

Thomas Werschlein

Frequenzbasierte  
Linienrepräsentationen für die  
kartographische Generalisierung

Diplomarbeit, ausgeführt am  
Geographischen Institut der Universität Zürich  
unter der Leitung von  
Prof. Dr. R. Weibel

Zürich 1996



# Vorwort

Die Römer haben keine Mathematik entwickelt, heisst es, weil sie ihre Zahlen falsch repräsentiert hätten. Aber was heisst schon falsch. Seit ich diese Arbeit geschrieben habe, weiss ich nun auch, dass eine Uhr, die stillsteht, mindestens so nützlich ist, wie eine, die um fünf Minuten vorgeht. Erstere zeigt nämlich zweimal am Tag die genaue Uhrzeit, während letztere immer die falsche Zeit anzeigt. Welches ist nun die bessere Approximation? Davon handelt diese Arbeit, von Approximationen und Repräsentationen. Oder mit anderen Worten: es kommt darauf an, unter welchem Blickwinkel etwas betrachtet wird.

Aber wie man es auch dreht und wendet: Es dürfte klar sein, dass diese Arbeit nicht ohne die Hilfe und Unterstützung anderer entstanden ist. Diesen Personen möchte ich an dieser Stelle meinen Dank aussprechen. Prof. Robert Weibel hatte stets ein offenes Ohr für meine Anliegen, waren sie nun konzeptioneller, inhaltlicher oder technischer Natur. Insbesondere für die sorgfältige Durchsicht und Korrektur des Manuskripts gebührt ihm mein Dank. Sehr fruchtbar waren auch die Diskussionen mit Andrej Vckovski, welche den Inhalt dieser Arbeit stark geprägt haben. Seine Unterstützung bei der Implementierung und seine Kompetenz in mathematischen Belangen waren schlicht unverzichtbar. Zusammen mit Joachim Hagger und Luzi Bernhard machte er sich auch als Linux-Guru unentbehrlich.

Mein Dank geht auch ans *Institut Géographique National (IGN)* in Paris, welches freundlicherweise die Liniendaten, auf welchen die Experimente in dieser Arbeit beruhen, zur Verfügung gestellt hat. Insbesondere die Zusammenarbeit mit Emmanuel Fritsch vom IGN im Rahmen seines Studienaufenthalts am GIUZ hat dieser Arbeit wesentliche Impulse verliehen. So stammt die Idee, kartographische Linien in Fourierreihen zu entwickeln, ursprünglich von ihm.

Über den Abschluss dieser Arbeit werden sich meine Eltern vermutlich ebenso freuen wie ich, nimmt doch damit die grosszügige finanzielle Unterstützung, welche ich über Jahre hinweg genoss, ein Ende. Ein Dankeschön geht schliesslich auch an meine Grossmutter, Maria Schächli, für die Finanzierung meines ersten Computers, zu einer Zeit notabene, als Hardware noch teuer bezahlt werden musste (vor allem jene mit dem Apfel-Logo).

Zu guter Letzt möchte ich all jenen Freundinnen und Freunden, Studienkolleginnen und -kollegen danken, welche ich hier nicht alle namentlich aufgeführt habe, von denen aber jeder und jede in irgendeiner Form ebenfalls zur Entstehung dieser Arbeit beigetragen hat. Ich habe mich hier aus Platzgründen für eine kompakte Repräsentation entschieden.

Thomas Werschlein, Januar MVMII



# Zusammenfassung

Die meisten Algorithmen zur kartographischen Liniengeneralisierung, seien es nun Vereinfachungs- oder Glättungsalgorithmen, werden auf Polylinien im Ortsraum ( $x, y$ -Ebene) angewandt. Diese Linienrepräsentation weist jedoch verschiedene Nachteile auf: Formmasse lassen sich weder direkt ablesen noch einfach berechnen, die Repräsentation ist nicht invariant gegenüber affinen Transformationen (Rotation, Translation, gleichförmige Skalierung) und es handelt sich im allgemeinen nicht um eine Funktion  $y = f(x)$ , was die mathematische Behandlung der Kurve erschwert. Insbesondere enthält eine Polyliniendarstellung keinerlei Information über die Komponenten, aus denen eine Kurve aufgebaut ist. Eine Zerlegung einer Linie in klar definierte “Bausteine” verschiedener Grösse wäre für die Generalisierung jedoch von Vorteil.

Die “Linienbausätze” welche in dieser Arbeit näher untersucht werden, sind einerseits bestückt mit Cosinusfunktionen verschiedener Frequenz (Fourierreihen), andererseits mit kleinen Wellenfunktionen verschiedener Länge (Wavelets). Schritt für Schritt wird aufgezeigt, wie von einer Polyliniendarstellung zu einer frequenzbasierten Linienrepräsentation gelangt werden kann. Nach der Abtastung und Normalisierung der Linie wird sie parametrisiert, um eine Funktion zu erhalten. Zum Zug kam eine auf dem Richtungswinkel basierende Parametrisierung. Es zeigte sich, dass ein sog. *Psi-s Plot* für unsere Zwecke aufgrund möglicher resultierender Diskontinuitäten unbrauchbar war. Stattdessen wurde der *t-alpha Plot* vorgeschlagen, dessen lokale Extrema den Wendepunkten der Linie im Ortsraum entsprechen. Eine *t-alpha* Repräsentation lässt sich sowohl in einer Fourierreihe entwickeln als auch einer Wavelettransformation (WT) unterziehen. Der Vorteil der letzteren besteht darin, dass sie *lokalisiert* ist, d.h. ein bestimmter Koeffizient beeinflusst lediglich einen genau bestimmten Bereich der Originalkurve. Das Degradationsverhalten von Fourier- und Wavelettransformation wurde anhand von Beispielen untersucht. Zu diesem Zweck wurde eine interaktive Software-Plattform entwickelt, welche die explorative Datenanalyse erst ermöglichte.

Es zeigte sich, dass die rekursive *t-alpha* Parametrisierung, welche lediglich auf die Richtungswinkel der einzelnen Liniensegmente abstellt, Lini verkürzungen nicht Rechnung tragen kann. Lini verkürzungen sind aber ein typisches Merkmal von Generalisierungslösungen. Sofern die diesbezüglichen Limitationen durch Einbezug zusätzlicher Lageinformation überwunden werden können, stellt insbesondere die Wavelettransformation eine für die Generalisierung sehr vielversprechende Repräsentation dar. Da es sich bei der WT um eine Mischung aus einer Mehrfachmassstabs-Repräsentation und einer klassischen spektralen Transformation handelt, ist deren Informationsdichte sehr hoch. So lassen sich aus der WT eines *t-alpha Plots* nach wie vor unmittelbar die Wendestellen bestimmen. Gleichzeitig enthält die Repräsentation aber auch Information über das Verhalten der Kurve in verschiedenen Massstäben. Es wird deshalb erwartet, dass sich die WT sehr gut als Grundlage für die Entwicklung lokal wirksamer Algorithmen eignet.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Zielsetzung</b>	<b>1</b>
1.1	Einleitung . . . . .	1
1.2	Problemstellung . . . . .	4
1.3	Zielsetzungen . . . . .	5
1.4	Begriffe . . . . .	6
<b>2</b>	<b>Linienrepräsentationen</b>	<b>7</b>
2.1	Zum Begriff der Linienrepräsentation . . . . .	7
2.2	Anforderungen an eine Linienrepräsentation . . . . .	9
2.3	Arten von Repräsentationen . . . . .	10
2.3.1	Im Ortsraum lokalisierte Repräsentationen . . . . .	11
2.3.2	<i>Scale-Space Images</i> . . . . .	12
<b>3</b>	<b>Abtastung</b>	<b>15</b>
3.1	<i>Walking Dividers</i> . . . . .	16
3.2	Approximationsqualität . . . . .	17
3.2.1	Flächenhafte Fehler . . . . .	18
3.2.2	Linienverkürzung . . . . .	19
3.3	Wahl der Schrittweite . . . . .	19
<b>4</b>	<b>Normalisierung</b>	<b>23</b>
4.1	Skalierungsinvarianz . . . . .	23
4.2	Translationsinvarianz . . . . .	23
4.3	Rotationsinvarianz . . . . .	25
4.4	Rücktransformation . . . . .	26
<b>5</b>	<b>Parametrisierung</b>	<b>27</b>
5.1	Der Parameter $t$ . . . . .	27
5.2	Parameterdarstellung der $x$ - und $y$ -Koordinaten . . . . .	28
5.3	<i>Psi-s Plot</i> . . . . .	29
5.3.1	Anwendung des <i>Psi-s Plots</i> auf diskret abgetastete Linien . . . . .	29
5.3.2	Vorteile des <i>Psi-s Plots</i> . . . . .	31
5.3.3	Nachteile des <i>Psi-S Plots</i> . . . . .	33
5.4	Die kumulativ-relative Repräsentation $t$ - $\alpha$ . . . . .	34
5.4.1	Anlehnung an Kurvatorelemente des <i>Chain Code</i> . . . . .	34
5.4.2	Herleitung der $t$ - $\alpha$ Repräsentation . . . . .	37
5.4.3	Rücktransformation in die $x, y$ -Ebene . . . . .	39

5.4.4	Wahl der Abtastschrittweite . . . . .	46
<b>6</b>	<b>Fourierreihen</b>	<b>49</b>
6.1	Konzept der Fourierreihe . . . . .	49
6.1.1	Die Behauptung . . . . .	49
6.1.2	Ein Beispiel . . . . .	50
6.1.3	Die Formeln . . . . .	53
6.1.4	Berücksichtigung von Symmetrieeigenschaften . . . . .	53
6.1.5	Nicht-periodische Funktionen . . . . .	56
6.1.6	Zusammenfassung . . . . .	58
6.2	Fourierreihen parametrisierter Linien . . . . .	59
<b>7</b>	<b>Wavelettransformation</b>	<b>73</b>
7.1	Konzept der Wavelettransformation . . . . .	73
7.1.1	Orthonormale Haar-Waveletbasis . . . . .	75
7.2	Wavelettransformation parametrisierter Linien . . . . .	78
7.2.1	Berechnung der Verschiebungskonstante $c$ . . . . .	79
7.2.2	Bestimmung der Anzahl Koeffizienten . . . . .	82
7.2.3	Inverse Wavelettransformation . . . . .	84
<b>8</b>	<b>Untersuchung von Linienpaaren</b>	<b>91</b>
<b>9</b>	<b>Schlussfolgerungen und Ausblick</b>	<b>99</b>
9.1	Was wurde erreicht? . . . . .	99
9.2	Ausblick auf zukünftige Entwicklungen . . . . .	100
	<b>Literaturverzeichnis</b>	<b>103</b>
<b>A</b>	<b>Die Softwareplattform <i>REPTILE</i></b>	<b>107</b>
<b>B</b>	<b>Daten</b>	<b>111</b>
B.1	Datenquelle . . . . .	111
B.2	Linienpaare . . . . .	111

# Abbildungsverzeichnis

1.1	Beispiele für Transformationen in der kartographischen Generalisierung . . . . .	2
1.2	Beispiel für die Probleme des Douglas–Peucker-Algorithmus . . . . .	3
1.3	Prozeduraler Ansatz basierend auf einer $x, y$ -Repräsentation. . . . .	4
1.4	Regelbasierter Ansatz basierend auf einer Komponenten-Repräsentation. . . . .	5
2.1	Extraktion von Regeln aus manuell generalisierten Linienpaaren . . . . .	8
2.2	Repräsentation einer Kurve durch Splines . . . . .	12
2.3	<i>Scale space</i> und <i>Scale-Space Image</i> nach Witkin . . . . .	13
2.4	Evolvierte Umrisse der Schweiz und dazugehöriges <i>CSSI</i> . . . . .	13
3.1	Kurvilineare Distanz . . . . .	15
3.2	Die <i>Walking-Dividers</i> -Abtastmethode . . . . .	16
3.3	Abtastung der Linie <i>587ungen</i> mit variierender Schrittweite . . . . .	17
3.4	Wahl der Schrittweite $s$ in Abhängigkeit der Auflösung $r$ . . . . .	19
3.5	Eliminierung von Kleinstformen mittels Sampling . . . . .	21
3.6	Die Linie <i>596ungen</i> in der $x, y$ -Ebene . . . . .	21
4.1	Normalisierung der Linie <i>501ungen</i> . . . . .	24
4.2	Berechnung des Richtungswinkels $\gamma$ . . . . .	25
5.1	Parametrisierung der $x$ - und $y$ -Koordinaten nach $t$ . . . . .	28
5.2	Nach $t$ parametrisierte Koordinaten der Linie <i>501ungen</i> . . . . .	29
5.3	Einfacher Freeman <i>Chain Code</i> . . . . .	30
5.4	Teil des <i>Hay River</i> und dazugehöriger <i>Psi-s Plot</i> . . . . .	30
5.5	Korrekte und inkorrekte Darstellung eines <i>Psi-s Plots</i> . . . . .	32
5.6	Korrekte und inkorrekte Integration eines <i>Psi-s Plots</i> . . . . .	32
5.7	Die Bedeutung linearer Abschnitte im <i>Psi-s Plot</i> . . . . .	33
5.8	Vollständiger <i>Psi-s Plot</i> des <i>Hay River</i> . . . . .	34
5.9	Die Diskontinuität im <i>Psi-s Plot</i> . . . . .	35
5.10	<i>Chain Code</i> , Krümmungselemente und kumulierte Krümmungselemente . . . . .	36
5.11	Darstellung der durch $\psi_i - \alpha_{i-1}$ aufgespannten Ebene . . . . .	38
5.12	Darstellung der durch $\gamma = \psi_i - \psi_{i-1}$ aufgespannten Ebene . . . . .	38
5.13	Transformation führt zum korrekten Wertebereich und eliminiert Sprungstelle . . . . .	39
5.14	Darstellung der Transformation $(\psi_i - \alpha_{i-1} + 3\pi) \bmod 2\pi - \pi$ . . . . .	40
5.15	<i>Psi-s Plot</i> und <i>t-alpha Plot</i> eines Teils der Linie <i>501ungen</i> . . . . .	41
5.16	Spirale und deren <i>t-alpha Plot</i> . . . . .	41
5.17	Mit Winkeltoleranz $T = 0.6$ rekonstruierte Linie <i>501ungen</i> . . . . .	43
5.18	Vergrößerte Ausschnitte aus Abb. 5.17 . . . . .	44

5.19	Rekonstruktion einer Kurve aus einem <i>t-alpha Plot</i> . . . . .	44
5.20	Schwerpunktbasierte Rekonstruktion aus einem <i>t-alpha Plot</i> . . . . .	45
5.21	Fixpunktbasierte Rekonstruktion aus einem <i>t-alpha Plot</i> . . . . .	46
5.22	Gegenüberstellung von fixpunkt- und schwerpunktbasierter Rekonstruktion .	47
5.23	Einfache Erkennung von Digitalisierfehlern im <i>t-alpha Plot</i> . . . . .	47
6.1	Die Schwingungskomponenten der Fourierreihe . . . . .	51
6.2	Approximationsfunktion für einen Ausschnitt der Ursprungskurve . . . . .	51
6.3	Eine Reihe rechteckiger Pulse . . . . .	53
6.4	Darstellung der Summe der ersten $n$ Harmonischen . . . . .	54
6.5	Integration einer periodischen Funktion . . . . .	55
6.6	Periodische <i>full-range</i> und <i>half-range</i> Erweiterungen einer Funktion . . . . .	57
6.7	Fourierkoeffizienten der Linien <i>596ungen</i> und <i>587ungen</i> . . . . .	60
6.8	Fourierapproximation des <i>t-alpha Plots</i> der Linie <i>596ungen</i> (Var. 1) . . . . .	62
6.9	Fourierapproximation des <i>t-alpha Plots</i> der Linie <i>596ungen</i> (Var. 2) . . . . .	63
6.10	Fourierapproximation des <i>t-alpha Plots</i> der Linie <i>587ungen</i> (Var. 1) . . . . .	64
6.11	Fourierapproximation des <i>t-alpha Plots</i> der Linie <i>587ungen</i> (Var. 2) . . . . .	65
6.12	Rücktransformation in die $x, y$ -Ebene (Teil v. <i>596ungen</i> , Var. 1) . . . . .	66
6.13	Rücktransformation in die $x, y$ -Ebene (Teil v. <i>596ungen</i> , Var. 2) . . . . .	66
6.14	Rücktransformation in die $x, y$ -Ebene (Teil v. <i>587ungen</i> , Var. 1) . . . . .	67
6.15	Rücktransformation in die $x, y$ -Ebene (Teil v. <i>587ungen</i> , Var. 2) . . . . .	67
6.16	Rücktransformation in die $x, y$ -Ebene ( <i>596ungen</i> , Var. 1) . . . . .	68
6.17	Rücktransformation in die $x, y$ -Ebene ( <i>596ungen</i> , Var. 2) . . . . .	69
6.18	Rücktransformation in die $x, y$ -Ebene ( <i>587ungen</i> , Var. 1) . . . . .	70
6.19	Rücktransformation in die $x, y$ -Ebene ( <i>587ungen</i> , Var. 2) . . . . .	71
7.1	Eine "kleine" Funktion, aber kein Wavelet . . . . .	74
7.2	Verschiedene Basiswavelets . . . . .	75
7.3	Das Haar-Basiswavelet und Ableitungen davon . . . . .	76
7.4	Zwei Haar-Wavelets mit verschiedener Skalierung . . . . .	77
7.5	Das "schmalste" Wavelet in Abhängigkeit von $\Delta t$ . . . . .	80
7.6	Das "breiteste" Wavelet in Abhängigkeit von $t + 1$ . . . . .	80
7.7	Bestimmung von $n_{max}(m)$ . . . . .	81
7.8	Waveletkoeffizienten einer Beispiellinie . . . . .	83
7.9	Waveletapproximation des <i>t-alpha Plots</i> der Linie <i>587ungen</i> (Var. 1) . . . . .	86
7.10	Waveletapproximation des <i>t-alpha Plots</i> der Linie <i>587ungen</i> (Var. 2) . . . . .	87
7.11	Rücktransformation in die $x, y$ -Ebene (Teil v. <i>587ungen</i> , Var. 1) . . . . .	88
7.12	Rücktransformation in die $x, y$ -Ebene (Teil v. <i>587ungen</i> , Var. 2) . . . . .	88
7.13	Rücktransformation in die $x, y$ -Ebene ( <i>587ungen</i> , Var. 1) . . . . .	89
7.14	Rücktransformation in die $x, y$ -Ebene ( <i>587ungen</i> , Var. 2) . . . . .	90
8.1	<i>t-alpha Plots</i> des Linienpaares <i>501</i> . . . . .	92
8.2	Linear gedehnter <i>t-alpha Plot</i> . . . . .	93
8.3	Das Linienpaar <i>488</i> . . . . .	94
8.4	Fourierkoeffizienten des Linienpaares <i>488</i> . . . . .	95
8.5	Waveletkoeffizienten des Linienpaares <i>488</i> . . . . .	97
8.6	Linie <i>488ungen</i> nach dem Editieren von Waveletkoeffizienten . . . . .	98

A.1 Funktionsdiagramm von <i>REPTILE</i> . . . . .	108
A.2 Die <i>REPTILE</i> -Arbeitsumgebung . . . . .	109
B.1 Auswahl einiger Linienpaare . . . . .	112



# Tabellenverzeichnis

2.1	Erfüllte und unerfüllte Kriterien von lokalisierten Repräsentationen. . . . .	12
3.1	Maximal mögliche und beobachtete Approximationsfehler . . . . .	20
5.1	Anzahl Rekonstruktionspunkte in Abhängigkeit von der Winkeltoleranz . . .	42



# Kapitel 1

## Einleitung und Zielsetzung

### 1.1 Einleitung

In den späten Achtzigerjahren wurde die Erkenntnis gewonnen, dass das Fehlen geeigneter Methoden zur automatischen Generalisierung räumlicher Daten eines der grössten Hindernisse für den flexiblen Einsatz von *Geographischen Informationssystemen (GIS)* darstellt. Weitaus am häufigsten werden Resultate, welche durch die Analyse der in einem *GIS* gespeicherten Daten generiert wurden, mittels Karten kommuniziert. Ohne entsprechende Generalisierungsverfahren bleibt eine solche Kartierung allerdings grundsätzlich auf den Massstab beschränkt, in welchem die Grundlagendaten zur Integration in ein *GIS* digitalisiert wurden. Diese Situation ist unbefriedigend, da den mit einem *GIS* erzeugten Resultaten meist eine zentrale Bedeutung bei der Entscheidungsfindung zukommt. Dient eine Karte als Entscheidungsgrundlage, dann sollte sie die in bezug auf die Fragestellung zentralen Elemente deutlich hervorheben und andere, im Kontext weniger wichtige Elemente allenfalls unterdrücken. Neben einer solchen *inhaltlichen Generalisierung*, welche mehr vom Thema der Karte als vom Massstab der Zielkarte abhängt, ist auch eine *graphische Generalisierung* vonnöten, insbesondere wenn Massstabsreduktionen vorgenommen werden. Die beiden Komponenten machen zusammen die *kartographische Generalisierung* aus (SGK 1980). Eine übersichtliche Darstellung der zur Verfügung stehenden Generalisierungsoperatoren findet sich in Shea und McMaster (1989), welche in Abbildung 1.1 wiedergegeben ist. Um die Komplexität der Fragestellung zu reduzieren, befasst sich diese Arbeit nur mit Linienelementen, welche zudem die überwiegende Mehrheit der Elemente einer topographischen Karte ausmachen. Wir beschränken uns somit ausschliesslich auf Operatoren, welche

- auf isolierte Elemente anwendbar sind (keine Verdrängung, engl. *Displacement*),
- auf lineare Elemente (Flüsse, Strassen, etc.<sup>1</sup>) anwendbar sind und
- die Objektgeometrie verändern und nicht in erster Linie die Signatur, wie z.B. das Betonen (*Enhancement*).

Somit verbleiben von den ursprünglich zwölf Operatoren noch deren drei, nämlich Vereinfachung (*Simplification*), Glättung (*Smoothing*) und Karikatur (*Exaggeration*). Für je-

---

<sup>1</sup>Eigentlich handelt es sich meist um flächenhafte Elemente, welche in einem GIS jedoch als Linie (z.B. als Mittelachse einer Strasse) repräsentiert werden und durch die Wahl einer geeigneten Signatur (Linienstärke) eine flächenhafte Ausdehnung erhalten.

