren kann. Die hohe Anzahl chinesischer Schriftzeichen und die dadurch resultierende teilweise sehr hohe strukturelle Ähnlichkeit zwischen verschiedenen Zeichen stellt jedoch eine große Herausforderung für holistische Verfahren dar. Ein Beispiel für ein Verfahren dieser Kategorie stellt der Ansatz von [Her07] dar, welcher auf der Erstellung sogenannter *Shape Contexts* beruht und zur Berechnung der Ähnlichkeit zweier

Formen verwendet werden kann. Bei diesem Verfahren wird die Kontur oder das Skelett eines Zeichens in Kontur- beziehungsweise Skelettpunkte aufgeteilt und für jeden solchen Punkt ein Shape Context erzeugt. Das Konzept eines Shape Contexts beschreibt die Position beziehungsweise Lage eines Punktes in Beziehung zu allen übrigen Punkten der Form. Über den Vergleich aller Shape Contexts zweier Zeichen kann die Ähnlichkeit ihrer Formen berechnet werden.

Radikalbasierte Verfahren

Radikalbasierte Ansätze versuchen, chinesische Schriftzeichen in ihre Radikale (siehe Abschnitt 2.1.2) zu zerlegen und anhand der gewonnenen Informationen über Vorkommen und Anordnung einzelner Radikale Schriftzeichen zu identifizieren oder zu analysieren. Da radikalbasierte Verfahren der Zeichenerkennung und -analyse häufig für antike Schriften verwendet werden, die deutlich vor der Qing-Dynastie datieren, muss bei diesen Ansätzen die höhere Anzahl möglicher auftretender Radikale berücksichtigt werden. Wie in Abschnitt 2.1.2 beschrieben, können in Schriftzeichen aus alten Manuskripten 540 verschiedene Radikale vorkommen. Trotzdem stellt diese Menge im Vergleich zur ungleich höheren Anzahl existierender Schriftzeichen eine starke Komplexitätsreduktion der zu untersuchenden Domäne dar. Bei der Segmentierung eines Schriftzeichens in seine Radikale kann daher eine Verringerung des Aufwandes zur Detektion einzelner Strukturen durch die Analyse der Anordnung der gefundenen Teilstrukturen innerhalb eines Zeichens erreicht werden. Abbildung 2.4 zeigt die Aufteilung mehrerer komplexer chinesischer Schriftzeichen in ihre Radikalstrukturen.

[SDG03] stellen ein radikalbasiertes Verfahren vor, welches in einem ersten Schritt das Skelett der Zeichenform konstruiert und anschließend sogenannte *landmark points* auf den Skelettkanten findet, welche zur Identifikation des Radikals dienen. Die Erkennung von Schriftzeichen wird dann mit Hilfe des Viterbi-Algorithmus' und den gefundenen Radikalen vorgenommen. Die in einem Schriftzeichen vorkommenden Radikale sowie ihre Positionen beziehungsweise Anordnungen stellen einen wichtigen Ansatzpunkt für die Identifikation eines Zeichens dar. Radikalbasierte Verfahren können gegenüber holistischen Verfahren den Vorteil haben, dass sie bezüglich lokaler Bildstörungen robuster sind, da die Fehlklassifizierung eines Radikales keinen Einfluss auf die Erkennung eines anderen in diesem Zeichen enthaltenen Radikales hat. Ein weiterer Vorteil ergibt sich durch die im Vergleich zu Schriftzeichen geringe Anzahl von Radikalen. Die Unterscheidung zweier strukturell sehr ähnlicher, aber nicht identischer, komplexer chinesischer Schriftzeichen kann sich in vielen Fällen als sehr schwierig herausstellen. Im Gegensatz dazu sind die meisten der 540 Radikale strukturell weit weniger komplex, wodurch sich der Vergleich auf Identität zwischen Radikalen erheblich einfacher gestalten kann als der zwischen kompletten Schriftzeichen. Aufgrund der geringen Menge an Radikalen im Vergleich zur Anzahl aller chinesischen Schriftzeichen folgt außerdem eine höhere Wahrscheinlichkeit für das Auftreten einzelner Radikale als für das kompletter Schriftzeichen auch in einzelnen Schriftstücken. Daraus ergibt sich eine vergrößerte Grundlage, um statistische Auswertungen über die Eigenschaften von Strukturen der Schrift aufzustellen.

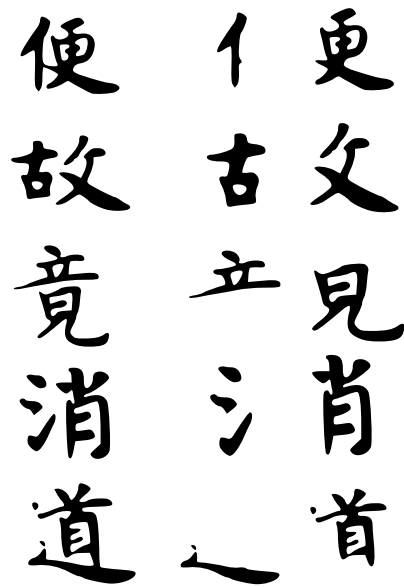


Abbildung 2.4: Chinesische Schriftzeichen und ihre Radikale

2.3.2 Strichbasierte Verfahren

Strichbasierte Ansätze gehen einen Schritt weiter und versuchen, ein Schriftzeichen in die Einzelstriche, aus denen es besteht, zu zerlegen. Damit stellen sie eine Weiterführung der Idee radikalbasierter Verfahren dar, wonach die Aufgabe der Erkennung hochkomplexer Schriftzeichenstrukturen auf die Erkennung weniger komplexer Teilstrukturen und deren relative Anordnung zueinander reduziert werden kann. Dafür werden Schriftzeichen zuerst in eine Komposition von Instanzen verschiedener, vorher definierter Grundstriche segmentiert und anschließend deren Anordnung, Ausprägungen sowie extrahierte Kreuzungen zur Klassifizierung der kompletten Zeichen verwendet. Viele Verfahren zur Strichextraktion basieren auf der Skelettierung einer ein Zeichen repräsentierenden Form als ersten Schritt. Skelettierungen vereinfachen die Repräsentation einer Form in der Ebene, bewahren dabei jedoch die wichtigsten Formeigenschaften und können auf unterschiedliche Weisen gewonnen werden, indem aus allen topologischen Flächen pixelbreite Skelettlinien erzeugt werden. Die häufigste Art der Skelettgewinnung ist dabei die Ausdünnung der Form, auf Englisch als *thinning* bezeichnet. Aus dem resultierenden Skelett werden dann üblicherweise die Kreuzungs- und Endpunkte als sogenannte *interest points* extrahiert, welche wiederum zur Extraktion von Strichen verwendet werden. Eine Schwäche dieser Verfahren ist allerdings, dass beim Ausdünnen der Form eines Schriftzeichens häufig unechte Kanten im Bereich von Strichkreuzungen und anderen Stellen entstehen, welche in nachfolgenden Schritten zu fehlerhafter Strichextraktion führen können. Die weißen Kreise in Abbildung 2.5 markieren einige Beispiele für solche unechten Kanten. Es gibt jedoch verschiedene Ansätze, solche überflüssigen Segmente innerhalb eines Skelettes zu entfernen.

[LHS97] beschreiben ein strichbasiertes Verfahren, welches auf der Skelettierung und anschließenden Extraktion von einer neu definierten Art von *interest points* basiert. Die

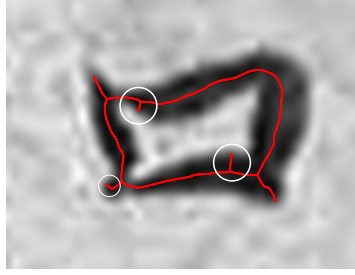


Abbildung 2.5: Skelett eines chinesischen Schriftzeichens (rot) mit unechten Skelettkanten (durch weiße Kreise markiert)

Autoren geben an, dass für ihr Verfahren kein bestimmter Skelettierungs-Algorithmus verwendet werden muss, sie jedoch den Algorithmus von [WZ89] gewählt haben, da dieser die Struktureigenschaften einer Originalform gut erhalte. Nach der Skelettierung einer Schriftzeichenform definieren sie Klassen *fork point* und *end point* als Merkmalspunkte beziehungsweise *interest points* auf dem Skelett. Dabei beinhaltet die Klasse *fork point* sowohl *T-junction points* als auch *crossing points*. Bezeichne im Folgenden P einen beliebigen Pixel des Bildes sowie S_b die Menge der extrahierten Skelettpixel. Dann ist die Menge der *interest points* S_n folgendermaßen definiert:

$$S_n = S_e \cup (S_3 \cap S_b) \quad (2.1)$$

S_e bezeichnet dabei die Menge der *end points*. Der zweite Teil der Vereinigung $S_3 \cap S_b$ beschreibt die Menge der *fork points* als Schnitt zwischen der Menge aller Skelettpixel S_b und der Menge S_3 , welche sich folgendermaßen berechnet:

$$S_3 = \{P | (N_c(P) \geq 3) \vee (N_b(P) \geq 4)\} \quad (2.2)$$

Hierbei gibt $N_b(P)$ die Anzahl der Skelettpixel in der 8er-Nachbarschaft des Punktes P an. In S_3 befinden sich damit alle Pixel P , deren sogenannte *crossing number* $N_c(P)$ mindestens 3 ist oder die 4 oder mehr Skelettpunkte in ihrer 8er-Nachbarschaft haben. Die *crossing number* $N_c(P)$ beträgt dabei die Hälfte der Wechsel zwischen Skelettpunkten und nicht-Skelettpunkten in der 8er-Nachbarschaft von P . Dabei ist $B(P) = 1$, wenn P ein Skelettpunkt ist und $B(P) = 0$ sonst.

$$N_c(P) = \frac{1}{2} \sum_0^8 |B(P_{i+1}) - B(P_i)| \quad (2.3)$$

Durch diese zwei alternativen Bedingungen für einen *fork point* lösen die Autoren das oft bei der Skelettierung auftretende Problem der Anfälligkeit für Störungen im Bereich von Kreuzungspunkten. Ansätze, welche sich auf die *crossing number* einzelner Skelettpixel zur Extraktion von *fork points* beschränken, versagen bei Pixelkonfigurationen wie der in Abbildung 2.6. Dabei haben die vier mittleren Bildpunkte, welche zusammen einen *interest point* darstellen, jeweils die *crossing number* 2 und qualifizieren sich dadurch

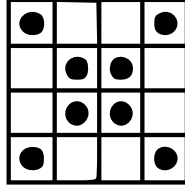


Abbildung 2.6: Die vier mittleren Bildpunkte haben jeweils die crossing number 2.

nicht als *fork point*. Die alternative Bedingung für einen *fork point* greift in diesem Fall jedoch. Da die Anzahl der Skelett-Nachbarpixel der vier mittleren Bildpunkte jeweils 4 beträgt, werden sie als *fork points* erkannt. Nachdem alle Verzweigungs- und Endpunkte des Skelettes gefunden wurden, extrahiert das Verfahren die Striche des Schriftzeichens mit Hilfe eines geometrischen Graphen G , dessen Knoten die extrahierten *interest points* sind. In einem ersten Schritt werden multiple *interest points*³, die für die gleiche Strichkreuzung im Originalbild stehen, verschmolzen und *interest points*, welche durch überflüssige Skelettzweige wie denen aus Abbildung 2.5 entstanden sind, gelöscht. Zu einem Strich gehörende Skelettsegmente werden anschließend durch eine Regelmenge identifiziert und zu einem kombinierten Segment zusammengefasst.

Wie die Autoren erklären, ist das Ziel ihres Verfahrens die Extraktion von Strichen und die anschließende auf den Strichen basierende Zeichenerkennung. Das Ziel dieser Arbeit ist jedoch die Strichanalyse handschriftlicher chinesischer Texte. Das vorgestellte Verfahren eignet sich dazu nur eingeschränkt, da die Repräsentation der Striche lediglich durch ihre Skelettlinien eine zu starke Abstraktion von den ursprünglichen Strichformen darstellt und die Berechnung von für die Manuskript- und Schreiberanalyse interessante Stricheigenschaften verhindert.

Ein weiteres Beispiel eines skelettbasierten Ansatzes ist das von [WH00] beschriebene OCCR-System (Offline Chinese Character Recognition), welches ein gegebenes Zeichen durch eine Lauflängen-Kodierung skelettiert und anschließend aus dem Skelett die zum Zeichen gehörenden Striche extrahiert. Danach wird das Zeichen anhand der gewonnenen Strichinformationen in einem zweistufigem Verfahren zuerst grob und anschließend genauer klassifiziert. Das konstruierte Skelett besitzt keine der in anderen Verfahren benutzten und oben erwähnten *interest points* wie Kreuzungs- oder Endpunkte, sondern nur separate Skelettlinien, wodurch Wei-Hsien das Problem der Störungen umgeht, welche bei Skelettierungen oft im Bereich sich kreuzender Linien entstehen.

Die Skelettierung des gegebenen Schriftzeichens wird durch die Erzeugung sogenannter *Laufabschnitte* (engl. „run slices“) in horizontaler, vertikaler und zwei diagonalen Richtungen erreicht. Ein Laufabschnitt besteht aus einer Liste von Punkten innerhalb der Zeichen-Kontur, welche von einer Konturseite des Zeichens in den angegebenen Richtungen (horizontal, vertikal und zweimal diagonal) bis zur gegenüberliegenden Seite geht. Bei der Erzeugung der Laufabschnitte in verschiedenen Richtungen wird dabei von jeweils unterschiedlichen Konturteilen ausgegangen. Die horizontalen Laufabschnitte wer-

³multiple *interest points* sind individuelle extrahierte *interest points*, die jedoch alle denselben tatsächlichen *interest point* repräsentieren

den ausgehend von der linken Kontur des Zeichens erzeugt, die vertikalen starten von der unteren Seite. Die Mengen der beiden diagonalen Laufabschnitte werden sowohl von der linken als auch von der unteren Konturseite einer Zeichenform erzeugt. Für die so gewonnenen Laufabschnitte berechnet der Algorithmus anschließend die *Laufängen* (engl. „*run lengths*“), also die Längen der einzelnen Abschnitte. Durch die Analyse der Laufängen aller erzeugten Laufabschnitte wird anschließend das Skelett des Zeichens gewonnen. Das Skelett wird aus den Mittelpunkten einzelner Laufabschnitte erzeugt. Welche der in vier verschiedenen Richtungen verlaufenden Laufabschnitte dazu verwendet werden, wird für jeden Punkt innerhalb der Zeichenkontur anhand der Laufängen der durch ihn verlaufenden Abschnitte entschieden. Generell sieht das Vorgehen so aus, dass immer die Mittelpunkte der kürzeren Laufabschnitte für das Skelett verwendet werden. So wird bei Punkten, deren horizontaler Laufabschnitt länger ist als ihr vertikaler der Mittelpunkt des vertikalen Abschnittes verwendet. Analog wählt der Algorithmus bei Punkten, deren vertikale Laufänge größer ist als ihre horizontale den Mittelpunkt des horizontalen Laufabschnittes für die Generierung des Skelettes. Die Extraktion diagonalen Skelettsegmente wird ähnlich vorgenommen. Bei diesem Vorgehen werden Kreuzungsregionen durch ihre hohen Laufängen sowohl in horizontaler als auch in vertikaler Richtung identifiziert. Für Punkte, die in Kreuzungsregionen liegen, werden keine Skelettpunkte erzeugt, wodurch der Autor das Problem der oftmals gestörten Kreuzungsregionen in Skelettierungen umgeht und paarweise voneinander getrennte einzelne Skelettsegmente erhält. Auch dieses Verfahren ist aufgrund seiner Verwendung von Skelettkanten als Strichrepräsentation für die Extraktion von Strichen gut geeignet, liefert jedoch keine Struktur, welche die Analyse der Stricheigenschaften ermöglicht.

[CT00] beschreiben ein Verfahren zur Strichextraktion, welches nicht auf dem Zeichenskelett sondern auf dem ursprünglichen, nicht ausgedünnten *thick-line-image* arbeitet. Nach der Konturextraktion werden die dicken Linien dieses Bildes in Endregionen, reguläre und singuläre Regionen unterteilt, welche mit den Endpunkten, mittleren Punkten und Verzweigungspunkten anderer, skelettbasierter Verfahren korrespondieren (siehe Abbildung 2.7). Um jeden Pixel des Bildes in eine dieser Kategorien einzuordnen, schlagen die Autoren ein Verfahren vor, welches den Grad des Pixels anhand sogenannter „PBODs“ (point-to-boundary orientation distances) ermittelt. Die Verteilungen dieser PBODs über einen Bereich von 0 bis 360 Grad geben an, wie groß die Entfernung des aktuell untersuchten Pixels zu einem Randpixel in jeder der 360 Richtungen ist. Hohe PBOD's werden somit für Richtungen bestimmt, in welche Striche, zu denen der aktuelle Pixel gehört, verlaufen. Der *Grad* eines jeden Pixels wird bestimmt als die Anzahl der lokalen Maxima der PBOD-Verteilungen, die oberhalb eines Schwellenwertes liegen. Daraus resultiert für Pixel in Endregionen ein Grad von 1, Pixel in regulären Regionen sind vom Grad 2 und Pixel in singulären Regionen beziehungsweise Kreuzungsregionen haben einen Grad von mindestens 3. Anschließend werden die End- und regulären Regionen entsprechend der bestimmten Pixelgrade extrahiert. Im folgenden Schritt wird für alle Pixel, welche einen Grad größer als 2 haben und damit zu den singulären Regionen gehören, ein sogenannter ρ -Space erstellt. Die Dimensionen dieses dreidimensionalen Raumes bestehen aus den beiden räumlichen Bilddimensionen sowie einer Richtungsdimension. Über den ρ -Space wird bestimmt, zu welchen Strichen das analysierte Pixel gehört. Die

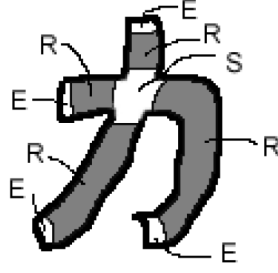


Abbildung 2.7: Pixeleinteilung in End-, reguläre und singuläre Regionen nach [CT00]

Autoren beschreiben das Verfahren als sowohl für gedruckte als auch für handschriftliche chinesische Schriftzeichen geeignet. Da es auf den ursprünglichen *thick-line-images* der chinesischen Schriftzeichen arbeitet, gehen während der Strichextraktion, im Gegensatz zu den auf Skelettierungen basierenden vorangegangenen Verfahren, keine für die Strichanalyse interessanten Informationen verloren. Die Idee von Regionen statt *interest points* als Repräsentation für Strichkreuzungen wird in der vorliegenden Arbeit aufgegriffen.

Andere strichbasierte Algorithmen beruhen auf der Konturanalyse von Zeichen, um dieses in einzelne Striche zu segmentieren. [HY00] beschreiben ein solches Verfahren, welches als Vorverarbeitungsschritt für die Verbesserung der Skelettierung eines Zeichens dessen Einzelstriche extrahiert. Das Verfahren basiert grundsätzlich auf der Aufteilung der Kontur eines Zeichens in Kontursegmente, sogenannte *Boundary Lines*, aufgrund von lokalen Krümmungseigenschaften und der anschließenden Klassifizierung dieser Segmente sowie deren Gruppierung zu Strichen. Im ersten Schritt werden mit einer Kantendetektion, welche geschlossene Kantenzüge liefert und nicht näher erläutert wird, die Konturen des Zeichens in einem Bild mit einem Schriftzeichen extrahiert. Anschließend werden für alle Punkte jeder extrahierten Kontur zwei $FDir(P(i, j))$ („forward direction“), bezeichnet mit $\frac{\partial y_f(i, j)}{\partial x_f(i, j)}$, beziehungsweise $ADir(P(i, j))$ („after direction“), bezeichnet mit $\frac{\partial y_a(i, j)}{\partial x_a(i, j)}$, genannte Werte folgendermaßen berechnet, wobei $P(i, j)$ den j ten Punkt auf der i ten Kontur und T die durchschnittliche Liniendicke beschreibt:

$$\frac{\partial y_f(i, j)}{\partial x_f(i, j)} = \frac{1}{T} \sum_{m=j-n}^{j-1} D[P(i, m), P(i, m+1)] \quad (2.4)$$

$$\frac{\partial y_a(i, j)}{\partial x_a(i, j)} = \frac{1}{T} \sum_{m=j}^{j+n-1} D[P(i, m), P(i, m+1)] \quad (2.5)$$

$D[P(i, j), P(s, t)]$ definiert den Winkel des Verbindungsvektors von Punkt $P(i, j)$ zu Punkt $P(s, t)$ im Bereich $[0, 2\pi]$. $FDir$ bezeichnet die Richtung von einem Punkt $P(i - k, j)$, welcher k Punkte vor dem aktuellen betrachteten Punkt auf der Kontur i liegt zum Punkt $P(i, j)$ auf derselben Kontur. Entsprechend steht $ADir$ für die Richtung von $P(i, j)$ zu einem auf der Kontur k Schritte weiter liegendem Punkt $P(i + k, j)$. Um eine *Boundary Line* zu extrahieren, werden die Punkte einer extrahierten Kontur

von einem beliebig gewähltem Startpunkt aus solange iteriert, bis drei von den Autoren definierte Bedingungen für den aktuellen Punkt nicht mehr gelten. Ist ein solcher Punkt erreicht, wird mit diesem als Anfangspunkt die nächste *Boundary Line* extrahiert. Die Bedingungen sind folgendermaßen definiert:

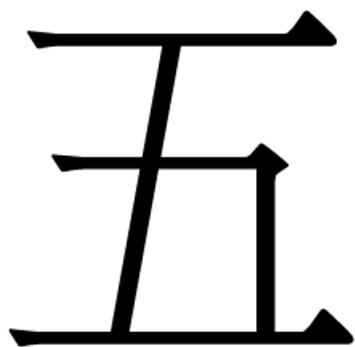
$$k \geq T, \quad (2.6)$$

$$\left| \frac{\partial y_f(i, j+k)}{\partial x_f(i, j+k)} - \frac{\partial y_a(i, j+k)}{\partial x_a(i, j+k)} \right| \leq \theta_1, \quad (2.7)$$

$$\left| \frac{\partial y_a(i, j)}{\partial x_a(i, j)} - \frac{\partial y_a(i, j+k)}{\partial x_a(i, j+k)} \right| \leq \theta_2 \cup \left| \frac{\partial y_a(i, j)}{\partial x_a(i, j)} - \frac{\partial y_f(i, j+k)}{\partial x_f(i, j+k)} \right| \leq \pi + \theta_2 \quad (2.8)$$

Bedingung 2.6 stellt sicher, dass die Schrittweite des Algorithmus mindestens der durch T symbolisierten durchschnittlichen Liniendicke entspricht. Durch Bedingung 2.7 wird garantiert, dass für jeden Punkt einer *Boundary Line* die Differenz zwischen $FDir$ und $ADir$ des Punktes im nächsten Iterationsschritt unterhalb einem festgelegten Schwellenwertes θ_1 liegt. Die dritte Bedingung 2.8 legt die obere Grenze der Differenz der Richtungen des aktuellen Punktes und des Vorgängerpunktes bzw. des aktuellen Punktes und des nachfolgenden Punktes fest. Die auf diese Weise extrahierten *Boundary Lines* einer Kontur werden mit CL_1, \dots, CL_i benannt und gemeinsam als *Boundary Line Chain* dieser Kontur bezeichnet. Die so erhaltenen *Boundary Lines* werden anhand von folgenden Kriterien in eine der Klassen *Singular Line*, *Connection Line* und *Opposite Line* eingeordnet. Zwei *Boundary Lines* sind *Opposite Lines*, wenn sie auf gegenüberliegenden Seiten der gleichen Linie des Linienbildes liegen und innerhalb eines Konfidenzintervalles parallel verlaufen. Eine *Crossing Line* stellt eine Verbindung zwischen zwei *Opposite Lines* auf einer Seite einer Linie dar, welche nicht einem im Bild sichtbaren Kontursegment sondern einer hypothetisierten Verbindung zweier extrahierter *Opposite Lines* entspricht. Alle nicht klassifizierten *Boundary Lines* werden abschließend als *Singular Lines* klassifiziert, welche den Verbindungen zweier *Opposite Lines* an den Strichenden entsprechen.