Spatial and Attribute Transformations (Generalization Operators)	Representation in the Original Map		Representation in the Generalized Map	
	At Scale of the Original Map		At 50% Scale	
<b>Simplification</b>				
<b>Smoothing</b>				
<b>Aggregation</b>				
<b>Amalgamation</b>				
<b>Merge</b>				
<b>Collapse</b>				
<b>Refinement</b>				
<b>Typification</b>				
<b>Exaggeration</b>				
<b>Enhancement</b>				
<b>Displacement</b>				
<b>Classification</b>	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	1-5, 6-10, 11-15, 16-20	Not Applicable	Not Applicable

Abbildung 1.1: Beispiele für Transformationen in der kartographischen Generalisierung (aus: Shea und McMaster 1989).

den Generalisierungsoperator existieren verschiedene Algorithmen<sup>2</sup>. Für die Linienvereinfachung wird beispielsweise der Douglas-Algorithmus (Douglas und Peucker 1973) weitaus am häufigsten eingesetzt. Daneben existieren aber andere Algorithmen, wie z.B. der Lang-Algorithmus (Lang 1969), der Reumann-Witkam-Algorithmus (Reumann und Witkam 1974), oder der Algorithmus von Visvalingam und Whyatt (1993). Einen breiten Überblick bieten McMaster und Shea (1992). Die meisten Algorithmen werden durch einen oder einige wenige Parameter kontrolliert. Häufig sind es globale Distanz- oder Winkeltoleranzen, welche den Vereinfachungs- oder Glättungsgrad bestimmen, was in vielen Fällen zu unbefriedigenden Resultaten führt (Abbildung 1.2).

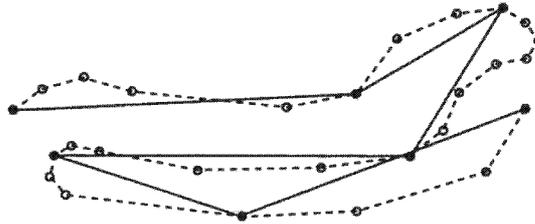


Abbildung 1.2: Beispiel für auftretende Probleme, wenn ein Douglas-Peucker-Algorithmus zur Linienvereinfachung eingesetzt wird. Die Ausgangskurve ist strichliert, die resultierende Linie (Kompression von 25%) ausgezogen dargestellt (aus: Fritsch und Lagrange 1995, 158).

Trotz der drastischen Vereinfachung der Problemstellung, welche insbesondere das Zusammenspiel verschiedener Kartenelemente ignoriert und stattdessen jede Linie isoliert betrachtet, bleibt die zu lösende Aufgabe nach wie vor reichlich komplex. Für die automatische Generalisierung stellen sich folgende Probleme:

1. Wahl der Generalisierungsoperatoren
2. Bestimmung der Reihenfolge der Operatoren (sog. *Sequencing*, vgl. dazu McMaster und Shea 1992, 71)
3. Wahl der Algorithmen
4. Wahl der Parameter

Um die Defizite dieses *prozeduralen* Ansatzes zu überwinden, wurden Mitte der 80er Jahre *regelbasierte* Ansätze propagiert. Mittels Expertensystemen wurde versucht, das kartographische Wissen, welches zur Beantwortung der obenstehenden Fragen 1-4 benötigt wird, in Generalisierungssysteme zu integrieren. Eine Zusammenfassung der Bemühungen in diesem Bereich liefert u.a. Reichenbacher (1995). Aber auch regelbasierte Generalisierungsansätze stellen zur geometrischen Umsetzung schlussendlich wieder auf algorithmische Lösungen ab. Solche wissensbasierte Methoden dienen bislang nur dazu, die vier oben angeführten Entscheidungen zu treffen. Zusätzlich stellt sich ein neues Problem: anders als z.B. in der medizinischen Diagnostik ist der Umfang an formalisiertem und strukturiertem Wissen im Bereich der kartographischen Generalisierung nur sehr gering. Zahlreiche neuere

<sup>2</sup>Um genau zu sein: zuerst entstanden die Algorithmen, erst dann die Klassifikation in Operatoren.

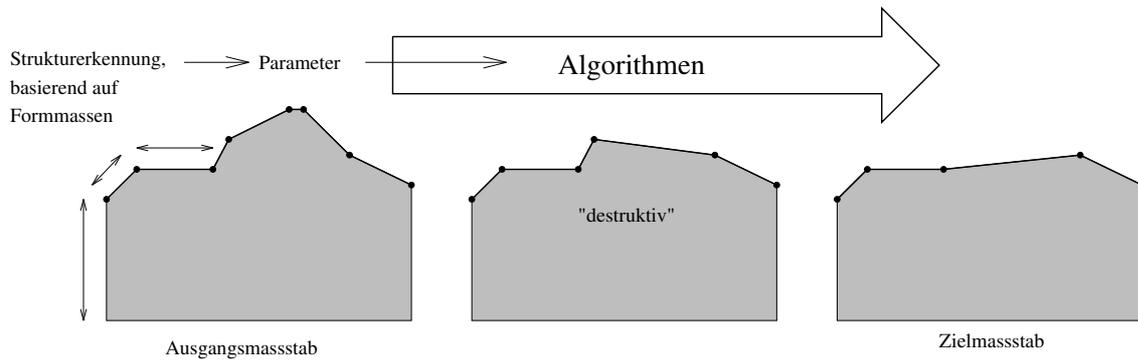


Abbildung 1.3: Prozeduraler Ansatz basierend auf einer  $x, y$ -Repräsentation.

Arbeiten befassen sich mit der Überwindung dieses *Knowledge Acquisition Bottleneck*, d.h. des “Flaschenhalses” bei der Integration von Know-How in ein digitales System (Weibel et al. 1995). Reichenbacher (1995) etwa beschreibt einen Ansatz mittels Interaktionsaufzeichnungen und induktivem Lernen.

## 1.2 Problemstellung

Zusammenfassend lassen sich im Bereich der digitalen Generalisierung im wesentlichen zwei hemmende Faktoren ausmachen:

1. die beschränkte Flexibilität bestehender Algorithmen
2. der *Knowledge Acquisition Bottleneck*

Beide Probleme – vor allem aber das erste – lassen sich bis zu einem gewissen Grad auf mangelnde Unterstützung durch die zugrundeliegende Linienrepräsentation zurückführen. Während algorithmische Lösungsansätze vielfach an der mangelnden Informationsdichte der Repräsentation scheitern, bekunden wissensbasierte Methoden Mühe mit der Unstrukturiertheit der Daten, welche eine eindeutige Abbildung von Abschnitten der Ausgangslinie auf die Ziellinie verunmöglichen. Wir postulieren nun, dass sich beide Probleme durch den Einsatz einer geeigneten Linienrepräsentation lindern lassen.

Die folgende Metapher soll zur Verdeutlichung des Lösungsansatzes beitragen: Man kann sich eine durch ihre  $x$ - und  $y$ -Koordinaten dargestellte Linie als Betonblock vorstellen, dessen Oberkante die eigentliche Linie repräsentiert. Dieser Sachverhalt ist in Abbildung 1.3 dargestellt. Soll diese Linie nun mit einem prozeduralen Ansatz generalisiert werden, müssen zuerst die Formmasse der Linie bestimmt werden. Im übertragenen Sinn bedeutet dies, dass der Betonblock vermasselt wird. Aufgrund dieser Größen werden die Parameter für den gewählten Generalisierungsalgorithmus bestimmt. Der Algorithmus muss nun, mangels anderer Information, von der Oberkante des Betonblocks ausgehen, und (im Falle der Linienvereinfachung) nicht signifikante Punkte “wegmeisseln”. Deshalb nennen wir diese Vorgehensweise auch “*destruktiv*”<sup>3</sup>. Dies ist die Situation, wenn Linien – wie meistens der Fall – durch Koordinatenketten (Polylinien) repräsentiert werden.

<sup>3</sup>Das Beispiel beschränkt sich der Einfachheit halber auf die Eliminierung einer konvexen Kleinform. Im allgemeinen Fall müsste allenfalls auch “aufbetoniert” werden.

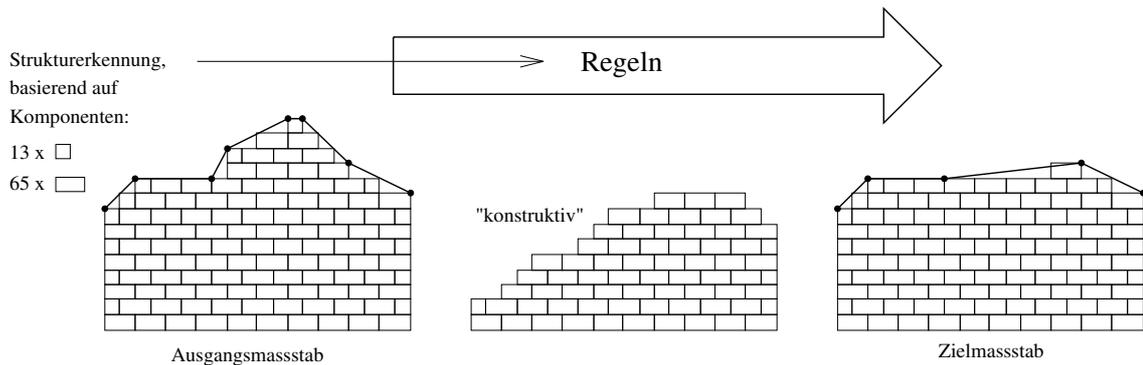


Abbildung 1.4: Regelbasierter Ansatz basierend auf einer Komponenten-Repräsentation.

Eine andere Möglichkeit wäre, sich eine "gemauerte" Linie vorzustellen, wie dies in Abbildung 1.4 illustriert ist. Anstelle des unstrukturierten Betonblocks treten Backsteine in zwei verschiedenen Grössen. Wir wissen nun, aus welchen *Komponenten* die Linie aufgebaut ist und nicht bloss, wie sie sich in der  $x, y$ -Ebene manifestiert. Sind wir nun überdies in der Lage, das Backsteinmuster zu deuten, können wir einen entsprechenden Bauplan (Regeln) zur Konstruktion einer generalisierten Version der Linie erstellen. Deshalb nennen wir diesen Ansatz auch "*konstruktiv*". Dabei liegt das Augenmerk nicht mehr auf der Modifikation der ursprünglichen Linie, sondern auf der Gestaltung eines eigenständigen, der Situation angepassten Linienelements. Die Identifizierung von "Normbauteilen" – in dieser Arbeit handelt es sich dabei um oszillierende Funktionen unterschiedlicher Bauart – würde nun allenfalls den Einsatz von künstlichen *Neuronalen Netzen (NN)* unterstützen. *NN* sind im Grunde nichts anderes, als fehlertolerante Klassifikatoren und werden deshalb vorwiegend für die Mustererkennung eingesetzt. Eine weitere Voraussetzung zum Einsatz von *NN* ist in der Kartographie ebenfalls gegeben: Linienpaare in der Form *Ausgangslinie*–(*generalisierte*) *Ziellinie* sind in grosser Fülle vorhanden und können grundsätzlich als Inputdaten für das Training von *NN* gebraucht werden.

Der Einsatz von *NN* in der Generalisierung wurde bereits von McMaster und Shea (1992) vorgeschlagen. Allerdings stellen diese Autoren das Finden von geeigneten Parametern für algorithmische Generalisierungslösungen in den Vordergrund. Ein anderes Beispiel für den Einsatz von *NN* in der Kartographie (nicht generalisierungsspezifisch) stellt die Arbeit von Meng (1993) dar.

### 1.3 Zielsetzungen

Ziel dieser Arbeit ist es, das Potential alternativer, *frequenzbasierter Linienrepräsentationen* (Fourierreihen, Wavelets) für die Liniengeneralisierung in theoretischer und experimenteller Hinsicht abzuklären. Zu diesem Zweck soll eine Software-Plattform entwickelt werden.

Ein zentrales Element dieser Plattform stellt die Verknüpfung der verschiedenen Repräsentationsebenen dar; wenn der Benutzer oder die Benutzerin in einer bestimmten (Zwischen-)Repräsentation einen Parameter verändert oder eine Operation ausführt, dann soll das Ergebnis dieser Manipulation durch alle Repräsentationsebenen hindurch zurück in die Ursprungsrepräsentation (Polylinie) propagiert werden. Die Auswirkungen einer Veränderung sollen also unmittelbar in der Kartenebene beurteilt werden können.

Daneben sollte die Software über einfache Editierfunktionen verfügen, um beispielsweise eine Linie manuell segmentieren zu können. Diese Anforderungen führten dazu, dass die zu entwickelnde Plattform den Arbeitstitel *REPTILE* – kurz für: “*RE*presentation *T*ool and *I*nteractive *L*ine-editing *E*nvironment” – erhielt.

Die betrachteten Linienrepräsentationen sollen mit *REPTILE* im Hinblick auf folgende Gesichtspunkte explorativ untersucht werden:

**Approximationsgenauigkeit:** Beurteilung der Abweichungen zwischen Originallinie und abgetasteter Linie einerseits, und der aus dem Frequenzraum rücktransformierten Linie andererseits.

**Robustheit:** Abschätzung des Degradationsverhaltens infolge Reduktion der Anzahl Fourier- und Waveletkoeffizienten während des Rekonstruktionsprozesses.

**Fehlerfortpflanzung:** Wie wirkt sich ein lokal auftretender Fehler bzw. ein bewusst vorgenommener Eingriff auf den Rest der rekonstruierten Linie aus?

Diese Experimente sollen nicht anhand *eines* Beispiels durchgeführt werden, sondern auf einem realen Liniendatensatz basieren. Dabei steht nicht die systematische Auswertung der Daten im Vordergrund. Vielmehr sollen Stichproben aus dieser Grundgesamtheit dazu dienen, auf Problemfälle aufmerksam zu machen und daraus Anforderungen an die Repräsentation zu formulieren.

## 1.4 Begriffe

### Englische Ausdrücke

Englische Ausdrücke, welche kein geläufiges deutsches Synonym aufweisen, werden *kursiv* gesetzt. Nach Möglichkeit werden englische Begriffe bei ihrem ersten Auftreten im Text ins Deutsche übersetzt.

### Nomenklatur

**Linie:** In dieser Arbeit steht der Begriff “Linie” durchwegs für die Mittelachse einer Strasse in der  $x, y$ -Ebene.

**Kurve:** Wird als Synonym für Linie verwendet. Wenn in dieser Arbeit von “Kurven” die Rede ist, sind damit immer planare Kurven gemeint.

**Funktion:** Eine Linie in der Parameterebene.

**Signal:** Wird als Synonym für Funktion gebraucht. Von Signal ist normalerweise die Rede, wenn die unabhängige Variable  $x$  als *Zeit* aufgefasst wird (und dann als  $t$  bezeichnet wird) im Gegensatz zur Betrachtung von  $x$  als *Ortsvariable*. Beide Sichtweisen sind austauschbar, wenn man sich vorstellt, dass man eine Kurve mit konstanter Geschwindigkeit abschreitet.

**Linienpaar:** Darunter wird immer eine Linie im Ursprungs- und die entsprechende Linie im Zielmasstab verstanden. In Anhang B wird näher auf die verwendeten Daten eingegangen.

# Kapitel 2

## Linienrepräsentationen

### 2.1 Zum Begriff der Linienrepräsentation

In Anlehnung an Foley et al. (1991, 471) können zwei Fälle unterschieden werden, in denen es nötig ist, Kurven zu repräsentieren:

1. *Modellierung existierender linearer Objekte:* In der Kartographie sind dies Flüsse, Strassen und Eisenbahnlinien, bzw. deren Mittelachsen, welche bei der Kartenerstellung, beispielsweise aufgrund eines Luftbildes, modelliert werden müssen.
2. *Modellierung neuer Objekte von Grund auf:* Kartographisch generalisierte Linien verfügen über kein Pendant in der realen Welt. Der Kartograph oder die Kartographin kreiert ein neues Kartenelement unter Berücksichtigung des Kontexts und unter Einbezug von kartographischem Wissen.

Im ersten Fall ist kaum eine mathematische Beschreibung des Objekts verfügbar<sup>1</sup>, weshalb sich als Modell natürlich die unendlich vielen Koordinaten des Objekts anbieten würden. Dieses Modell ist aber offensichtlich ungeeignet für einen Computer mit seinem endlichen Speicher. In der Praxis wird deshalb das Objekt meist aus mathematisch einfach zu beschreibenden Stücken approximiert (Geraden im Fall von Polylinien, kubische Polynome im Fall von Splines), so dass Punkte aus unserem Modell genügend nahe an entsprechende Punkte auf dem realen Objekt zu liegen kommen. Wir nennen diese Repräsentation fortan die *Kartierungs-Repräsentation*.

Im zweiten Fall kann die kartierende Person das Objekt im Modellierungsprozess erzeugen. Das Objekt ist somit deckungsgleich mit der Repräsentation, denn das Objekt manifestiert sich einzig durch seine Repräsentation. Bei der manuellen Generalisierung am Bildschirm wird sich diese *Generalisierungs-Repräsentation* kaum von der Kartierungs-Repräsentation unterscheiden. Der Grund liegt in der Einfachheit von stückweise approximierten Liniensegmenten, welche es der generalisierenden Person unmittelbar erlauben, ihre voreinst mental vereinfachte Linie in die Kartenebene umzusetzen.

Bislang wurden auch in der computergestützten, (semi-)automatischen Generalisierung Kartierungs- und Generalisierungs-Repräsentation als identisch betrachtet. Es ist jedoch zu vermuten, dass sich die ideale Repräsentation für die computergestützte Generalisierung von

---

<sup>1</sup>Es sei denn, es handle sich um geometrische, anthropogene Objekte wie z.B. einen Strassenkreisel.

der idealen Kartierungs-Repräsentation unterscheidet. Diese Erkenntnis wird in anderen Bereichen von *GIS* bereits seit langem umgesetzt. So werden Linien für topologische Analysen nicht als Polylinien, sondern als Knoten und Vektoren repräsentiert und für Ausbreitungsmodellierungen werden die Daten als Rasterzellen dargestellt. Laurini und Thompson (1992, 334) bestätigen diese Sicht der Dinge: "It is unlikely too that a single way explicitly to encode spatial phenomena can efficiently provide for all the diverse needs and uses of spatial databases." Wir gehen deshalb davon aus, dass sich die idealen Linienrepräsentationen für die Kartierung bzw. die Generalisierung unterscheiden. Dieses Modell ist in Abbildung 2.1 links dargestellt.

Ist jedoch beabsichtigt, kartographisches Wissen aus manuell generalisierten Linienpaaren zu extrahieren, dann genügt dieses Modell bereits nicht mehr. Es stellen sich zwei Probleme:

1. Es ist fraglich, ob es gelingt, repräsentationsübergreifende Regeln zu extrahieren.
2. Falls ja, müssten diese Regeln in der Kartierungs-Repräsentation angewandt werden, d.h. in einer für die wissensbasierte Generalisierung vermutlich suboptimalen Repräsentation.

Abhilfe schafft die Konvertierung der Ursprungslinie in die Generalisierungs-Repräsentation, wie in Abbildung 2.1 rechts dargestellt. Die Regeln können innerhalb der Generalisierungs-Repräsentation extrahiert werden und auch in dieser Repräsentation angewandt werden. Erst dann wird das Resultat wieder in die Kartierungs-Repräsentation konvertiert. Dieses Modell erfordert jedoch, dass die Generalisierungs-Repräsentation invertierbar ist (im Gegensatz zum einfacheren Modell, wo sich die Generalisierung immer in der Kartierungs-Repräsentation abspielt).

Da wir geeignete Linienrepräsentationen für die Generalisierung auch im Hinblick auf den Einsatz in wissensbasierten Systemen (wie z.B. Neuronalen Netzen) suchen, nehmen

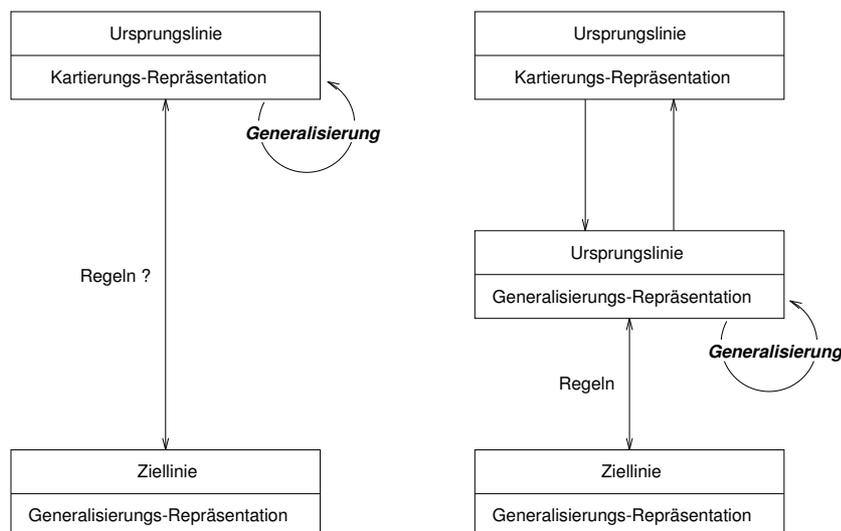


Abbildung 2.1: Extraktion von Regeln aus manuell generalisierten Linienpaaren. Links ohne Konvertierung der Ursprungslinie in die Generalisierungs-Repräsentation, rechts mit entsprechender Konvertierung.

wir letzteres Modell als Grundlage. Neben der Invertierbarkeit stellen sich aber noch andere Anforderungen an eine Linienrepräsentation. Darauf wird im nächsten Abschnitt eingegangen.

## 2.2 Anforderungen an eine Linienrepräsentation

Eine brauchbare Linienrepräsentation für die kartographische Generalisierung muss notwendigerweise verschiedene Kriterien erfüllen. Ein Grossteil dieser Kriterien deckt sich mit solchen, wie sie aus dem Bereich der *Computer Vision*<sup>2</sup> bekannt sind. Mokhtarian und Mackworth (1992) stellen folgende Anforderungen an eine in der Praxis einsetzbare Repräsentation für planare Kurven. Wenn dabei von Kurven die Rede ist, welche *dieselbe Form* aufweisen, dann bedeutet dies, dass eine Transformation, bestehend aus gleichförmiger Skalierung, Rotation und Translation, existiert, welche zur Deckungsgleichheit der beiden Kurven führt.

**Invarianz (*invariance*):** Wenn zwei Kurven dieselbe Form aufweisen, dann sollten sie auch dieselbe Beschreibung<sup>3</sup> aufweisen.

**Einmaligkeit (*uniqueness*):** Wenn zwei Kurven nicht dieselbe Form aufweisen, dann sollten sich auch ihre Beschreibungen unterscheiden.

**Stabilität (*stability*):** Wenn sich die Form zweier Kurven nur geringfügig unterscheidet, dann sollten sich auch ihre Beschreibungen nur geringfügig unterscheiden. Die Umkehrung dieser Bedingung muss ebenso erfüllt sein. Daraus folgt, dass, wenn sich zwei Beschreibungen ähnlich sind, auch die Kurven, welche sie repräsentieren, ähnlich sind.

**Effizienz (*efficiency*):** Die Repräsentation sollte einfach zu berechnen und zu speichern sein. Dies ist insbesondere dann von Bedeutung, wenn die Repräsentation in Echtzeit-Systemen (z.B. zur Generalisierung) eingesetzt werden soll.

**Einfach zu implementieren (*ease of implementation*):** Wenn zwei oder mehrere gleichwertige Repräsentationen zur Auswahl stehen, dann soll jene gewählt werden, welche am wenigsten Zeit für die Implementierung in Anspruch nimmt.

**Berechnung von Formmassen (*computation of shape properties*):** Es kann von Nutzen sein, wenn Formeigenschaften einer Kurve aufgrund ihrer Beschreibung bestimmt werden können. Wenn eine Kurve z.B. eine symmetrische Form aufweist, wäre es wünschenswert, wenn dieser Umstand aus der Beschreibung ersichtlich wäre (das *Symmetrie*-Kriterium). Es ist ebenfalls zu begrüssen, wenn aus lokalen Extrema der Beschreibung auf Formeigenschaften der Kurve geschlossen werden kann (z.B. auf Wendepunkte).

---

<sup>2</sup>Diese Forschungsrichtung beschäftigt sich in erster Linie mit der Identifizierung von Objekten innerhalb von digitalen Bildszenen.

<sup>3</sup>Die Autoren benutzen in der Aufzählung jeweils den Begriff *Repräsentation*, welcher in diesem Zusammenhang jedoch unangebracht ist. Eine *Repräsentation* ist ein Satz von Konventionen zur Beschreibung einer *Klasse* von Dingen. Eine *Beschreibung* benutzt diese Konventionen zur Charakterisierung *eines bestimmten* Elements aus dieser Klasse (Winston 1984).

Ein Kriterium, welches Mokhtarian und Mackworth (1992) nicht erwähnen (und welches die von ihnen vorgeschlagene Repräsentation auch nicht erfüllt, vgl. 2.3.2), welches aber für eine Generalisierungs-Repräsentation von eminenter Wichtigkeit<sup>4</sup> ist, ist die *Invertierbarkeit*. Dies bedeutet, dass die Kurve aufgrund ihrer Beschreibung rekonstruierbar sein muss. In der *Computer Vision* ist dieses Kriterium nicht von Bedeutung, da es in erster Linie darum geht, Objekte zu klassieren. Die bloße Zuordnung einer Kurve zu ihrer Beschreibung ist für diesen Zweck hinreichend.

Weitere Kriterien, insbesondere wenn die Repräsentation auch als Input-Darstellung für Neuronale Netze Verwendung finden soll, sind:

**Fixe Länge:** Der Umfang der Beschreibung sollte nicht von der Länge der Linie abhängen, sondern lediglich von deren Komplexität. Mit einer genügend grossen Anzahl Werte sollte somit jede Linie hinreichend genau approximiert werden können.

**Beschränkter Wertebereich:** Im Idealfall weisen die Werte der Beschreibung einen beschränkten Wertebereich auf. Ein bekannter Wertebereich erleichtert die Skalierung.

Im weiteren Verlauf dieser Arbeit wird immer wieder Bezug auf die hier genannten Kriterien genommen, um die Qualität einer Repräsentation abschätzen zu können. Auch die im nächsten Abschnitt kurz besprochenen Repräsentationen werden im Hinblick auf diese Kriterien bewertet.

## 2.3 Arten von Repräsentationen

Linienrepräsentationen lassen sich zunächst in zwei grundlegend verschiedene Klassen unterteilen:

**Lokalisierte Repräsentationen:** Einige ausgewählte Samplingpunkte in der  $x, y$ -Ebene definieren die Lokalisierung der Linie in der Kartenebene. Sämtliche Repräsentationen in der  $x, y$ -Ebene gehören in diese Klasse, sowohl diskrete (Raster) als auch stetige (Polylinien, Splines<sup>5</sup>). Lokalisierte Repräsentationen werden in Abschnitt 2.3.1 behandelt.

**Parametrische Repräsentationen:**  $x$ - und  $y$ -Koordinaten werden implizit oder explizit als Funktion eines Parameters  $t$ , meist die kurvilineare Distanz entlang der Kurve, ausgedrückt. *Chain-Codes* (Freeman 1974) und der *Psi-S-Plot* (Ballard und Brown 1982) sind Beispiele dafür. Parametrisierte Repräsentationen werden in Kapitel 5 ausführlich behandelt.

Lokalisierte Repräsentationen eignen sich schlecht als Grundlage für andere, höher entwickelte Linienrepräsentationen. Die Gründe dafür werden im nächsten Abschnitt erläutert. Diese Aussage trifft nicht auf parametrisierte Liniendarstellungen zu, welche als Grundlage für viele andere Repräsentationen herbeigezogen werden:

<sup>4</sup>Sofern sie auf dem Modell in Abbildung 2.1 rechts basiert.

<sup>5</sup>Streng genommen handelt es sich bei diesen beiden Repräsentationen um *stückweise* parametrische Darstellungen. Da aber nicht ein kontinuierlicher Parametersatz für die ganze Linie verwendet werden kann, wird sie als lokalisierte Repräsentation eingestuft.

**Scale-Space Images:** Dabei handelt es sich um eine qualitative Beschreibung einer parametrisierten Kurve. Ein Beispiel dafür ist das *Curvature Scale-Space Image (CSSI)* (Mokhtarian und Mackworth 1992), welches in Abschnitt 2.3.2 kurz beschrieben wird.

**Frequenzbasierte Repräsentationen:** Liniendarstellungen, welche auf den Frequenzgehalt einer Funktion abstellen. Im Falle von Fourierreihen handelt es sich um Sinus- und Cosinus-Kurven, im Falle von Wavelets um eine grössere Palette von oszillierenden Funktionen. Diese beiden spektralen Repräsentationen werden in den Kapiteln 6, bzw. 7 ausführlich behandelt.

### 2.3.1 Im Ortsraum lokalisierte Repräsentationen

Lokalisierte Repräsentationen unterscheiden sich hauptsächlich durch die Anzahl Stützpunkte die gebraucht werden, um die Linie mit einer hinreichend guten Approximationsgenauigkeit zu modellieren. Diese Anzahl hängt ihrerseits ab von der Ordnung der Polynome, welche zur Interpolation zwischen diesen Punkten zur Anwendung gelangen. Im einfachsten Fall der *Polylinie* sind dies Polynome erster Ordnung, also Geraden. Bei der Modellierung handelt es demnach um nichts anderes als um eine stückweise lineare Approximation der Kurve. Sobald die zu approximierende Kurve nicht stückweise linear ist, muss eine grosse Zahl von Koordinaten erfasst werden, um eine genügend gute Approximation zu erzielen. Das Editieren einer solchen Linie gestaltet sich dementsprechend mühsam, weil eine grosse Zahl von Punkten exakt umplaziert werden muss.

Kompakter und besser manipulierbar als Polylinien sind *Splines*. Die Approximation zwischen Punkten erfolgt dabei durch polynomiale Kurven höherer Ordnung. Für die meisten Anwendungen genügen kubische Polynome der Form

$$\begin{aligned}x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x, \\y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y, \quad 0 \leq t \leq 1.\end{aligned}$$

Die verschiedenen Spline-Arten unterscheiden sich durch die Randbedingungen, welche zur Auflösung des obigen Gleichungssystems beigezogen werden. Bei den am weitesten verbreiteten Splines, den *Bézier-Kurven*, werden diese durch die beiden Endpunkte des Segments und zwei weitere Punkte, welche die Tangenten in den Endpunkten bestimmen, geliefert. Da nur die Parameter der Gleichungen gespeichert werden müssen, ermöglichen Splines eine effiziente Datenhaltung. Dies aber auch nur, wenn die zu modellierende Kurve einen stetigen Verlauf aufweist. Zur Approximation administrativer Grenzen beispielsweise eignen sich Splines eher schlecht, da sie immer einen glatten Verlauf aufweisen (vgl. Abbildung 2.2).

Wie sich lokalisierte Repräsentationen in bezug auf die in Abschnitt 2.2 definierten Kriterien verhalten, ist in Tabelle 2.1 zusammengefasst.

Weil es sich um einen Spezialfall handelt, wenn eine planare Kurve durch eine explizite Funktion der Form  $y = f(x)$  beschrieben werden kann, eignen sich lokalisierte Linienrepräsentationen schlecht als Grundlage für andere Repräsentationen. Im allgemeinen handelt es sich nämlich bei einer kartographischen Kurve nicht um eine Funktion. Obwohl manchmal durch die Rotation des Koordinatensystems (Laurini und Thompson 1992, 146) eine Funktion erzielt werden kann, ist es im Normalfall schwierig, eine Linie in einer solchen Repräsentation mathematisch zu behandeln. Treten beispielsweise vertikale Liniensegmente auf, wird die Steigung (1. Ableitung) an diesem Punkt unendlich. Erst die Parametrisierung (vgl. Kapitel 5) transformiert eine lokalisierte Repräsentation in eine Funktion.

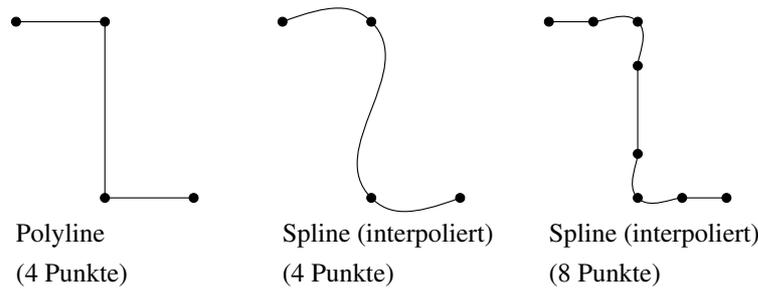


Abbildung 2.2: Splines eignen sich schlecht für die Repräsentation unstetiger Kurven.

Invarianz	nicht erfüllt
Einmaligkeit	erfüllt
Stabilität	erfüllt
Effizienz	erfüllt <sup>a</sup>
Einfach zu implementieren	erfüllt
Berechnung von Formmassen	nicht erfüllt
Invertierbarkeit	erfüllt
fixe Länge	nicht erfüllt
definierter Wertebereich	nicht erfüllt

<sup>a</sup>Die computationale Komplexität als Funktion der Länge der Kurve ist linear.

Tabelle 2.1: Erfüllte und unerfüllte Kriterien von lokalisierten Repräsentationen.

### 2.3.2 Scale-Space Images

*Multiscale Representations* (Repräsentationen in Mehrfachmassstäben) für eindimensionale *Funktionen* wurden erstmals von Stansfield (1980) vorgeschlagen und später durch Witkin (1983) entwickelt. Dabei wird ein Signal  $f(x)$  mit einer Gauss'schen Funktion gefaltet, was zu einer geglätteten Version des Signals führt<sup>6</sup>. Die Faltung hängt ab von  $x$ , der unabhängigen Variable des Signals, und von  $\sigma$ , der Standardabweichung der Gauss'schen Funktion. Die Faltung ist gegeben durch:

$$(2.1) \quad F(x, \sigma) = f(x) \circ g(x, \sigma) = \int_{-\infty}^{\infty} f(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}} du,$$

wobei “ $\circ$ ” die Faltung in bezug auf  $x$  bezeichnet. Diese Funktion definiert eine Oberfläche über der  $(x, \sigma)$ -Ebene. Bei jedem Profil dieser Oberfläche entlang einem konstanten  $\sigma$  handelt es sich um eine geglättete Version von  $f(x)$  (Abbildung 2.3 (a)). Witkin nennt die  $(x, \sigma)$ -Ebene *Scale Space* und die in Gleichung 2.1 definierte Funktion  $F$  das *Scale Space Image* von  $f$ . In einem solchen *Scale-Space Image* lassen sich nun Extrema des geglätteten Signals durch verschiedene Werte von  $\sigma$  verfolgen. Bei den Nullstellen der zweiten Ableitung ( $F_{xx} = 0$ ) handelt es sich um Steigungsextrema, welche die Lokalisierung der Wendepunkte der Funktion erlauben. In Abbildung 2.3 (b) sind die Pfade, welche Wendepunkte bei der zunehmenden Glättung des Signals zurücklegen, wiedergegeben. Die Schnittpunkte einer horizontalen Linie mit diesen Pfaden definieren die Wendepunkte des Signals für

<sup>6</sup>Mokhtarian und Mackworth (1992) nennen diesen Vorgang *Evolution* und sprechen deshalb auch von der *evolvierten* (entwickelten) Version einer Funktion.

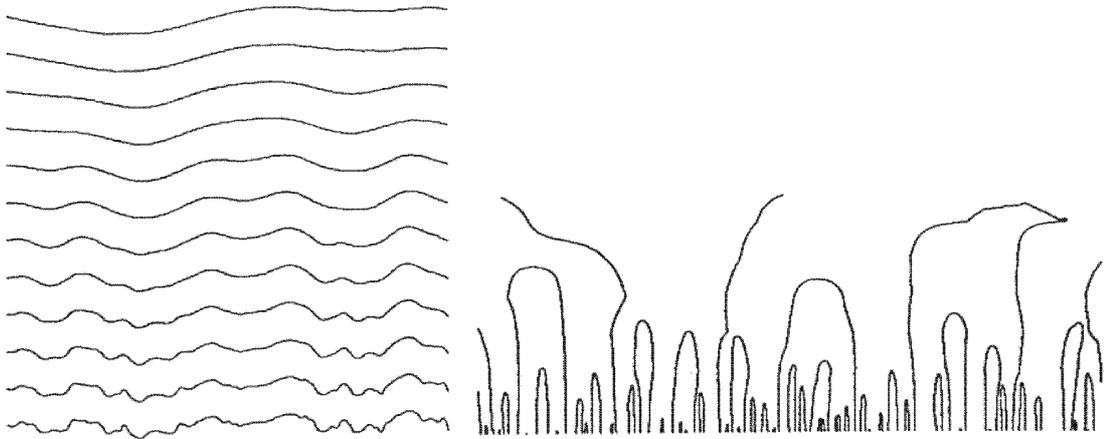


Abbildung 2.3: (a) Eine Sequenz von Gauss'schen Glättungen einer Funktion, mit von oben nach unten abnehmendem  $\sigma$ . Jeder dieser Graphen entspricht dem Querschnitt durch das *Scale-Space Image* entlang einem gegebenen (konstanten)  $\sigma$ -Wert.

(b) Äquilinien der Wendepunkte in einem *Scale-Space Image*. Die  $x$ -Achse verläuft horizontal;  $\sigma$  nimmt von unten nach oben zu (aus Witkin 1983).

ein bestimmtes  $\sigma$ . Durch das Verschieben dieser horizontalen Linie nach oben und unten können die entsprechenden Extrema für kleinere und grössere Werte von  $\sigma$  abgelesen werden. Mokhtarian und Mackworth (1992) dehnten das Konzept von *Scale-Space Images*, in der Version von Witkin lediglich auf Funktionen beschränkt, auf planare Kurven beliebiger Form aus (auch auf geschlossene). Die Autoren nennen ihre neue Repräsentation *Curvature Scale-Space Image (CSSI)*. Diese unterscheidet sich im wesentlichen dadurch, dass die zu repräsentierende Kurve zuerst parametrisiert wird (kurvilineare Parametrisierung, siehe Abschnitt 5.1), womit die für ein *Scale-Space Image* nötige Funktion erhalten wird. Ein weiterer Unterschied besteht in der Nomenklatur: Das *CSSI* von Mokhtarian und Mackworth (1992) beschreibt im Gegensatz zu Witkins *Scale Space-Image* keine Oberfläche, sondern gleich die *Pfade*, welche Wendepunkte während der Evolution einer Kurve beschreiben (Abbildung 2.4).

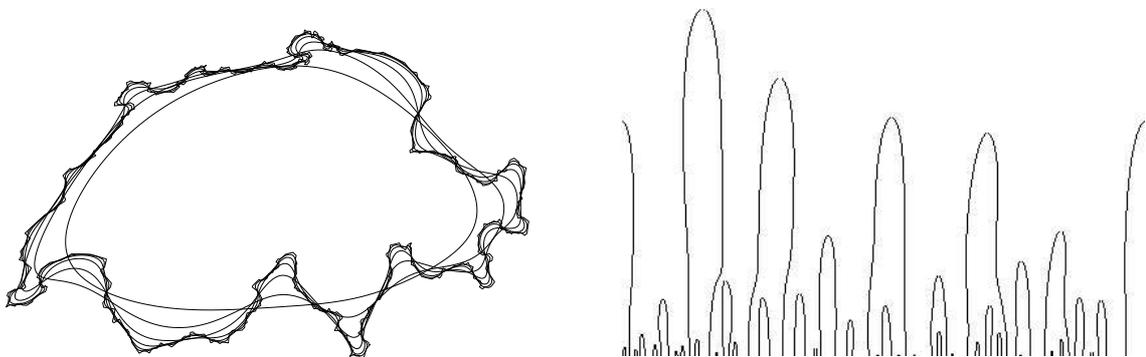


Abbildung 2.4: Umriss der Schweiz überlagert mit den entwickelten Versionen für die  $\sigma$ -Werte 2, 4, 8, 16, 32 und 64 (links) und dazugehöriges *Curvature Scale-Space Image* (rechts).

Die Autoren des *CSSI* halten fest, dass es sich beim *CSSI* um eine unter Rotation, gleichförmiger Skalierung und Translation invariante Repräsentation handle. Eigene Tests basierend auf dem Programmcode von Mokhtarian (1994) und eine ausführliche Behandlung durch Fritsch (1994) im Rahmen eines Aufenthalts am GIUZ haben jedoch ergeben, dass zumindest das Kriterium der Rotationsinvarianz nicht erfüllt werden konnte. Der eigentliche Grund, weshalb diese Repräsentation in dieser Arbeit nicht weiterverfolgt wurde, ist jedoch die Nicht-Erfüllung des Invertierbarkeits-Kriteriums. Die erforderliche Invarianz gegenüber affinen Transformationen hätte durch eine sorgfältigere und robustere Normalisierung der Daten, als sie im verfügbaren Code vorgefunden wurde, erzielt werden können (vgl. auch Kapitel 4). Da es aber gemäss Mokhtarian (E-Mail Korrespondenz, 1994) nicht möglich ist, aus einem *CSSI* die ursprüngliche Kurve zu rekonstruieren, fällt sie als Generalisierungs-Repräsentation ausser Betracht.

# Kapitel 3

## Abtastung

Die für diese Arbeit zur Verfügung stehenden Liniendaten liegen ursprünglich in einer Polylinien-Repräsentation vor, d.h. als geordnete Liste von Koordinaten-Paaren. Die Länge der (linearen) Liniensegmente, welche diese Punkte miteinander verbinden, ist unterschiedlich. Anders ausgedrückt: in bezug auf die *kurvilineare Distanz* der Kurve liegt keine äquidistante Abtastung (engl. *Sampling*) der Kurve vor. Bei der *kurvilinearen Distanz*  $u$  eines Punktes  $P$  handelt es sich um die Länge des Weges, welcher entlang der Linie zurückgelegt werden muss, um von deren Anfangspunkt zu diesem Punkt zu gelangen (Abbildung 3.1). Die Frage, ob sich eine äquidistante Abtastung einer Polylinien-Kurve aufdrängt, lässt sich unter dem Gesichtspunkt der in Abschnitt 2.2 aufgeführten Anforderungen an eine Repräsentation betrachten:

**Effizienz:** Das Kriterium der *Speichereffizienz* spricht klar gegen eine äquidistante Abtastung der Kurve, da redundante Punkte in die Beschreibung eingefügt würden. Die gleichzeitig geforderte *rechnerische Effizienz* andererseits rechtfertigt die Abtastung, insbesondere wenn abgeleitete Repräsentationen berechnet werden sollen, welche auf einer numerischen Integration beruhen.

**Einfachheit der Implementation:** Die numerische Integration der Linienlänge ist einfacher zu implementieren, wenn die Datenwerte in äquidistanten Abständen vorliegen.

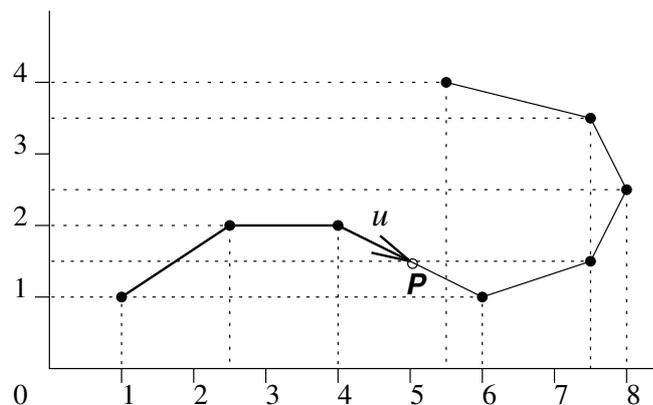


Abbildung 3.1: Die kurvilineare Distanz  $u$  bis zum Punkt  $P$  beträgt in diesem Beispiel  $u = \sqrt{1.5^2 + 1} + 1.5 + \sqrt{0.5^2 + 1} = 4.42$ .

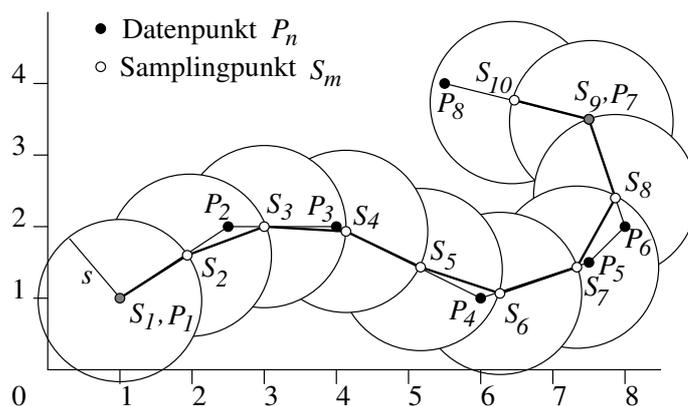


Abbildung 3.2: Die *Walking Divider*-Abtastmethode mit einer Schrittweite  $s$ . Datenpunkte sind mit  $P_n$  bezeichnet, Samplingpunkte mit  $S_m$ . Siehe Text für Erläuterungen.

Aufgrund dieser Überlegungen wird in einem ersten Vorverarbeitungsschritt jede zu verarbeitende Linie einer Abtastung unterzogen. Für die äquidistante Abtastung bietet sich speziell die Methode der *Walking Dividers* an, welche im nächsten Abschnitt behandelt wird.