Die Autoren erwähnen nicht ausdrücklich, ob das Verfahren für handschriftliche oder gedruckte Schriftzeichen entwickelt wurde, sondern schreiben lediglich: „This paper presents a method which segments the complicated Chinese characters into simpler patterns [...]“ Die dargestellten Beispiele legen jedoch nahe, dass das Verfahren im Hinblick auf gedruckte Schrift entwickelt wurde. Angewendet auf handschriftliche Zeichen werden die Schwächen des Verfahrens deutlich. Da gedruckte Schriftzeichen vor allem an Strichkreuzungen andere Krümmungseigenschaften aufweisen als handschriftliche Zeichen an denselben Stellen, können für letztere die aus lokalen Richtungen einzelner Konturpunkte hergeleiteten Krümmungen nur begrenzt zur Aufspaltung einer Kontur in einzelne Segmente verwendet werden. Abbildung 2.8 macht diese Problematik deutlich. Gedruckte Zeichen besitzen an Strichkreuzungen und -enden deutlich identifizierbare Krümmungsmaxima, während Strichseiten üblicherweise größtenteils ungekrümmt sind. Handschrift auf der anderen Seite tendiert dazu, die Konturen sowohl gerader Striche als auch an



(a) gedrucktes Zeichen



(b) handschriftliches Zeichen

Abbildung 2.8: Unterschiedliche Konturkrümmungen bei handschriftlichen und gedruckten Schriftzeichen

Stellen von *interest points* wie Strickkreuzungen und -Enden abzurunden, wodurch sich Krümmungsmaxima an diesen Stellen weniger abheben und schwieriger zu detektieren sind. Die Verwendung von Pinseln als am häufigsten verwendetes Schreibwerkzeug in antiken chinesischen Manuskripten verstärkt diesen Effekt noch zusätzlich, da das Erscheinungsbild der Kontur von mit Pinseln gezeichneten Formen aus physikalischen Gründen von vorneherein weniger ausgeprägte Ecken beziehungsweise plötzliche Richtungsänderung beinhaltet als beispielsweise mit Stiften erzeugte Schriftzeichen.

Ein weiteres strichbasiertes Verfahren stellen [SDRI09] vor, welches zur Analyse von Handschrift als zentrales Element *Strich-Loops* beziehungsweise *-Schleifen* beschreibt. Die Detektion und Analyse solcher Schleifen stellt das Hauptziel der Arbeit dar. Um dies zu erreichen, propagieren die Autoren einen doppelten Ansatz, der sowohl pixel- als auch konturabschnittsbasiert arbeitet. Ein Konturabschnitt, hier *Section* genannt, ist definiert als möglichst lange Verbindungslinie zwischen zwei Konturpunkten mit der Eigenschaft, dass kein zwischen Start- und Endpunkt liegender Konturpunkt weiter als ein Pixel entfernt von der Linie ist. Der Algorithmus wurde für die Handschrift-Analyse entwickelt, bezieht sich dabei allerdings auf westliche Kursiv- beziehungsweise Schreibschrift. In chinesischen Schriftzeichen gibt es jedoch keine nach der Definition aus [SDRI09] natürlichen Schleifen, da sich Einzelstriche hier nicht selber schneiden (siehe Abschnitt 2.1.3). Folglich sind alle in chinesischen Schriftzeichen vorkommenden Schleifen „artifizielle“ Schleifen, welche durch Kreuzungen einzelner Striche entstehen. Aus diesem Grund scheint das Verfahren zur Schleifendetektion für die vorliegende Arbeit ohne grundsätzliche Umstrukturierung nicht geeignet.

3 Motivation und Vorverarbeitung

Das folgende Kapitel beschreibt die Motivation und die in dem Verfahren, welches in dieser Arbeit vorgestellt wird, verwendeten Vorverarbeitungsschritte, die genutzt werden, um die Extraktion einzelner Striche in Bildern chinesischer Schriftzeichen zu unterstützen und vorzubereiten. Zu diesen gehört neben einer Größennormalisierung, welche hauptsächlich zur verbesserten Darstellung der Zeichen in der bereitgestellten graphischen Benutzeroberfläche verwendet wird, die Glättung des gegebenen Bildes, um möglicherweise vorhandenes Rauschen an den Konturen zu entfernen. Im zweiten Teil dieses Kapitels wird zunächst die zur Kantendetektion verwendete Marching Squares-Methode beschrieben. Anschließend wird die Delaunay-Triangulierung erklärt, die als Eingabe für die im nächsten Kapitel vorgestellte Strichextraktion dient. Der dieses Kapitel abschließende Abschnitt 3.4.4 beschreibt, durch welche Eigenschaften die Delaunay-Triangulierung für die Strichextraktion zu einem geeigneten Verfahren wird.

3.1 Motivation

Die im letzten Kapitel beschriebenen Verfahren zur Strichextraktion basieren entweder auf der Skelettierung der Kontur eines Zeichens oder auf der Konturaufspaltung in einzelne Abschnitte, welche zu Strichkonturen zusammengesetzt werden. Die konturbasierten Verfahren teilen die Kontur aufgrund ihrer lokalen Krümmungseigenschaften auf. Da die lokalen Konturkrümmungen bei handschriftlichen Zeichen, wie in Abbildung 2.8b zu sehen, an Strichkreuzungen geringer ausfallen als bei gedruckten Zeichen, sind derartige konturbasierte Strichextraktionsverfahren für handschriftliche Schriftzeichen nur bedingt nutzbar. Das in dieser Arbeit vorgestellte Verfahren umgeht diese Problematik konturbasierter Strichextraktionsverfahren, indem es die Zeichenkontur nicht abhängig von lokalen Krümmungseigenschaften aufsplittet, sondern die in diesem Kapitel vorgestellte Delaunay-Triangulierung nutzt, um Strichkreuzungen zu identifizieren und an diese anschließende Teilstriche zu extrahieren.

3.2 Größen- und Grauwertnormalisierung

Zur Veranschaulichung der Vorverarbeitungsschritte zeigt Abbildung 3.1 ein schlecht aufgelöstes Digitalbild eines handschriftlichen chinesischen Schriftzeichens, wie es für die Strichanalyse zur Verfügung stehen kann. Den ersten Schritt der Vorverarbeitung



Abbildung 3.1: Chinesisches Schriftzeichen „kommen“ vor der Vorverarbeitung.

stellt die Normalisierung der Bildgröße dar. Um der Strichextraktion im nächsten Verarbeitungsschritt möglichst einheitliche Daten zu liefern, werden alle Eingabebilder auf eine einheitliche Größe mit einer Höhe von 300 Pixeln skaliert. Diese Größe bietet einen akzeptablen Kompromiss zwischen hoher Verarbeitungsgeschwindigkeit und ausreichend hoher Auflösung, um auch relativ komplexe Schriftzeichen optisch erkennbar in der Benutzungsoberfläche darzustellen. Die Wahl des Schwellenwertes für die im nächsten Kapitel beschriebene Marching Squares-Methode zur Kantendetektion wird außerdem durch die Normalisierung des Bildes auf den gesamten Grauwertbereich zwischen 0 und 255 unterstützt.

3.3 Glättung

Um mit der im nächsten Abschnitt beschriebenen Kantendetektion für die Weiterverarbeitung verwendbare Konturen eines Schriftzeichens aus einem digitalen Bild zu extrahieren, muss sichergestellt werden, dass die zu bearbeitenden Bilder möglichst rauschfrei sind, da sonst auch die extrahierten Konturen verrauscht sind. Als *Bildrauschen* bezeichnet man eine Art von Bildstörung, bei der vom Bildinhalt unabhängige Wertabweichungen vereinzelter Bildpunkte auftreten. Rauschen ist üblicherweise räumlich unkorreliert, das heißt es tritt zufällig verteilt in einem Bild auf und stellt sich in Form von in ihren Werten von der Umgebung im Wertebereich nach oben oder unten abweichenden Pixeln dar. Abbildung 3.2 zeigt Rauschen in einem niedrig aufgelöstem Bild eines chinesischen Schriftzeichens. In der Bildverarbeitung wird Glättung digitaler Bilder zur Unterdrückung von Rauschen beziehungsweise zum Weichzeichnen verwendet. Hierzu können zum Beispiel sogenannte *Glättungsfilter* benutzt werden. Sie funktionieren nach

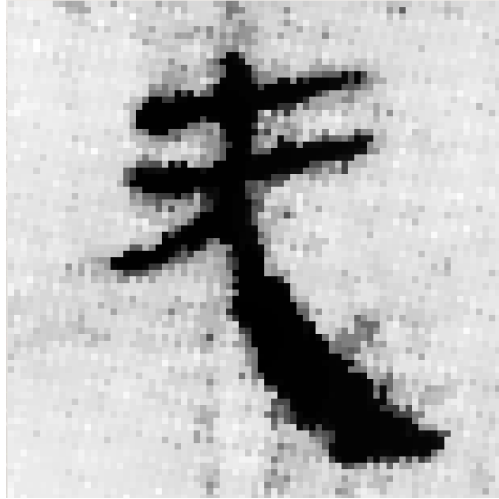


Abbildung 3.2: Rauschen in einem niedrig aufgelöstem Bild

dem Prinzip, ein verrauschtes Bild mit einem Tiefpassfilter zu falten und so einzelne von der Umgebung abweichende Pixel auszufiltern. Realisiert werden solche Filter häufig als Masken, die die Pixelwerte eines geglätteten Ausgabebildes aus den Werten der Nachbarpixel des entsprechenden Bildpunktes im Eingabebild berechnen. Es existieren verschiedene Algorithmen zur Glättung in der Bildverarbeitung. Der *Mittelwertfilter* berechnet den Wert eines Bildpunktes $p_{ij_{out}}$ im geglätteten Ausgabebild als Mittelwert der Werte seiner Nachbarpixel:

$$p_{ij_{out}} = \frac{1}{k^2} \sum_{n=i-k}^{i+k} \sum_{m=j-k}^{j+k} p_{nm_{in}} \quad (3.1)$$

Dies bedeutet, dass die Entfernungen der $k^2 - 1$ Nachbarpixel zum aktuellen Pixel $p_{ij_{in}}$ sich nicht in deren Gewichtung bei der Berechnung des neuen Pixelwertes $p_{ij_{out}}$ im Ausgabebild widerspiegelt. Stattdessen werden alle k^2 in die Berechnung einfließenden Nachbarpixel gleich (z.B. hier mit $\frac{1}{k^2}$) gewichtet. Bei der Glättung mit einem *Medianfilter* wird statt des Mittelwertes benachbarter Pixel deren Median verwendet. Der Median einer Verteilung von Werten ist der Wert, der eine sortierte Menge so in zwei Hälften teilt, dass alle Werte der einen Hälfte niedriger sind als der Median und alle Werte der zweiten Hälfte höher.

Der Median einer Menge von Werten kann gefunden werden, indem diese Menge sortiert und anschließend der mittlere Wert dieser sortierten Menge gewählt wird. Der Vorteil des Medianfilters gegenüber dem Mittelwertfilter ist die höhere Robustheit gegenüber Ausreißern. Ausreißer sind Werte, die sich extrem von den übrigen Pixelwerten eines Digitalbildes unterscheiden und somit einen über alle Werte berechneten Mittelwert verfälschen können. Eine weitere Möglichkeit der Glättung ist die *Gauss-Glättung*, bei der ein Eingabebild mit einem *Gauss-Filter* gefaltet wird, welcher durch eine zweidimensionale Gauss-Funktion mit einer Varianz von σ^2 realisiert wird. Sie entspricht damit



(a) Eingabebild vor Glättung



(b) Eingabebild nach Gauss-Glättung

Abbildung 3.3: Gauss-Glättung mit einer Varianz von 2

einem Tiefpassfilter im Frequenzbereich, der hohe Frequenzen (feine Strukturen im Bild) rausfiltert und niedrige Frequenzen (grobe Strukturen) erhält. Die Impulsantwort eines solchen Filters sieht folgendermaßen aus ([Boy07]):

$$h(x, y) = \frac{1}{\sqrt{\sigma^2 2\pi}} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (3.2)$$

In die Berechnung des Wertes eines Pixels $p_{i_{out}}$ des gefilterten Ausgabe-Bildes fließen dabei die Bildwerte der umgebenden Bildpunkte des Eingabe-Bildes mit Faktoren entsprechend der Varianz der Gauss-Funktion ein. Eine höhere Varianz entspricht daher einer stärkeren Glättung des Bildes und eine niedrigere Varianz einer schwächeren Glättung. Abbildung 3.3 zeigt ein niedrig aufgelöstes Eingabe-Bild eines chinesischen Schriftzeichens vor und nach der Glättung mit einem Gauss-Filter.

3.4 Konturextraktion und Triangulierung

Die folgenden zwei Abschnitte beschreiben das *Marching Squares*-Verfahren zur Extraktion von Konturpunkten des gegebenen Zeichens sowie die Funktionsweise der verwendeten *Delaunay-Triangulierung*, welche als Ausgangspunkt für die im nächsten Kapitel vorgestellte Strichextraktion dient. Die *Marching Squares*-Methode ist eine einfache und effektive Art der Kantendetektion, die mit Hilfe eines Schwellenwertes für jeden Bildausschnitt entscheidet, ob er von einer Kante durchzogen wird oder nicht und auf diese Weise geschlossene, subpixelgenaue Kantenzüge einer Form liefert. Desweiteren wird die auf der *Delaunay-Triangulierung* aufbauende Skelettierung der *Chordal Axis Transform* erläutert, deren Klassifizierung der *Delaunay-Dreiecke* in der vorliegenden Arbeit übernommen wird. Das Ergebnis dieser Verarbeitungsschritte stellt die Basis für die in Kapitel 4 vorgeschlagene Extraktion und Verschmelzung von Teilstrichen dar.



Abbildung 3.4: Das chinesische Zeichen für “kommen“ nach der in den letzten Abschnitten beschriebenen Vorverarbeitungsschritten.

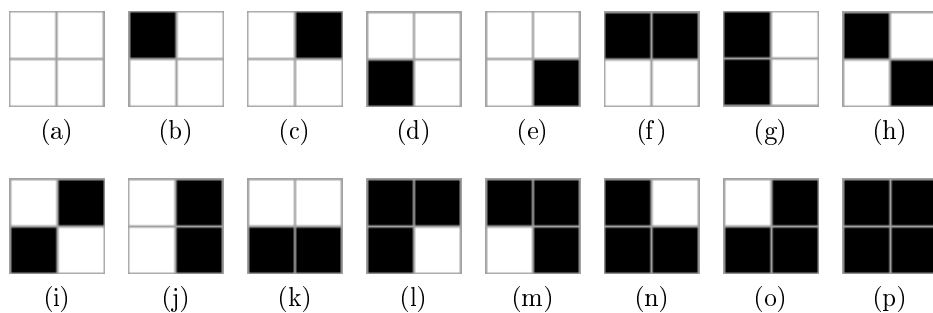


Abbildung 3.5: 16 mögliche Konfigurationen

3.4.1 Kantendetektion mit Marching Squares

Zur Kantendetektion der chinesischen Schriftzeichen wird die *Marching Squares*-Methode verwendet. Dieses Verfahren extrahiert subpixelgenaue Kanten in digitalen Bildern. Es basiert auf dem Ansatz, eine Maske der Größe 2x2 Bildpunkte („marching square“) über das Bild wandern zu lassen und für jede Maskenposition anhand ihres Grau- beziehungsweise Farbwertes zu bestimmen, welche der vier abgedeckten Pixel zum Objekt beziehungsweise Vordergrund und welche zum Hintergrund gehören. Der Übergang zwischen Vorder- und Hintergrundelementen stellt dann eine gefundene Kante dar. Bei zwei möglichen Werten pro Bildpunkt (Vordergrund, Hintergrund) ergeben sich dadurch $2^4 = 16$ mögliche sogenannte *Konfigurationen*. Abbildung 3.5 zeigt die möglichen Konfigurationen. Dabei stellen genau solche Konfiguration, welche sowohl Vorder- als auch Hintergrundpixel enthalten, einen Bildausschnitt dar, durch den eine Kante beziehungsweise

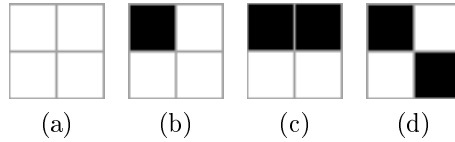


Abbildung 3.6: eindeutige Konfigurationen

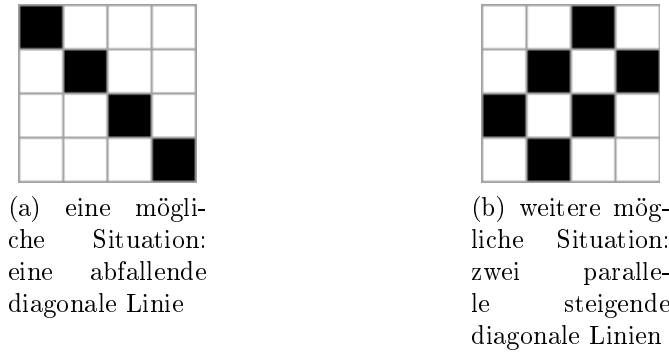


Abbildung 3.7: Zwei mögliche Interpretationen der vierten Konfiguration.

ein Kantenabschnitt verläuft. Die Reduzierung der Anzahl möglicher Konfigurationen auf maximal vier kann erreicht werden, indem alle Konfigurationen aussortiert werden, die durch die Rotation oder Invertierung anderer Konfigurationen erzeugt werden können ([Mei08]). Abbildung 3.6 zeigt die Menge der resultierenden eindeutigen Konfigurationen. Anhand der jeweiligen Konfiguration kann interpretiert werden, auf welche Weise eine Kante durch den von der Maske abgedeckten Bereich verläuft. Bei drei der vier Konfigurationen ist ein solcher Kantenverlauf eindeutig. Konfiguration 3.6a repräsentiert einen Bildbereich, durch den keine Kante verläuft, Konfiguration 3.6b beschreibt eine Situation, bei der eine Kante diagonal durch das Bild läuft, diese dabei jedoch nur an einer Ecke schneidet und daher nur eines der vier Pixel einnimmt. Die dritte Konfiguration 3.6c repräsentiert einen Bildausschnitt, durch den eine Kante horizontal oder vertikal verläuft (durch Rotation der Konfiguration). Bei Konfiguration 3.6d kommt es zu Mehrdeutigkeiten, da diese zwei Interpretationen bezüglich der durch das Bild laufenden Kanten zulässt (siehe Abbildung 3.7). Die erste Interpretation ist die eines Bildausschnittes, durch den eine fallende diagonale Kante verläuft (Abbildung 3.7a), während in der zweiten Interpretation dieser Konfiguration zwei steigende diagonale Kanten das Bild durchziehen (Abbildung 3.7b). Um diese Mehrdeutigkeit zu entfernen, beschreibt [Mei08] eine Möglichkeit, durch Interpolation den Grauwert des Mittelpunktes der vier aktuellen Pixel zu berechnen und diesen anhand des definierten Schwellenwertes als Vorder- oder Hintergrund zu klassifizieren. Wird der Wert dieses Mittelpunktes als Vordergrund beziehungsweise Objekt klassifiziert, ist die Interpretation aus Abbildung 3.7a mit einer diagonalen Kante korrekt, ansonsten die Interpretation aus Abbildung 3.7b.

Für jede in einem Bild detektierte Konfiguration kann durch eine Tabelle genau fest-



Abbildung 3.8: Mit Marching Squares extrahierte Konturen des Zeichens “kommen“.

gelegt werden, in welche Richtung die Pixelmaske im nächsten Schritt verschoben wird, wodurch sich ein deterministischer Ablauf ergibt. Beispielsweise kann bei Auftreten von Konfiguration 3.5f, deren zwei untere Pixel zum Hintergrund und zwei obere Pixel zum Vordergrund gehören, eine Verschiebung der Pixelmaske um einen Pixel nach rechts festgelegt werden. Auf diesen Weg ergibt sich für jede Konfiguration eine begrenzte Zahl möglicher Folgekonfigurationen. Durch dieses Vorgehen umläuft der Algorithmus das im Bild enthaltene Objekt beziehungsweise die als Vordergrund klassifizierten Bildpunkte entlang dessen Kontur in Umlaufrichtung bis er seinen Ausgangspunkt erreicht oder an eine Bildgrenze stößt. Im ersten Fall wird auf diese Weise ein geschlossener Kantenzug erzeugt.

3.4.2 Delaunay-Triangulierung

Die *Delaunay-Triangulierung (DT)*, benannt nach dem russischen Mathematiker Boris Delaunay, ist eine Methode, um aus einer gegebenen Punktmenge P in der Ebene ein Netz aus Dreiecken zu generieren, für welche die sogenannte *Delaunay-Eigenschaft* gilt. Diese Eigenschaft besagt, dass der Umkreis eines Dreiecks keine weiteren Punkte aus P außer den Eckpunkten des Dreiecks enthält. Diese Triangulierung muss jedoch nicht eindeutig sein, da mehr als drei Punkte auf einem Umkreis liegen können.

Definition 1 (Delaunay-Triangulierung (DT)) Die *Delaunay-Triangulierung* DT einer gegebenen Punktmenge P enthält genau dann die Kante $\overrightarrow{p_1 p_2}$ zweier Punkte $\overrightarrow{p_1}, \overrightarrow{p_2} \in P$, wenn es einen Kreis durch die beiden Punkte gibt, der durch keine weiteren Punkte aus P geht und keine Punkte aus P enthält. Desweiteren enthält jeder Umkreis eines durch die Delaunay-Triangulierung erzeugten Dreiecks keine Punkte aus P .

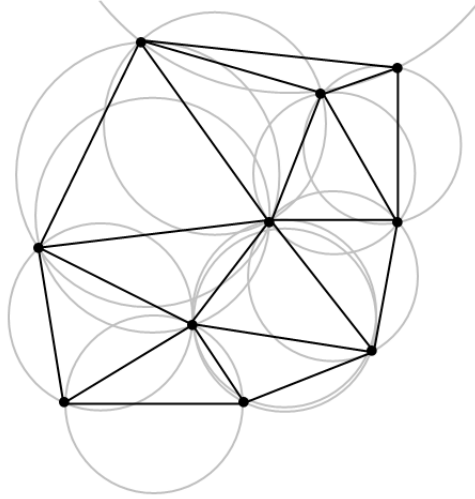


Abbildung 3.9: Delaunay-Triangulierung einer Punktemenge

Aus der Umkreiseigenschaft folgt die Minimierung des maximalen Winkels aller erzeugten Dreiecke ([BE95]). Dies bedeutet, dass die Triangulierung die gegebenen Punkte so zu Dreiecken verbindet, dass der maximale Innenwinkel jedes Dreiecks kleiner ist als bei jeder anderen Triangulierung der Punktemenge, bei der die erzeugten Dreiecke die Delaunay-Eigenschaft nicht besitzen.