### 3.1 *Walking Dividers*

Der *Walking-Dividers-Algorithmus* ist ein verbreitetes Verfahren zur Abtastung von Kurven, das beispielsweise auch zur Bestimmung der fraktalen Dimension eingesetzt wird (Goodchild und Mark 1987). Die folgenden Ausführungen beziehen sich auf Abbildung 3.2, in der die Funktionsweise des Algorithmus' anhand eines Beispiels aufgezeigt wird. Einziger Parameter des *Walking-Dividers-Algorithmus* ist die Schrittweite  $s$ , mit der die Kurve abgescritten wird. Der erste Samplingpunkt ( $S_1$ ) fällt immer mit dem ersten Datenpunkt ( $P_1$ ) zusammen. Um den nächstfolgenden Samplingpunkt  $S_2$  zu bestimmen, wird ein Kreis mit Radius  $s$  und Mittelpunkt  $S_1$  gezogen. Vereinfacht ausgedrückt bestimmt der Schnittpunkt dieses Kreisbogens mit dem "nächsten" Liniensegment den gesuchten Punkt  $S_2$ . Im Falle des gesuchten Samplingpunkts  $S_2$  handelt es sich dabei um das Segment  $\overline{P_1P_2}$ . Beim "nächsten" Liniensegment für einen gesuchten Samplingpunkt  $S_m$  kann es sich im allgemeinen Fall handeln um:

- dasselbe Segment, auf welchem sich  $S_{n-1}$  befindet (die Punkte  $S_2, S_5, S_7$  und  $S_{10}$  in Abb. 3.2)
- das dem Segment, auf welchem sich  $S_{n-1}$  befindet, unmittelbar folgende Segment (die Punkte  $S_3, S_4$  und  $S_6$ )
- eines der dem Segment, auf welchem sich  $S_{n-1}$  befindet, folgenden Segmente, jedoch nicht das unmittelbar folgende Segment (der Punkt  $S_8$ )
- einen Datenpunkt, der ein Liniensegment definiert (der Punkt  $S_9$  und, als Spezialfall, der Anfangspunkt  $S_1$ )

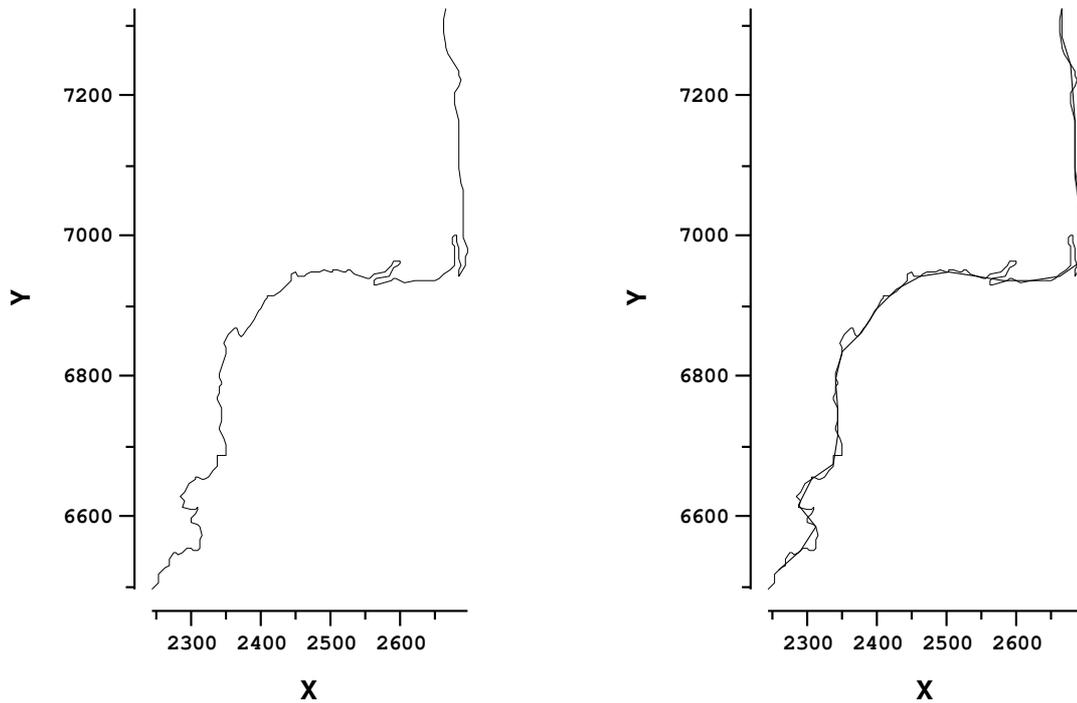


Abbildung 3.3: Die Linie *587ungen* im Maßstab 1:100'000 mit überlagerter abgetasteter Linie (Schrittweite 0.5). Bedingt durch die kleine Schrittweite und den kleinen Maßstab erscheinen die beiden Kurven deckungsgleich (links). Mit einer Schrittweite von 40 Einheiten nimmt die Approximationsgenauigkeit jedoch ab und es entsteht eine rudimentär vereinfachte Version der Linie (rechts).

Der letzte Fall tritt sehr selten auf. Im allgemeinen beinhalten die durch *Walking Dividers* erzeugten Samplingpunkte die ursprünglichen Datenpunkte, abgesehen vom Anfangspunkt, nämlich nicht. Daraus folgt, dass die ursprüngliche Polylinie aufgrund der resultierenden Samplingpunkte nicht exakt rekonstruiert, sondern lediglich approximiert werden kann. Dies ist der Preis, der für die äquidistante Anordnung der Samplingpunkte zu bezahlen ist. In Abschnitt 3.3 wird jedoch gezeigt, dass die Eliminierung von Datenpunkten auch positive Effekte zeitigen kann.

## 3.2 Approximationsqualität

In der implementierten Experimentierplattform *REPTILE* kann die Abtastschrittweite beliebig variiert werden. Unmittelbar nach einer Änderung der Schrittweite werden die Samplingpunkte neu berechnet und zusammen mit der Ausgangslinie in der  $x, y$ -Ebene dargestellt. Auf diese Weise kann die Veränderung der Approximationsqualität in Echtzeit verfolgt werden. Wie aus Abbildung 3.3 ersichtlich ist, kann durch eine Vergrößerung der Schrittweite eine rudimentäre Vereinfachung der Linie erzielt werden. Es ist jedoch dringend davor abzuraten, die *Walking-Dividers*-Methode als "Generalisierungs"-Algorithmus einzusetzen. In technischer Hinsicht besteht das Haupthindernis vor allem darin, dass dem Algorithmus die

nötige Stabilität fehlt: Eine Form (z.B. eine Haarnadelkurve) kann bei einer bestimmten Schrittweite  $s$  eliminiert werden, was jedoch nicht heisst, dass sie nach einem Anheben der Samplingdistanz nicht plötzlich wieder zum Vorschein kommt. Dieses Verhalten rührt daher, dass eine Veränderung der Schrittweite die Position *sämtlicher* Abtastpunkte verändert. Wird der *Walking-Dividers*-Algorithmus, wie in dieser Arbeit, lediglich als Abtastmethode verwendet, ist diese Instabilität jedoch unerheblich.

In den folgenden beiden Abschnitten wird auf die Ursache von Approximationsfehlern eingegangen und versucht, deren maximalen Betrag in Abhängigkeit der Schrittweite zu bestimmen. Die auftretenden Näherungsfehler sind unterschiedlicher Natur: Es handelt sich um *flächenhafte* Fehler einerseits und um eine *Linienverkürzung* andererseits. Positionsfehler (d.h. “falsche” Abtastpunkte) können nicht auftreten, da jeder Abtastpunkt algorithmusbedingt auf der Ursprungslinie liegen muss.

### 3.2.1 Flächenhafte Fehler

Flächenhafte Fehler entstehen durch mangelnde Information darüber, wie die Abtastpunkte an Unstetigkeitsstellen (d.h. Orten, wo die Kurve ihre Richtung ändert) verbunden werden müssen. Ein Beispiel dafür ist das Segment  $\overline{S_2S_3}$  in Abbildung 3.2. Da der Punkt  $P_2$  nicht zur Menge der Samplingpunkte gehört, entsteht ein flächenhafter Fehler in der Grösse der Dreiecksfläche  $P_2S_2S_3$ . Der Betrag  $A$  dieser Fläche ist abhängig vom Winkel, welche die beiden Liniensegmente  $\overline{P_1P_2}$  und  $\overline{P_2P_3}$  beschreiben, kann aber maximal gleich der Fläche des gleichseitigen<sup>1</sup> Dreiecks mit Seitenlänge  $s$  (= Schrittweite) sein:

$$(3.1) \quad A < \frac{s^2}{4}\sqrt{3}$$

Mit wievielen solcher Residualdreiecke haben wir es nun zu tun? Diese Zahl hängt vom Verhältnis der Auflösungsgenauigkeit  $r$  der Ursprungsdaten zur Schrittweite  $s$  ab. Die Grösse  $r$  bestimmt den Minimalabstand zweier Datenpunkte und demzufolge auch die minimale Länge, welche ein Segment der Ausgangslinie aufweisen kann. Links in Abbildung 3.4 ist dargestellt, welche Probleme auftreten können, wenn  $r = s$  ist. Die etwas stärkere Linie stellt die Ausgangslinie mit ihren Stützpunkten  $A$ ,  $B$  und  $C$  dar. Die Strecke  $\overline{BC}$  entspricht dabei genau der maximalen Auflösung  $r$  der Linie (d.h. dem kürzestmöglichen Liniensegment). Die schwarzen Punkte markieren bereits definierte Samplingpunkte. Ein Kreis mit Mittelpunkt  $S$ , dem zuletzt bestimmten Samplingpunkt, und Radius  $s_1$  wird dazu benutzt, den Folgepunkt zu finden. Bedingt durch die Kürze des Segments  $\overline{BC}$  und den spitzen Winkel zwischen  $\overline{AB}$  und  $\overline{BC}$  existiert aber kein Schnittpunkt zwischen Kreisbogen und dem Segment  $\overline{BC}$ . Dies bedeutet, dass kein Samplingpunkt auf letzteres Segment zu liegen kommt, sondern erst wieder auf das – in der Abbildung nicht dargestellte – Segment  $\overline{CD}$ . Ein Beispiel dafür ist auch das Liniensegment  $\overline{P_5P_6}$  in Abbildung 3.2, welches leer ausgeht. Eine Garantie dafür, dass auf jedes Segment der Ursprungslinie mindestens ein Samplingpunkt zu liegen kommt, besteht nur dann, wenn gilt:

$$(3.2) \quad \frac{r}{s} \geq 2$$

Diese Untergrenze der Abtaststrategie entspricht genau der *Nyquist-Frequenz* wie sie aus der *Sampling-Theorie* hinlänglich bekannt ist (Ballard und Brown 1982).

<sup>1</sup>Genau genommen haben wir es nie mit einem gleichseitigen Dreieck zu tun, weil die Strecke  $\overline{S_2P_2}$  immer kleiner als  $s$  ist. Wir drücken diese Tatsache im weiteren Verlauf durch eine Ungleichung aus.

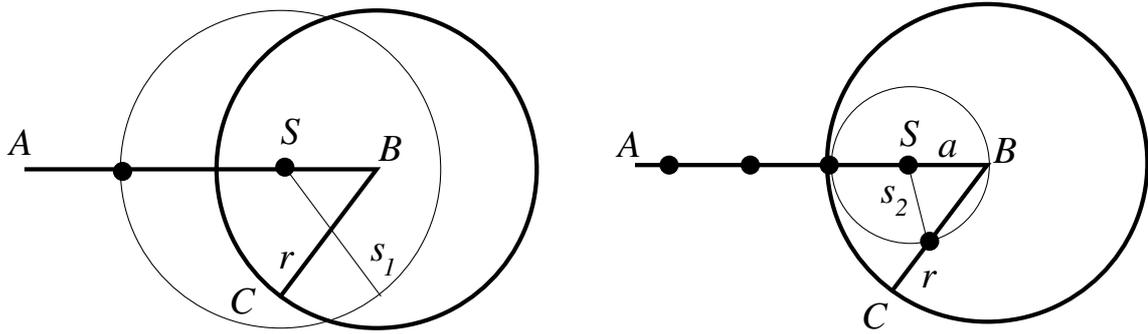


Abbildung 3.4: Die Schrittweite  $s$  muss kleiner als die Hälfte der Auflösung  $r$  der ursprünglichen Polylinie sein, damit auf jedes Liniensegment mindestens ein Abtastpunkt entfällt. Siehe Text für genauere Erläuterungen.

In Abbildung 3.4 rechts ist Bedingung 3.2 veranschaulicht. Zu bemerken ist, dass es sich dabei um einen Beinahe-Spezialfall handelt, indem gilt  $a + \varepsilon = s_2$ ,  $\varepsilon \rightarrow 0$ . Mit anderen Worten:  $a$  ist beinahe gleich  $s_2$ . Dieser Fall wurde deshalb gewählt, um zu zeigen, dass unter Voraussetzung von Bedingung 3.2 jederzeit der ganze Kreis  $K_1(S, s_2)$  im Kreis  $K_2(B, r)$  enthalten ist und somit der gesuchte Schnittpunkt immer existiert. Nun lässt sich der maximal auftretende flächenhafte Approximationsfehler  $E_{Amax}$  in Abhängigkeit von der Anzahl Datenpunkte  $n$  der Polylinie wie folgt bestimmen:

$$(3.3) \quad E_{Amax} < (n - 2) \frac{s^2}{4} \sqrt{3}, \quad \text{für } 0 < s \leq \frac{r}{2}$$

Wie zu erwarten, tendiert dieser Fehler für  $s \rightarrow 0$  gegen Null.

### 3.2.2 Linienverkürzung

Die Linienverkürzung, welche aus der Abtastung mit *Walking Dividers* resultiert, hat zwei Ursachen: Zum einen bedingt die Dreiecksungleichung, dass jede Unstetigkeitsstelle der Polylinie einen Beitrag zur Verkürzung liefert. Pro Datenpunkt strebt dieser gegen  $2s$  (im Falle von aufeinanderfolgenden Liniensegmenten mit entgegengesetztem Richtungssinn). Zum anderen fallen im allgemeinen der letzte Datenpunkt und der letzte Abtastpunkt nicht zusammen (vgl.  $P_8$  und  $S_{10}$  in Abb. 3.2), woraus ebenfalls eine Verkürzung der Linie um maximal  $s$  resultiert. Die maximale Linienverkürzung  $E_{Lmax}$  in Abhängigkeit von der Anzahl Punkte der ursprünglichen Polylinie  $n$  ergibt sich demnach wie folgt:

$$(3.4) \quad E_{Lmax} < (n - 2)2s + s, \quad \text{für } 0 < s \leq \frac{r}{2}$$

## 3.3 Wahl der Schrittweite

Die für diese Arbeit zur Verfügung stehenden Liniendaten (vgl. Anhang B) weisen eine geometrische Genauigkeit von 10 Metern auf. Kodiert sind die Daten jedoch in Dekameter-Einheiten, was bedeutet, dass der Auflösungsparameter  $r$  den Wert 1 aufweist. Um Bedingung 3.2 zu erfüllen, darf die Schrittweite  $s$  also nicht grösser als 0.5 sein.

Ein Blick auf Tabelle 3.1 genügt, um festzustellen, dass die in den beiden vorhergehenden Abschnitten beschriebenen maximalen Approximationsfehler  $E_{Amax}$  und  $E_{Lmax}$ , auf

Linie	$n$ , Länge	Schritt- weite	$E_{Amax}/n$	tatsächl. $E_A/n$	in % des max. Wertes	$E_{Lmax}/n$	tatsächl. $E_{Lmax}/n$	in % des max. Wertes
587	184, 1511.06	0.5	0.1071	0.0106	9.9	0.9918	0.0193	1.9
		1 <sup>a</sup>	0.4283	0.0479	11.2	1.9837	0.0492	2.5
596	98, 1189.93	0.5	0.1060	0.0088	8.3	0.9846	0.0095	1.0
		1 <sup>a</sup>	0.4242	0.0327	7.7	1.9693	0.0197	1.0

<sup>a</sup>Diese Schrittweite erfüllt die in Ungleichung 3.2 aufgestellte Bedingung formell nicht mehr. Es wurde jedoch überprüft, dass auf jedes Liniensegment ein Abtastpunkt zu liegen kommt. Insofern ist die Bedingung materiell erfüllt.

Tabelle 3.1: Gegenüberstellung von maximal möglichen flächenhaften Approximationsfehlern  $E_{Amax}$  und beobachteten Abweichungen  $E_A$ , sowie der entsprechenden Werte für entstehende Linienverkürzungen ( $E_{Lmax}$ , bzw.  $E_L$ ).

kartographische Linien angewandt, rein theoretischer Natur sind. In dieser Tabelle sind für zwei Linien die theoretischen Maximalwerte pro Datenpunkt<sup>2</sup> den tatsächlich beobachteten gegenübergestellt. Es zeigt sich, dass die flächenhaften Abweichungen kaum mehr als 10% des theoretisch möglichen Wertes ausmachen, in bezug auf die Linienverkürzung sogar nicht über 2.5% hinauskommen. Die beiden gewählten Beispiellinien *587ungen* (Abb. 3.3) und *596ungen* (Abb. 3.6) unterscheiden sich vor allem durch ihre Angularität. Als einfaches Mass dafür wurde die Summe der absoluten Winkeländerungen, dividiert durch die Länge der Linie, beigezogen. Die Linie *587ungen* weist demnach eine (hohe) Angularität von 0.0591 auf, die Linie *596ungen* eine (kleinere) von 0.0318. Wie zu erwarten, wirken sich Änderungen der Schrittweite auf Linien mit hoher Angularität stärker aus: Infolge Erhöhung der Schrittweite von 0.5 auf 1 nimmt der beobachtete flächenhafte Fehler  $E_A$  für Linie *587ungen* um 352% zu, während die entsprechende Zunahme für Linie *596ungen* lediglich bei 272% liegt. Dasselbe lässt sich für die Linienverkürzung beobachten, wo sich die Zunahmen auf 154% (*587ungen*), respektive 107% (*596ungen*) belaufen. In Anbetracht der relativ kleinen Fehler<sup>3</sup> erscheint eine Abtastung mit einer Schrittweite von der Hälfte der geometrischen Präzision der Ausgangsdaten für unsere Zwecke als genügend. Es kann jedoch durchaus von Vorteil sein, die Schrittweite bewusst grösser als diesen Schwellwert zu wählen, um kleinste Knitterungen zu eliminieren. Solche Kleinstformen sind häufig das Produkt von Digitalisierfehlern. Ein Beispiel eines solchen möglichen Aufnahmefehlers ist in Abbildung 3.5 wiedergegeben. Allerdings ist es schwierig, solche Mikroinflexionen in einer  $x, y$ -Repräsentation einer Kurve zu entdecken. In einem  $t, \alpha$ -Plot, wie er in Abschnitt 5.4 behandelt wird, sind solche Unstetigkeiten jedoch einfach auszumachen. In der Praxis wird deshalb die Schrittweite nach Begutachtung der parametrisierten Version mit Vorteil nochmals angepasst (vgl. Abschnitt 5.4.4).

Für den Rest dieser Arbeit wird nun von äquidistant abgetasteten Linien ausgegangen. Dies bedeutet u.a., dass sich auch die kurvilineare Distanz nunmehr auf die abgetastete

<sup>2</sup>Diese Normalisierung wurde vorgenommen, damit die beiden Linien auch untereinander verglichen werden können.

<sup>3</sup>Der absolute flächenhafte Fehler für *587ungen* mit  $s = 0.5$  entspricht einem Quadrat mit Seitenlänge 1.4, bei einer Linienlänge von 1511.

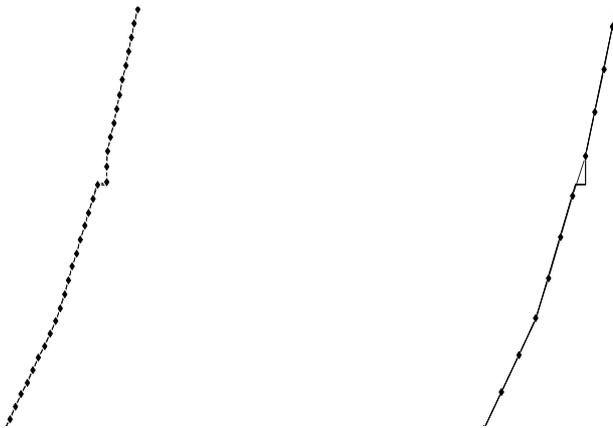


Abbildung 3.5: Ausschnitt der Linie *536ungen* im Massstab 1:2200. Links: Ein Abtastschritt von 1 führt zur Rekonstruktion eines mutmasslichen Digitalisierfehlers. Rechts: Mit Schrittweite 3 fallen keine Abtastpunkte mehr auf die fraglichen Segmente.

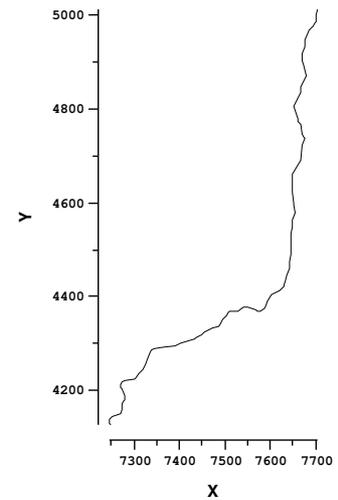


Abbildung 3.6: Linie *596ungen*, deren Approximationsverhalten in Tabelle 3.1 zusammengestellt ist.

Linie bezieht und gleichzeitig diskretisiert wurde. Der neue kurvilineare Distanzparameter  $t$  der abgetasteten Kurve kann nur noch Werte annehmen, welche ein Vielfaches der Samplingdistanz  $s$  sind.



# Kapitel 4

## Normalisierung

Ein in Abschnitt 2.2 gefordertes Kriterium, welchem eine Linienrepräsentation gerecht werden sollte, ist die Invarianz gegenüber affinen Transformationen. Dazu zählen:

- Translation
- Rotation
- lineare Skalierung.

Wie sich in Abschnitt 4.3 zeigen wird, muss für die Liniengeneralisierung ein weiteres Kriterium erfüllt werden, jenes der

- Richtungsinvarianz (Invarianz in bezug auf den Richtungssinn).

### 4.1 Skalierungsinvarianz

Das drittgenannte Kriterium, die Skalierungsinvarianz, ist in der Generalisierung jedoch nicht erforderlich, da der Massstab jeweils explizit vorgegeben ist. Bei zwei formidentischen, aber skalierten Objekten auf einem Kartenblatt desselben Massstabs handelt es sich daher um Objekte *verschiedener* Ausprägung. Demzufolge müssen sich auch ihre Beschreibungen unterscheiden. Dasselbe gilt für den Vergleich von Objekten über verschiedene Massstäbe hinweg: auch wenn eine Linie dadurch generalisiert wurde, dass sie lediglich skaliert wurde (was bei Linien mit simpler Geometrie durchaus der Fall sein kann), muss sich ihre Beschreibung von derjenigen im grösseren Massstab unterscheiden. Die Skalierung sollte im Idealfall einzig durch eine vereinfachte Transformationsabbildung zum Ausdruck kommen.

### 4.2 Translationsinvarianz

Invarianz gegenüber Translationen wird erreicht, indem eine Linie derart verschoben wird, dass ihr arithmetisches Mittelzentrum (Schwerpunkt) auf den Ursprung des Koordinatensystems zu liegen kommt. Die Koordinaten des Schwerpunkts  $\bar{P}$  berechnen sich einfach aus

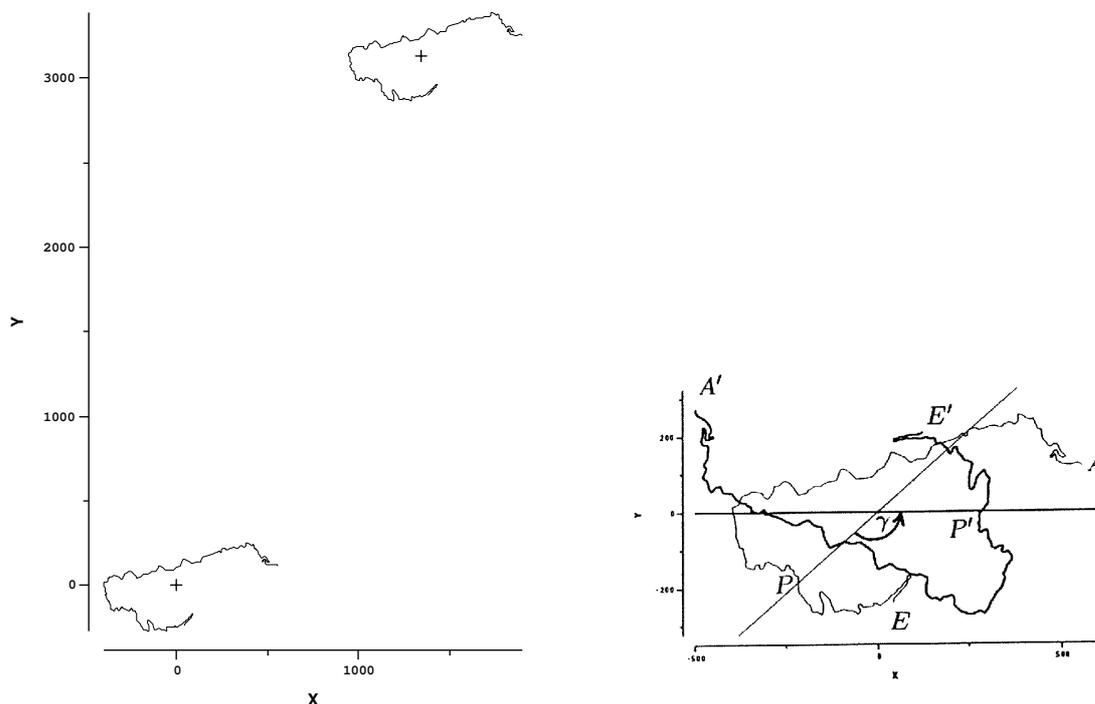


Abbildung 4.1: Normalisierung der Linie 501ungen. Links: Zuerst wird die Kurve derart verschoben, dass ihr Schwerpunkt auf den Ursprung des Koordinatensystems zu liegen kommt. Rechts: Nun wird der Trendvektor dieser Kurve bestimmt (der Übersichtlichkeit wegen ist allerdings die Trendlinie eingezeichnet) und die Kurve im Gegenuhrzeiger rotiert, so dass die Trendlinie mit der  $x$ -Achse zusammenfällt. Die Punkte  $A$  und  $E$  markieren Anfangs- und Endpunkt der Linie,  $A'$  und  $E'$  deren rotierte Bildpunkte. Auf Punkt  $P$  wird in Abb. 5.15 Bezug genommen.

den arithmetischen Mittelwerten der Koordinaten der  $n$  Punkte (Bahrenberg et al. 1990)<sup>1</sup>:

$$(4.1) \quad \bar{P} = (\bar{x}, \bar{y}), \quad \text{mit}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Die Lage von  $\bar{P}$  ist unabhängig von den gewählten Koordinatenachsen. Beim gesuchten Verschiebungsvektor handelt es sich folglich um  $\vec{p} = (-\bar{x}, -\bar{y})$  (Abbildung 4.1 links).

<sup>1</sup>Die Berechnung des wahren Schwerpunkts auf diese Art und Weise ist nur möglich, weil wir es mit äquidistant abgetasteten Linien zu tun haben.

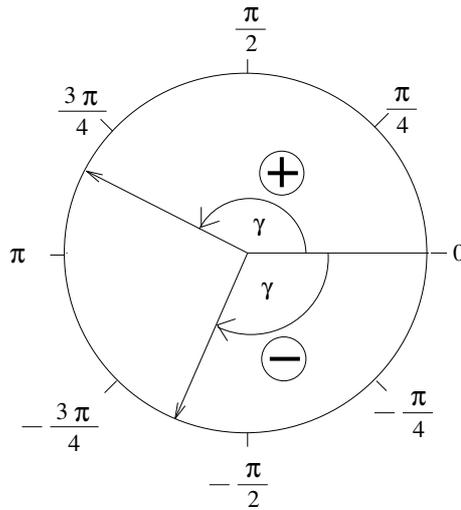


Abbildung 4.2: Berechnung des Richtungswinkels  $\gamma$  im Bogenmass für einen gegebenen (Trend-)Vektor. Dasselbe Berechnungsschema gelangt beim *Psi-s Plot* zum Einsatz, welcher in Kapitel 5.3 behandelt wird.

### 4.3 Rotationsinvarianz

Rotationsinvarianz wird erzielt, indem der Trendvektor der Linie berechnet und die Kurve so rotiert wird, dass dieser parallel zur  $x$ -Achse verläuft (bzw. mit der  $x$ -Achse zusammenfällt, im Falle einer bereits an den Ursprung verschobenen Kurve). Etwas salopp ausgedrückt definiert der Trendvektor die “durchschnittliche” Richtung eines Segments der abgetasteten Linie. Je nach Trendvektor ergeben sich verschiedene Richtungswinkel  $\gamma$ , was in Abbildung 4.2 dargestellt ist. Beim eigentlichen Rotationswinkel  $\alpha_R$  handelt es sich schliesslich um  $-\gamma$  (Abbildung 4.1 rechts).

Die “durchschnittliche” Richtung eines Segments der abgetasteten Kurve ergibt sich durch eine separate lineare Regression der  $x$ - und  $y$ -Koordinaten nach dem kurvilinearen Distanzparameter  $t^2$ . Die Regressionskoeffizienten  $b_x$  und  $b_y$  berechnen sich wie folgt (nach Bahrenberg et al. 1990):

$$(4.2) \quad b_x = \frac{\sum_{i=1}^n (t_i - \bar{t})(x_i - \bar{x})}{\sum_{i=1}^n (t_i - \bar{t})^2}$$

$$(4.3) \quad b_y = \frac{\sum_{i=1}^n (t_i - \bar{t})(y_i - \bar{y})}{\sum_{i=1}^n (t_i - \bar{t})^2}$$

Weil der Schwerpunkt der Kurve an den Ursprung verschoben wurde, verlaufen die Regressionsgeraden durch den Nullpunkt und somit fallen die Regressionskonstanten sowohl für  $x$  als auch für  $y$  weg. Die Regressionskoeffizienten sagen etwas aus über die durchschnittliche Veränderung der  $x$ - bzw.  $y$ - Koordinate, wenn von einem Samplingpunkt  $S_n$  zum nächsten Punkt  $S_{n+1}$  geschritten wird. Dies bedeutet nichts anderes, als dass die Regressionskoeffizienten  $b_x$  und  $b_y$  die  $x$ -, bzw.  $y$ -Koordinate des gesuchten Trendvektors liefern.

<sup>2</sup> $t$  kann diskrete Werte annehmen, welche ein Vielfaches der Samplingdistanz  $s$  sind. Siehe auch Kapitel 5.1 und 5.2.

Nun kann der Rotationswinkel berechnet werden. Der Richtungssinn des Trendvektors hängt vom Richtungssinn der Kurve ab. Bei der Linie in Abbildung 4.1 (rechts) verläuft dieser beispielsweise im Gegenuhrzeigersinn. Aufgrund von Experimenten mit verschiedenen Linienpaaren wurde bemerkt, dass zahlreiche Linienpaare unterschiedliche Richtungen aufwiesen, d.h. ungeneralisierte und generalisierte Linie wurden nicht in derselben Richtung digitalisiert. Die resultierenden Rotationswinkel unterschieden sich um 180 Grad, was den Vergleich der beiden Linienbeschreibungen verunmöglichte. Es wurde deshalb nötig, in einem Vorverarbeitungsschritt den Richtungssinn eines jeden Linienpaars zu harmonisieren.

#### 4.4 Rücktransformation

Der Datensatz einer normalisierten Linie umfasst neben den (normalisierten)  $x$ - und  $y$ -Koordinaten die Koordinaten des Translationsvektors  $\vec{p}$  sowie den Rotationswinkel  $\alpha_R$ . Die Rücktransformation der normalisierten Linie an ihre ursprüngliche Lage in der Ebene erfolgt durch Anwendung der oben beschriebenen Transformationen in umgekehrter Reihenfolge.

# Kapitel 5

## Parametrisierung

Mit der Parametrisierung der Linie verlassen wir die  $x, y$ -Ebene der Karte, in welcher wir die Abtastung und die Normalisierung vorgenommen haben. Die Motivation zum Verlassen des Ortsraums besteht in erster Linie darin, die Kurve als *Funktion* darstellen zu können, bei der jedem Wert auf der Abszisse *genau ein* Ordinatenwert entspricht. Diese Absicht liegt ihrerseits darin begründet, dass Funktionen mathematisch und numerisch bedeutend besser handhabbar sind als Graphen. Dies gilt vor allem auch im Hinblick auf die Verwendung von Parameterdarstellungen als Grundlage für andere, weiterentwickelte Repräsentationen.

Eine Linie in geeigneter Parameterdarstellung kann aber auch ohne direkte Weiterverarbeitung bereits wertvolle Informationen für die Generalisierung liefern. Hier kommen wiederum die in Kapitel 2.2 aufgeführten Kriterien ins Spiel. Für die Berechnung von Formmassen, z.B. zur Beurteilung der Generalisierungsqualität (Ehrliholzer 1996) oder zur Segmentierung der Linie, dürfte eine Parameterdarstellung am besten geeignet sein. Diese Vermutung liegt einmal mehr darin begründet, dass es sich um eine Funktion handelt. Die Ableitungen und deren Nullstellen erlauben es, wichtige Formmasse zu extrahieren.

Schliesslich dürfte es auch möglich sein, speziell entwickelte Generalisierungsalgorithmen direkt auf die parametrisierte Linie anzuwenden. O'Neill und Mark (1987), deren *Psi-s Plot* in Abschnitt 5.3 beschrieben wird, glauben jedenfalls dass "the Psi-s curve appears to have potential both for line generalization and for pattern recognition in cartographic lines." Wie sich herausstellte, ist der *Psi-s Plot* jedoch mit Problemen behaftet, welche durch die in Abschnitt 5.4 beschriebene Weiterentwicklung allerdings behoben werden können.

### 5.1 Der Parameter $t$

Alle im folgenden beschriebenen Parametrisierungen bedienen sich desselben Parameters  $t$ , welcher auf der in Kapitel 3 erstmals erwähnten *kurvilinearen Distanz*  $u$  beruht. Im Gegensatz zu  $u$  ist der Parameter  $t$  nicht mehr stetig, sondern kann nur noch diskrete Werte annehmen:

$$(5.1) \quad \begin{aligned} &0 \leq i \leq n \\ t &= i s, \quad s : \text{Abtastschrittweite} \\ &n : \text{Anzahl Abtastpunkte} \end{aligned}$$

Der Parameter  $t$  stellt die unabhängige Variable dar und wird auf der Abszisse abgetragen. Die nun folgenden Parametrisierungen unterscheiden sich einzig von der abhängigen Grösse, welche auf der Ordinate abgetragen wird.

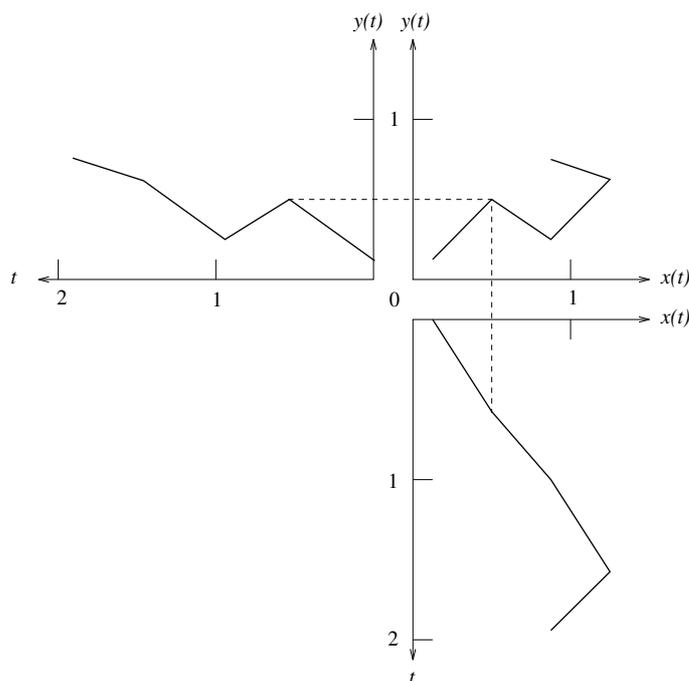


Abbildung 5.1: Parametrisierung von  $x$ - und  $y$ -Koordinaten nach  $t$ . Die strichlierte Linie verbindet einen Punkt im  $x, y$ -Plot mit den entsprechenden Punkten im  $x(t)$ -, bzw.  $y(t)$ -Plot (nach Foley et al. 1991).

## 5.2 Parameterdarstellung der $x$ - und $y$ -Koordinaten

Am naheliegendsten – und trivial zu implementieren – ist die separate Parametrisierung der  $x$ - und  $y$ -Koordinaten nach  $t$ , wie sie schon von Buttenfield (1985) besprochen wurde. Diese Transformation wurde bereits in Abschnitt 4.3 kurz erwähnt, als es darum ging, die Regressionsgeraden zwecks Normalisierung der Linie zu berechnen. Das Vorgehen ist denkbar einfach: der Reihe nach wird für jeden Samplingpunkt  $i$  die  $x$ - und  $y$ -Koordinate gegenüber  $t$  abgetragen (Abbildung 5.1). Daraus folgen zwei Funktionen:

$$(5.2) \quad \begin{aligned} x &= x(t), \\ y &= y(t). \end{aligned}$$

Zur Veranschaulichung finden sich in Abbildung 5.2 die parametrisierten Funktionen der (normalisierten) Linie *501ungen*.

Die Nachteile dieser Repräsentation liegen darin, dass *zwei* Funktionen nötig sind, um die Kurve zu beschreiben. Dies macht die Repräsentation schwer handhabbar. Formmasse lassen sich, abgesehen von den globalen Extrema, welche die minimalen und maximalen  $x$ -, bzw.  $y$ -Koordinaten definieren (nützlich zur Bestimmung der *Bounding Box*), ebenfalls nicht ablesen. Trotzdem wurde diese Repräsentation in *REPTILE* implementiert, hauptsächlich um Fourierreihen darauf zu entwickeln (Kapitel 6).

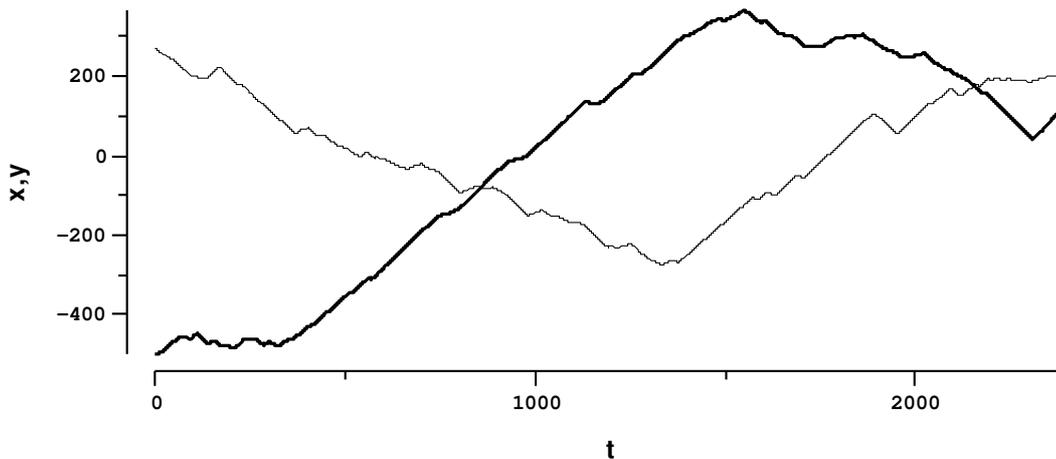


Abbildung 5.2: Nach  $t$  parametrisierte Koordinaten der normalisierten Linie *501ungen*. Die dünne Linie markiert die Funktion  $y = y(t)$ , die fette Linie  $x = x(t)$ .

### 5.3 *Psi-s Plot*

Die Anwendung von *Psi-s Plots* in der Generalisierung wurde erstmals von O’Neill und Mark (1987) vorgeschlagen. Es handelt sich dabei gewissermassen um das stetige Analogon zum diskreten *Freeman Chain Code* (Freeman 1974; Freeman 1978). Beim Chain Code werden normalerweise die Koordinaten des ersten Punkts explizit angegeben, während es sich bei den nachfolgenden Elementen der Datenstruktur um Ziffern handelt, welche Richtung und Distanz zwischen jedem Punkt der Linie und dessen Nachbarn definieren. Wie aus Abbildung 5.3 ersichtlich, definieren die Ziffern 0 bis 7 die Richtung im Gegenuhrzeigersinn (Thapa 1988). Durch die Beschränkung auf acht Richtungen werden *Chain Codes* normalerweise nur auf Rasterdaten angewandt. Der *Psi-s Plot* schliesst diese Lücke, indem er die Kodierung eines beliebigen Richtungsvektors  $\psi$  (Psi) erlaubt. Es gelangen dieselben Winkel im Bogenmass zur Anwendung, wie sie bereits zur Erzielung der Rotationsinvarianz gebraucht wurden (Abbildung 4.2). Der *Psi-s Plot* wird erhalten, indem der Richtungswinkel für jeden Punkt der Kurve gegenüber  $s$  abgetragen wird. Um die Notation der Autoren beizubehalten, steht der Parameter  $s$  an dieser Stelle ausnahmsweise für die kurvilineare Distanz entlang der Kurve (in dieser Arbeit ansonsten mit  $u$  bezeichnet) und *nicht* für die Schrittlänge. Abbildung 5.4 entstammt dem Artikel von O’Neill und Mark und zeigt einen Flusslauf sowie einen Teil von dessen *Psi-s Plot*.

#### 5.3.1 Anwendung des *Psi-s Plots* auf diskret abgetastete Linien

In Abbildung 5.4 wird die *Psi-s*-Repräsentation auf eine stetige Kurve angewandt, was in einem zumindest stückweise differenzierbaren *Psi-s*-Plot resultiert. In dieser Arbeit – wie wohl meistens in der Praxis – haben wir es jedoch nicht mit stetigen Kurven zu tun, sondern mit Polylinien, deren Segmente stückweise linear sind. Jeder Stützpunkt einer solchen Linie ist ein Unstetigkeitspunkt<sup>1</sup>. Die Abtastung einer solchen Kurve verändert diese

<sup>1</sup>Es sei denn, die Richtung der Linie ändert sich nicht, womit der Stützpunkt eigentlich unnötig wäre.

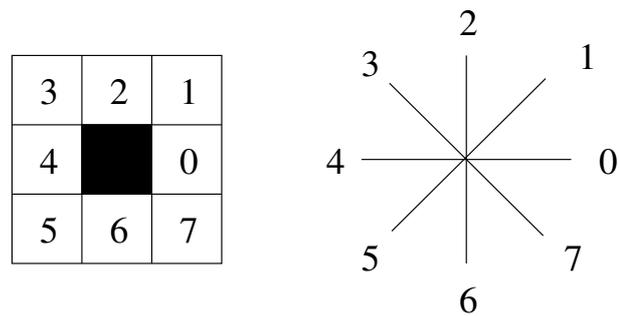


Abbildung 5.3: Einfacher Freeman *Chain Code* am Beispiel von Rasterdaten. Das Pixel in der Mitte hat 8 Nachbarn mit den Codes 0 bis 7 (nach Thapa 1988).

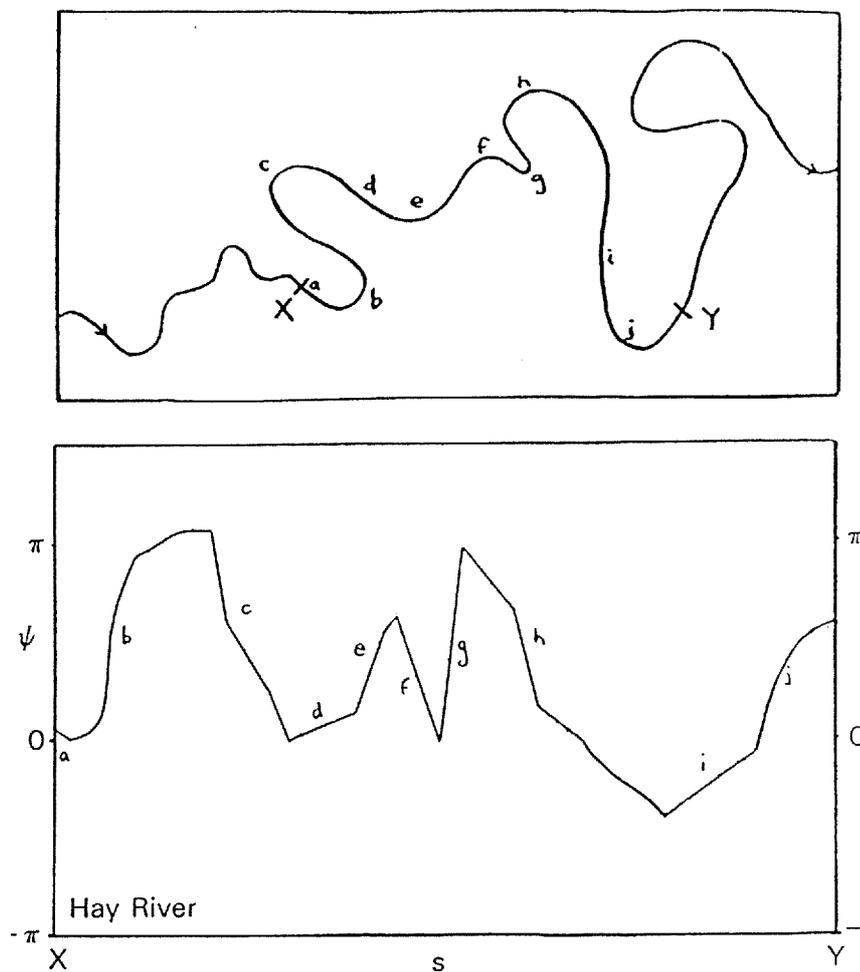


Abbildung 5.4: Teil des *Hay River* in Alberta, Kanada, (oben) und der dazugehörige *Psi-s Plot* für das Teilstück *XY* (unten). Entsprechende Abschnitte der beiden Diagramme sind mit Kleinbuchstaben gekennzeichnet (aus: O'Neill und Mark, 1987).