Die Eingeschränkte Delaunay-Triangulierung (*Constrained Delaunay Triangulation*, „*CDT*“) ist eine Erweiterung der Delaunay-Triangulierung. Sie erlaubt es, neben Punkten auch Linien als Eingabe für die Berechnung der Triangulierung zu verwenden. Die Einschränkung, die bei dieser Art der Triangulierung erfüllt sein muss, besagt, dass die übergebenen Linien als Dreiecksseiten in der Triangulierung verwendet werden müssen. Dies ist dann sinnvoll, wenn die Kontursegmente einer gegebenen Form bei der Triangulierung als Seiten für die erzeugten Dreiecke erhalten bleiben sollen, das heißt, wenn P nicht nur eine Punktemenge sondern eine Form beschreibt, die trianguliert werden soll. Durch diese Bedingung kommt es jedoch zu einer Einschränkung der Umkreiseigenschaft der Delaunay-Triangulierung. Diese besagt, dass Dreiecksumkreise keine *sichtbaren* Punkte enthalten dürfen. Dabei wird Sichtbarkeit folgendermaßen definiert [Mei08]:

Definition 2 (Sichtbarkeit) Sei G ein Graph aus Punkten (Knoten) V und geraden Verbindungslinien (Kanten) E , dann ist ein Punkt \vec{p}_1 sichtbar für einen Punkt \vec{p}_2 , wenn die Linie $\overrightarrow{p_1 p_2}$ keine Kante von G schneidet. Ein Punkt ist sichtbar für eine Linie $\overrightarrow{p_1 p_2}$, wenn er für einen Punkt auf $\overrightarrow{p_1 p_2}$ sichtbar ist.

Die Eingeschränkte Delaunay-Triangulierung ist für das vorgestellte Verfahren vorzuziehen, um die im vorherigen Schritt gewonnenen Konturen bei der Dreiecksgenerierung beizubehalten.

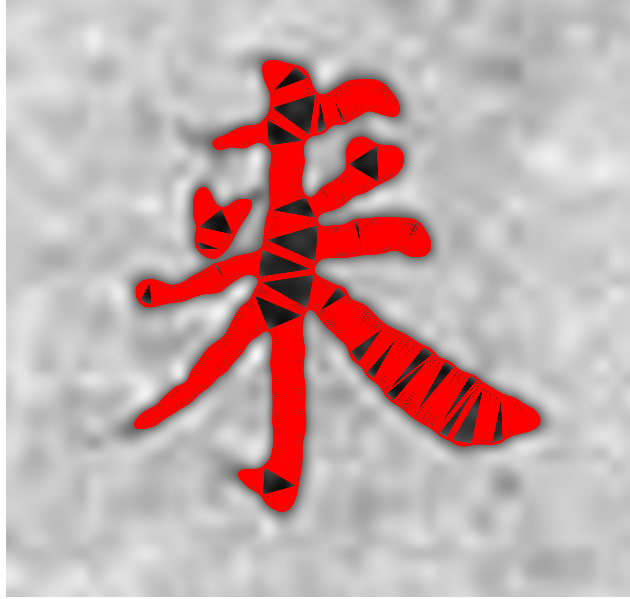


Abbildung 3.10: Eingeschränkte Delaunay-Triangulierung des chinesischen Schriftzeichens für “kommen“.

Definition 3 (Erweiterte Delaunay-Triangulierung (CDT)) Für eine gegebene Punktmenge P enthält die *Erweiterte Delaunay-Triangulierung (Constrained Delaunay Triangulation, CDT)* genau dann die Kante $\overrightarrow{p_1 p_2}$ zweier Punkte $\overrightarrow{p_1}, \overrightarrow{p_2} \in P$, wenn $\overrightarrow{p_1}$ für $\overrightarrow{p_2}$ sichtbar ist und es einen Kreis durch $\overrightarrow{p_1}$ und $\overrightarrow{p_2}$ gibt, der keinen anderen für $\overrightarrow{p_1 p_2}$ sichtbaren Punkt aus P enthält.

Abbildung 3.10 zeigt die Triangulierung der Kontur eines chinesischen Schriftzeichens. Viele nebeneinanderliegende, schmale Dreiecke erscheinen hier aufgrund des geringen Abstandes zwischen den Konturpunkten als zusammenhängende rote Flächen.

[BE95] stellen eine Reihe von möglichen Algorithmen zur Erstellung einer Delaunay-Triangulierung aus einer gegebenen Punktmenge vor. Eine einfache Methode liefert der sogenannte *Flip*-Algorithmus. Zunächst werden dabei aus den eingegebenen Punkten beliebige Dreiecke erstellt. Anschließend überprüft das Verfahren für je zwei Dreiecke, die eine gemeinsame Seite besitzen, ob der kleinste Kreis, der durch die beiden Endpunkte der gemeinsamen Dreiecksseite geht, keine weiteren Punkte der Punktmenge enthält. Ist dies nicht der Fall, wird ein Flip durchgeführt. Das bedeutet, dass die gemeinsame Seite der beiden Dreiecke entfernt und eine neue Seite zwischen den beiden Eckpunkten, die bisher nicht verbunden waren, eingefügt wird. Abbildung 3.11 veranschaulicht dieses Vorgehen. Der Flip-Algorithmus hat den Vorteil, dass durchgeführte Flips die Triangulierung nur lokal beeinflussen, wodurch eine nachträgliche Korrektur anderer, bereits überprüfter Dreiecke nicht notwendig ist.

Die beschriebenen Ansätze umfassen außerdem unter anderem einen Teile-und-hersche-Algorithmus ([SH75], [GS83]), der die eingegebene Punktmenge rekursiv in zwei Mengen aufteilt, welche einzeln trianguliert und anschließend unter Beibehaltung der Delaunay-

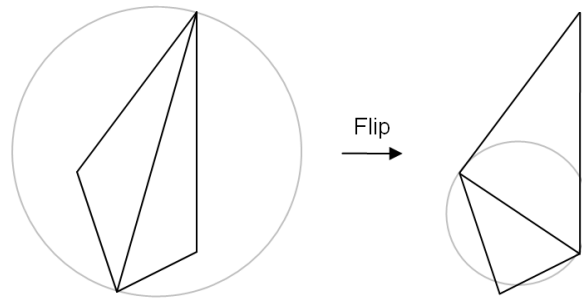


Abbildung 3.11: Die beiden Dreiecke links erfüllen die Umkreiseigenschaft der Delaunay-Triangulierung nicht. Daher wird die gemeinsame Kante entfernt und das so entstehende Viereck durch seine zweite Diagonale in zwei neue Dreiecke aufgeteilt, die die Delaunay-Eigenschaft besitzen.

Eigenschaft zu einer gesamten Triangulation verschmolzen werden. Ein anderes Verfahren ist der von [For86] entwickelte *Sweep*-Algorithmus, welcher Delaunay-Dreiecke iterativ entlang einer sogenannten *Sweep*- oder *Scan*-Geraden erzeugt. In jedem Iterationsschritt werden dabei nur solche Dreiecke hinzugefügt, die über eine gemeinsame Seite mit einem bereits erzeugten Dreieck verfügen.

In dieser Arbeit wird das von [She96] entwickelte Programm „Triangle“ verwendet, welches die Eingeschränkte Delaunay-Triangulierung eines planaren Graphen (*planar straight-line graph*, PLSG) mit Hilfe der Verfahren von [For86] und [GS83] generiert. Die Knoten des PLSG entsprechen den Eingabepunkten und die Kanten den Liniensegmenten, welche sich nicht gegenseitig schneiden dürfen und die in der Eingeschränkten Delaunay-Triangulierung als Dreiecksseiten verwendet werden müssen.

Das Programm ist eine Umsetzung des von Ruppert in [Rup95] beschriebenen Verfahrens zur Delaunay-Triangulierung und besteht aus vier Verarbeitungsschritten. Im ersten Schritt wird die Delaunay-Triangulierung aus den Knoten des eingegebenen Graphen erzeugt, ohne dessen Liniensegmente zu berücksichtigen. Der zweite Schritt fügt alle Liniensegmente des PLSG zu der Triangulierung hinzu, die im ersten Schritt nicht erzeugt wurden, und generiert so die Eingeschränkte Triangulierung (CDT) des Graphen. Dazu werden die Liniensegmente als neue Seiten in die Triangulierung eingefügt. Alle Dreiecksseiten, die die Liniensegmente schneiden, werden entfernt und die Bereiche auf beiden Seiten der Liniensegmente neu trianguliert. Im anschließenden dritten Schritt werden Dreiecke von konkaven Wölbungen entfernt. Dabei handelt es sich um alle Dreiecke, welche Seiten am Rand des Dreiecksgitters besitzen, die nicht als Liniensegmente im eingegebenen PLSG vorkommen. Dieser Schritt stellt eine Erweiterung des Verfahrens von Ruppert und eine bedeutende Verbesserung dar. Er garantiert, dass die Verfahren nur innerhalb von zu triangulierenden Formen Dreiecke generiert, welche zur anschließenden Weiterverarbeitung verwendet werden. Der abschließende vierte Schritt des Algorithmus verfeinert die erstellte CDT bezüglich der Winkeleigenschaften ihrer Dreiecke. Alle Dreiecke, deren maximale und minimale Innenwinkel nicht bestimmten Bedingungen entsprechen, werden in diesem Schritt überarbeitet.

3.4.3 Chordal Axis Transform

Die Eingeschränkte Delaunay-Triangulierung kann außerdem zur Skelettierung einer zweidimensionalen Form verwendet werden, die, wie in Kapitel 2 beschrieben, für viele Strichextraktionsverfahren die Grundlage darstellt. Um aus den in der CDT erzeugten Dreiecken das Skelett einer Form zu gewinnen, verwendet [Pra97] das Verfahren der *Chordal Axis Transform (CAT)*. Dieser Algorithmus extrahiert die einzelnen Punkte des Skelettes, indem er zunächst die in der CDT berechneten Dreiecke in eine der drei folgenden Klassen einsortiert:

1. Terminale Dreiecke (*terminal triangles*) haben maximal eine Seite, welche innerhalb der Form liegt. Die restlichen Seiten sind Teile der Kontur der Form.
2. Ärmeldreiecke (*sleeve triangles*) besitzen genau zwei Seiten, welche nicht Teil der Kontur sind.
3. Verbindungsdreiecke (*junction triangles*) liegen vollständig innerhalb der Form, das heißt sie besitzen keine Seiten, welche Teil der Kontur sind.

Nach der Klassifikation der einzelnen Dreiecke werden die Mittelpunkte benachbarter Ärmeldreiecke miteinander verbunden und bilden auf diese Weise das Skelett der eingegebenen Form.

Das in dieser Arbeit vorgestellte Verfahren ähnelt in seiner Funktionsweise der Chordal Axis Transform und nutzt die abgeleitete Definition der Verbindungsdreiecke, untersucht jedoch statt des Skelettes die ursprüngliche Form des Zeichens.

3.4.4 Delaunay-Triangulierung als Basis der Grundstrichextraktion

Bei der Zerlegung eines Schriftzeichens in seine Grundstriche sind besonders die Stellen von Bedeutung, an denen sich Striche berühren oder kreuzen. Diese Regionen können analog zu den *interest points* skelettbasierter Verfahren, wie dem in Abschnitt 2.3.2 beschriebenen, betrachtet werden. Da Grundstriche chinesischer Schriftzeichen keine wesentlichen Richtungsänderungen enthalten, deuten hohe Krümmungen der Zeichenkontur auf einen Übergang von der Kontur eines Striches zur Kontur eines anderen Striches hin. Daher ist es im Rahmen der Strichextraktion von Interesse, diese Bereiche zu finden und die an sie angrenzenden Strichsegmente zu identifizieren und zu analysieren. Wie bereits bei der Besprechung alternativer Arbeiten zur Strichextraktion in Kapitel 2 erwähnt, reicht dazu die Analyse der Konturkrümmung bei handschriftlichen Schriftzeichen allein oft nicht aus.

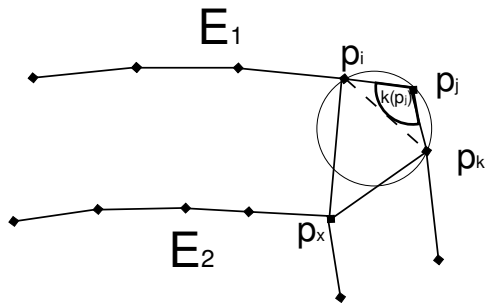
Die Triangulierung der Kontur eines Schriftzeichens ist für die Strichextraktion aus dem Grund vielversprechend, dass sie die Krümmung einer Strichkontur nicht einzeln sondern in Zusammenhang mit der Breite des Striches berücksichtigt. Die durch die Triangulierung erzeugten Dreiecke können wie bei der in Abschnitt 3.4.3 beschriebenen *Chordal Axis Transform* in drei Klassen eingeteilt werden. Besonders interessant für die Strichextraktion sind dabei die Verbindungsdreiecke, da diese genau in den Regionen mit den deutlichsten Richtungsänderungen entstehen und somit als Repräsentation für Stellen fungieren können, an denen Striche sich kreuzen oder berühren.

Zur Entstehung eines Verbindungsdreiecks wird ein strichförmiger Ausschnitt aus zwei sich gegenüberliegenden Zeichenkonturen E_1 und E_2 betrachtet. Verbindungsdreiecke entstehen zwischen diesen Konturen an Stellen hoher Krümmung. Die lokale Krümmung $k(E_1, p_j)$ der Kante E_1 im Punkt p_j entspricht der Umkreisradius des Dreiecks mit den Eckpunkten p_i , p_j und p_k und kann als Kehrwert des Umkreisradius' R berechnet werden:

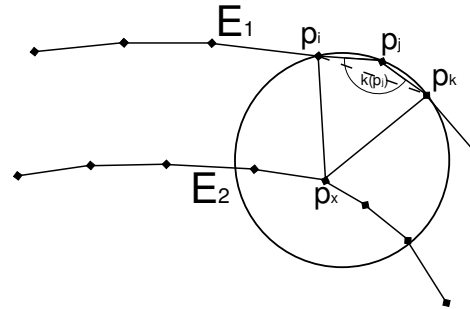
$$k(E_1, p_j) = \frac{1}{R} \quad (3.3)$$

Da die Eingeschränkte Delaunay-Triangulierung verwendet wird, müssen die Kanten-segmente $[p_i, p_j]$ und $[p_j, p_k]$ bei der Triangulierung als Dreiecksseiten verwendet werden.

Der Umkreisradius R des Dreiecks mit den Eckpunkten p_i , p_j und p_k verringert sich mit der Krümmung $k(E, p_j)$ in p_j . Wenn nun die Abstände $d(p_x, p_i)$, $d(p_x, p_j)$ und $d(p_x, p_k)$ aller Punkte p_x der gegenüberliegenden Kontur E_2 zu den Dreieckseckpunkten größer sind als R , bedeutet dies, dass kein weiterer Konturpunkt p_x innerhalb des Dreiecksumkreises liegt und für das Dreieck mit den Eckpunkten p_i , p_j und p_k somit die Delaunay-Eigenschaft gilt. Die Dreiecksseite $[p_i, p_k]$ muss gleichzeitig Seite eines weiteren Dreiecks mit den Eckpunkten p_i , p_k und p_x werden. Abbildung 3.12a verdeutlicht die beschriebene Situation. Zusammenfassend hängt die Entstehung eines durch die *Chordal Axis Transform* als Verbindungsdreieck klassifizierten Dreiecks also von der lokalen Krümmung der Kontur und deren Entfernung zur gegenüberliegenden Kontur ab. Da sich die Konturkrümmung in Regionen oft weniger stark von der Krümmung entlang eines Striches



(a) Der Umkreis des Dreiecks $p_i p_j p_k$ enthält keine weiteren Punkte



(b) Der Umkreis des Dreiecks $p_i p_j p_k$ enthält den Punkt p_x

Abbildung 3.12: Die Entstehung eines Verbindungsdreiecks ist abhängig von der lokalen Krümmung einer Kante sowie der Entfernung zwischen beiden Kanten.

unterscheidet (siehe Abbildung 3.12b), liefert die Hinzunahme der Entfernung zweier gegenüberliegender Konturen und die resultierende Generierung eines Verbindungsdreiecks nach der CAT-Definition einen verlässlichen Indikator, um Strichkreuzungen und -berührungen zu finden.

4 Strichextraktion

In diesem Kapitel wird das Verfahren zur Extraktion einzelner Striche aus Bildern einzelner chinesischer Schriftzeichen beschrieben, für welche eine Vorverarbeitung wie in Kapitel 3 durchgeführt wurde. Die folgenden Abschnitte erklären das Verfahren zur Strichextraktion im Detail und beschreiben die einzelnen Schritte, ausgehend von der Eingabe der im letzten Kapitel erläuterten Delaunay-Triangulierung der Schriftzeichenkontur. Die Verarbeitungsschritte gehen von der Extraktion der als Junctions bezeichneten Verbindungsstellen über die Extraktion von Teilstrichen und die Verschmelzung von Junctions bis zur Verschmelzung von Teilstrichen zu Grundstrichen der chinesischen Schrift. Die Teilstrichverschmelzung ist abhängig von einem zu berechnenden Verschmelzbarkeitsmaß, dessen Berechnung auf den Teilstricheigenschaften beruht. Die Ergebnisse der einzelnen Verarbeitungsschritte werden an dem chinesischen Zeichen für „kommen“ aus Abbildung 3.4 veranschaulicht. Abschließend werden die extrahierten und verschmolzenen Teilstriche in die sechs in Kapitel 2 beschriebenen Grundstrichklassen eingeordnet.

4.1 Junction-Extraktion

Die im vorherigen Abschnitt erklärte Delaunay-Triangulierung der Zeichenkontur dient als Eingabe für den Schritt der *Junction*-Extraktion. Junctions stellen die Verbindungsstellen zwischen Strichen dar. Dazu werden zunächst alle Dreiecke der Triangulierung herausgefiltert, welche in Abschnitt 3.4.3 als Verbindungsdreiecke bei der Berechnung der Chordal Axis Transform klassifiziert wurden. Hierbei handelt es sich genau um die Dreiecke, die über alle drei Seiten mit weiteren Dreiecken der Triangulierung verbunden sind und damit keine Teile der durch das Marching Squares-Verfahren extrahierten Zeichenkontur enthalten. Die Kontur wird ausschließlich durch Ärmeldreiecke und Terminale Dreiecke definiert, die im letzten Abschnitt eingeführt wurden, da nur diese Dreiecke Teile der Kontur als Dreiecksseiten beinhalten.

Als Junction im Sinne der vorgestellten Strichextraktion werden Dreiecke oder Kombinationen von Dreiecken der Triangulierung bezeichnet, die zwei oder mehr *Teilstriche* verbinden. Das Konzept der Teilstriche sowie ihre Extraktion wird im folgenden Abschnitt 4.2 erklärt und bildet zusammen mit den Junctions die atomaren Strukturen, aus denen anschließend komplette Grundstriche der chinesischen Schriftzeichen zusammengesetzt werden. Zu diesem Zweck müssen die Konzepte der Striche und Junctions erläutert werden. Eine Junction J ist ein Dreieck der Delaunay-Triangulierung, welches wie in Abschnitt 3.4.3 beschrieben als Verbindungsdreieck des Chordal Axis Transform-Algorithmus' klassifiziert wurde. Eine Junction ist über jede ihrer drei Seiten mit einem

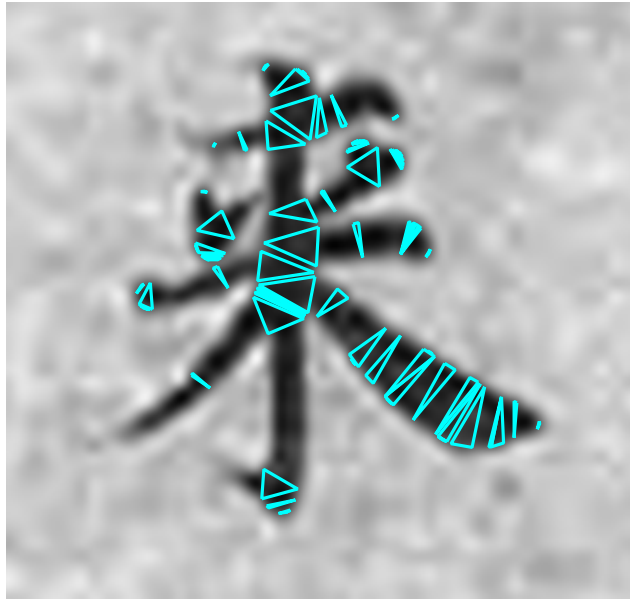


Abbildung 4.1: Verbindungsdreiecke der Delaunay-Triangulierung

Teilstrich nach Definition 4 verbunden und dient damit als Verbindungsstück zwischen Teilstrichen.

Die initial extrahierten Junctions des Strichextraktionsverfahrens entsprechen also den während der Chordal Axis Transform als Verbindungsdreiecke klassifizierten Dreiecken der Delaunay-Triangulierung. Abbildung 4.1 zeigt alle Verbindungsdreiecke der Triangulierung des Schriftzeichens „kommen“. Wie in Abschnitt 3.4.4 beschrieben, ist die Entstehung eines Verbindungsdreiecks zwischen zwei Konturen von ihrer Entfernung zueinander sowie von ihren Krümmungseigenschaften abhängig. Da die Strichbreiten und -krümmungen entlang eines Striches in einem handschriftlichen Zeichen jedoch stärker variieren als bei gedruckten Zeichen, übersteigt die Anzahl der generierten Verbindungsdreiecke die der tatsächlich im Zeichen vorkommenden Strichkreuzungen deutlich. Aus diesem Grund werden die Verbindungsdreiecke nach der folgenden Teilstrichextraktion auf verschiedene geometrische Eigenschaften geprüft, um solche Dreiecke identifizieren zu können, die tatsächlich den vorhandenen Strichkreuzungen entsprechen.

Die an eine Junction angrenzenden Teile der Striche, aus denen ein Zeichen aufgebaut ist, stellen im in dieser Arbeit vorgestellten Verfahren die elementaren Einheiten dar, aus denen Grundstriche der chinesischen Schrift zusammengesetzt werden. Zu diesem Zweck werden die Eigenschaften aller mit einer Junction beziehungsweise Verbindungsstelle verbundenen Teilstriche untersucht, um die Kompatibilität dieser Teilstriche zu berechnen und zu entscheiden, ob sie zu einem Strich verschmolzen werden oder nicht.

4.2 Teilstrichextraktion und -klassifizierung

Das in dieser Arbeit vorgestellte Verfahren zur Strichextraktion basiert auf der Darstellung von chinesischen Schriftzeichen als Aggregation von Grundstrichen, wie sie in Kapitel 2.1 beschrieben wurden. Das bedeutet, dass jeder einzelne der Striche eines Zeichens in eine der folgenden Klassen eingeordnet werden kann (in Klammern die aus [Ber97] entnommenen Grundstrichbezeichnungen).