Ausgangslage nur insofern, als dass die Liniensegmente nach erfolgter Abtastung dieselbe Länge aufweisen. Bis anhin stellte die Unstetigkeit der Ursprungslinie kein Problem dar, da es sich bei den behandelten Entitäten ausschliesslich um die Koordinaten von Punkten in der  $x, y$ -Ebene handelte. Mit dem *Psi-s Plot* kommen nun aber Winkel ins Spiel, welche die Frage aufwerfen, welcher Entität (Abtastpunkten oder Liniensegmenten) diese zugeordnet werden sollen. Zwei Fälle müssen unterschieden werden:

**Physische Datenstruktur:** Der Datensatz eines *Psi-s Plots* wird in *REPTILE* als Liste mit den Elementen  $\{\text{Abtastpunkt } i, \psi_i\}$  gespeichert. Der *Header* des Datensatzes beinhaltet zudem Abtastschrittweite, Translationsvektor und Rotationswinkel. Die Zuordnung des Winkels zu einem Punkt ist die kompakteste, redundanzfreie interne Repräsentation.

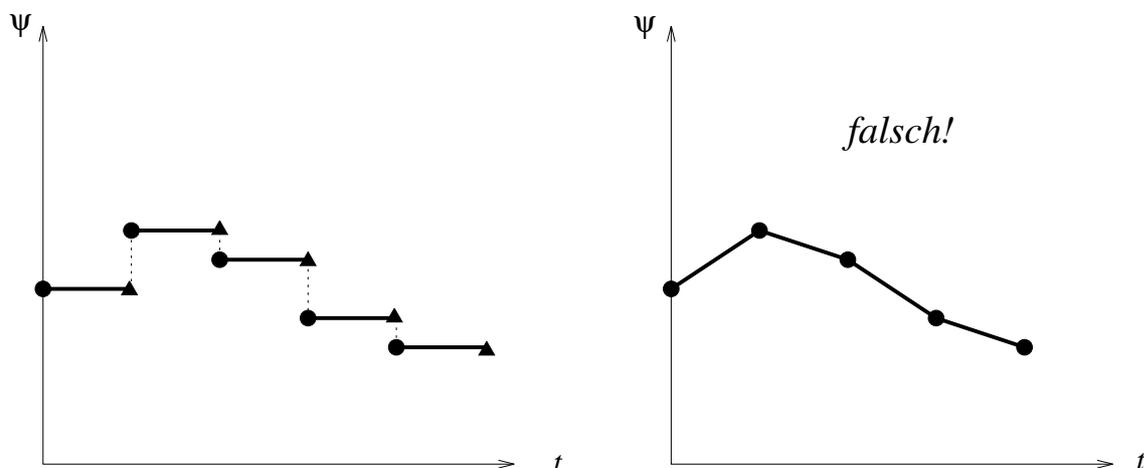
**Logische Datenstruktur:** Die logische Datenstruktur zieht das Wissen über die Qualität der Daten (sog. Metadaten) mit ein, wenn es um (a) die grafische Darstellung der Daten und (b) um die Entwicklung von Algorithmen geht. Im Fall des *Psi-s Plots* besteht die *logische* Datenstruktur aus Elementen der Form  $\{\text{Segment } i, \psi_i\}$ . Weil wir wissen, dass es sich bei den Ausgangsdaten um stückweise lineare Liniensegmente handelt, können wir den Abtastpunkten keine Winkel zuordnen – wie dies die physische Datenstruktur tut –, weil die Kurve an dieser Stelle nicht differenzierbar ist. Die grafische Darstellung muss immer auf der logischen Datenstruktur beruhen, wie dies in Abbildung 5.5 links dargestellt ist. Um klar zu machen, dass der Parameter  $s$  des *Psi-s Plots* (die kurvilineare Distanz) im Falle von Polylinien als Ausgangsrepräsentation nur diskrete Werte annehmen kann, wird er fortan mit  $t$  bezeichnet. Als Beispiel für einen auf der logischen Datenstruktur basierenden Algorithmus sei hier die Integration eines *Psi-s Plots* erwähnt, wie sie für die Transformation in frequenzbasierte Repräsentationen gebraucht wird<sup>2</sup> (Abbildung 5.6).

### 5.3.2 Vorteile des *Psi-s Plots*

Einen Vorteil sehen O'Neill und Mark in der unmittelbaren geometrischen Interpretierbarkeit der Beschreibung<sup>3</sup>. So stehen Geraden im *Psi-s Plot* für Kreisbogen in der  $x, y$ -Ebene. Wird der Richtungswinkel  $\psi$  im Bogenmass gemessen, dann entspricht der Radius dieses Kreisbogens exakt dem Kehrwert der Steigung der *Psi-s* Kurve. Diese Eigenschaften lassen sich aus Abbildung 5.7 ersehen. Gegeben sei der links in der Abbildung dargestellte lineare Abschnitt eines *Psi-s Plots*. Eingezeichnet sind Steigungsdreiecke mit der Basislänge  $t$ . Unter Annahme einer solchen diskreten Schrittlänge  $t$  könnte die Kurve in der  $x, y$ -Ebene wie im mittleren Teil der Abbildung dargestellt rekonstruiert werden. Pro Schritt nimmt der Winkel im Vergleich zum vorhergehenden Segment um  $\alpha$  zu. Es ist unschwer zu erkennen, dass diese Kurve für  $t \rightarrow 0$  einen Kreisbogen beschreibt, wie im rechten Teil dargestellt. An jedem Punkt dieser Kurve ist die Richtung derselben bestimmt durch den Tangentenvektor in diesem Punkt. Der für den *Psi-s Plot* massgebende Winkel  $\beta$  wird bestimmt durch

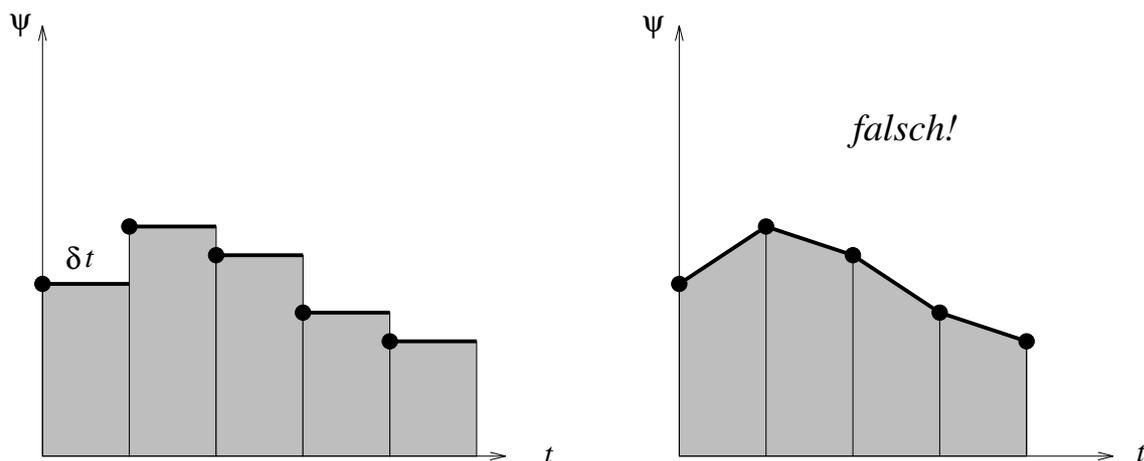
<sup>2</sup>Tatsächlich werden allerdings *t-alpha Plots* integriert, wie sie in Abschnitt 5.4 besprochen werden. Das Prinzip bleibt sich jedoch gleich.

<sup>3</sup>Die folgenden Ausführungen beziehen sich wieder auf *Psi-s Plots* von stetigen Kurven, wie von den Autoren beschrieben. Die gewonnenen Erkenntnisse lassen sich in beschränktem Masse auch auf diskrete Ausgangskurven übertragen.



- *Abtastpunkte*
- ▲ *für grafische Darstellung eingefügte Punkte*

Abbildung 5.5: Links: Korrekte Darstellung eines *Psi-s Plots* mit diskreter Schrittweite. In *REPTILE* werden die punktiert dargestellten vertikalen Verbindungslinien ebenfalls eingezeichnet. Rechts: Falsche Darstellung basierend auf der physischen Datenstruktur. Es wird suggeriert, dass sich  $\psi$  zwischen zwei Abtastpunkten verändert, was aber nicht der Fall ist. Vergleiche auch Abschnitt 5.3.2 zur Bedeutung der Steigung in einem *Psi-s Plot*.



- *Abtastpunkte*

Abbildung 5.6: Links: Korrekte Integration eines *Psi-s Plots* mit diskreter Schrittweite  $\Delta t$ . Rechts: Falsche Integration unter Einbezug der Steigungsdreiecke.

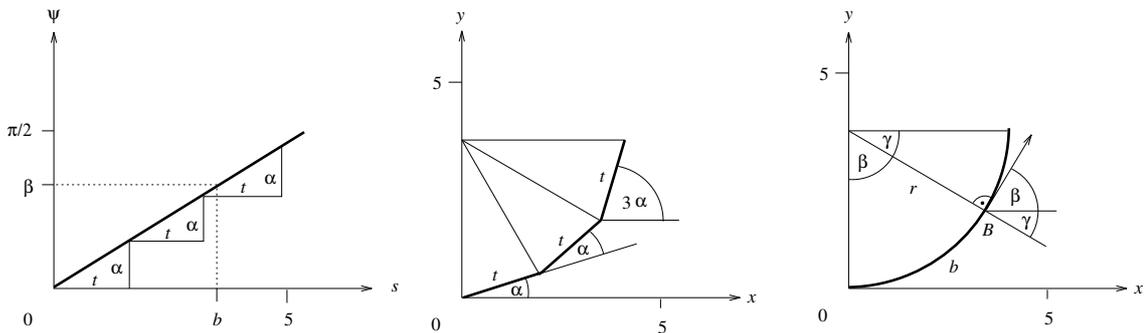


Abbildung 5.7: Lineare Abschnitte im *Psi-s Plot* (links) stellen Kreisbogensegmente in der  $x,y$ -Ebene dar (rechts). Die mittlere Abbildung wurde zur Herleitung gebraucht. Erläuterungen dazu finden sich im Text.

den Tangentenvektor und eine Parallele zur  $x$ -Achse. Mit anderen Worten: beim Winkel  $\beta$  handelt es sich um den  $\psi$ -Wert an der Stelle  $B$ , wie es im *Psi-s Plot* mit der punktierten Linie angedeutet ist. Was nun noch fehlt, ist die Herleitung, weshalb der Radius  $r$  des Kreisbogens gleich dem Reziprokwert der Steigung  $a$  der *Psi-s*-Kurve ist. Es gilt:

$$(5.3) \quad a = \frac{\beta}{b}$$

$$(5.4) \quad b = r\beta$$

Durch Einsetzen von 5.3 in 5.4 und Auflösung nach  $r$  ergibt sich schliesslich:

$$(5.5) \quad r = \frac{1}{a}$$

Für horizontale Abschnitte ( $a = 0$ ) im *Psi-s Plot* ist obige Gleichung nicht definiert. Solche Fälle bedeuten aber nichts anderes, als dass entlang der Linie keine Richtungsänderungen auftreten, d.h. horizontale, lineare Abschnitte des *Psi-s Plots* manifestieren sich als Geraden im Ortsraum. Zum Schluss sei ein weiterer Vorteil dieser Repräsentation erwähnt: Das Kriterium des *beschränkten Wertebereichs* (vgl. 2.2) ist erfüllt, weil mögliche *Psi*-Werte auf das Intervall  $]-\pi, \pi]$  beschränkt sind.

### 5.3.3 Nachteile des *Psi-S Plots*

Aufgrund des *Psi-s Plots* in Abbildung 5.4 ist man versucht, eine weitere nützliche Eigenschaft in den *Psi-s Plot* hineininterpretieren zu wollen. Alle Beispiele in O'Neill und Marks Artikel sind nämlich so gewählt worden, dass die lokalen Extrema des *Psi-s Plots* immer genau den Wendepunkten der Kurve in der  $x,y$ -Ebene entsprechen. Die Lokalisierung von Wendepunkten alleine durch das Finden von lokalen Extrema einer Funktion wäre ein sehr einfach zu implementierendes Formmass, welches z.B. für die Liniensegmentierung von grosser Bedeutung ist (vgl. dazu Plazanet 1995). Im allgemeinen handelt es sich bei lokalen Extremwerten eines *Psi-s Plots* jedoch *nicht immer* um die Wendepunkte der Kurve. Dies kann pikanterweise gleich an dem von O'Neill und Mark verwendeten Beispiel des *Hay River* (Abb. 5.4) gezeigt werden. Wird der *Psi-s Plot* der Flusslinie nämlich nicht nur für den Abschnitt zwischen den eingezeichneten Punkten  $X$  und  $Y$  berechnet, sondern für die ganze

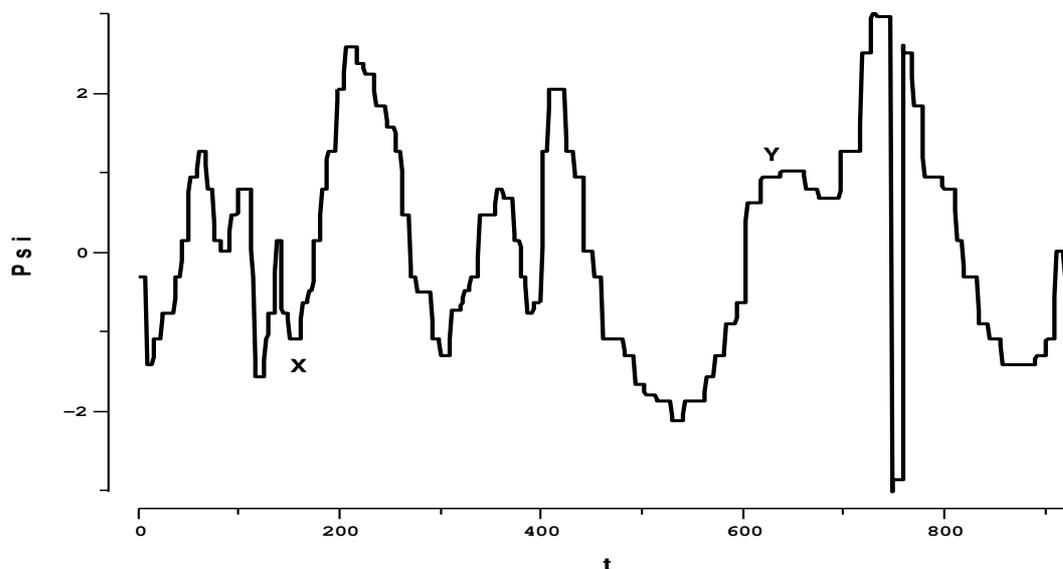


Abbildung 5.8: *Psi-s Plot* des digitalisierten und abgetasteten *Hay River* über die Punkte X und Y hinaus. Deutlich ist die Diskontinuität an der Stelle  $t = 750$  erkennbar.

Linie, treten Probleme zutage (Abbildung 5.8). Die Diskontinuität an der Stelle  $t = 750$  wird durch die Unstetigkeitsstelle im Kodierungsschema des Richtungswinkels (Abb. 4.2) verursacht. Wie aus Abbildung 5.9 entnommen werden kann, ist die Amplitude der Sprungstelle am grössten, wenn sich das Vorzeichen von  $\psi$  in der Nähe der Unstetigkeitsstelle  $\pi$  ändert. Grundsätzlich tritt die Sprungstelle aber immer dann in Erscheinung, wenn die Summe der beiden Richtungswinkel  $\psi_{\bar{a}}$  und  $\psi_{\bar{b}}$  einen stumpfen Winkel einschliesst:

$$(5.6) \quad \begin{aligned} |\psi_{\bar{a}}| + |\psi_{\bar{b}}| > \pi, \quad \psi_{\bar{a}} \geq 0 \\ \psi_{\bar{b}} < 0 \end{aligned}$$

Wie bereits erwähnt, verunmöglichen solche Sprungstellen die Interpretation der Extrema als Wendepunkte. Sie erschweren aber auch die Entwicklung der Funktion in Frequenzzerlegungen wie Fourierreihen und Wavelets. Wie in den Kapiteln 6 und 7 gezeigt wird, basieren diese Transformationen auf der Aufmodulierung von Schwingungen zunehmend höherer Frequenz auf ein Trägersignal. Eine Sprungstelle in der zu entwickelnden Funktion wirkt sich dabei auf den gesamten Frequenzbereich aus, was eine optimale Approximation mit möglichst wenigen Koeffizienten verunmöglicht.

## 5.4 Die kumulativ-relative Repräsentation *t-alpha*

### 5.4.1 Anlehnung an Kurvaturelemente des *Chain Code*

Der *Psi-s Plot* wurde eingangs dieses Kapitels als stetiges Analogon zum Freeman *Chain Code* bezeichnet. Es ist nicht weiter erstaunlich, dass auch der *Chain Code* dieselben Diskontinuitäten wie der *Psi-s Plot* aufweist. Die Sprungstellen im Freeman *Chain Code* können – unter Voraussetzung, dass eine Kodierung gemäss Abbildung 5.3 vorliegt – jedoch mit

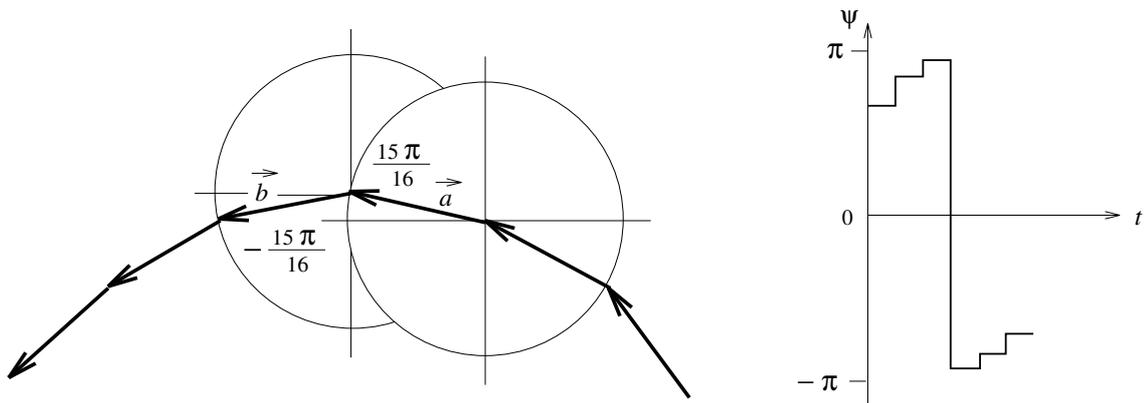


Abbildung 5.9: Die Amplitude der Sprungstelle im *Psi-s Plot* ist am grössten in der Nähe der Unstetigkeitsstelle  $\pi$ . Obwohl sich die Richtungen der beiden Vektoren  $\vec{a}$  und  $\vec{b}$  nur um  $2\pi/16$  unterscheiden, liegen sie im *Psi-s Plot*  $30\pi/16$  auseinander.

folgender Formel eliminiert werden (Thapa 1988; Pavlidis 1978; Eccles et al. 1977):

$$(5.7) \quad c(i) = (e(i) - e(i - 1) + 11) \bmod 8 - 3$$

Dabei handelt es sich bei  $e(i)$  ( $i = 1, 2, \dots, N$ ) um die Freeman *Chain Codes* und bei  $c(i)$  um sogenannte *Kurvaturelemente*. Die  $c(i)$  beschreiben die *relative* Richtungsänderung der Linie von einem Pixel zum nächsten, wobei diese in Anzahl Pixel angegeben wird. Eine Abweichung im Gegenuhrzeigersinn hat ein positives Vorzeichen, eine Abweichung im Uhrzeigersinn ein negatives. Sowohl der *Chain Code* als auch die Kurvaturelemente für den Ausschnitt einer Rasterlinie sind im oberen Teil von Abbildung 5.10 wiedergegeben. Es zeigt sich wohl, dass die durch Kurvaturelemente repräsentierte Linie die Diskontinuitäten des *Chain Code* nicht mehr aufweist. Diese Art der Repräsentation ist aber für unsere Zwecke aus zwei Gründen nicht von Vorteil:

1. Die *lokalen* Extrema sind als Formmass unbrauchbar. Sie sagen lediglich etwas über die maximale Krümmung zwischen zwei geraden Liniensegmenten aus. Die *globalen* Extrema lokalisieren die Punkte mit maximaler Krümmung, was für die Liniensegmentierung jedoch nur von untergeordneter Bedeutung ist.
2. Das Diagramm besteht aus lauter kleinen Ausschlägen. Sobald sich die Krümmung nicht verändert, fällt der Graph auf die Abszisse zurück. Es dürfte kaum möglich sein, eine solche Funktion adäquat in frequenzbasierte Repräsentationen zu transformieren, weil sich das Signal lediglich aus hochfrequenten Komponenten zusammensetzt.

Diese beiden Vorbehalte fallen jedoch dahin, sobald die Kurvaturelemente zu  $k(i)$  *aufaddiert* werden, wie dies in Abbildung 5.10 unten dargestellt ist. Die Sprungstelle des *Chain Code* konnte unterdrückt werden und die lokalen Maxima bezeichnen nun erstmals die Wendepunkte der Rasterlinie. Darüberhinaus fällt der Graph bei unveränderter Krümmung nicht gleich in sich zusammen, was in einer Vergrößerung des Frequenzspektrums resultiert.