- Punkt („dot“)
- horizontaler Strich („horizontal“)
- von links oben nach rechts unten fallender diagonaler Strich („press down“)
- von rechts oben nach links unten fallender diagonaler Strich („throw away“)
- vertikaler Strich („vertical“)
- kurzer von links unten nach rechts oben steigender diagonaler Strich („rise“)

Diese sechs Klassen chinesischer Grundstriche haben gemeinsam, dass sie alle relativ gerade verlaufende grafische Grundelemente darstellen. Sie unterscheiden sich vor allem in ihrer prinzipiellen Schreibrichtung. Aus diesem Grund ist es das Ziel des vorgestellten Verfahrens, die im folgenden Schritt extrahierten *Teilstriche* anschließend so zu verschmelzen, dass sie der Struktur der vorgestellten Grundstriche möglichst gut entsprechende gerade Kombinationen von Teilstrichen ergeben.

Definition 4 (Teilstrich) *Ein Teilstrich S besitzt eine linke Kontur C_{left} und eine rechte Kontur C_{right} sowie folgende Eigenschaften:*

- *Start- und Endpunkt $P_s(S)$ und $P_e(S)$*
- *Strichlänge $len(S)$*
- *Strichbreite $width(S, i)$, als Werteliste mit i als Index für die Konturpunkte der Strichkontur*
- *Hauptrichtung $\alpha_{main}(S)$*
- *Lage $pos(S)$*
- *Start- und Endrichtung $\alpha_{start}(S)$ und $\alpha_{end}(S)$*

Zusätzlich kann ein Teilstrich eine Start- und eine End-Junction $J_s(S)$ beziehungsweise $J_e(S)$ besitzen, über welche er mit anderen Teilstrichen verbunden ist.

$C_{left}(S)$ und $C_{right}(S)$ bestehen aus geordneten Punktmengen, deren Extraktion aus den Konturpunkten des Schriftzeichens in Abschnitt 4.2.1 erklärt wird.

Die Berechnung der oben aufgezählten Eigenschaften eines Striches wird in Abschnitt 4.2.2 beschrieben. Teilstriche werden weiterhin in folgende vier Unterklassen aufgeteilt, welche sich nach der Anzahl der mit ihnen verbundenen Junctions sowie ihren Eigenschaften ergeben.

Definition 5 (Einzelstrich) *Ein Einzelstrich ist ein Teilstrich, der weder eine Start- noch eine End-Junction besitzt. Einzelstriche werden in den nachfolgenden Algorithmusschritten nicht mit anderen Strichen verschmolzen und stellen somit bereits fertige Grundstriche dar, wie sie der Algorithmus am Ende liefert.*

$$S = \text{Einzelstrich} \text{ gdw. } (J_s(S) = \emptyset \wedge J_e(S) = \emptyset)$$

Definition 6 (Endstrich) *Endstriche sind nur an einem ihrer beiden Enden mit einer Junction verbunden und haben die Eigenschaft, dass ihre Länge größer ist als ihre maximale Breite.*

$$S = \text{Endstrich} \text{ gdw. } (J_s(S) = \emptyset \oplus J_e(S) = \emptyset) \wedge \text{len}(S) \geq \max(\text{width}(S, i))$$

Definition 7 (Störung) *Eine Störung hat genau wie ein Endstrich nur an einem Ende eine mit ihr verbundene Junction. Im Gegensatz zu einem Endstrich ist eine Störung jedoch breiter als lang.*

$$S = \text{Störung} \text{ gdw. } (J_s(S) = \emptyset \oplus J_e(S) = \emptyset) \wedge \text{len}(S) < \max(\text{width}(S, i))$$

Definition 8 (Verbindungsstrich) *Verbindungsstriche sind an beiden Enden über Junctions mit weiteren Teilstrichen verbunden. Die beiden an einen Verbindungsstrich angeschlossenen Junctions können wie in Abschnitt 4.3.3 miteinander verschmolzen werden, wenn die Länge des Striches einen Längenschwellenwert l_{thres} unterschreitet.*

$$S = \text{Verbindungsstrich} \text{ gdw. } (J_s(S) \neq \emptyset \wedge J_e(S) \neq \emptyset)$$

Abbildung 4.2 zeigt die aus einem chinesischen Schriftzeichen extrahierten Teilstriche. Einzelstriche sind hier blau, Endstriche gelb, Störungen rot und als Verbindungsstriche klassifizierte Teilstriche grün markiert.

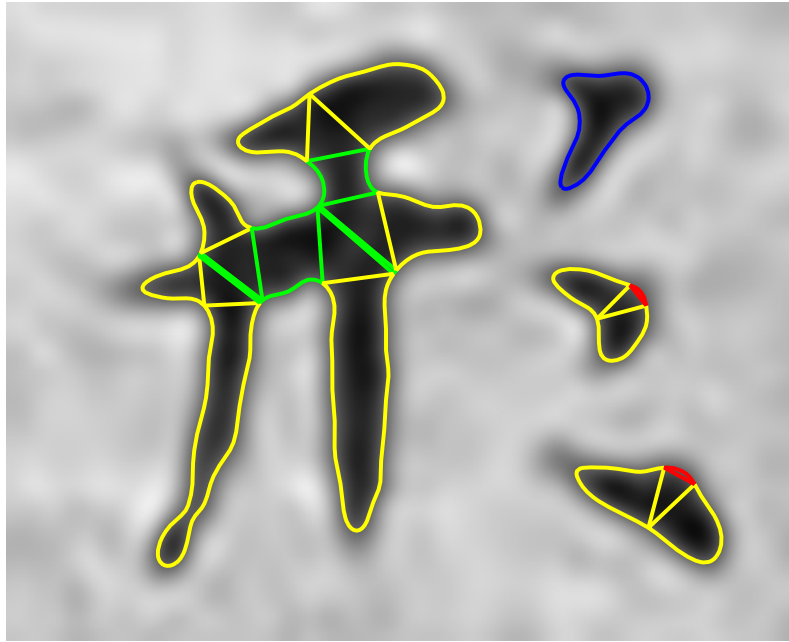


Abbildung 4.2: Teilstrichklassen: Einzelstriche (blau), Endstriche (gelb), Störungen (rot), Verbindungsstriche (grün)

4.2.1 Teilstrichextraktion

Die Extraktion und Klassifizierung von Teilstrichen wird im folgenden Abschnitt erklärt und in zwei Schritten vorgenommen. Der erste Schritt extrahiert *Einzelstriche*, also Striche, die nicht über Junctions mit anderen Strichen verbunden sind. Anschließend werden alle Teilstriche extrahiert, welche von den vorher gewonnenen Junctions ausgehen. Dazu verfolgt der Algorithmus die von den Start- und Endpunkten der Junction-Seiten ausgehenden Zeichenkonturen, welche in dieselbe Richtung gehen beziehungsweise gegenüberliegende Seiten einer Fläche darstellen, und erstellt aus diesen Teilstriche. Die auf diese Weise gewonnen Teilstriche werden in die in Abschnitt 4.2 definierten Klassen *Verbindungsstrich*, *Endstrich* und *Störung* eingeordnet.

Striche in Schriftzeichen können andere Striche im selben Schriftzeichen berühren, schneiden oder von diesen topologisch getrennt sein. Abbildung 4.3a zeigt die Teilstrichextraktion eines handschriftlichen chinesischen Schriftzeichens, welches die Bedeutung „verlieren“ trägt. In diesem Zeichen kommen sowohl sich gegenseitig schneidende, also topologisch miteinander verbundene, als auch einfache, mit anderen Strichen des Zeichens topologisch nicht verbundene Striche vor. Der blau markierte Einzelstrich im rechten unteren Teil von Abbildung 4.3a stellt ein Beispiel für die letztere Kategorie dar. Der gleiche Strich bildet jedoch in der druckschriftlichen Variante des gleichen Zeichens in Abbildung 4.3b mit den anderen Strichen des Zeichens eine gemeinsame topologische Fläche. Dieses Beispiel verdeutlicht, dass die topologische Struktur durch die Variationen, die durch die Eigenschaften einzelner Merkmale entstehen, bei gleichen Zeichen unterschiedlich sein kann. Daher kann ein Strich in manchen Vorkommen eines Schrift-

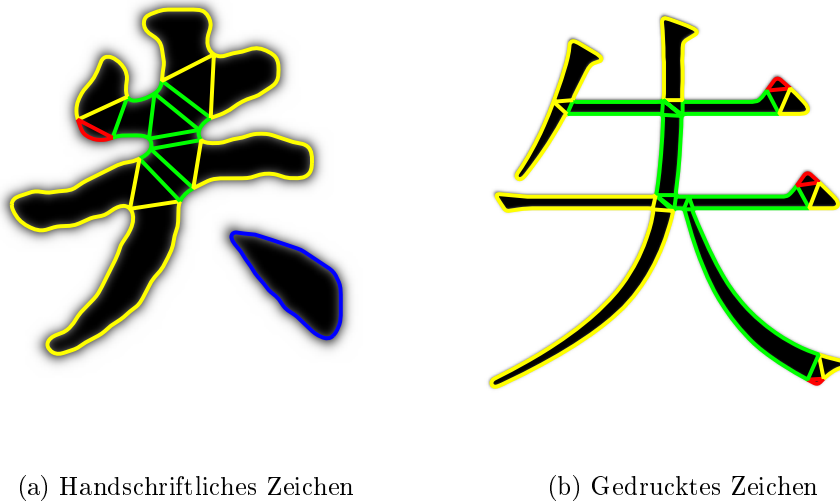


Abbildung 4.3: Teilstrichextraktion: Unterschiedliche topologische Strukturen des Zeichen für „verlieren“

zeichen mit anderen Strichen über eine Junction verbunden sein, während er in anderen Vorkommen desselben Zeichens als Einzelstrich extrahiert wird.

Auf der Strichextraktion basierende Verfahren zur Zeichenerkennung stellen Schriftzeichen häufig als Graph dar, dessen Knoten den Strichen eines Zeichens und Kanten den geometrischen und topologischen Relationen zwischen den Strichen entsprechen. Diese Art der Repräsentation eines Zeichens kann aufgrund der einheitlichen Darstellung gedruckter Zeichen hilfreich sein. Gleichzeitig kann sie jedoch auch zu Problemen bei handschriftlichen Zeichen führen, wenn die Topologie identischer Zeichen wie gezeigt variiert.

Verglichen mit dem handschriftlichen Zeichen in Abbildung 4.3a hat dessen druckschriftliches Pendant also eine andere topologische Struktur. Aus diesem Grund können strichbasierte Zeichenerkennungsverfahren, welche ein Zeichen über die topologischen Relationen seiner erzeugenden Einzelstriche beziehungsweise -teile repräsentieren, anfälliger für solche Bildstörungen sein, welche die topologische Struktur beeinflussen. Da dies bei Handschriften der Fall sein kann, verzichtet das hier vorgestellte Verfahren auf die Einbeziehung von Informationen über topologische Relationen zwischen Strichen und beschränkt sich bei der formalen Darstellung eines Zeichens durch seine Striche auf deren geometrische Eigenschaften und Relationen. Allerdings geht das Verfahren von der Annahme aus, dass die zu erkennenden Grundstriche für sich topologisch korrekt sind.

Die Unterscheidung zwischen den in Abschnitt 4.2 erläuterten Teilstrichkategorien ist für das Verfahren dennoch entscheidend, da die zentrale Struktur der Junction als Verbindung zwischen Teilstrichen nicht in Einzelstrichen vorkommt. Aus diesem Grund werden Instanzen der letzteren Art besonders behandelt. Topologisch stellt jeder Einzelstrich,

genau wie jede aus mehreren sich berührenden oder schneidenden Strichen bestehende Teilstruktur eines Schriftzeichens, eine zusammenhängende, mit dem restlichen Zeichen nicht verbundene Region aus Vordergrundpixeln des Bildes dar. Nach der Kantendetektion durch das Marching Squares-Verfahren und der anschließenden Eingeschränkten Delaunay-Triangulierung der Konturen aller extrahierter Regionen präsentieren sich Einzelstriche als Regionen mit Konturen, deren Triangulierung keine Verbindungsdreiecke nach der Definition der Chordal Axis Transform enthalten. Da die Extraktion von Teilstrichen jedoch von den aus Verbindungsdreiecken gewonnenen Junctions als Ausgangspunkt startet, kann dieses Verfahren bei Einzelstrichen nicht angewendet werden. Stattdessen werden Regionen, deren Konturtriangulierungen keine Verbindungsdreiecke enthalten, als komplette Teilstriche ohne Start- und End-Junction interpretiert, und ihre Kontur wird an den Bereichen der höchsten Krümmung in linke Kontur C_{left} und rechte Kontur C_{right} aufgeteilt. Algorithmus 1 veranschaulicht das beschriebene Vorgehen.

Algorithm 1 Einzelstrichextraktion

```

for  $R \in \text{Vordergrundregionen}$  do
  if  $\neg(\text{R enthält Verbindungsdreieck})$  then
    Krümmungen = berechne Krümmungen von  $R.Kontur$ 
     $n = \text{len}(R.Kontur)$ 
     $\text{index}_{p_1} = \text{index}(\text{max}(\text{Krümmungen}))$ 
     $\text{index}_{p_2} = \text{index}(\text{max}(\text{Krümmungen})) + \frac{n}{2}$ 
     $C_{left} = R.Kontur_{\text{index}_{p_1}}, \dots, R.Kontur_{\text{index}_{p_2}}$ 
     $C_{right} = R.Kontur_{\text{index}_{p_2}}, \dots, R.Kontur_{\text{index}_{p_1}}$ 
    invertiere Reihenfolge von Punkten in  $C_{right}$ 
    füge neuen Teilstrich  $S$  aus  $C_{left}$  und  $C_{right}$  zu Teilstrichliste hinzu
  end if
end for

```

Im anschließenden Verarbeitungsschritt werden alle Teilstriche extrahiert, die topologisch miteinander verbunden sind, ausgehend von den im vorigen Schritt extrahierten Junctions. Für diesen Zweck wird über alle Seiten aller initial extrahierten Junctions J_i iteriert und für jede Seite ein angrenzender Teilstrich extrahiert. Die Extraktion des angrenzenden Teilstriches wird durch Traversierung der beiden von den Endpunkten der Junction-Seite ausgehenden Konturen realisiert, welche in die zwei sich gegenüberliegenden Seiten des zu extrahierenden Striches konvertiert werden. Dazu wird ein zweistufiges Verfahren verwendet, welches die beiden Konturen nacheinander bearbeitet. In der ersten Stufe traversiert der Algorithmus die Kontur, welche dem Startpunkt der aktuell bearbeiteten Dreiecksseite entspringt. Über die Punkte dieser Kontur wird solange iteriert, bis ein Eckpunkt eines Junction-Dreiecks erreicht wird. Zwei mögliche Fälle müssen hier unterschieden werden:

1. Bei dem gefundenen Endpunkt handelt es sich um einen Eckpunkt des Ausgangsdreiecks, weshalb die zweite Stufe übersprungen und der extrahierte Strich als

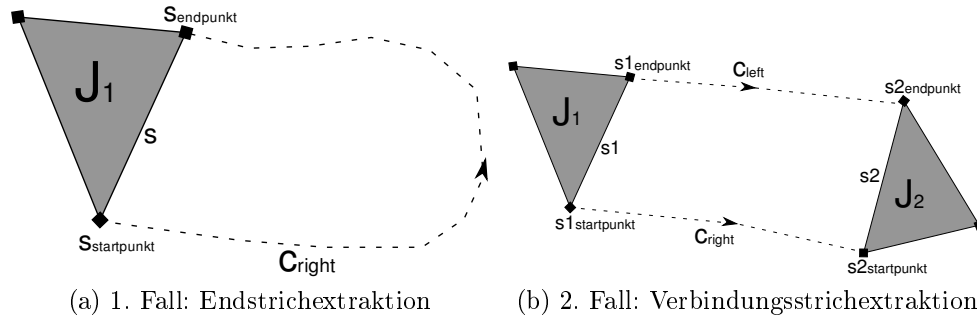


Abbildung 4.4: Zwei mögliche Fälle bei der Extraktion der Strichkontur

Endstrich klassifiziert wird, welcher nur an einem Ende über eine Junction mit weiteren Teilstrichen verbunden ist. In diesem Fall wird die extrahierte Strichkontur an einem angemessenem Punkt in linke und rechte Strichkontur aufgetrennt. Abbildung 4.4a visualisiert diesen Fall.

2. Ist der erreichte Konturpunkt jedoch als Eckpunkt ein Teil einer anderen Junction J_j , $j \neq i$, handelt es sich bei dem aktuellen Strich um einen Verbindungsstrich zwischen den beiden Junctions J_i und J_j . Die in Stufe 1 extrahierte Kontur entspricht dann nur der einen Seite des aktuellen Striches. Stufe 2 gewinnt auf äquivalente Weise die gegenüberliegende Strichkontur: über die Punkte der Kontur, zu der der Endpunkt der aktuellen Junction-Seite gehört, wird ausgehend von diesem iteriert, bis ein weiterer Eckpunkt einer Junction erreicht wird. Bei diesem Eckpunkt handelt es sich um den zweiten Endpunkt der Junction-Seite, deren erster Endpunkt in Stufe 1 erreicht wurde. Dieser Fall wird in Abbildung 4.4b dargestellt.

Nach der Extraktion der Strichkonturen wird ein Resampling der sie repräsentierenden Polygone vorgenommen werden, um die gleiche Anzahl an Punkten in beiden Konturen zu gewährleisten.

Der Algorithmus aus Pseudo-Code-Ausschnitt 2 verdeutlicht das beschriebene Vorgehen.

Abbildung 4.5a zeigt eine türkis markierte Junction und die rot markierte Kontur des Zeichens „kommen“, über die der Algorithmus ausgehend von den Dreiecksseiten der Junctions jeweils solange iteriert, bis er einen weiteren Junctionendpunkt erreicht. Ausgehend von der rechten Dreiecksseite wird dadurch der in Abbildung 4.5b gelb markierte Endstrich extrahiert.

4.2.2 Berechnung der Stricheigenschaften

Nach ihrer Extraktion werden die folgenden Eigenschaften der Striche berechnet: Start- und Endpunkt, Länge, Breite, durchschnittliche Breite, Lage, Hauptrichtung sowie Richtungen in den Endregionen der Striche. Sei S im Folgenden der Teilstrich, dessen Eigenschaften berechnet werden sollen. Seien desweiteren $C_{left}(S)$ und $C_{right}(S)$ die linken

Algorithm 2 Teilstrichextraktion

```
for  $j \in \text{Junctions}$  do
  for  $s \in j.\text{seiten}$  do
     $p_1 = s.\text{startpunkt}$ 
     $\text{Strichkontur} = []$ 
    while  $p_1 \notin \text{Junctionendpunkte}$  do
       $p_1 = \text{nächster Punkt der Zeichenkontur}$ 
      hänge  $p_1$  an  $\text{Strichkontur}$  an
    end while
    if  $p_1 == s.\text{endpunkt}$  then
       $n = \text{len}(\text{Strichkontur})$ 
       $C_{\text{left}} = \text{Strichkontur}_1, \dots, \text{Strichkontur}_{\frac{n}{2}}$ 
       $C_{\text{right}} = \text{Strichkontur}_n, \dots, \text{Strichkontur}_{\frac{n}{2}+1}$ 
    else
       $p_2 = s.\text{endpunkt}$ 
       $\text{Strichkontur}_2 = []$ 
      while  $p_2 \notin \text{Junctionendpunkte}$  do
         $p_2 = \text{nächster Punkt der Zeichenkontur}$ 
        hänge  $p_2$  an  $\text{Strichkontur}_2$  an
      end while
       $C_{\text{left}} = \text{Strichkontur}_1$ 
       $C_{\text{right}} = \text{Strichkontur}_2$ 
    end if
    füge neuen Teilstrich  $S$  aus  $C_{\text{left}}$  und  $C_{\text{right}}$  zu Teilstrichliste hinzu
  end for
end for
```



(a) Zeichenkontur (rot) und aktuell bearbeitete Junction (türkis)



(b) Aus der Junction (türkis) extrahierter Endstrich (gelb)

Abbildung 4.5: Bei der Teilstrichextraktion wird die Zeichenkontur (rot) ausgehend von den Dreiecksseiten der Junctions (türkis) traversiert. Der Endstrich (gelb) rechts wird auf diese Weise extrahiert.

und rechten Konturseiten des Striches, sowie n und m die Punktemengen in C_{left} beziehungsweise C_{right} , und bezeichne C_i den i ten Punkt auf der Kontur C . Dann werden die erwähnten Stricheigenschaften folgendermaßen definiert.

Definition 9 (Startpunkt, Endpunkt) Die Start- und Endpunkte $P_s(S)$ und $P_e(S)$ eines Striches werden berechnet als Vektoren der Mittelpunkte der Verbindungsstrecken zwischen den ersten beziehungsweise letzten Punkten der linken bzw. rechten Strichkontur.

$$P_s(S) = \frac{1}{2}(C_{left}(S)_0 + C_{right}(S)_0)$$

$$P_e(S) = \frac{1}{2}(C_{left}(S)_{n-1} + C_{right}(S)_{m-1})$$

Definition 10 (Strichlänge) Die Länge $len(S)$ eines Strichs wird berechnet als der Durchschnitt der Bogenlängen seiner beiden Konturen:

$$len(S) = \frac{1}{2}(\text{arcLength}(C_{left}(S)) + \text{arcLength}(C_{right}(S)))$$

Im Fall eines digitalen Bildes wird die Bogenlänge einer durch ein Polygon repräsentierten Kurve über die aufsummierten Abstände aller aufeinanderfolgenden Punkte des

Polygons berechnet. Sei

$$\begin{aligned} d(P, Q) &= \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \\ &= \sqrt{\sum_{i=0}^n (p_i - q_i)^2, p_i \in P, q_i \in Q} \end{aligned}$$

die Euklid'sche Distanz zwischen den Punkten P und Q . Dann kann die Bogenlänge $arcLength(C)$ des eine Konturseite repräsentierenden Polygons C mit n Punkten folgendermaßen berechnet werden:

Definition 11 (Bogenlänge)

$$arcLength(C) = \sum_{i=0}^{n-2} d(C_i, C_{i+1})$$

Definition 12 (Strichbreite) *Seien m und n die Punktemengen in den in den Konturen $C_{left}(S)$ beziehungsweise $C_{right}(S)$ und sei $m = n$. Dann berechnet sich die Strichbreite $width(S, i)$ des Striches S an dem Punkt mit dem Index i als Abstand der i ten Punkte $C_{left}(S)_i$ und $C_{right}(S)_i$ von $C_{left}(S)$ beziehungsweise $C_{right}(S)$:*

$$width(S, i) = d(C_{left}(S)_i, C_{right}(S)_i)$$

Aus der Strichbreite kann zusätzlich die durchschnittliche Strichbreite ermittelt werden:

Definition 13 (Durchschnittliche Strichbreite)

$$averageWidth(S) = \frac{1}{n} \sum_{i=0}^{n-1} (width(S, i))$$

Als Richtungen werden in dieser Arbeit die Neigungswinkel zwischen Ortsvektoren und einer festen Orientierungsachse verwendet, für welche die Abszisse eines nicht invertierten Koordinatensystems gewählt wird. Die Haupt- und Endrichtungen von Strichen stellen die Haupteigenschaften zur Berechnung der Kompatibilität zweier Teilstriche bei deren Verschmelzung sowie für die Einordnung in eine der Grundstrichklassen der chinesischen Schrift dar.

Definition 14 (Hauptrichtung) *Als Hauptrichtung von S wird der Winkel α_{main} zwischen Abszisse und dem Verschiebungsvektor $\overrightarrow{v_{start2end}} = \overrightarrow{P_s(S)P_e(S)}$ von Start- zu Endpunkt des Striches verwendet. Bezeichnet man einen die Richtung der Abszisse beschreibenden Einheitsvektor $\overrightarrow{(0,0)(1,0)}$ mit $\overrightarrow{v_{absc}}$, berechnet sich die Hauptrichtung α_{main} folgendermaßen:*

$$\alpha_{main} = \arccos\left(\frac{\langle \overrightarrow{v_{start2end}}, \overrightarrow{v_{absc}} \rangle}{\|\overrightarrow{v_{start2end}}\| * \|\overrightarrow{v_{absc}}\|}\right)$$

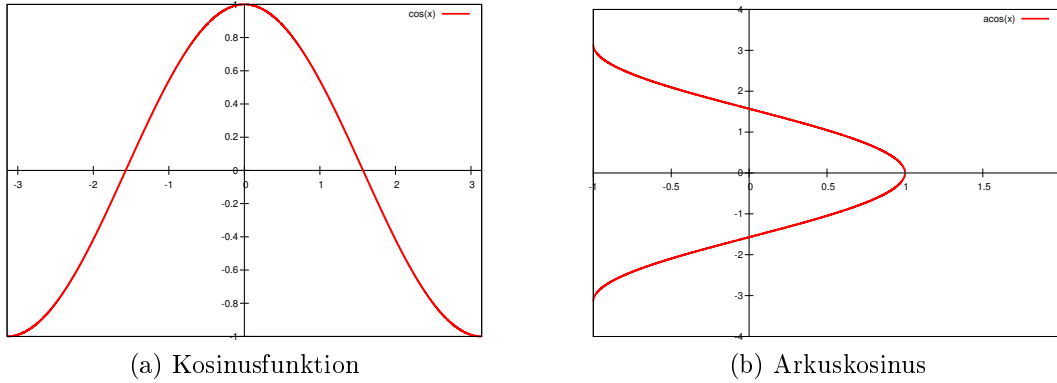


Abbildung 4.6: Links: Kosinusfunktion im Definitionsbereich $[-\pi, \pi]$, rechts: Arkuskosinus als Umkehrabbildung der Kosinusfunktion im Definitionsbereich $[-\pi, \pi]$

Für \vec{v}_{absc} als Einheitsvektor gilt $\|\vec{v}_{absc}\| = 1$. Durch diese Eigenschaft und die Wahl der Abszissenrichtung als Grundrichtung, zu der die Strichrichtung relativ berechnet werden soll, lässt sich die Gleichung zur Berechnung von α_{main} folgendermaßen vereinfachen:

$$\alpha_{main} = \arccos\left(\frac{\vec{v}_{start2end}}{\|\vec{v}_{start2end}\|}\right)$$

Die Kosinusfunktion ist 2π -periodisch, woraus folgt, dass sie nicht injektiv ist, wie in Abbildung 4.6a zu sehen ist. Das bedeutet, dass die Umkehrabbildung des Kosinus, der Arkuskosinus, keine eindeutig definierte Funktion ist, wenn der Definitionsbereich der Kosinusfunktion nicht eingeschränkt wird (siehe Abbildung 4.6).

Um die Mehrdeutigkeit des Arkuskosinus aufzulösen, muss seine Urbildfunktion Kosinus auf einen Definitionsbereich eingeschränkt werden, in dem sie injektiv ist. Dazu wird der Bereich $[0, \pi]$ gewählt. Die resultierende injektive Kosinusfunktion ist in Abbildung 4.7a zu sehen. Bildet man den Arkuskosinus als Umkehrabbildung dieser Kosinusfunktion, handelt es sich um die Funktion in Abbildung 4.7b.

Die *Start-* und *Endrichtungen* beschreiben lokale Richtungseigenschaften des Striches in der Nähe seiner Endpunkte und berechnen sich auf die gleiche Weise wie dessen Hauptrichtung. Statt des Winkels α_{main} zwischen der Abszisse und dem Vektor $\vec{v}_{start2end}$ zwischen Start- und Endpunkt werden jedoch die Winkel α_{start} zwischen der Abszisse und einem Vektor \vec{v}_{start} vom Startpunkt zu einem Punkt $P_x(S)$ in der Nähe des Startpunkts sowie α_{end} zwischen Abszisse und einem Vektor \vec{v}_{end} vom Endpunkt zu einem Punkt $P_y(S)$ in der Nähe des Endpunktes berechnet. Dabei bezeichnen \vec{v}_{start} und \vec{v}_{end} zwei Vektoren $\vec{P}_s(S)\vec{P}_x(S)$ und $\vec{P}_y(S)\vec{P}_e(S)$, welche die Richtungen des Striches in dessen Start- und Endregionen beschreiben.

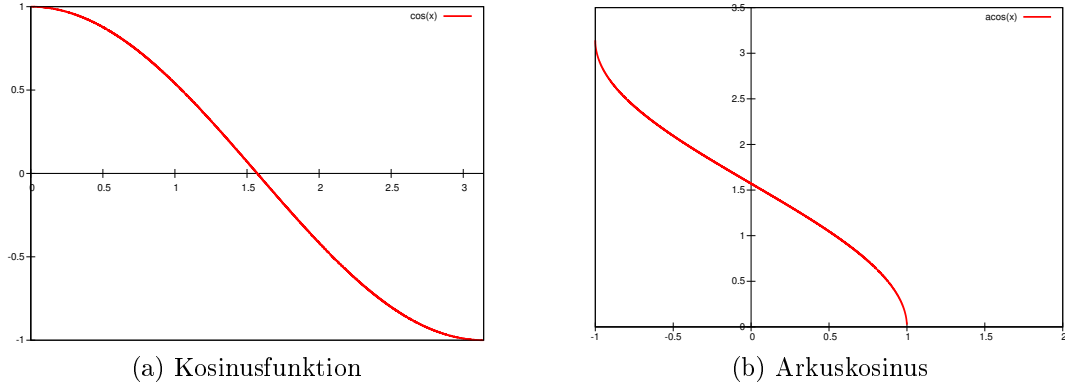


Abbildung 4.7: Links: Kosinusfunktion im eingeschränkten Definitionsbereich $[0, \pi]$, rechts: Arkuskosinus als Umkehrabbildung der Kosinusfunktion im eingeschränkten Definitionsbereich $[0, \pi]$

Definition 15 (Start- und Endrichtungen)

$$\alpha_{start} = \arccos\left(\frac{\langle \vec{v}_{start}, \vec{v}_{absc} \rangle}{\|\vec{v}_{start}\| * \|\vec{v}_{absc}\|}\right) \tag{4.1}$$

$$\alpha_{end} = \arccos\left(\frac{\langle \vec{v}_{end}, \vec{v}_{absc} \rangle}{\|\vec{v}_{end}\| * \|\vec{v}_{absc}\|}\right) \tag{4.2}$$

Die Wahl der Punkte $P_x(S)$ und $P_y(S)$ als Endpunkte der Vektoren \vec{v}_{start} und \vec{v}_{end} ist dabei abhängig von verschiedenen Faktoren und kann das Ergebnis stark beeinflussen. Der Abstand der Punkte $P_x(S)$ und $P_y(S)$ von Start- beziehungsweise Endpunkt wird in Abhängigkeit von der vorher ermittelten durchschnittlichen Strichdicke gesetzt. Dazu iteriert der Algorithmus ausgehend von den jeweils ersten Punkten über die linken und rechten Konturen des Striches bis zu den Punkten $P_{l_{asw}}$ beziehungsweise $P_{r_{asw}}$, welche abhängig von der durchschnittlichen Strichbreite $asw = averageWidth(S)$ sind. Aus den Ortsvektoren $\vec{P}_{l_{asw}}$ und $\vec{P}_{r_{asw}}$ dieser zwei Punkte berechnet sich anschließend der Ortsvektor $\vec{P}_x(S)$ des gesuchten Punktes auf der Strichachse als deren Mittelpunkt: $\vec{P}_x(S) = (\vec{P}_{l_{asw}} + \vec{P}_{r_{asw}})/2$.

Die Berechnung von $\vec{P}_y(S)$ läuft dazu analog, ausgehend von den letzten Punkten der linken und rechten Konturen: $\vec{P}_y(S) = (\vec{P}_{m-l_{asw}} + \vec{P}_{n-r_{asw}})/2$.

Die Wahl der Indices x und y zur Berechnung der lokalen Start- und Endrichtungen stellt dabei den für die Qualität der Berechnung entscheidenden Parameter dar. Ein zu groß gewählter Abstand führt eventuell zu einem Punkt, der zu weit vom Ausgangspunkt der Kontur entfernt ist und durch eine eventuell hohe Gesamtkrümmung des Striches die lokale Richtungseigenschaft des Striches nicht korrekt wiedergibt. Das gleiche Problem kann auftreten, wenn x beziehungsweise y zu klein gewählt wird. In diesem Fall können Konturrauschen oder große Krümmungen, welche an Strickkreuzungen über größere

Konturstrecken auftreten, das Ergebnis verfälschen. Als ein guter Richtwert hat sich die durchschnittliche Strichdicke asw herausgestellt. Durch diesen für jeden Strich individuell berechneten Wert kann sichergestellt werden, dass die Richtung über eine Strecke berechnet wird, welche groß genug ist, um Störungen zu vermeiden und gleichzeitig die Gefahr vermindert, die Richtung über einen gebogenen Teil des Striches zu berechnen.

Die Lage $pos(S)$ eines Striches S wird als dessen Schwerpunkt (*center of gravity, cog*) berechnet. Die Koordinaten eines Schwerpunktes einer Punktemenge ergeben sich als Durchschnittswerte der X- beziehungsweise Y-Koordinaten aller Punkte in der Menge. Im Falle der Lageberechnung eines Striches wird aus Performanzgründen der Schwerpunkt der Strichkontur berechnet. Durch das nach der Teilstrichextraktion durchgeführte Resampling der Strichkonturen und die daraus resultierende gleiche Anzahl von Konturpunkten in $C_{left}(S)$ und $C_{right}(S)$ wird dies ermöglicht.

Definition 16 (Lage) Sei $M(S)$ die vereinigte Menge aller Konturpunkte der linken und rechten Konturen von Strich S .

$$\begin{aligned} pos(S) &= (x_c, y_c) \\ x_c &= \frac{1}{|M(S)|} \sum_i p_i(x) \in M(S) \\ y_c &= \frac{1}{|M(S)|} \sum_i p_i(y) \in M(S) \end{aligned}$$

Diese Stricheigenschaften werden sowohl für Teilstriche als auch für verschmolzene Striche berechnet.

Nach der Berechnung der genannten Stricheigenschaften werden alle Striche in einer in Abschnitt 4.2 definierten drei Klassen Verbindungsstrich, Endstrich und Störung eingeordnet. Die Klassifizierung basiert, wie im ersten Abschnitt dieses Kapitels beschrieben, auf der Anzahl der mit dem Strich verbundenen Junctions sowie auf den zuvor extrahierten Eigenschaften Strichlänge und Strichbreite.

4.3 Junctionklassifizierung, -Reduktion und -Verschmelzung

In den letzten beiden Schritten wurde beschrieben, wie initiale Junctions und Teilstriche extrahiert werden. Die folgenden Abschnitte erklären, wie der Algorithmus aus ihnen komplexere Teilstriche konstruiert und abschließend als Grundstriche der chinesischen Schrift klassifiziert.

Im nächsten Abschnitt wird zunächst das Vorgehen zum Klassifizieren und anschließendem Verschmelzen von Junctions erläutert, bevor das darauffolgende Kapitel die Verschmelzung der Teilstriche beschreibt. Bei den initial extrahierten Junctions handelt es sich, wie in Abschnitt 4.1 beschrieben, um die Verbindungsdreiecke der triangulierten Form des Schriftzeichens. Abbildung 4.1 zeigt die aus einem Bild des chinesischen Schriftzeichens für „kommen“ auf diese Weise extrahierten Junctions als türkise Dreiecke.

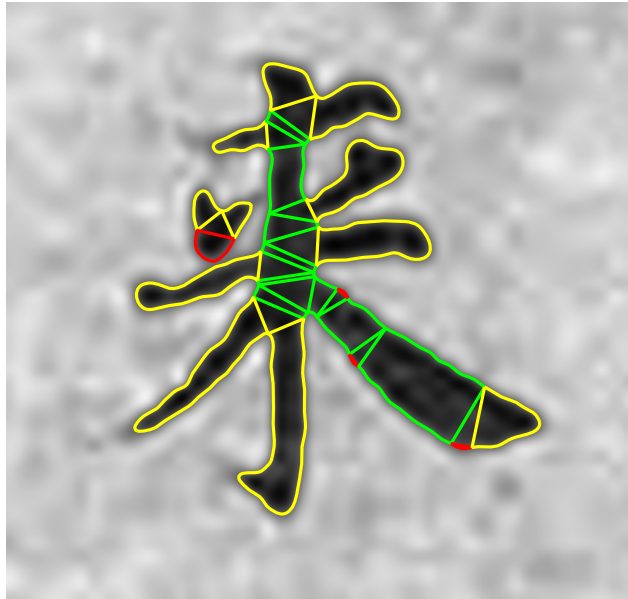


Abbildung 4.8: Ausgehend von den zuvor extrahierten initialen Junctions werden die initialen Teilstriche des Zeichens gewonnen. Verbindungsstriche zwischen zwei Junctions werden grün, Endstriche gelb und Störungen rot dargestellt.

4.3.1 Junction-Klassifizierung

Die Einteilung der in Abschnitt 4.1 extrahierten Junctions aus den Dreiecken der Delaunay-Triangulierung in verschiedene Klassen stellt den folgenden Verarbeitungsschritt dar. Diese Klassifizierung wird mit Hilfe der im letzten Schritt gewonnen Teilstriche vorgenommen. Die Klasse einer Junction hängt dabei von den Klassen der mit ihr verbundenen Teilstriche ab. Entscheidend ist die Anzahl der echten Teilstriche, das heißt solcher Teilstriche, die keine Störung sind, sondern entweder zu den Verbindungsstrichen oder zu den Endstrichen gehören.

Sei im Folgenden $|dist(J)|$ die Anzahl der mit der Junction J verbundenen Störungsstriche.

Definition 17 (Einzelstrich-Junction) *Eine Einzelstrich-Junction liegt vor, wenn bei der Triangulierung eines nicht mit anderen Teilstrichen verbundenen kurzen Striches ein Verbindungs-dreieck entsteht, das drei Störungsstriche miteinander verbindet.*

$$J = \text{Einzelstrich-Junction gdw. } |dist(J)| = 3$$

Definition 18 (Strichende-Junction) *Strichende-Junctions markieren das Ende eines Striches und verbinden den eigentlichen Strich, welcher vom Typ Endstrich oder Verbindungsstrich ist, mit zwei Störungsstrichen.*

$$J = \text{Strichende-Junction gdw. } |dist(J)| = 2$$

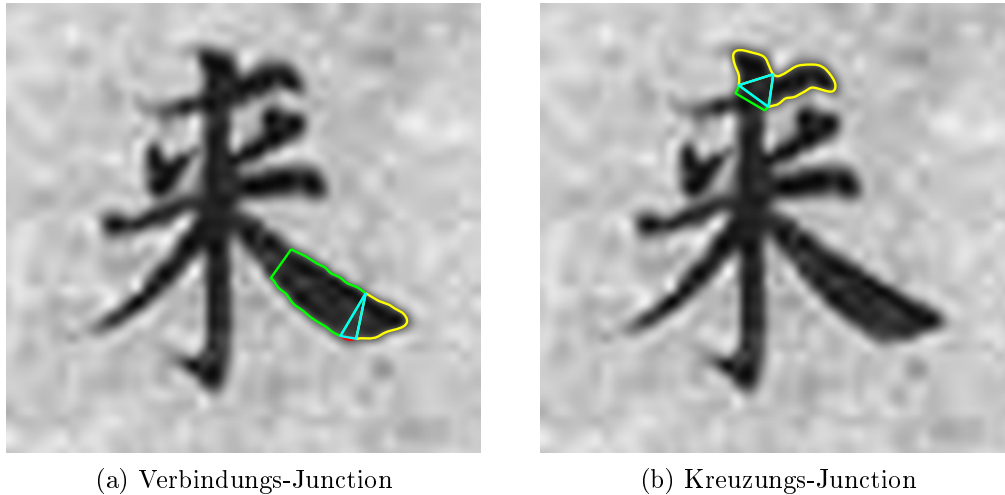


Abbildung 4.9: Junction-Klassen

Definition 19 (Verbindungs-Junction) *Eine Verbindungs-Junction stellt eine Verbindung zwischen zwei Strichen der Klassen Endstrich oder Verbindungsstrich und einem Störungsstrich dar.*

$$J = \text{Verbindungs-Junction gdw. } |dist(J)| = 1$$

Definition 20 (Kreuzungs-Junction) *Über eine Kreuzungs-Junction werden nur echte Teilstriche, das heißt keine Störungen miteinander verbunden.*

$$J = \text{Kreuzungs-Junction gdw. } |dist(J)| = 0$$

Abbildung 4.9 zeigt Beispiele von Verbindungs- und Kreuzungs-Junctions und die mit ihnen verbundenen Teilstriche.

Die auf die beschriebene Weise vorgenommene Klassifizierung wird für die folgenden Schritte der Junction-Reduktion und anschließenden Junction-Verschmelzung verwendet.

4.3.2 Junction-Reduktion

Da die Form vieler handschriftlicher chinesischer Schriftzeichen auch nach der in Kapitel 3 durchgeführten Vorverarbeitung nicht vollständig glatt ist, entsteht bei ihrer Delaunay-Triangulierung eine hohe Anzahl von Verbindungs-dreiecken. An Abbildung 4.1 lässt sich erkennen, dass nicht alle dieser Verbindungs-dreiecke als Junctions zu zählen sind und eine Reduzierung der Verbindungs-dreiecke redundante Verschmelzungen zwischen Teilstrichen verhindert werden können.

Um die Anzahl der Verbindungs-dreiecke zu verringern und aus der Menge aller Verbindungs-dreiecke solche herauszufiltern, die für die weitere Verarbeitung als Kreuzungs- oder Verbindungsstellen mehrerer Striche benötigt werden, wird ein zweistufiges Vorgehen verwendet, welches über die geometrischen Eigenschaften der Dreiecke sowie die

Junction-Klassen und somit die Typen der mit ihnen verbundenen Teilstriche eine Liste von zu entfernenden Junctions erstellt. Mit der durch diese Liste bereinigten Menge der Delaunay-Verbindungs-dreiecke wird anschließend der vorangegangene Schritt der Teilstrich-Extraktion wiederholt.

Sei J die aktuelle Junction und A_J ihr Flächeninhalt sowie ϕ_J die Menge der Seitenverhältnisse der Junction.

1. **Stufe:** in der ersten Stufe werden alle Einzelstrich- und Strichende-Junctions entfernt, da die mit Junctions dieser Klassen verbundenen Striche immer Teilstriche desselben Gesamtstriches darstellen.
2. **Stufe:** die zweite Stufe filtert aus den verbleibenden Junctions solche heraus, welche die folgenden drei Bedingungen erfüllen:
 1. $J = \text{Verbindungs-Junction}$
 2. $A_J > A_t$
 3. $\max(\phi_J) < \phi_t$

Für den Flächeninhalt A_J wird ein Schwellenwert A_t berechnet, der alle Verbindungs-dreiecke der Triangulierung in zwei Gruppen aufteilt. Die erste Gruppe enthält Verbindungs-dreiecke mit einem Flächeninhalt $A_J > A_t$, für welche anhand des zweiten Kriteriums der Seitenverhältnisse entschieden wird, ob es sich um Junctions handelt oder nicht. Die zweite Gruppe enthält alle Verbindungs-dreiecke mit einem Flächeninhalt $A_J < A_t$, deren Klassifikation als Junction aufgrund ihres geringen Flächeninhaltes ausgeschlossen wird und die daher nicht weiter betrachtet werden. Um den Schwellenwert A_t für den Flächeninhalt zu berechnen, bieten sich mehrere Ansätze an. Ein naheliegender Gedanke ist es, ein Histogramm über alle in einer Triangulierung vorkommenden Flächeninhalte aufzustellen und den Schwellenwert an einer Stelle eines ausgeprägten lokalen Minimums zwischen zwei ausgeprägten lokalen Maxima zu wählen. Des Weiteren kann der Schwellenwert auch als Median aller vorkommenden Flächeninhalte berechnet werden. Eine weitere Möglichkeit, den gesuchten Schwellenwert zu finden, stellt die Quotientenbildung zweier aufeinander folgender Flächeninhalte bei vorheriger Sortierung der Dreiecke nach ihrem Flächeninhalt dar. Auf diese Weise wird der größte Sprung zwischen zwei Flächeninhalten ermittelt, welcher einen guten Indikator für die Klassifizierung als Junction bietet. Allerdings kann der so errechnete Schwellenwert nur als Richtwert verwendet werden. Um aus den herausgefilterten Dreiecken mit einem über dem Schwellenwert liegendem Flächeninhalt die tatsächlichen Junctions zu gewinnen, ist also ein weiterer Verarbeitungsschritt notwendig. Hierfür hat sich das Verhältnis der Seitenlängen der Dreiecke zueinander als ein guter Indikator zur Klassifizierung erwiesen.

Besitzt ein Dreieck einen Flächeninhalt A_j größer als der Schwellenwert A_t , werden die Längenverhältnisse zwischen seinen drei Seiten berechnet, um zu bewerten, ob es sich bei dem Dreieck um eine Junction handelt oder nicht. Das Längenverhältnis zwischen der längsten und der kürzesten Seite einer Junction muss kleiner als ein Schwellenwert ϕ_t sein. Bei Versuchen hat sich ein Schwellenwert von $\phi_t = 2$ als angemessen erwiesen. Dadurch werden nur solche Dreiecke als Junction klassifiziert, deren längste Seite maximal doppelt so lang ist wie ihre kürzeste Seite.

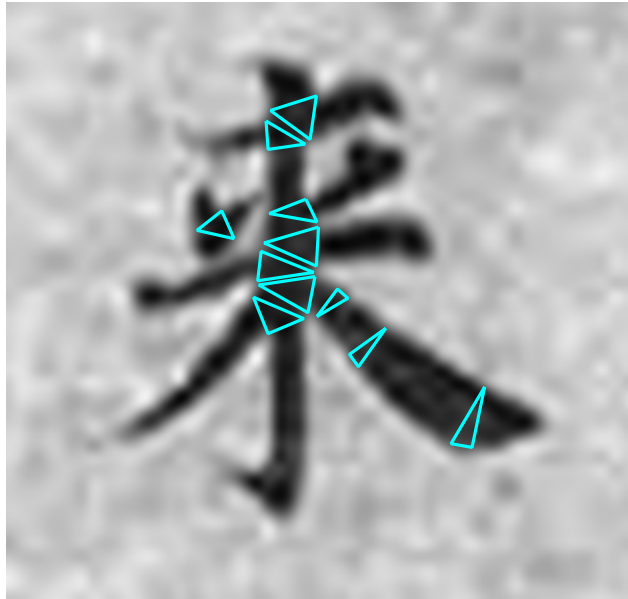


Abbildung 4.10: Nach der Junction-Reduktion übrig bleibende Junctions.