### 5.4.2 Herleitung der *t-alpha* Repräsentation

In dieser Arbeit wurde nun versucht, das Analogon zu den kumulativen Kurvaturelementen auch für den *Psi-s Plot* zu berechnen. Der kumulierte relative Richtungswinkel wird *alpha* genannt, die unabhängige Variable wird wie bereits beim diskreten *Psi-s Plot* mit  $t$  bezeichnet. Die folgende Formel liefert – in Anlehnung an Formel 5.7 – das gewünschte Resultat für das Segment  $i$  der abgetasteten Linie (mit Schrittweite  $s$  und  $i = t/s$ ):

$$(5.8) \quad \alpha_i = \alpha_{i-1} + ((\psi_i - \alpha_{i-1} + 3\pi) \bmod 2\pi - \pi)$$

Der zweite Summand berechnet den relativen, *spitzen* Richtungswinkel  $\beta$  zwischen den Segmenten  $i$  und  $i - 1$  ( $-\pi < \beta \leq \pi$ ). Der etwas mystisch anmutende Term basiert auf der Subtraktion der kumulierten, relativen Richtungswinkel aller vorangegangenen Segmente ( $\alpha_{i-1}$ ) vom absoluten Richtungswinkel des aktuellen Segments ( $\psi_i$ ). Die Ebene, welche das Resultat dieser Subtraktion für verschiedene  $\psi_i$  und  $\alpha_{i-1}$  aufspannt, ist in Abbildung 5.11 dargestellt. Die Figur beschränkt sich auf das  $\alpha$ -Intervall  $[-3\pi, 3\pi]^4$ . Ausserhalb dieses Intervalls werden die durch  $\psi_i - \alpha_{i-1}$  definierten Richtungswinkel grösser als  $2\pi$ , bzw. kleiner als  $-2\pi$ . Ein Richtungswinkel der Grösse  $2\pi + \beta$  definiert aber denselben Richtungsvektor wie der Winkel  $\beta$ . Wie später gezeigt wird, trägt die Formel 5.8 dieser zyklischen Beschaffenheit der Richtungsvektoren Rechnung. Zur Vereinfachung der nachfolgenden Erläuterungen soll der Definitionsbereich von  $\alpha_{i-1}$  aber noch weiter eingeschränkt werden. Ein Blick auf Abbildung 5.11 zeigt, dass auch auf einem  $\alpha$ -Intervall von lediglich  $[-\pi, \pi]$  sämtliche durch  $\psi_i - \alpha_{i-1}$  bestimmten Richtungsvektoren zwischen  $-2\pi$  und  $2\pi$  erhalten werden. Der kumulierte relative Richtungswinkel  $\alpha$  an der Stelle  $(i - 1)$  wird damit dem Winkel  $\psi_{i-1}$  gleichgestellt (Abbildung 5.12 links). Aus der Abbildung wird deutlich, dass die Bestimmung des relativen Richtungsvektors basierend auf der Differenz zweier aufeinanderfolgender Richtungswinkel nicht praktikabel ist. Zum einen variiert der resultierende Richtungswinkel im Intervall  $[-2\pi, 2\pi]$ , was auch (unerwünschte) stumpfe Winkel ermöglicht. Zum anderen befindet sich die Sprungstelle um  $(-\pi, \pi)$  bzw.  $(\pi, -\pi)$  an einem arbiträren Ort. An dieser Stelle weisen nämlich beide Vektoren annähernd dieselbe Richtung auf, was eine Diskontinuität mit der grossen Amplitude von  $4\pi$  nicht rechtfertigt (Abbildung 5.12 rechts).

Durch eine geschickte Transformation der Ebene  $\psi_i - \psi_{i-1}$  kann aber sowohl der gewünschte Wertebereich  $[-\pi, \pi]$  als auch eine plausible Sprungstelle erhalten werden. Das Resultat einer solchen Transformation ist in Abbildung 5.13 dargestellt. Bei der gesuchten Transformationsabbildung handelt es sich natürlich um  $(\psi_i - \alpha_{i-1} + 3\pi) \bmod 2\pi - \pi$  aus Formel 5.8. Abbildung 5.14 kann entnommen werden, welche Auswirkungen die einzelnen Operatoren des Terms haben. Daraus dürfte auch ersichtlich sein, dass die zur Vereinfachung der Abbildungen eingeführte Beschränkung auf das  $\alpha$ -Intervall  $[-\pi, \pi]$  ohne weiteres aufgehoben werden kann. Mit dem *t-alpha Plot* steht nun also eine Repräsentation zur Verfügung, welche die Sprungstelle des *Psi-s Plots* eliminiert und es erlaubt, lokale Extrema als Wendepunkte der Linie in der Geometrieebene aufzufassen.

Die Unterschiede der beiden Repräsentationen werden in Abbildung 5.15 anhand eines Beispiels aufgezeigt. Es wird deutlich, dass die beiden Parametrisierungen über weite Strecken deckungsgleich sind. Einzig beim Vorliegen einer zirkulären Ausgangskurve, beispielsweise einer Spirale (Abbildung 5.16), unterscheiden sich die Beschreibungen durch

<sup>4</sup>Der gesamte Definitionsbereich von  $\alpha$  reicht von  $-\infty$  bis  $\infty$ .

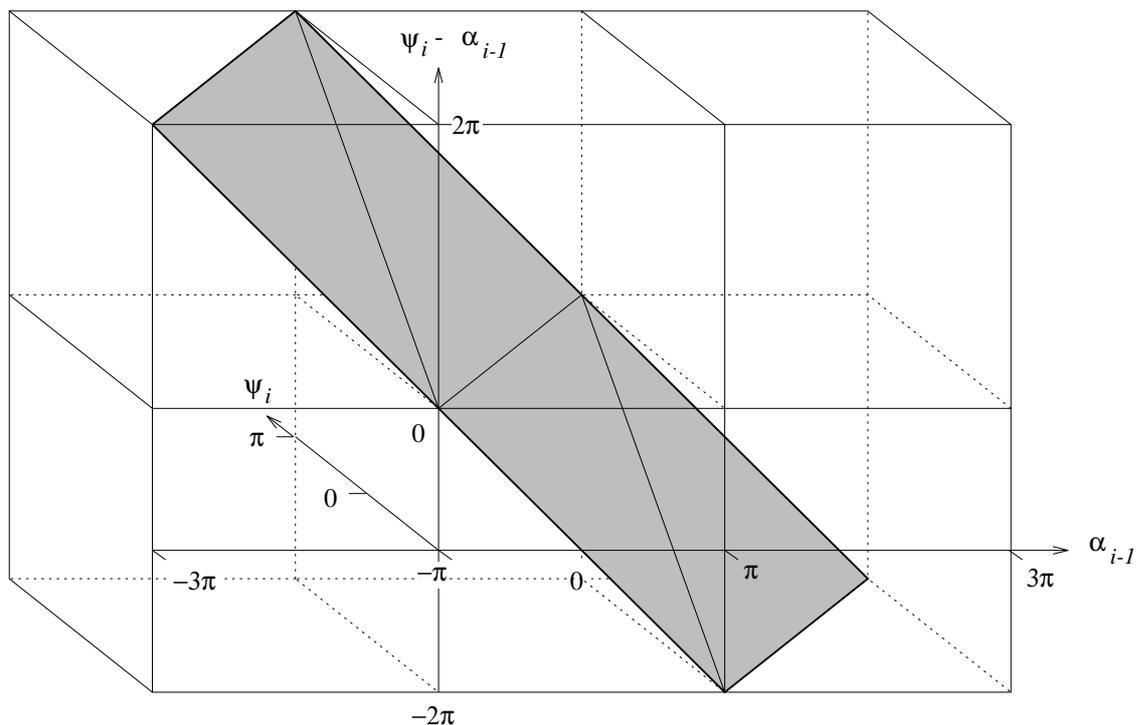


Abbildung 5.11: Darstellung der durch  $\psi_i - \alpha_{i-1}$  aufgespannten Ebene über dem  $\alpha$ -Intervall  $[-3\pi, 3\pi]$ . Siehe Text für weitere Erläuterungen.

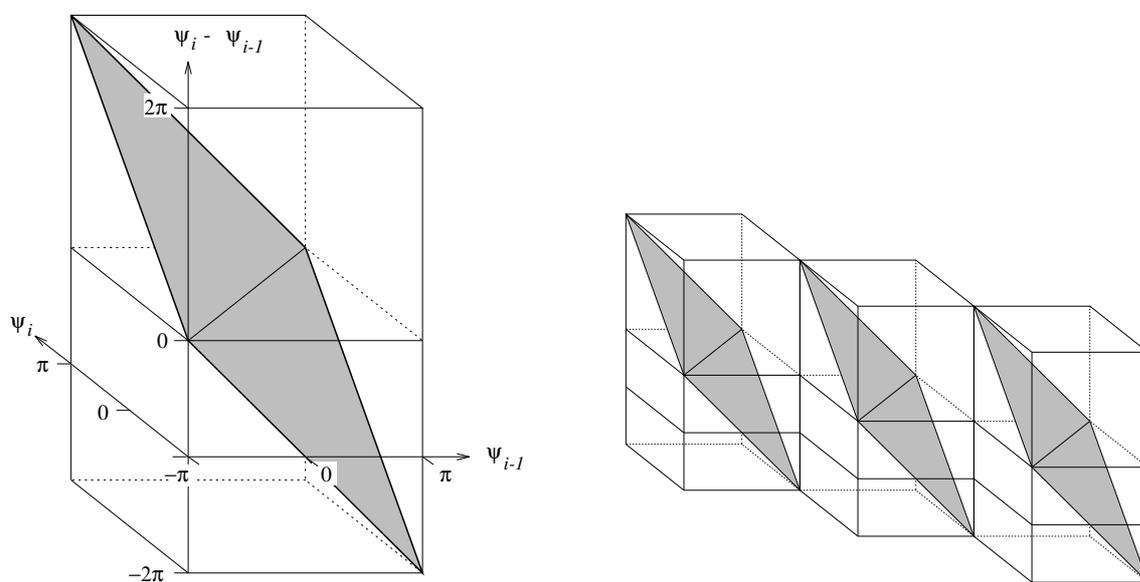


Abbildung 5.12: Links: Darstellung der durch  $\gamma = \psi_i - \psi_{i-1}$  aufgespannten Ebene. Der Richtungswinkel  $\gamma$  variiert zwischen  $-2\pi$  und  $2\pi$ . Rechts: Sprungstelle an der Stelle  $-\pi, \pi$  bzw.  $\pi, -\pi$  mit Amplitude  $4\pi$ . Die Dimensionen entsprechen der Figur links.

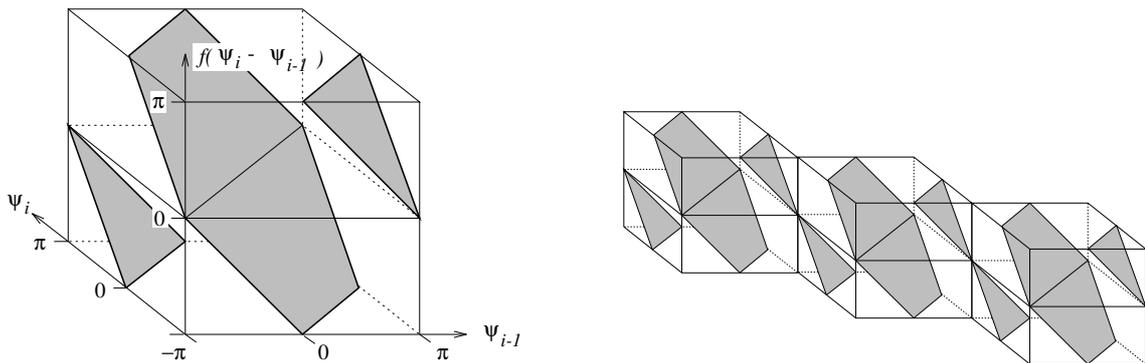


Abbildung 5.13: Links: Die Ebene aus Abb. 5.12 wird derart transformiert, dass der Wertebereich von  $f(\psi_i - \psi_{i-1})$  dem Intervall  $]-\pi, \pi]$  entspricht. Rechts: Die Sprungstelle an der Stelle  $-\pi, \pi$  bzw.  $\pi, -\pi$  konnte eliminiert werden. Stattdessen findet sich eine plausible Diskontinuität mit Amplitude  $2\pi$  an Stellen, wo gilt  $|\psi_i - \psi_{i-1}| = \pi$ . In solchen Fällen weisen zwei aufeinanderfolgende Segmente die entgegengesetzte Richtung auf. Die Dimensionen entsprechen der Figur links.

eine *t-alpha* bzw. *Psi-s* Repräsentation grundlegend voneinander. Das Beispiel der Spirale verdeutlicht, dass mit dem Übergang vom *Psi-s Plot* zum *t-alpha Plot* mit den Nachteilen auch gleich ein kleiner Vorteil verschwindet: der Wertebereich der Funktion ist nicht mehr auf ein bestimmtes Intervall beschränkt. Da Spiralförmigkeiten in der Kartographie relativ selten sind, werden die *alpha*-Werte in der Praxis aber kaum das Intervall  $[-2\pi, 2\pi]$  verlassen.

### 5.4.3 Rücktransformation in die $x, y$ -Ebene

#### Rücktransformation eines unveränderten *t-alpha Plots*

Die Rücktransformation geht in zwei Schritten vor sich. Zuerst wird aufgrund des *t-alpha Plots* die normalisierte Linie rekonstruiert und diese dann an ihre ursprüngliche Lage in der  $x, y$ -Ebene transformiert. Weil in einem *t-alpha Plot* keine explizite Lageinformation abgespeichert ist, sondern nur relative Richtungswinkel in bezug auf das vorangehende Liniensegment, muss die Linie rekursiv aufgebaut werden. Der Endpunkt des zuletzt rekonstruierten Liniensegments bildet den Ausgangspunkt  $P$  für die Rekonstruktion des aktuellen Segments. Die Berechnung der relativen Koordinaten des aktuellen Segments in bezug auf  $P$  gestaltet sich äusserst einfach ( $s$ : Abtastschrittweite):

$$(5.9) \quad \begin{aligned} \Delta y &= s \sin(\alpha) \\ \Delta x &= s \cos(\alpha) \end{aligned}$$

Dass es sich bei  $\alpha$  um kumulierte Richtungswinkel handelt, dessen Wert demzufolge nicht im Intervall  $]-\pi, \pi]$  liegen muss, stellt kein Problem dar, da Sinus und Cosinus zyklische Funktionen sind.

Eine dermassen rekonstruierte Linie enthält viele redundante Datenpunkte<sup>5</sup>, weil sie ja auf einer abgetasteten Version der Ursprungslinie beruht. Jedes gerade Liniensegment, das länger als die Abtastschrittweite ist, enthält solche nunmehr überflüssigen Punkte. Solche

<sup>5</sup>vorausgesetzt, die Schrittweite wurde genügend klein gewählt.

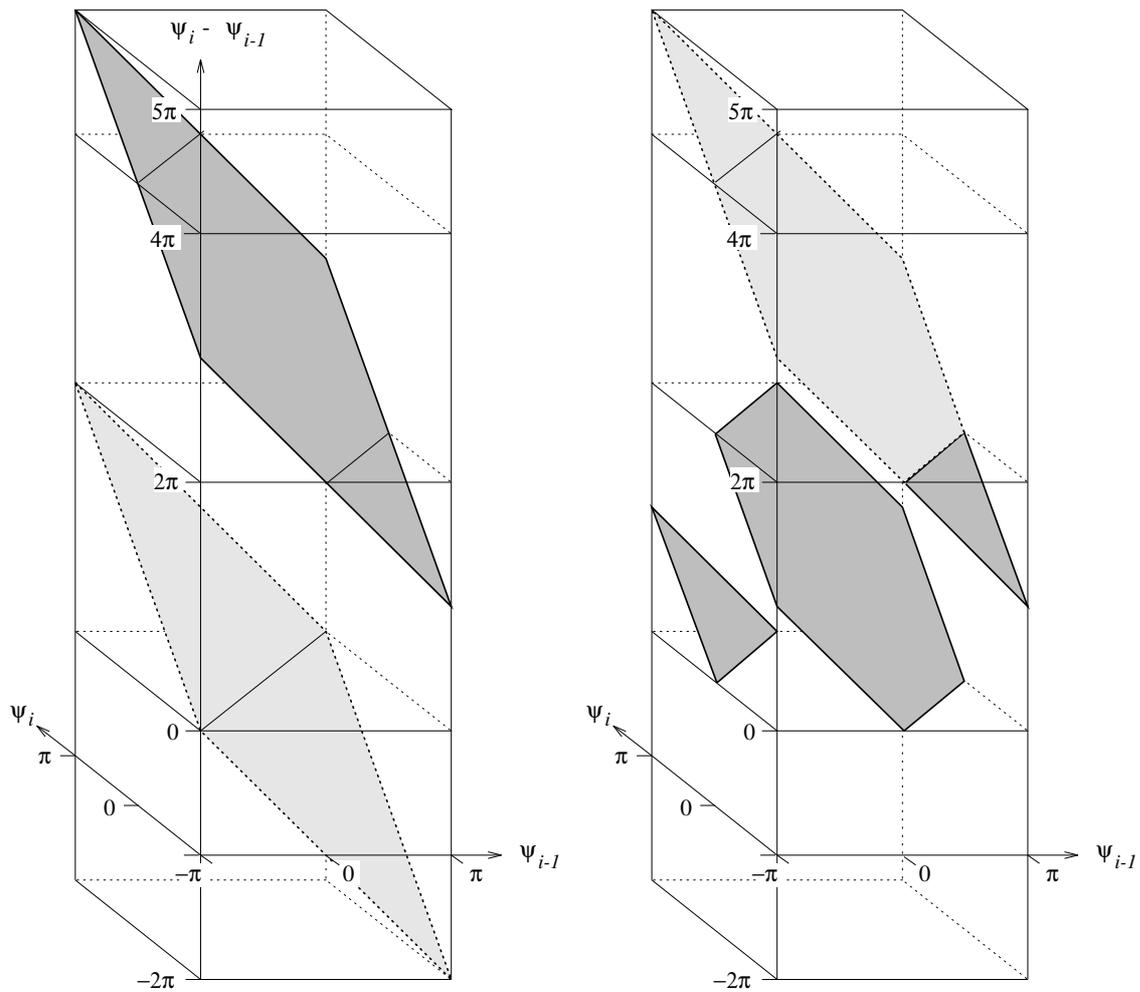


Abbildung 5.14: Darstellung der Transformation  $(\psi_i - \alpha_{i-1} + 3\pi) \bmod 2\pi - \pi$ . Links: Auswirkungen der Verschiebung um  $+3\pi$ . Rechts: Der Modulo-Operator subtrahiert ganze Vielfache von  $2\pi$  vom jeweiligen Funktionswert. Eine Verschiebung um  $-\pi$  (hier nicht dargestellt) führt schliesslich zu den gewünschten Funktionswerten, wie sie in Abb. 5.13 dargestellt sind.

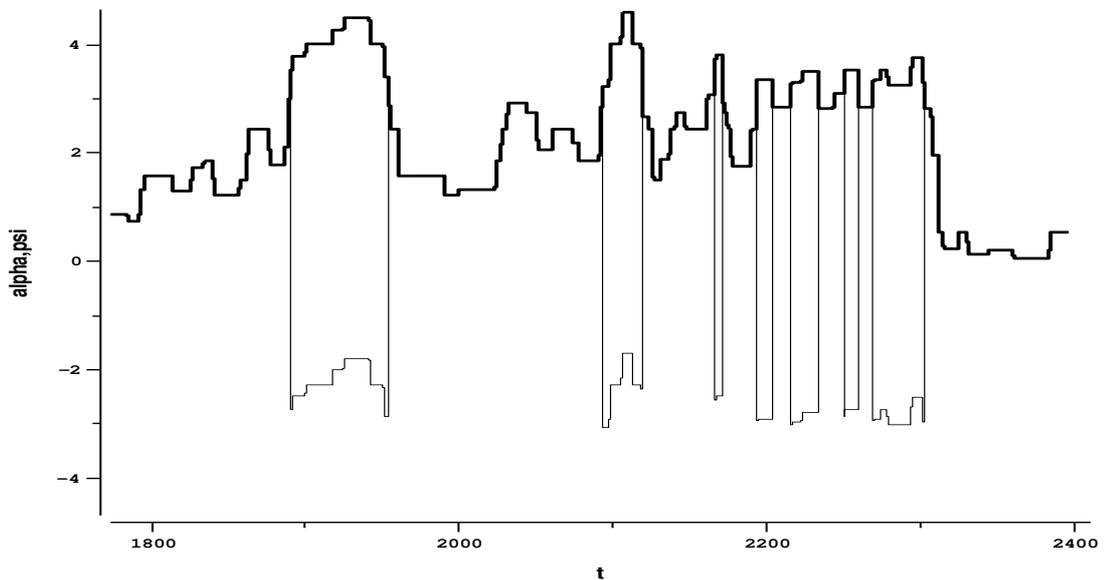


Abbildung 5.15: *Psi-s Plot* (dünne Linie) und *t-alpha Plot* (fette Linie) eines Teils der *normalisierten Linie 501ungen*. Der gewählte Ausschnitt umfasst den Linienabschnitt  $P'E'$  in Abb. 4.1.

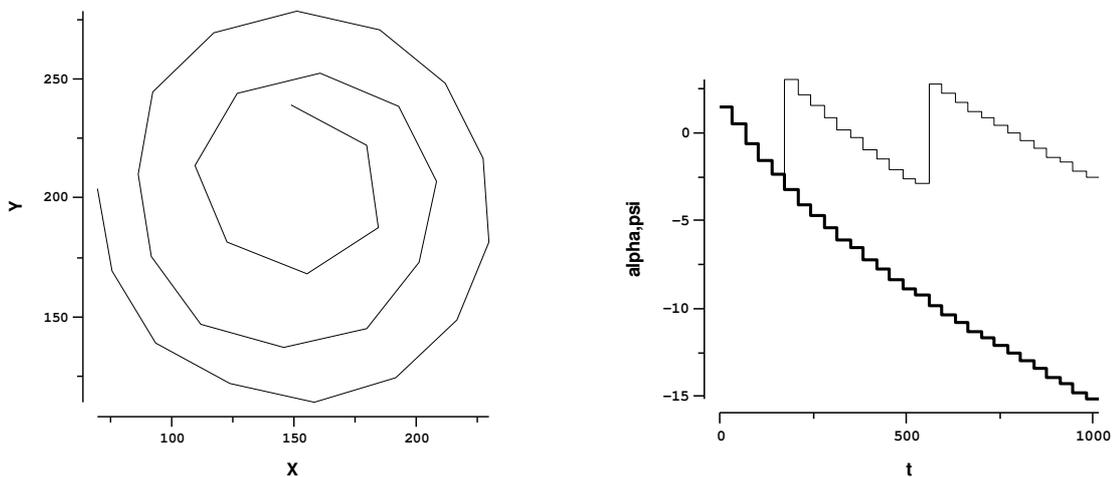


Abbildung 5.16: Eine spiralförmige Ausgangskurve (links) weist keinen Wendepunkt auf und deren *t-alpha Plot* demzufolge auch keine lokalen Extrema (fette Linie, rechts). Im Gegensatz zum *Psi-s Plot* (dünne Linie, rechts) kann *alpha* aber ausserhalb des Intervalls  $]-\pi, \pi]$  liegen.