Zusammengefasst wird also durch die Kombination der Flächeninhalts- und Seitenverhältniseigenschaften effektiv geprüft, ob es sich bei einem Verbindungs-dreieck aus der Triangulierung um eine Junction handelt oder nicht. Abbildung 4.10 zeigt das Ergebnis der initialen Junction-Extraktion. Die Menge der extrahierten Junctions wird im Folgenden wie Kreuzungspunkte beziehungsweise *interest points* in skelett-basierten Verfahren zur Strichextraktion und Zeichenerkennung behandelt.

Mit der von redundanten Verbindungs-dreiecken bereinigten Menge der Junctions wird anschließend der Schritt der Teilstrich-Extraktion wiederholt. Die dadurch entstehende Teilstrichliste wird als Grundlage für die folgenden Schritte verwendet.

4.3.3 Junction-Verschmelzung

Eine Junction im Sinne des hier vorgestellten Algorithmus' stellt eine Schnittstelle oder Berührstelle mehrerer Striche dar. Stellt man sich eine Junction beispielsweise als klassische Kreuzung aus zwei sich orthogonal schneidenden Strichen vor, ist leicht ersichtlich, dass die Junction als Viereck repräsentiert werden kann, das über seine vier Seiten mit den Strichfortsätzen auf beiden Seiten verbunden ist. Da die initial extrahierten Junctions jedoch aus Dreiecken der Delaunay-Triangulierung bestehen, können die beschriebenen Schnittstellen nicht durch initial extrahierte Junctions dargestellt werden. Um dieses Problem zu beheben, können Bedingungen für eine Verschmelzung zweier Junctions aufgestellt werden, um komplexere, den beschriebenen Anforderungen angemessene Junctions zu erzeugen und die Junction-Definition flexibler zu gestalten.

Im letzten Abschnitt wurde die Extraktion und Klassifizierung von Teilstrichen beschrieben. Dabei ist unter anderem die Klasse der Verbindungsstriche erläutert worden, deren Instanzen sowohl am Strichanfang als auch am Strichende mit einer Junction ver-



Abbildung 4.11: Verschmelzung zweier Junctions zu einer neuen Junction.

bunden sind. Die Länge dieser verbindenden Striche ist durch den Abstand zwischen den Junctions definiert, mit denen sie verbunden sind. Die beiden zentralen Junctions in Abbildung 4.11 sind durch einen sehr kurzen Verbindungsstrich miteinander verbunden. Die beiden Junctions stellen Teile einer kompletten Junction beziehungsweise Kreuzung der beiden Striche des Zeichens dar. In diesem Fall ist es wünschenswert, die beiden Teil-Junctions durch eine komplette Junction zu ersetzen und mit dieser weiterzuarbeiten. Um für zwei Junctions zu entscheiden, ob sie Teile einer komplexeren Junction-Struktur sind und miteinander verschmolzen werden können oder nicht, bietet sich die Längeeigenschaft des zwischen den Teilstrukturen liegenden Verbindungsstriches an. Unterschreitet diese einen von der durchschnittlichen Strichdicke abgeleiteten Schwellenwert, werden die Junctions verschmolzen, ansonsten gelten sie als vollständige Junctions.

4.4 Teilstrichverschmelzung zu Grundstrichen der chinesischen Schrift

Die bisher erarbeiteten Strukturen stellen die Grundlage für den folgenden Schritt der Teilstrichverschmelzung dar. An dieser Stelle kann das Verfahren an die Eigenschaften der Grundstriche verschiedener Schriften angepasst werden, um so verschiedene Arten von Strichklassen extrahieren zu können. Da das Ziel des vorliegenden Ansatzes die Analyse chinesischer Schriftzeichen ist, werden für die Verschmelzung von Teilstrichen entsprechende Kriterien angelegt, die die Erzeugung solcher Grundstriche erlauben, wie sie in Kapitel 2 beschrieben wurden. Demnach können chinesische Grundstriche in die Hauptkategorien „Punkt“, „Horizontaler Strich“, „Vertikaler Strich“ sowie in drei Klassen



Abbildung 4.12: Zwei Endstriche (gelb), die über die verschmolzene Junction (türkis) verbunden sind.

diagonaler Striche unterteilt werden. Die Klassifizierung der extrahierten Striche basiert auf ihren im letzten Teilkapitel berechneten Eigenschaften und wird im nächsten Abschnitt beschrieben.

Die Verschmelzung der extrahierten Teilstriche geht von den Junctions als zentralen Elementen aus. Aus diesem Grund durchläuft der Algorithmus die Liste aller Junctions, die nach dem vorangegangenen Junction-Verschmelzungsschritt konstruiert wurden, und erstellt für jede Junction J eine Matrix J_P , deren Einträge $J_{P_{i,j}}$ die *Verschmelzbarkeiten* aller mit J verbundenen Teilstrichpaare S_i, S_j enthalten. Die Verschmelzbarkeit $P(S_1, S_2)$ zweier Teilstriche S_1 und S_2 ist ein Wert zwischen 0 und 1 und soll angeben, mit welcher Wahrscheinlichkeit S_1 und S_2 Teile des gleichen Gesamtstriches sind.

Zur Berechnung der Verschmelzbarkeit zweier Teilstriche werden ihre in Abschnitt 4.2.2 berechneten Richtungseigenschaften verwendet. Aus den Haupt- und Endrichtungen α_{main} , α_{start} und α_{end} zweier Teilstriche S_i und S_j werden hierzu die drei Werte *Hauptrichtungskompatibilität* $C_{main}(S_i, S_j)$, *Endrichtungskompatibilität* $C_{end}(S_i, S_j)$ und *Achsenkontinuität* $C_{axis}(S_i, S_j)$ beider Teilstriche berechnet.

Die Kompatibilität zweier Richtungen wird über ihre Differenz ermittelt.

Definition 21 (Richtungsdifferenz) Seien α_1 und α_2 zwei Richtungen. Dann ergibt sich ihre Richtungsdifferenz $\alpha_{diff_1}(\alpha_1, \alpha_2)$ folgendermaßen:

$$\alpha_{diff_1}(\alpha_1, \alpha_2) = |\alpha_1 - \alpha_2|$$

Für den Fall, dass eine der Richtungen größer als 180 Grad ist, kann $\alpha_{diff_1}(\alpha_1, \alpha_2)$ größer als 180 werden. Da die Richtungsdifferenz zweier über eine gemeinsame Junction

verbundenen Teilstriche jedoch maximal 180 Grad betragen kann, müssen für diesen Fall folgende, angepasste Werte α_1' und α_2' berechnet werden:

$$\alpha_1' = \begin{cases} \alpha_1 & \text{wenn } \alpha_1 < \alpha_2 \\ 360 - \alpha_1 & \text{sonst} \end{cases}$$

$$\alpha_2' = \begin{cases} 360 - \alpha_2 & \text{wenn } \alpha_1 < \alpha_2 \\ \alpha_2 & \text{sonst} \end{cases}$$

Aus diesen Werten ergibt sich die angepasste Richtungsdifferenz $\alpha_{diff_2}(S_1, S_2)$:

$$\alpha_{diff_2}(\alpha_1, \alpha_2) = \alpha_1' + \alpha_2'$$

Abhängig von den berechneten Werten wird die endgültige Richtungsdifferenz α_{diff_3} ausgewählt:

$$\alpha_{diff_3}(\alpha_1, \alpha_2) = \begin{cases} \alpha_{diff_1}(\alpha_1, \alpha_2) & \text{wenn } \alpha_{diff_1}(\alpha_1, \alpha_2) < 180 \\ \alpha_{diff_2}(\alpha_1, \alpha_2) & \text{sonst} \end{cases}$$

Um die Richtungsdifferenz zweier Richtungen für die Berechnung ihrer Kompatibilität verwenden zu können, wird sie anschließend auf einen Wert zwischen 0 und 1 normalisiert.

$$\alpha_{diff}(\alpha_1, \alpha_2) = \frac{\alpha_{diff_3}(\alpha_1, \alpha_2)}{180}$$

Definition 22 (Richtungskompatibilität) Sei $\alpha_{diff}(\alpha_1, \alpha_2)$ die normalisierte Richtungsdifferenz zweier Richtungen α_1 und α_2 nach Definition 21. Dann ergibt sich die entsprechende Richtungskompatibilität folgendermaßen:

$$C(\alpha_1, \alpha_2) = 1 - \alpha_{diff}(\alpha_1, \alpha_2)$$

Bei der Berechnung der Richtungskompatibilität muss dabei beachtet werden, dass die Richtung eines Striches S_i prinzipiell die beiden Werte α und $(\alpha + 180) \bmod 360$ haben kann. Berechnet man beispielsweise die Hauptrichtung des linken markierten Endstriches S_e aus Abbildung 4.12, ergibt sich der Wert $\alpha_{main}(S_e) = 168$. Bei dieser Berechnung würde S_e von rechts nach links verlaufen. Der Strich kann jedoch auch in der entgegengesetzten Richtung, also von links nach rechts verlaufen, was einer Vertauschung seiner Start- und Endpunkte entspricht. In diesem Fall wäre seine Hauptrichtung $(\alpha_{main}(S_e) + 180) \bmod 360 = 348$.

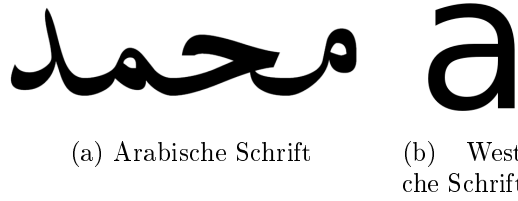


Abbildung 4.13: Runde Striche in anderen Schriften

Diese Mehrdeutigkeit wird bei der Berechnung der Haupt- und Endrichtungskompatibilitäten zweier Teilstriche eingeschränkt, da sie über eine gemeinsame Junction verbunden sind. Indem die Richtung eines Teilstriches festgelegt wird, ergibt sich die Richtung des zweiten Striches automatisch. Während einer der beiden Striche die Junction als End-Junction hat, stellt dieselbe Junction die Start-Junction des anderen Striches dar.

1. *Haupttrichtungs-Kompatibilität*: Wie in Sektion 2.1.3 beschrieben, handelt es sich bei den Grundstrichen der chinesischen Schrift um Striche ohne wesentliche Richtungsänderung. Im Gegensatz zu Einzelstrichen anderer Schriften, wie beispielsweise der arabischen Schrift, enthalten chinesische Schriftzeichen also keine runden Grundstriche oder wesentlichen Richtungsänderungen innerhalb einzelner Grundstriche. Abbildung 4.13 zeigt ein paar Beispiele für runde Striche in anderen Schriften. Daher eignet sich der Vergleich der Haupttrichtung zweier Striche zur Beurteilung, ob ein Teilstrich die Fortsetzung eines anderen Teilstriches darstellt oder nicht, beziehungsweise ob beide Teilstriche Teile des gleichen Grundstriches sind oder nicht. Aus diesem Grund kann für alle chinesischen Grundstriche eine einzelne Haupttrichtung angegeben werden, welche der tatsächlichen Schreibrichtung des Striches entspricht.

Die Haupttrichtungskompatibilität C_{main} zweier Striche S_i und S_j wird nun wie oben beschrieben über die Differenz der Strichhaupttrichtungen $\alpha_{main}(S_i)$ und $\alpha_{main}(S_j)$ ermittelt.

Definition 23 (Haupttrichtungskompatibilität) Seien S_1 und S_2 zwei Teilstriche und sei $\alpha_{diff}(\alpha_{main}(S_1), \alpha_{main}(S_2))$ die normalisierte Richtungs­differenz ihrer Haupttrichtungen nach Definition 21.

Dann ergibt sich ihre Haupttrichtungskompatibilität $C_{main}(S_1, S_2)$ folgendermaßen:

$$C_{main}(S_1, S_2) = C(\alpha_{main}(S_1), \alpha_{main}(S_2)) = 1 - \alpha_{diff}(\alpha_{main}(S_1), \alpha_{main}(S_2))$$

Die Haupttrichtungskompatibilität stellt für die meisten in chinesischen Schriftzeichen vorkommenden Striche einen guten Indikator für die Entscheidung dar, ob zwei Teilstriche verschmolzen werden sollen oder nicht. Falls jedoch in einem gebogenen Strich eines chinesischen Zeichens durch das vorgestellte Verfahren kein Dreieck der Delaunay-Triangulierung als Junction klassifiziert wurde und dieser gebogene Strich daher als ein einzelner Teilstrich extrahiert wird, entspricht die Haupttrichtung, welche als Richtung

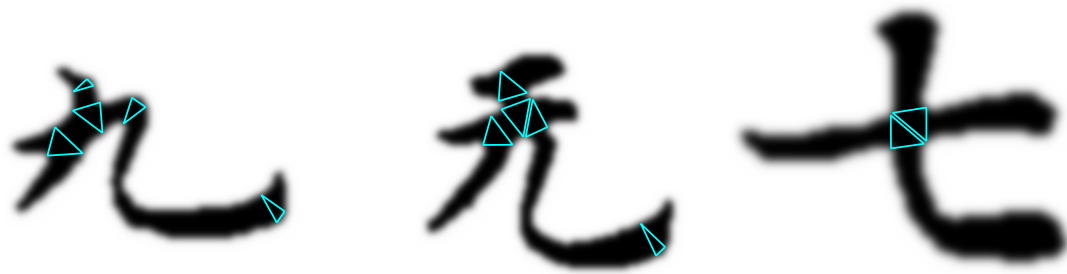


Abbildung 4.14: gebogene Striche ohne Junctions (türkis markiert)

von Start- zu Endpunkt berechnet wird, nicht mehr zwangsläufig der lokalen Richtung des Striches in der Nähe seiner Start- und Endpunkte und kann daher nicht zur Kompatibilitätsberechnung mit einem als Verschmelzungskandidaten in Frage kommenden zweiten Teilstrich verwendet werden. Abbildung 4.14 zeigt entsprechende Beispiele.

In einem solchen Fall ist es erforderlich, statt der Hauptrichtung des Teilstriches ein weiteres Kompatibilitätsmaß zu berechnen.

2. *Endrichtungskompatibilität*: Statt der Hauptrichtungen zweier Striche S_i und S_j können auch deren Start- und Endrichtungen zur Kompatibilitätsberechnung verwendet werden. Dazu wird die Richtungskompatibilität der Endrichtung des einen Striches und der Startrichtung des anderen Striches berechnet. Im Falle dieser lokalen Strichrichtungen muss zusätzlich berücksichtigt werden, auf welche Weise S_i und S_j miteinander verbunden sind. Stimmt die End-Junction von S_i mit der Start-Junction von S_j überein, muss dementsprechend die Kompatibilität der Endrichtung von S_i und der Startrichtung von S_j berechnet werden. Umgekehrt wird die Richtungskompatibilität der Endrichtung von S_j und der Startrichtung von S_i berechnet, wenn ihre gemeinsame Junction die End-Junction von S_j und die Startjunction von S_i ist. Allgemein müssen jeweils die lokalen Richtungen zweier Striche berücksichtigt werden, die der Junction entsprechen, über die die Striche verbunden sind.

Definition 24 (Endrichtungskompatibilität) Seien S_1 und S_2 zwei Teilstriche und sei J die End-Junction von S_1 und die Start-Junction von S_2 . Sei weiterhin $\alpha_{diff}(\alpha_{end}(S_1), \alpha_{start}(S_2))$ die normalisierte Richtungsdivergenz der Endrichtung $\alpha_{end}(S_1)$ von S_1 und der Startrichtung $\alpha_{start}(S_2)$ von S_2 . Dann ergibt sich ihre Endrichtungskompatibilität $C_{end}(S_1, S_2)$ folgendermaßen:

$$C_{end}(S_1, S_2) = C(\alpha_{end}(S_1), \alpha_{start}(S_2)) = 1 - \alpha_{diff}(\alpha_{end}(S_1), \alpha_{start}(S_2))$$

Die Berechnung der Endrichtungskompatibilität wird für $J = J_s(S_1) = J_e(S_2)$ analog durchgeführt.

3. *Achsenkontinuität*: die beiden bisher beschriebenen Komponenten zur Berechnung eines Kompatibilitätsmaßes bewerten die Ähnlichkeit der Richtungseigenschaften zweier Teilstriche. Die Richtungsähnlichkeit stellt eine Hauptbedingung für die Teilstrichverschmelzung dar. Jedes Paar von Teilstrichen muss in diesem Kriterium gut bewertet werden, um letztendlich zu einem neuen Strich verschmolzen werden zu können. Zusätzlich muss allerdings ein Maß berechnet werden, das angibt, wie stark die Strichenden beider Teilstriche an der Junction, über welche sie verschmolzen werden sollen, voneinander versetzt sind. Dieses Maß ist notwendig, um Striche, deren Richtung übereinstimmt, die jedoch stark versetzt sind, so zu kennzeichnen, dass sie nicht verschmolzen werden.

Um das beschriebene Merkmal für die Achsen zweier Teilstriche zu berechnen, gibt es verschiedene Herangehensweisen. Ein Ansatz ist es, die Richtung $\alpha_c(S_1, S_2)$ des Verbindungsvektors $\overrightarrow{P_e(S_1)P_s(S_2)}$ zwischen dem Endpunkt $P_e(S_1)$ des ersten Teilstriches S_1 und dem Startpunkt $P_s(S_2)$ des zweiten Teilstriches S_2 zu extrahieren. Die Richtung dieses Verbindungsvektors kann nun mit den Hauptrichtungen $\alpha_{main}(S_1)$ und $\alpha_{main}(S_2)$ beider Striche in Verbindung gebracht werden. Dazu werden die Richtungskompatibilitäten zwischen der Hauptrichtung $\alpha_{main}(S_1)$ des ersten Striches und der Richtung $\alpha_c(S_1, S_2)$ des Verbindungsvektors zwischen den Strichen sowie zwischen der Verbindungsvektorrichtung $\alpha_c(S_1, S_2)$ und der Hauptrichtung $\alpha_{main}(S_2)$ des zweiten Striches berechnet:

$$\begin{aligned} C(\alpha_{main}(S_1), \alpha_c(S_1, S_2)) &= 1 - \alpha_{diff}(\alpha_{main}(S_1), \alpha_c(S_1, S_2)) \\ C(\alpha_c(S_1, S_2), \alpha_{main}(S_2)) &= 1 - \alpha_{diff}(\alpha_c(S_1, S_2), \alpha_{main}(S_2)) \end{aligned}$$

Aus diesen Richtungskompatibilitäten lässt sich anschließend die Achsenkontinuität der beiden Striche kombinieren. Abbildung 4.15 visualisiert die Richtungen, die zur Berechnung der Achsenkontinuität zweier Striche verwendet werden.

Definition 25 (Achsenkontinuität) *Seien S_1 und S_2 zwei Teilstriche. Seien weiterhin $\overrightarrow{P_e(S_1)P_s(S_2)}$ der Verbindungsvektor zwischen dem Endpunkt $P_e(S_1)$ von S_1 und dem Startpunkt $P_s(S_2)$ von S_2 und $\alpha_c(S_1, S_2)$ die Richtung des Verbindungsvektors. Dann ergibt sich die Achsenkontinuität $C_{axis}(S_1, S_2)$ von S_1 und S_2 über die Richtungskompatibilitäten ihrer Hauptrichtungen und der Richtung des Verbindungsvektors folgendermaßen:*

$$C_{axis}(S_1, S_2) = \frac{1}{2}(C(\alpha_{main}(S_1), \alpha_c(S_1, S_2)) + C(\alpha_c(S_1, S_2), \alpha_{main}(S_2)))$$

Die vorgeschlagene Methode zur Berechnung der Achsenkontinuität liefert also höhere Werte für Strichpaare, deren Hauptrichtungen gemäß Definition 22 kompatibler mit der Richtung des Verbindungsvektors beider Striche sind als bei anderen Strichpaaren. Das bedeutet, dass die Achsenkontinuität zweier Striche umso höher ist, je weiter beide Striche voneinander entfernt sind, da die Richtungsunterschiede entsprechend abnehmen.

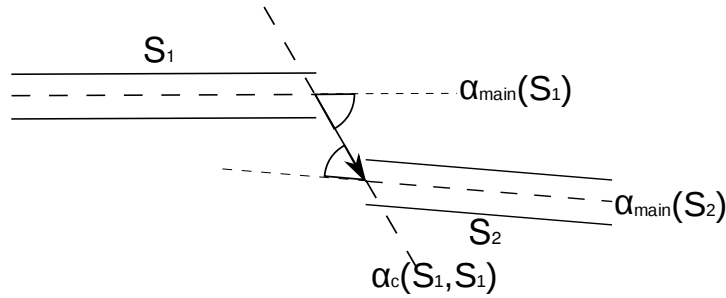


Abbildung 4.15: Die Berechnung der Achsenkontinuität zweier Striche wird über ihre Haupttrichtungen vorgenommen.

Nach den Definitionen der Haupt- und Endrichtungskompatibilitäten sowie der Achsenkontinuität zweier Teilstriche lässt sich nun der Begriff der Verschmelzbarkeit definieren.

Definition 26 (Verschmelzbarkeit) Die Verschmelzbarkeit $M(S_1, S_2)$ zweier Teilstriche S_1 und S_2 berechnet sich aus ihren Haupt- und Endrichtungskompatibilitäten $C_{main}(S_1, S_2)$ und $C_{end}(S_1, S_2)$, sowie aus ihrer Achsenkontinuität $C_{axis}(S_1, S_2)$.

$$M(S_1, S_2) = w_{main} * C_{main}(S_1, S_2) + w_{end} * C_{end}(S_1, S_2) + w_{axis} * C_{axis}(S_1, S_2)$$

Dabei bezeichnen w_{main} , w_{end} und w_{axis} drei Gewichtungsfaktoren, für die gilt

$$w_{main} + w_{end} + w_{axis} = 1$$

4.4.1 Verschmelzung von Verbindungs- und Endstrichen

Die Berechnung der Verschmelzbarkeit zwischen Verbindungs- und Endstrichen, die über eine gemeinsame Junction verbunden sind, wird wie im vorangegangenen Abschnitt beschrieben vorgenommen. Abbildung 4.12 zeigt eine türkis markierte Junction, die aus zwei im ersten Schritt extrahierten Junctions verschmolzen wurde, sowie zwei gelb markierte über diese Junction verbundene Endstriche, deren Verschmelzbarkeit auf die in diesem Abschnitt beschriebene Weise berechnet wird.

Nach der Berechnung dieses Wertes für alle möglichen Teilstrichkombinationen über alle Junctions wählt der Algorithmus die Strichkombination mit dem höchsten Verschmelzbarkeitswert aus und überprüft, ob dieser über einem Schwellenwert M_{thres} liegt. Ist dies der Fall, werden die beiden Teilstriche miteinander zu einem neuen Strich verschmolzen und aus der Liste der an die aktuelle Junction angrenzenden Teilstriche entfernt. Dieser Vorgang wird solange wiederholt, bis keine Strichkombination mehr eine Verschmelzbarkeit über dem Schwellenwert M_{thres} besitzt. Die resultierenden Striche stellen die aus dem Schriftzeichen extrahierten Grundstriche dar. Abbildung 4.17 zeigt die auf die beschriebene Vorgehensweise gewonnen Grundstriche des Schriftzeichens für „kommen“.



Abbildung 4.16: Ergebnis der Teilstrichverschmelzung zweier Endstriche

4.4.2 Verarbeitung von Störungen

Nach der Verschmelzung von Verbindungs- und Endstrichen im letzten Abschnitt müssen anschließend auch die über Junctions mit anderen Strichen verbundenen Störungen verarbeitet werden.

Störungen sind genau wie Endstriche nur über eine Junction mit anderen Strichen verbunden. Die Verarbeitung einer Störung S geschieht in zwei Schritten.

1. Im ersten Schritt wird versucht, S mit einem anderen Strich S_i zu verschmelzen, der mit der gleichen Junction verbunden ist wie die Störung. In diesem Fall wird wie in Definition 26 beschrieben die Verschmelzbarkeit $M(S, S_i)$ der Störung S und des Teilstriches S_i berechnet. Da eine Störung per Definition jedoch alleine keinen Grundstrich darstellen kann, muss sie mit einem anderen Strich verbunden werden, unabhängig davon, ob die berechnete Verschmelzbarkeit überhalb des Schwellenwertes M_{thres} liegt oder nicht.

2. Der zweite Schritt verarbeitet alle Störungen, die mit einer Junction verbunden sind, mit der kein weiterer Strich verbunden ist. Solche Störungen können demnach nicht mit anderen Teilstrichen verschmolzen werden. Stattdessen werden in diesem Fall die Konturpunkte der Störung in die Kontur eines bestehenden Teilstriches eingefügt. Dazu sucht das Verfahren den Strich S_j aus allen zu diesem Zeitpunkt extrahierten und verschmolzenen Teilstrichen, welcher die geringste Entfernung zu den ersten und letzten Punkt in der Liste der Störungspunkte besitzt und fügt die Störungspunkte an der entsprechenden Stelle in die Kontur von S_j ein.

Nachdem durch das in diesem Kapitel vorgestellte Verfahren alle Teilstriche solange verschmolzen wurden, bis keine weiteren Verschmelzungen mehr möglich sind, werden die so entstandenen Striche mit Hilfe ihrer Eigenschaften in eine der sechs Klassen chi-

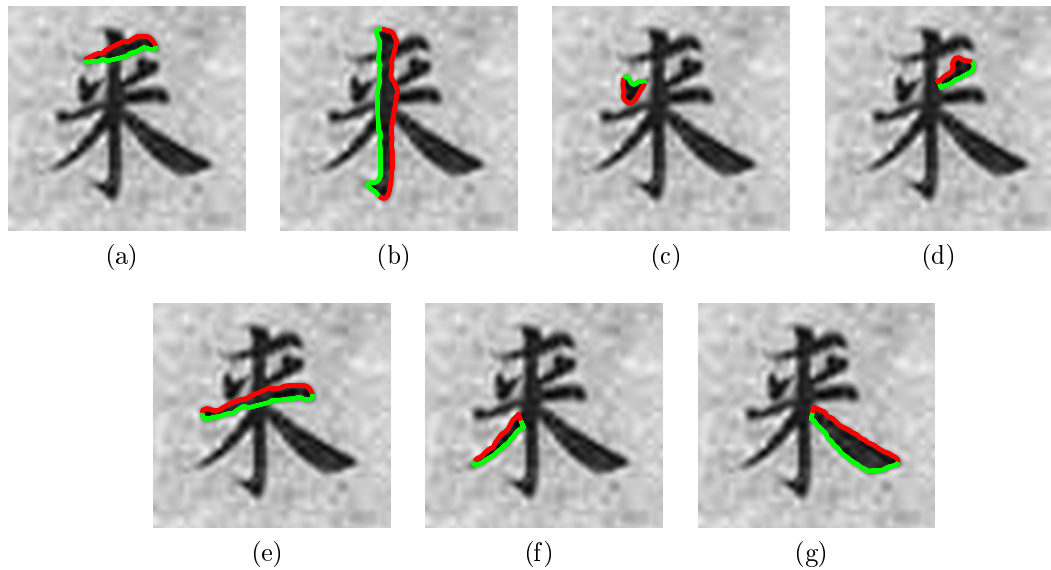


Abbildung 4.17: Extrahierte Grundstriche des chinesischen Schriftzeichens für „kommen“

nesischer Grundstriche klassifiziert. Dies ist Thema des folgenden Abschnittes.

4.5 Grundstrichklassifizierung

Die Klassifizierung der durch die Teilstrichverschmelzung gewonnenen Striche wird mit Hilfe ihrer in Abschnitt 4.2.2 berechneten Richtungs-, Längen- und Breitereigenschaften vorgenommen. Die Hauptrichtung dient dabei als wichtigstes Kriterium für die Grundstrichklassifizierung. Teilt man den die Ebene beschreibenden Vollwinkel von 360 Grad in acht Sektoren auf, welche jeweils einem Winkel von 45 Grad entsprechen, erhält man eine Winkelpartitionierung wie in Abbildung 4.19a. Diese acht Sektoren stellen eine Einteilung aller möglichen Richtungen in acht horizontale, vertikale und diagonale Grundrichtungen dar, die ein extrahierter Strich haben kann. Nimmt man die in Abschnitt 2.1.3 vorgestellten sechs Grundstriche der chinesischen Schrift als Basis, verringert sich die Menge der möglichen Grundrichtungen. Da horizontale Striche immer von links nach rechts und vertikale Striche immer von oben nach unten geschrieben werden, fallen die Sektoren 4 und 6, welche vertikalen und horizontalen Schreibrichtungen von rechts nach links beziehungsweise von unten nach oben entsprechen, als mögliche Richtungen weg. Ebenso werden diagonale Striche der Klassen „dot“ und „press down“, deren einer Endpunkt jeweils rechts unterhalb des anderen liegt, immer von links oben nach rechts unten geschrieben, wodurch auch die Winkel aus Sektor 5 als mögliche Schreibrichtungen wegfallen. Durch diese Regeln der Schreibrichtung wird die Menge der möglichen Richtungssektoren eingeschränkt, da keine chinesischen Grundstriche mit den in den Sektoren 4, 5 und 6 zusammengefassten Richtungen als Schreibrichtungen existieren. Aus dieser Einschränkung ergeben sich die fünf in Abbildung 4.19b grün unterlegten

Sektor	Winkel	Aktion	Grundstrichklassen
0	337 - 21	-	„horizontal“
1	22 - 66	Strichbreitenanalyse	„dot“ oder „press down“
2	67 - 111	-	„vertical“
3	112 - 156	Strichbreitenanalyse	„throw away“, „rise“
4	157 - 201	drehe Strich	„horizontal“
5	202 - 246	drehe Strich, Strichbreitenanalyse	„dot“ oder „press down“
6	247 - 291	drehe Strich	„vertical“
7	292 - 336	Strichbreitenanalyse	„throw away“, „rise“

Tabelle 4.1: Strichklassifizierung

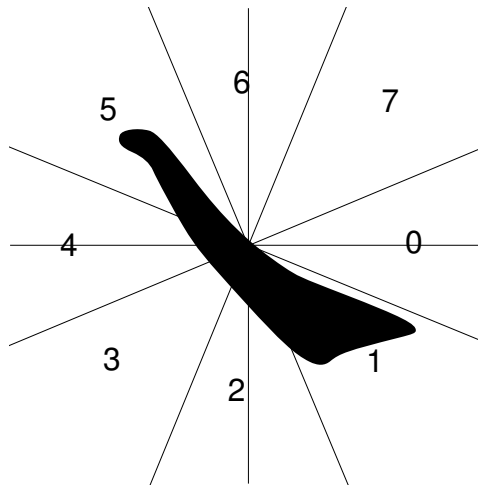


Abbildung 4.18: Grundstrich der Klasse „press down“ mit Winkelpartitionierung

Sektoren als mögliche Grundrichtungen der extrahierten Striche.

Da die bis zu diesem Punkt extrahierten und verschmolzenen Striche jedoch noch nicht die Einschränkungen der Schreibrichtungen chinesischer Grundstriche berücksichtigen, kann der die Hauptrichtung eines Striches S beschreibende Neigungswinkel $\alpha_{main}(S)$ jeden Wert zwischen 0 und 360 Grad haben. Zur Klassifizierung der bis zu diesem Punkt extrahierten und verschmolzenen Teilstriche müssen deren Richtungseigenschaften daher an die Schreibrichtungen chinesischer Grundstriche angepasst werden. Tabelle 4.1 listet die Verarbeitungsschritte für alle extrahierten Striche abhängig von ihren Hauptrichtungen auf. Horizontale und vertikale Striche müssen demnach gegebenenfalls umgedreht werden, können jedoch ohne weitere Analyse direkt klassifiziert werden. Da die Hauptrichtungen von Strichen der Klassen „dot“ und „press down“ jeweils in den Sektor eins fallen, müssen zur Unterscheidung und korrekten Klassifizierung solcher Striche weitere Eigenschaften mitberücksichtigt werden. Gleiches gilt für „rise“- und „throw away“-Striche, deren Hauptrichtungen in die Sektoren drei und sieben fallen. Das entsprechende Vorgehen bei solchen Grundstrichen wird im Folgenden erklärt. Abbildung 4.18 zeigt als Beispiel einen Strich der Klasse „press down“ mit hinterlegter Winkelpartitionierung.

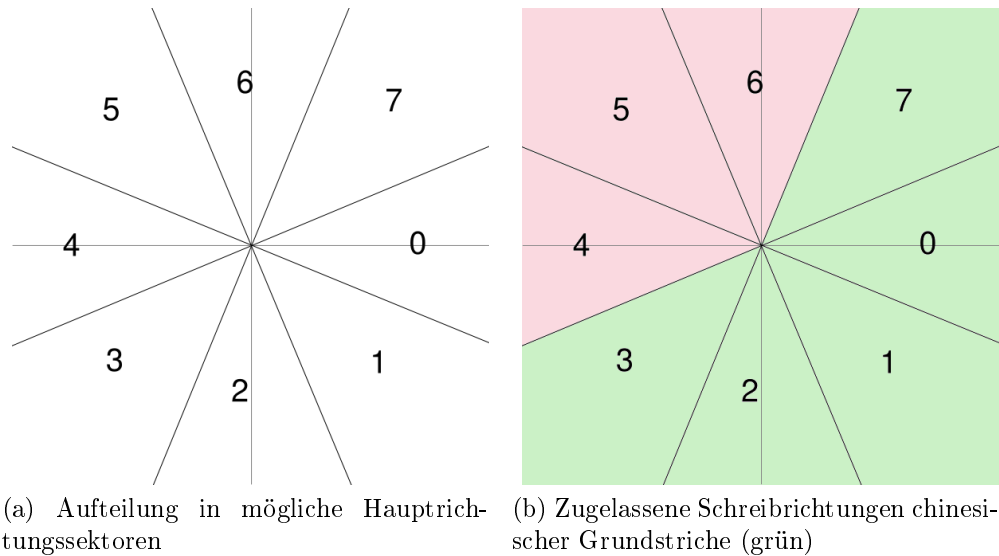


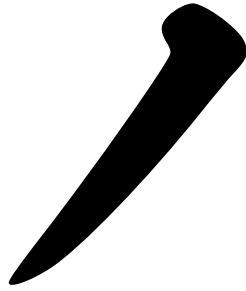
Abbildung 4.19: Winkelpartitionierung in acht Sektoren

Wie aus Tabelle 4.1 ersichtlich wird, ist die Klassifizierung bei Strichen eindeutig, deren Hauptrichtungen in einem der Sektoren 0, 2, 4 und 6 liegen. Die Sektoren 0 und 4 stehen für Grundstriche der Klasse „horizontal“, während die Sektoren 2 und 6 vertikale Grundstriche repräsentieren. Um gegebenenfalls die Schreibrichtung bei solchen Strichen zu korrigieren, müssen solche, deren Hauptrichtung in Sektor 4 oder 6 fällt, noch gedreht werden. Dabei werden Start- und Endpunkt des Striches vertauscht und seine restlichen Stricheigenschaften entsprechend angepasst.

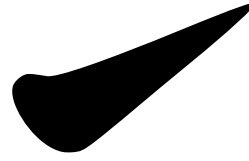
Zur Unterscheidung von Strichen der Klassen „dot“ und „press down“, deren Grundrichtungen beide im selben Sektor 1 liegen, werden ihre Breiten- und Längeneigenschaften ausgewertet. Während Striche der ersten Klasse kurz und relativ breit sind, besitzen „press down“-Striche eine größere Länge relativ zu ihrer maximalen Breite. Aus diesem Grund lässt sich die Unterscheidung von Strichen dieser beiden Klassen durch das Verhältnis ihrer Länge zu ihrer maximalen Breite vornehmen.

Als Beispiel werden hierbei die Strichbilder der Klassen „dot“ und „press down“ aus Tabelle 2.3 verwendet. Für das größennormalisierte Bild des ersten Striches der „dot“-Klasse berechnet das Verfahren eine maximale Strichbreite von 56 Pixeln und eine Länge von 136 Pixeln. Demgegenüber beträgt die maximale Breite des Striches der „press down“-Klasse 67 Pixel und seine Länge 304 Pixel. Als Unterscheidungsmerkmal zwischen Grundstrichen der „dot“-Klasse und solchen der „press down“-Klasse kann daher das unterschiedliche Verhältnis von Strichlänge zu Strichbreite genutzt werden, welches bei ersterem bei 2,4 und bei letzterem bei 4,5 liegt.

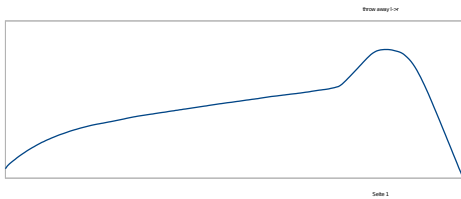
Auch zur Unterscheidung von Grundstrichen der Klassen „throw away“ und „rise“ werden deren Breitereigenschaften verwendet. Erstellt man eine Breitenprojektion von Strichen beider Strichklassen über ihre jeweiligen Strichlängen, sind sie jeweils in ihren Startregionen breiter als in ihren Endregionen. Kombiniert man diese Breitenbe-



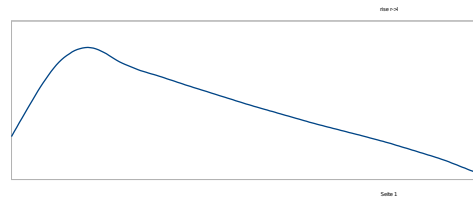
(a) Grundstrich der Klasse „throw away“



(b) Grundstrich der Klasse „rise“



(c) Breitenprojektion des Striches der „throw away“-Grundstrichklasse



(d) Breitenprojektion des Striches der „rise“-Grundstrichklasse

Abbildung 4.20: Breitenprojektionen von „rise“ und „throw away“ bei gleicher Richtung

dingungen mit den bei beiden Strichen unterschiedlichen Hauptrichtungen, ergeben sich daraus eindeutige Merkmalsvektoren für beide Grundstrichklassen. Die Informationen zur Erstellung der Breitenprojektionen werden durch die in Abschnitt 4.2.2 berechneten Breitenlisten bereitgestellt. Abbildung 4.20 zeigt Grundstriche der Klassen „throw away“ und „rise“, deren Hauptrichtung vereinheitlicht wurde, zusammen mit den dazugehörigen Breitenprojektionen. Die Vereinheitlichung der Hauptrichtung beider Striche wurde hier vorgenommen, um die unterschiedliche Breitenprojektion deutlich zu machen. Hier gehen beide Striche von links unten nach rechts oben, obwohl Striche der „throw away“-Klasse tatsächlich in die entgegengesetzte Richtung geschrieben werden.

Wie man an den Breitenprojektionen beider Striche ablesen kann, ist das linke untere Ende des „rise“-Grundstriches breiter als das rechte obere Ende des Striches. Umgekehrt ist es bei dem Grundstrich der „throw away“-Klasse. In diesem Fall ist der Startbereich im linken unteren Bereich des Striches schmaler als sein rechts oben gelegener Endbereich. Da die tatsächliche Richtung eines Striches der „throw away“-Klasse jedoch von rechts oben nach links unten geht, muss der Strich gedreht werden. Obwohl sich dadurch eine ähnliche Breitenprojektion wie bei Strichen der „rise“-Klasse ergibt, ist die Unterscheidung der beiden Grundstriche nun aufgrund der gegensätzlichen Hauptrichtungen möglich. Diese Unterscheidungsmöglichkeit für Striche dieser beiden Klassen wäre über ihre Hauptrichtungen alleine nicht möglich.

Durch die hier vorgestellten Möglichkeiten, die Mehrdeutigkeiten bei der Klassifikation

Grundstrichklasse	Hauptrichtungs- sektor	Breiten-/Längen- verhältnis	Breitenprojektion
„dot“	1	$\phi_l(S) < \phi_t$	-
„horizontal“	0	-	-
„press down“	1	$\phi_l(S) > \phi_t$	-
„throw away“	3	-	$proj(S)$
„vertical“	2	-	-
„rise“	7	-	$proj(S)$

Tabelle 4.2: Eindeutige Grundstrichklassifizierung

diagonaler Striche aufzulösen, wurden für alle der sechs in Abschnitt 2.1.3 vorgestellten Grundstrichklassen eindeutige Klassifizierungsmerkmale definiert. Tabelle 4.2 listet diese eindeutigen Klassifizierungsbedingungen für alle sechs Grundstrichklassen auf. Dabei steht S für einen zu klassifizierenden Strich und $\phi_l(S)$ für das Verhältnis zwischen Länge und Breite des Striches. Weiterhin beschreibt ϕ_t einen Schwellenwert für dieses Verhältnis. $proj(S)$ bezeichnet die Breitenprojektion eines Striches, bei der dessen Startbereich breiter als dessen Endbereich ist. Sind Einträge in der Tabelle mit „-“ markiert, bedeutet dies, dass die jeweilige Bedingung für die Klassifizierung der Grundstrichklasse nicht berücksichtigt wird.

5 Ergebnisse und Ausblick

Dieses abschließende Kapitel beschreibt die Ergebnisse, die das vorgestellte Verfahren angewendet auf eine Menge von 339 digitalen Bildern einzelner chinesischer Schriftzeichen erzielt hat. Diese Bilder wurden aus einem Ausschnitt eines antiken chinesischen Manuskripts entnommen und bieten relativ gute Qualität, aber vergleichsweise niedrige Auflösung. Im anschließenden Teil dieses Kapitels werden die Stärken und Schwächen des Verfahrens geschildert, sowie mögliche Weiterentwicklungen, um die Qualität der Extraktionsergebnisse weiter zu steigern.

5.1 Experimente

Um das vorstellte Verfahren zur Strichextraktion zu testen, wurde es auf die 339 aus dem chinesischen Manuskriptausschnitt der Abbildung 5.1 stammenden handschriftlichen Zeichen angewendet. Die Strichextraktion wurde für diese Zeichen mit festen Segmentierungs- und Junction-Verschmelzungsparametern durchgeführt, um bewerten zu können, wie das Verfahren automatisch und ohne Korrektur durch den Benutzer Striche aus handschriftlichen Zeichen extrahieren kann. Anschließend wurde das Verfahren auf einzelne Zeichen angewandt, für die die Parameter zur Berechnung der Konturen durch das Marching Squares-Verfahren sowie für die Verschmelzung von Junctions angepasst wurden. Auf diese Weise konnte festgestellt werden, wie abhängig das Verfahren von einer Parameteranpassung für einzelne Zeichen ist und in welchen Bereichen durch Weiterentwicklung zukünftige Verbesserungen in den Ergebnissen erzielt werden können.

5.1.1 Statistische Auswertung

Die in dem gezeigten Manuskriptausschnitt 5.1 enthaltenen Schriftzeichen sind aus insgesamt 3070 einzelnen Grundstrichen zusammengesetzt. Bei dieser Zählung wurden alle Grundstriche miteinbezogen, aus denen die jeweiligen Schriftzeichen formal bestehen. Aufgrund des Erscheinungsbildes von mit Pinseln erzeugten Schriftzeichenformen können dabei deren Bestandteile wie Radikale und Striche weniger deutlich ausgeprägt sein als es in gedruckten Zeichen der Fall ist. Abbildung 5.2 zeigt ein Beispiel für das beschriebene Szenario. In 5.2a ist das Bild eines chinesischen Schriftzeichens nach der in Kapitel 3 beschriebenen Vorverarbeitung zu sehen. In 5.2b wurde ein Radikal des Schriftzeichens markiert, dessen Erscheinungsbild als gedrucktes Zeichen in 5.2c zu sehen ist. In der Druckvariante ist deutlich zu erkennen, dass das fragliche Radikal aus zwei horizontalen und zwei vertikalen Grundstrichen aufgebaut ist und eine rechteckige bis quadratische

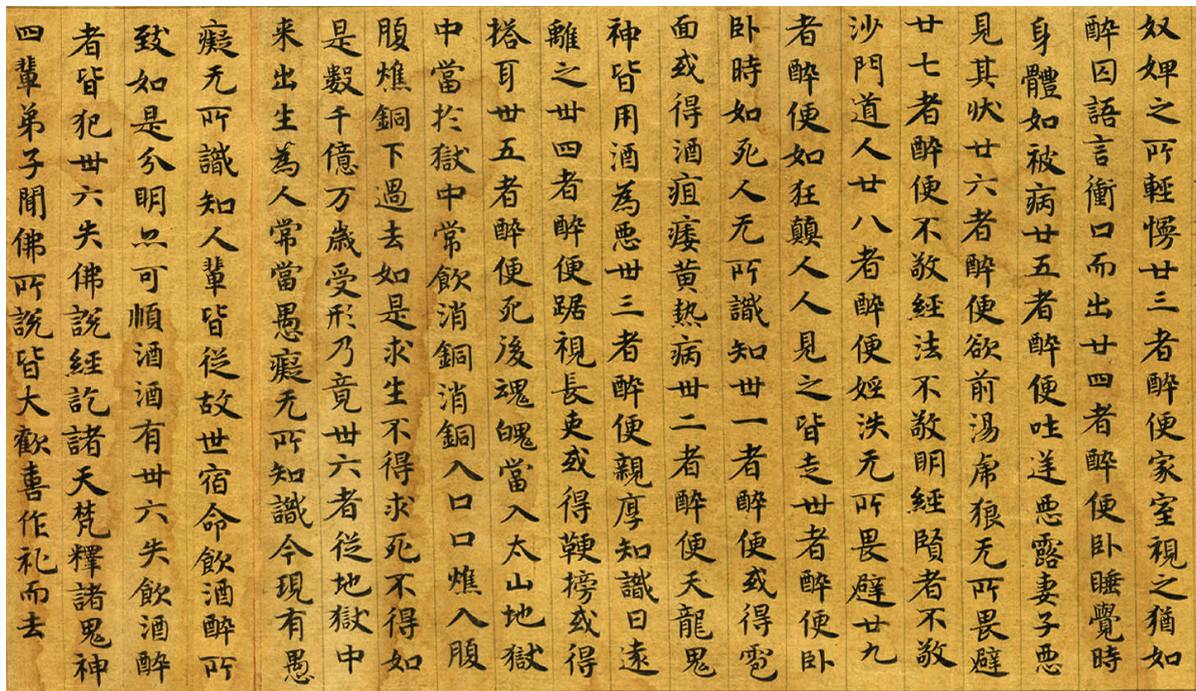


Abbildung 5.1: Manuskriptausschnitt

Form besitzt. In der handschriftlichen Variante dagegen ist vor allem die Länge des unteren der beiden horizontalen Grundstriche des Radikals so stark verkürzt, dass dieser nicht mehr als eigenständiger Strich durch das Verfahren erkannt und extrahiert wird.

Von den 3070 Grundstrichen konnte das Strichextraktionsprogramm 2220 korrekt extrahieren. Dies entspricht einer Stricherkennungsrate von 72,48 Prozent. Die Zeichen des Manuskriptausschnittes sind durchschnittlich aus 9 Strichen aufgebaut, von denen durchschnittlich 6,4 korrekt extrahiert wurden. Die Erkennungsrate steigt dabei deutlich antiproportional zur Komplexität eines Zeichens: Während sie bei Zeichen mit maximal 8 Grundstrichen bei durchschnittlich 86,1 Prozent liegt, werden von komplexeren Schriftzeichen mit mehr als 8 Strichen noch durchschnittlich 68,2 Prozent ihrer Grundstriche korrekt extrahiert. Abbildung 5.3 zeigt die Erkennungsraten des Verfahrens angewendet auf die unterschiedlich komplexen Schriftzeichen.

Tabelle 5.1 listet einige Schriftzeichen auf, aus denen das Verfahren erfolgreich alle Grundstriche extrahieren konnte. In der ersten Spalte der Tabelle wird das jeweilige Schriftzeichen angezeigt, während die restlichen Spalten die extrahierten Striche im Zeichen markiert darstellen.

5.1.2 Aufgetretene Probleme

Das vorgestellte Verfahren hat gute Ergebnisse bei Zeichen mit ausgeprägten Strichen erzielt und sich durch die Verwendung der Delaunay-Triangulierung als Vorverarbeitungs-

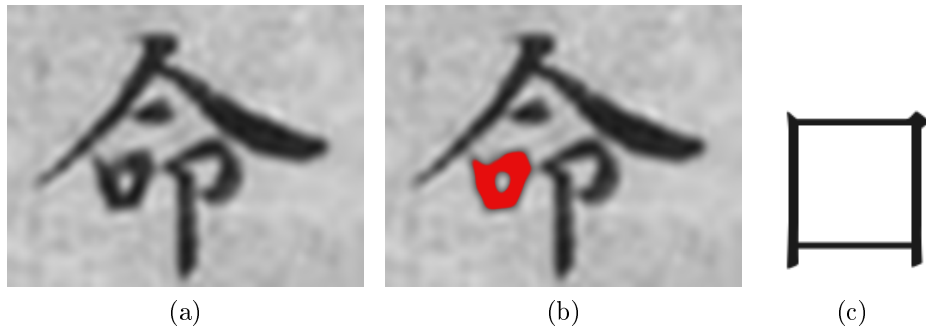


Abbildung 5.2: Degeneration eines Grundstrichs

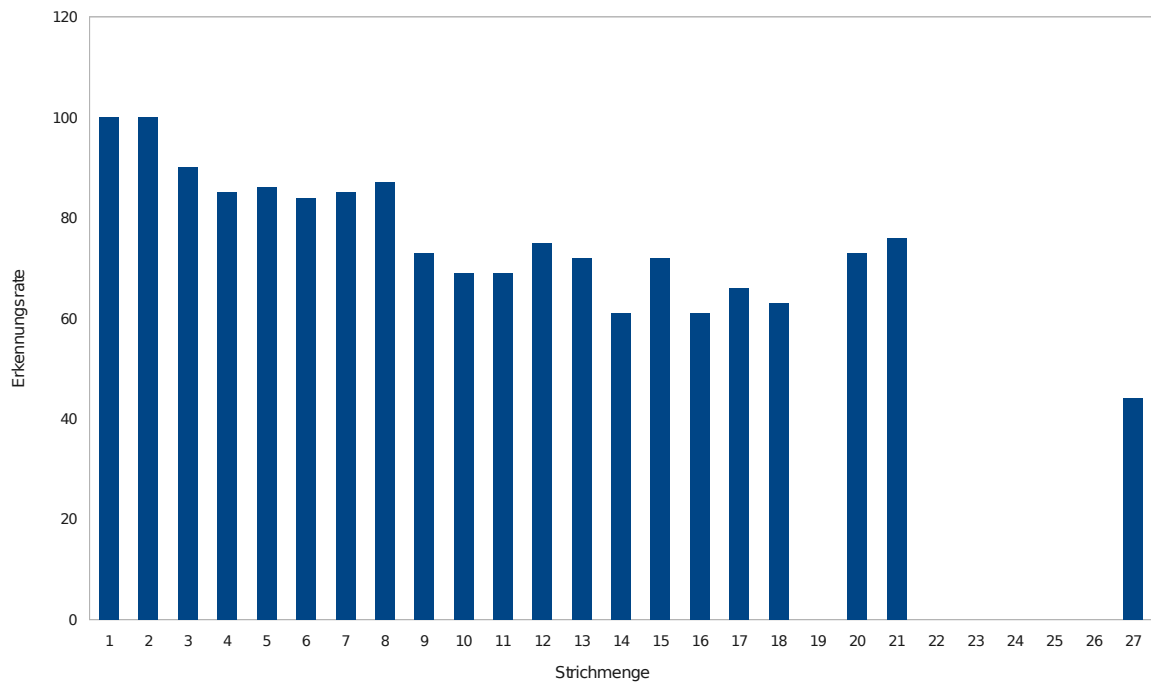


Abbildung 5.3: Die Erkennungsrate sinkt mit der Komplexität der chinesischen Schriftzeichen

Zeichen	Korrekt extrahierte Striche						
前	前	前	前	前	前	前	前
	前	前	前				
喜	喜	喜	喜	喜	喜	喜	喜
	喜	喜	喜	喜	喜		
所	所	所	所	所	所	所	所
竟	竟	竟	竟	竟	竟	竟	竟
	竟	竟	竟	竟			
卅	卅	卅	卅	卅	卅		
作	作	作	作	作	作	作	作
耳	耳	耳	耳	耳	耳	耳	
沙	沙	沙	沙	沙	沙	沙	沙
受	受	受	受	受	受	受	受
	受	受	受				

Tabelle 5.1: Korrekt extrahierte Grundstriche

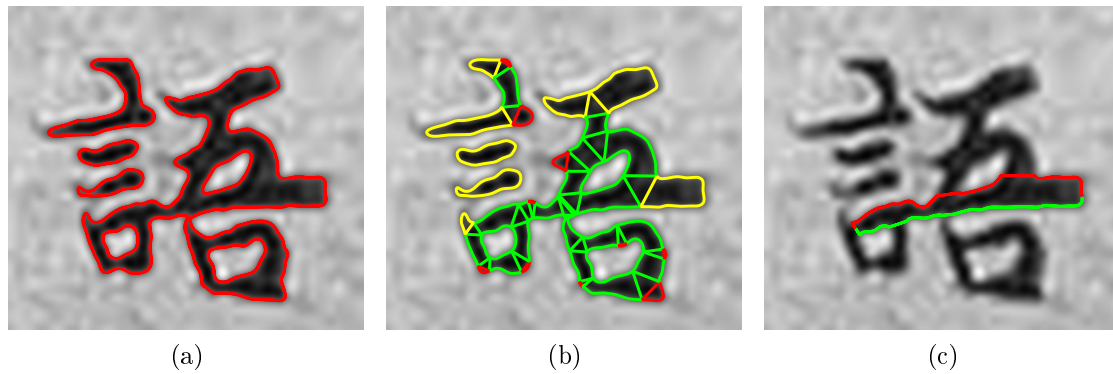


Abbildung 5.4: Fehlerhafte Ausgangsdaten und die Auswirkung auf die Strichextraktion

schritt für die Extraktion chinesischer Striche als relativ robust gegenüber Problemen wie gleichmäßiger Krümmung entlang der Konturen handschriftlicher Zeichen im Bereich von Strichkreuzungen erwiesen. Die Verwendung von Verbindungsdreiecken nach Definition der Chordal Axis Transform als Bereiche, die Strichkreuzungen innerhalb eines Zeichens entsprechen, erlaubt eine der Extraktion von *interest points* in skelettbasierten Verfahren ähnliche Vorgehensweise, ohne die Nachteile, die skelettbasierte Ansätze zum Beispiel durch überflüssige Kanten und Störungen im Bereich von Strichkreuzungen haben.

Das Verfahren hat jedoch auch einige Möglichkeiten zur Verbesserung und Optimierung offenbart. Diese Schwächen sollen im Folgenden mit Hilfe einiger Beispielzeichen erläutert werden.

Segmentierungsfehler

Wie jeder Algorithmus in der Bildverarbeitung ist auch das vorgestellte Verfahren zur Strichextraktion abhängig von der Qualität der zugrunde liegenden Ausgangsdaten. Durch in handschriftlichen Werken miteinander verbundene Striche, welche in dem jeweiligen Zeichen nicht verbunden sein sollten, entstehen Fehler in der weiteren Extraktion der Teilstriche, die durch das vorliegende Verfahren nicht ausgeglichen werden können und die Qualität der Ergebnisse spürbar negativ beeinflussen können. Abbildung 5.4 zeigt ein Beispiel für die Auswirkung fehlerhafter Ausgangsdaten auf die Qualität der Strichextraktion. Da die Lage und die Richtung der beiden horizontalen Striche in Abbildung 5.4b starke Übereinstimmung zeigen, führt die Kantendetektion (siehe Abbildung 5.4a) zu einer zusammenhängenden Kontur dieser Striche, durch die diese verbunden sind, und zu einer Verschmelzung beider zu einem horizontalen Grundstrich, wie in Abbildung 5.4c zu sehen ist.

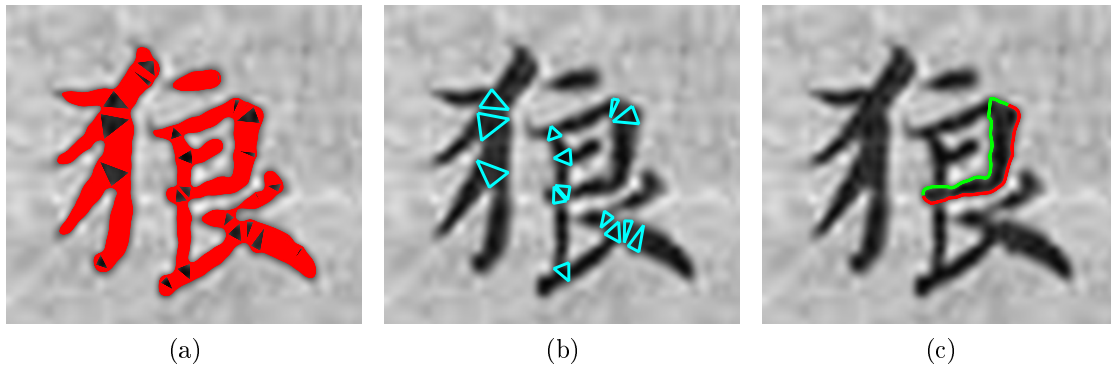


Abbildung 5.5: Fehlende Verbindungsdreiecke an Strichberührung

Fehlende Verbindungsdreiecke

Ein weiterer Punkt, welcher unerwünschte Ergebnisse liefern kann, ist die Abhängigkeit des Verfahrens in seiner gegenwärtigen Form von den Ergebnissen der Delaunay-Triangulierung. Durch die Nutzung der als Verbindungsdreiecke klassifizierten Dreiecke der Triangulierung führt die Abwesenheit solcher Verbindungsdreiecke in Regionen, welche als Verbindung mehrerer Striche angesehen werden sollten, zu einer Untersegmentierung und einer Verbindung zwischen Strichen, welche nicht verbunden sein sollten. Dieses Problem tritt auf, wenn die in Abschnitt 3.4.4 beschriebenen Bedingungen für die Entstehung eines Verbindungsdreiecks nicht erfüllt sind. Abbildung 5.5 zeigt die Auswirkungen fehlender Verbindungsdreiecke auf die Strichextraktion.

Abhängigkeit der Strichextraktion von der Junction-Verschmelzung

Zusätzlich hat sich die Verschmelzung von Junctions als ein Punkt erwiesen, welcher durch Weiterentwicklung die Qualität der Strichextraktionsergebnisse zu steigern verspricht. Da für die Verschmelzung zweier Junctions zu einer neuen die Länge des sie verbindenden Teilstriches als entscheidendes Kriterium geprüft wird, ist die Wahl eines passenden Schwellenwertes für die Strichlänge ausschlaggebend für die erzielten Ergebnisse. Wird dieser Schwellenwert jedoch zu niedrig gewählt, werden zwei Junctions, welche eigentlich verschmolzen werden sollten, eventuell nicht verschmolzen. Abbildung 5.6 zeigt die Auswirkung einer fehlerhaften Junctionverschmelzung auf die Strichextraktion. Der grün markierte Strich in 5.6a ist 15 Pixel lang, während für die durchschnittliche Strichdicke des Zeichens ein Wert von 19,6 Pixeln berechnet wurde. Wird der den Schwellenwert für die Junction bestimmende Faktor auf den Standardwert 0,5 gesetzt, ergibt sich eine maximale Länge $L_T = \frac{1}{2} * 19 = 9,5$ eines Verbindungsstriches, bei der zwei angrenzende Junctions verschmolzen werden. Da für die Länge des Verbindungsstriches L_S jedoch gilt $L_S = 15 > L_T = 9,5$, werden die beiden Junctions nicht verschmolzen. Aus diesem Grund werden aus den an beide Junctions angeschlossenen Teilstrichen keine Kandidaten für eine gemeinsame Verschmelzung.

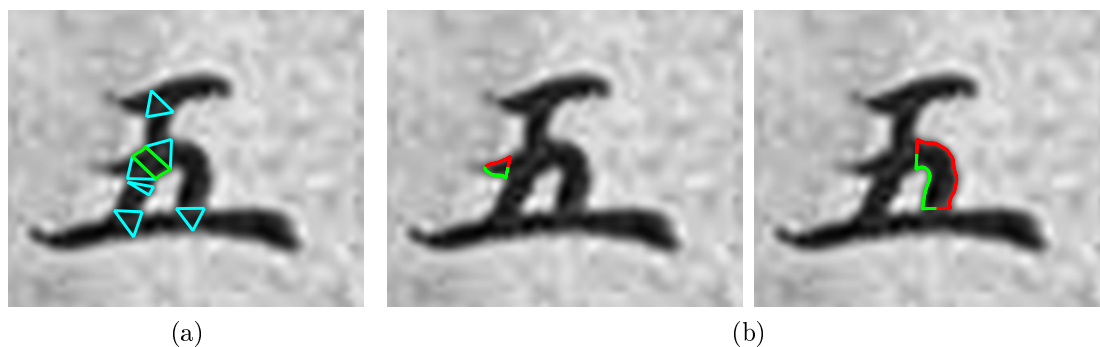


Abbildung 5.6: Fehlerhafte Strichextraktion wegen Abhängigkeit von Strichlängenschwellenwert.

Abbildung 5.7 zeigt, dass das Problem für ausgewählte Zeichen mit einer Anpassung des Strichlängenschwellenwertes für die Verschmelzung zweier Junctions gelöst werden kann. Im dargestellten Fall wurde der Faktor für die Berechnung des Schwellenwertes auf 1 gesetzt, wodurch sich ein Strichlängenschwellenwert $L_T = 1 * 19 = 19$ ergibt. Da der fragliche Verbindungsstrich eine Länge von $L_S = 15$ hat, werden die beiden Junctions zu einer neuen verschmolzen und die mit dieser verbundenen Teilstriche werden somit zu Kandidaten für die Teilstrichverschmelzung, die letztendlich zur Erzeugung des Striches in Abbildung 5.7b führt. Da eine individuelle Anpassung des den Schwellenwert berechnenden Faktors jedoch die Automatisierung des vorgestellten Verfahrens einschränkt, ist es in diesem Bereich wünschenswert, eine Erweiterung der Bedingungen zur Verschmelzung zweier Teilstriche zum Beispiel über die Einbeziehung weiteren Kontextes statt nur der an einer Verbindungsstelle lokal auftretenden Teilstriche zu erreichen. Eine Möglichkeit dazu könnte beispielsweise die Miteinbeziehung von Strichen in der näheren Umgebung der aktuell betrachteten Junction sein.

Die Struktur des Verfahrens müsste dabei so angepasst werden, dass auch Teilstriche, die nicht direkt sondern über einen Pfad von mehreren dazwischen liegenden Junctions und anderen Teilstrichen verbunden sind, auf ihre Verschmelzbarkeit verglichen und gegebenenfalls verschmolzen werden können.

5.2 Fazit

Das vorgestellte Verfahren zur Extraktion von Strichen aus handschriftlichen chinesischen Schriftzeichen hat in den durchgeführten Experimenten gute Ergebnisse erzielt und 72,48 Prozent der in den Schriftzeichen enthaltenen Grundstriche korrekt extrahiert. Die Ergebnisse haben jedoch auch Spielraum für Verbesserungen des Extraktionsverfahrens wie durch Einbeziehung von Kontext bei der Teilstrichverschmelzung oder die zusätzliche Berücksichtigung von reinen Krümmungsinformationen der Kontur bei der Extraktion initialer Teilstriche aufgezeigt. Desweiteren ist das Verfahren stark von der

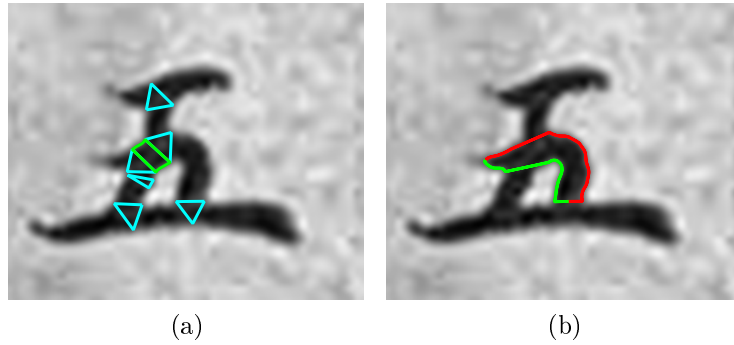


Abbildung 5.7: Anpassung des Strichlängenschwellenwert zur Verschmelzung von Junctions kann das Ergebnis verbessern.

Qualität des Ausgangsmaterials abhängig. Auch die Ausprägung einzelner (Teil-)Striche hat großen Einfluss auf die Ergebnisse. Die Strichextraktion arbeitet desweiteren bei einfacheren Zeichen zuverlässiger als bei komplexen Zeichen, deren einzelne Striche in handschriftlichen Quellen von minderer Qualität häufig nicht mehr zu identifizieren sind.

Insgesamt liefert das vorgestellte Verfahren einen vielversprechenden Ansatz, um Striche aus handschriftlichen chinesischen Schriftzeichen zu extrahieren. Die Verwendung der Delaunay-Triangulierung als Ausgangspunkt für die Strichextraktion macht das Verfahren im Vergleich zu konturbasierten Ansätzen, die ausschließlich die Informationen über die Konturkrümmung zur Extraktion von Strichen verwenden, stabiler bei der Verarbeitung handschriftlicher Schriftzeichen. Deren Krümmungsinformationen sind in Strichkreuzungsregionen schwieriger von denen entlang einer im Idealfall ungekrümmten Strichkontur zu unterscheiden als bei gedruckten Zeichen.

Da für die Identifikation von Schreibern in chinesischen Manuskripten keine vollständige Erkennung aller in einem Schriftzeichen theoretisch enthaltenen Striche notwendig ist, sondern die Analyse der erkennbaren Striche und die Berechnung ihrer Eigenschaften im Vordergrund steht, ist eine vollständige Erkennung aller in einem Zeichen vorkommenden Grundstriche nicht notwendig. Auch das Fehlen von Strichen in Schriftzeichen kann als Indiz für die Schreiberidentifikation verwendet werden.

Literaturverzeichnis

- [BE95] Marshall Wayne Bern and David Eppstein. Mesh generation and optimal triangulation. In Ding-Zhu Du and Frank Kwang-Ming Hwang, editors, *Computing in Euclidean Geometry*, number 4 in Lecture Notes Series on Computing, pages 47–123. World Scientific, second edition, 1995.
- [Ber97] Eckhard Berkenbusch. *Praktisches Chinesisch 1. Übungsbuch der chinesischen Schriftzeichen*. China Book Trading GmbH, 1997.
- [Boy07] Milan Sonka; Vaclav Hlavac; Roger Boyle. *ISE Image Processing, Analysis and Machine Vision*. Nelson Engineering, 2007.
- [CT00] Ruini Cao and Chew Lim Tan. A model of stroke extraction from chinese character images. *Pattern Recognition, International Conference on*, 4:4368, 2000.
- [Faz91] Edoardo Fazzioli. *Gemalte Wörter, 214 chinesische Schriftzeichen - vom Bild zum Begriff*. Fourier Verlag, 1991.
- [For86] S Fortune. A sweepline algorithm for voronoi diagrams. In *SCG '86: Proceedings of the second annual symposium on Computational geometry*, pages 313–322, New York, NY, USA, 1986. ACM.
- [GS83] Leo J. Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. In *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 221–234, New York, NY, USA, 1983. ACM.
- [Han93] Chad Hansen. Chinese ideographs and western ideas. *The Journal of Asian Studies*, 52(2):373–399, 1993.
- [Her07] Rainer Herzog. Analyse historischer chinesischer manuskripte mit hilfe des shape context verfahrens. Diplomarbeit, Universität Hamburg, 2007.
- [HY00] Rong He and Hong Yan. Stroke extraction as pre-processing step to improve thinning results of chinese characters. *Pattern Recogn. Lett.*, 21(8), 2000.
- [LHS97] Ke Liu, Yea S. Huang, and Ching Y. Suen. Robust stroke segmentation method for handwritten chinese character recognition. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, 1997.

- [Mei08] Hans Meine. *The GeoMap Representation: On Topologically Correct Sub-pixel Image Analysis*. PhD thesis, Universität Hamburg, 2008.
- [Pra97] Lakshman Prasad. Morphological analysis of shapes. *CNLS Newsletter*, 139:1–18, 1997.
- [Ric04] Matthias Richter. Tentative criteria for discerning individual hands in the guodian manuscripts. *Rethinking Confucianism: Selected Papers from the Third International Conference on Excavated Chinese Manuscripts*, 2004.
- [Rup95] Jim Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18(3):548–585, 1995.
- [Sch90] Wolfgang G.A. Schmidt. *Einführung in die chinesische Schrift- und Zeichenkunde*. Bushke, 1990.
- [SDG03] Daming Shi, Robert I. Damper, and Steve R. Gunn. Offline handwritten chinese character recognition by radical decomposition. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(1):27–48, 2003.
- [SDRI09] T. Steinherz, D. Doermann, E. Rivlin, and N. Intrator. Off-Line Loop Investigation for Handwriting Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):193–209, February 2009.
- [SH75] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *SFCS '75: Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, pages 151–162, Washington, DC, USA, 1975. IEEE Computer Society.
- [She96] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.
- [SYB07] Sargur N. Srihari, Xuanshen Yang, and Gregory R. Ball. Offline chinese handwriting recognition: A survey. In *Frontiers of Computer Science in*, 2007.
- [WH00] Wu Wei-Hsien. *Off-Line Chinese Character Recognition Based on Stroke Features*. PhD thesis, National Central University Taiwan, 2000.
- [WZ89] P.S.P. Wang and Y.Y. Zhang. A fast and flexible thinning algorithm. *IEEE Transactions on Computers*, 38(5):741–745, 1989.