Winkeltoleranz $T$ im Bogenmass	Anzahl Punkte		
	gefiltert	abgetastet	original
0	2397	2397	281
0.000001	1166		
0.00001	550		
0.0001	550		
0.001	546		
0.01	528		
0.1	368		
0.2	259		
0.3	186		
0.4	129		
0.5	82		
0.6	55		
0.7	36		
0.8	25		
0.9	11		
1.0	5		
1.1	4		
1.2	3		
1.3	2		

Tabelle 5.1: Anzahl Punkte, welche die aus einem  $t$ - $\alpha$  Plot rekonstruierte Linie  $501$ ungen in Abhängigkeit vom Winkeltoleranzwert  $T$  aufweist. Es ist zu beachten, dass zwischen den  $T$ -Werten 0.1 und 0.2 ein Übergang von exponentiellen zu linearen  $\Delta T$  stattfindet.

redundanten Punkte zeichnen sich dadurch aus, dass sich ihr  $\alpha$ -Wert nicht von demjenigen des nachfolgenden Segments unterscheidet. Dies bedeutet, dass es sehr einfach ist, solche Punkte bereits während der Rekonstruktion der Linie zu unterdrücken. Werden aufeinanderfolgende  $\alpha$ -Werte nicht nur auf Gleichheit getestet, sondern, allgemeiner, auf die Unterschreitung einer Differenztoleranz  $T$ , dann handelt es sich indirekt um die Implementation eines einfachen Winkeltoleranzalgorithmus, wie er von McMaster und Shea (1992, 74-76) beschrieben wurde. Für die Rekonstruktion aus einem  $t$ - $\alpha$  Plot bedeutet dies, dass Punkt  $i$  dann unterdrückt wird, wenn gilt:

$$(5.10) \quad |\alpha_{i+1} - \alpha_i| < T, \quad 0 < T < \pi$$

Mit einer Winkeltoleranz  $T = 0$  werden somit alle Abtastpunkte in die rekonstruierte Linie übernommen. Eine Toleranz nahe bei Null führt dazu, dass lediglich redundante Punkte unterdrückt werden. Grössere Werte von  $T$  führen schliesslich zu einer vereinfachten Version der Linie. In Tabelle 5.1 ist anhand eines Beispiels zusammengestellt, wieviele Stützpunkte eine rekonstruierte Linie in Abhängigkeit von der Winkeltoleranz aufweist. Abbildung 5.17 zeigt die Linie, wie sie sich nach einer Rekonstruktion mit Toleranzwert 0.6 präsentiert. Wie erwartet, werden für  $T = 0$  gleichviele Rekonstruktionpunkte wie Abtastpunkte erhalten. Für  $T = 10^{-6}$  werden offensichtlich nicht alle redundanten Punkte eliminiert, was auf numerische Ungenauigkeiten zurückzuführen ist. Für  $T = 10^{-5}$  werden jedoch alle überflüssigen Punkte unterdrückt. Es ist durchaus korrekt, dass immer noch ungefähr doppelt so viele

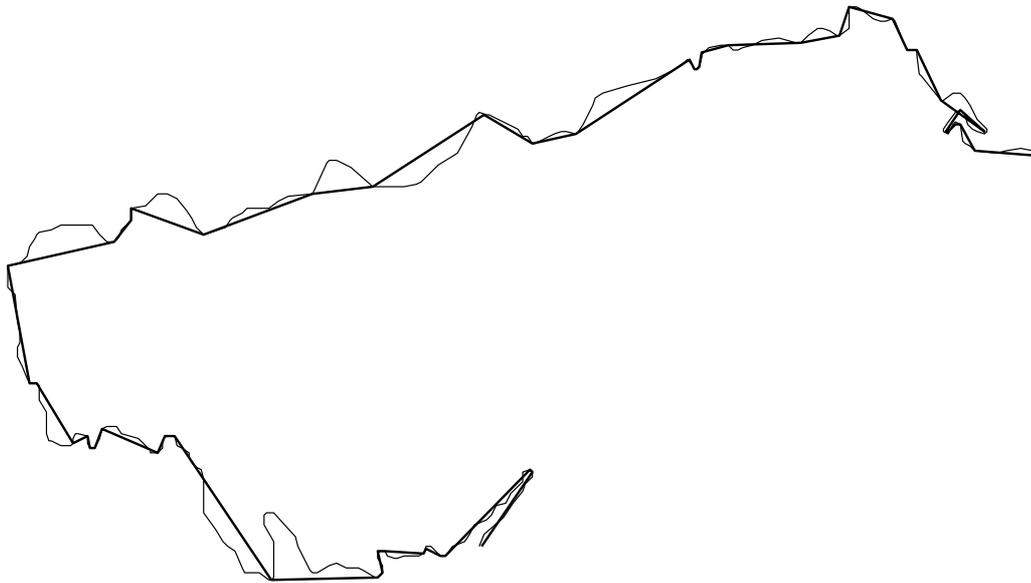


Abbildung 5.17: Die mit einer Winkeltoleranz  $T = 0.6$  aus einem *t-alpha Plot* rekonstruierte Linie *501ungen*.

Rekonstruktionspunkte wie Originalpunkte erhalten werden, weil die ursprünglichen Punkte ja nicht zur Menge der Abtastpunkte gehören (siehe 3.2). Somit verbleiben auch nach der Filterung von redundanten Punkten jeweils zwei rekonstruierte Punkte pro Ausgangspunkt (Abbildung 5.18 links). Der vergrößerte Ausschnitt in Abbildung 5.18 rechts zeigt, dass sich der Winkeltoleranzalgorithmus schlecht zur Linienvereinfachung eignet, weil es zu Überschneidungen kommen kann.

### Rücktransformation eines veränderten *t-alpha Plots*

Im vorherigen Abschnitt wurde nicht näher darauf eingegangen, an *welchem Ort* in der  $x, y$ -Ebene die normalisierte Linie aus einem *t-alpha Plot* rekonstruiert wird. Diese Frage ist nicht unerheblich, weil mit einer *t-alpha* Repräsentation keinerlei Lageinformation abgespeichert wird. Das bedeutet, dass als Ausgangspunkt der rekursiven Rekonstruktion ein arbiträrer Punkt in der Ebene gewählt werden muss, z.B. der Ursprung des Koordinatensystems. Erst wenn die Kurve fertig rekonstruiert ist, kann ihr Schwerpunkt berechnet werden und dieser sodann an den Ursprung verschoben werden (dorthin wurde die Ausgangslinie im Normalisierungsschritt ja verschoben). Die Kurve wird dann in ihre ursprüngliche Lage zurückrotiert und schliesslich an die Ausgangslage in der  $x, y$ -Ebene verschoben. Dieser Vorgang ist in Abbildung 5.19 schematisch dargestellt. Dieses Vorgehen ist unproblematisch, wenn der *t-alpha Plot* nach dessen Berechnung nicht verändert wurde. Sobald diese Voraussetzung nicht mehr gegeben ist<sup>6</sup>, kann nicht mehr garantiert werden, dass der Schwerpunkt von ursprünglicher Linie und rekonstruierter Linie am selben Ort liegen. Eine

<sup>6</sup>Da beabsichtigt ist, *Generalisierungsoperatoren* oder *-regeln* basierend auf einer *t-alpha* Repräsentation zu entwickeln, wird diese Voraussetzung zwangsläufig nicht erfüllt werden können.

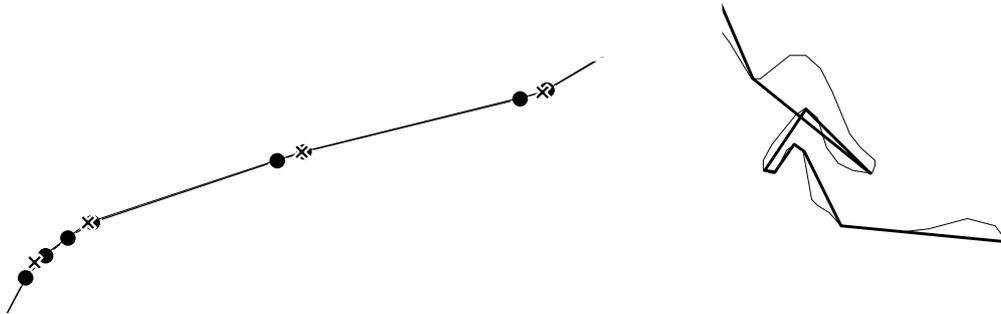


Abbildung 5.18: Vergrosserte Ausschnitte aus Abb. 5.17. Links: Auch nach der Eliminierung redundanter Punkte verbleiben zwei Rekonstruktionspunkte (Kreissignatur) pro ursprünglichen Stützpunkt (Kreuzsignatur). Rechts: Der Winkeltoleranzalgorithmus eignet sich schlecht zur Linienvereinfachung, weil es zu Überschneidungen kommen kann.

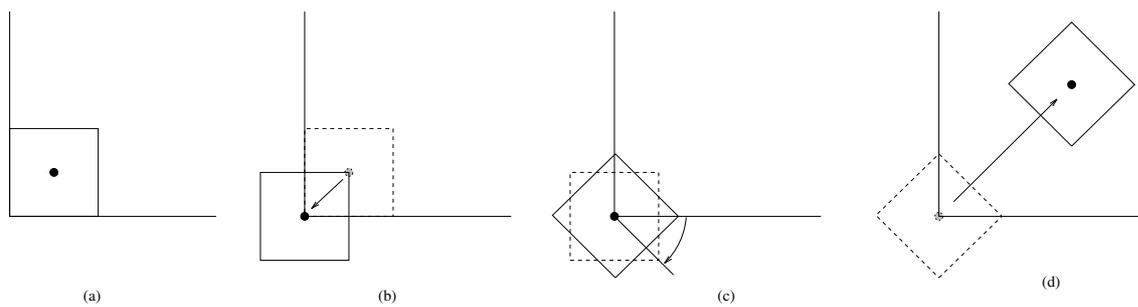


Abbildung 5.19: Schematische Darstellung der Rekonstruktion einer Kurve aus einem *t-alpha Plot*. (a) Rekursive Rekonstruktion am Ursprung und anschließende Berechnung des Schwerpunkts. (b) Verschiebung des Schwerpunkts an den Ursprung. (c) Rotation zurück in die ursprüngliche Hauptrichtung. (d) Verschiebung zurück an die ursprüngliche Lage.

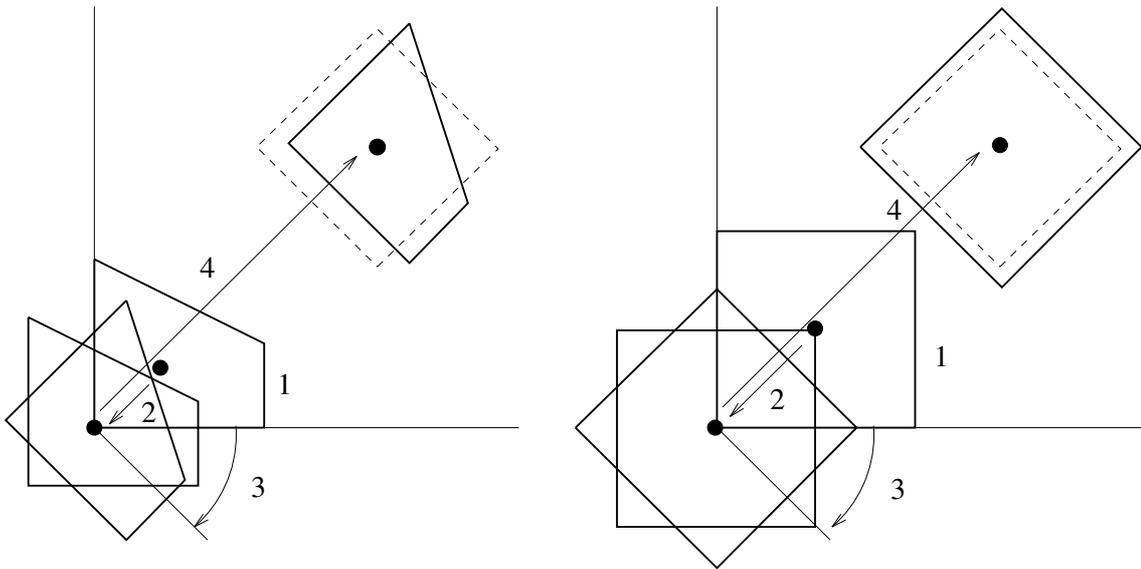


Abbildung 5.20: Schwerpunktbasierte Rekonstruktion gemäss Abb. 5.19. Links: Der *t-alpha Plot* wurde derart verändert, dass der rekonstruierten Kurve eine “Ecke” fehlt (*lokale Veränderung*). Rechts: Die Veränderung der *alpha*-Werte führt zu längeren Kanten des Quadrats (*globale Veränderung*). Die strichlierten Linien markieren die Lage des ursprünglichen, unveränderten Quadrats.

Veränderung führt zu einem anderen Rotationszentrum und zu einem anderen Ansatzpunkt für den Verschiebungsvektor. Die Kurve kann somit nicht mehr an ihre “korrekte” Lage in der  $x, y$ -Ebene zurücktransformiert werden. Dabei stellt sich allerdings die Frage, welches die “korrekte” Lage einer veränderten Kurve sein soll. Dieser Ort ist abhängig von der Art der Veränderung, welche die Kurve erfährt:

**Lokale Veränderungen:** Sie treten z.B. beim Editieren einzelner Waveletkoeffizienten auf (siehe Kapitel 7) und können sich in der Elimination einzelner Kleinformen der Kurve äussern. Links in Abbildung 5.20 ist schematisch dargestellt, zu welchem Resultat eine Rekonstruktion nach oben beschriebenem Schema führt.

**Globale Veränderungen:** Sie treten z.B. bei der Rekonstruktion eines *t-alpha Plots* aus einer beschränkten Anzahl Fourierkoeffizienten auf (siehe Kapitel 6). Abbildung 5.20 (rechts) vermittelt einen Eindruck, zu welchem Resultat die schwerpunktbasierte Rekonstruktion global veränderter Kurven führt.

Abbildung 5.20 kann entnommen werden, dass die “korrekte” Lage von der Art der Veränderung abhängig ist. Bei lokalen Veränderungen resultiert die schwerpunktbasierte Rekonstruktion in einer unbefriedigenden Rücktransformation der Kurve, während im Falle einer globalen Veränderung ein ansprechendes Resultat erzielt wird.

Beim Vorliegen lokaler Veränderungen kann nur ein gutes Resultat erzielt werden, wenn zusammen mit der *t-alpha*-Repräsentation die Lage des Anfangspunktes gespeichert wird<sup>7</sup>, von welchem die rekursive Rekonstruktion der normalisierten Linie ausgehen soll. Dieser

<sup>7</sup>Bei offenen Kurven wird es sich dabei typischerweise um den Anfangspunkt der normalisierten Linie handeln.

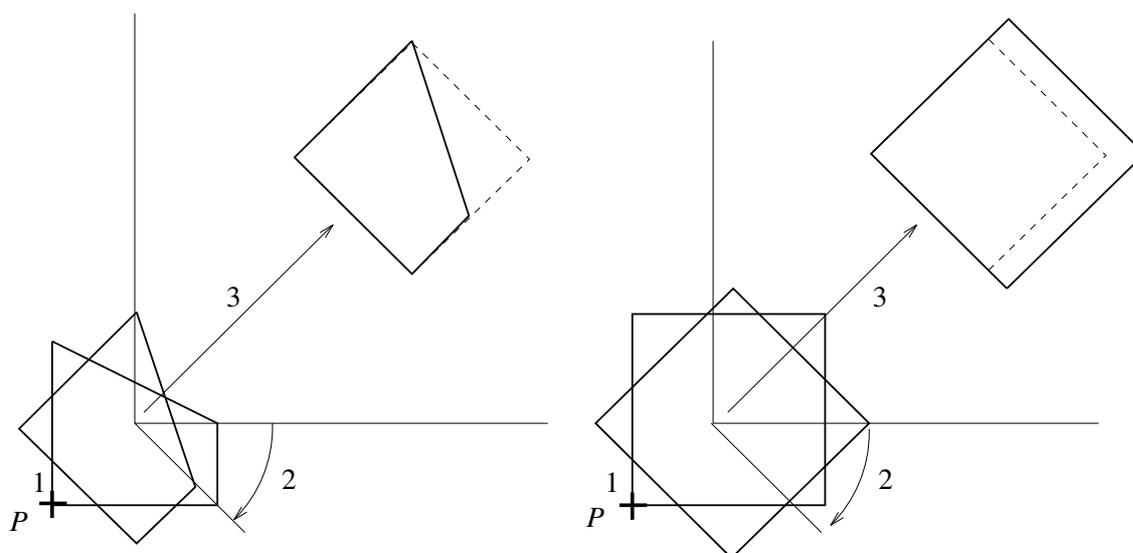


Abbildung 5.21: Fixpunktbasierte Rekonstruktion aus einem  $t$ - $\alpha$  Plot. Der rekursive Aufbau der normalisierten Kurve beginnt am Punkt  $P$ . Links: Fixpunktbasierte Rekonstruktion und Rücktransformation einer *lokal* veränderten Kurve. Rechts: Dasselbe für eine *global* veränderte Kurve. Die strichlierten Linien markieren die Lage des ursprünglichen, unveränderten Quadrats.

Ansatz ist in Abbildung 5.21 schematisch dargestellt. Wie nicht anders zu erwarten, führt diese Methode jedoch zu einem unbefriedigenden Ergebnis für global veränderte Kurven.

Folglich geht es bei der Wahl der Rekonstruktionsmethode um die Frage, welcher Art die am häufigsten auftretenden Veränderungen sind. Repräsentationsbedingt werden dies *globale Veränderungen* sein, auch wenn die Eingriffe in die  $t$ - $\alpha$ -Repräsentation lediglich lokaler Natur sind. Dies deshalb, weil keine Linienverkürzung möglich ist. Durch lokale Veränderungen der  $\alpha$ -Werte, welche auf die Eliminierung einer Kleinform abzielen, werden nur die Richtungswinkel von Segmenten der abgetasteten Linie verändert. Es ist dadurch jedoch nicht möglich, Segmente zu verkürzen oder gar zu eliminieren. Folglich wirken sich derartige Veränderungen auch auf die Lage der übrigen Liniensegmente aus. Aus diesem Grund wurde in *REPTILE* die schwerpunktbasierte Rekonstruktionsmethode implementiert. In Abbildung 5.22 findet sich abschliessend ein Beispiel, welches diesen Entscheid untermauern soll.

#### 5.4.4 Wahl der Abtastschrittweite

In Abschnitt 3.3 wurde darauf hingewiesen, dass es sehr schwierig ist, die Digitalisierfehler einer Kurve in der  $x, y$ -Ebene auszumachen. Im  $t$ - $\alpha$  Plot gestaltet sich dies hingegen sehr einfach. Digitalisierfehler weisen grosse Abweichungen im Richtungswinkel bezogen auf nur wenige Abtastsegmente auf, was sie im  $t$ - $\alpha$  Plot als Ausreisser erscheinen lässt (Abbildung 5.23 links). Die Schrittweite kann dann inkrementell vergrössert werden, bis die Sprungstelle verschwindet (Abb. 5.23 rechts).

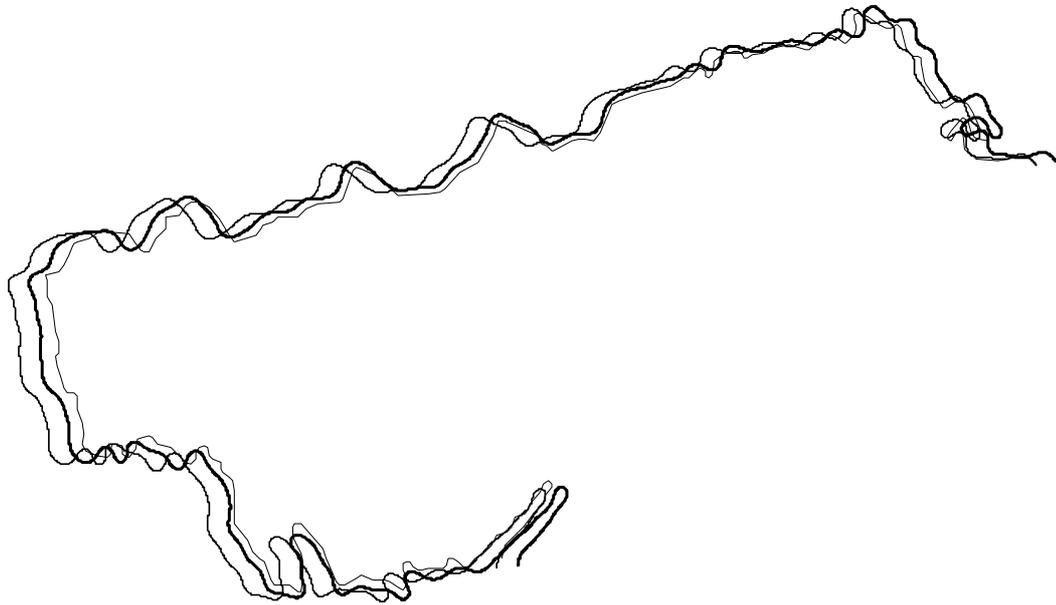


Abbildung 5.22: Linie *501ungen* (dünne Linie) und zwei – via *t-alpha Plot* – aus lediglich 150 Fourierkoeffizienten rekonstruierte Versionen der Linie. Bei der halb fetten Linie handelt es sich um eine fixpunktbasierte (= Anfangspunkt *A*) Rekonstruktion aus *t-alpha*, bei der fetten Linie um eine schwerpunktbasierte Rekonstruktion.

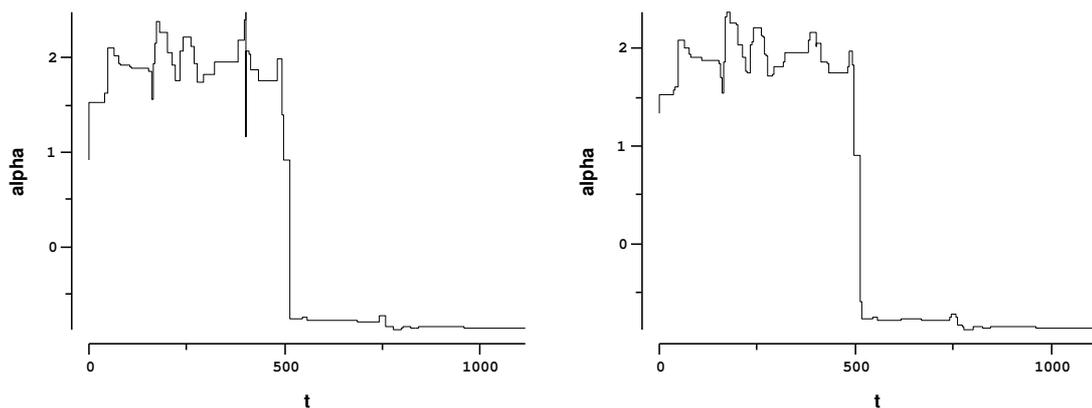


Abbildung 5.23: Links: Der mutmassliche Digitalisierfehler in Linie *536ungen* an der Stelle *P* (vgl. Abb. 3.5) kommt im *t-alpha Plot* mit einer Länge von nur einer Einheit deutlich zum Vorschein. Rechts: Eine Erhöhung der Schrittweite auf 3 Einheiten bringt die Sprungstelle *P* zum Verschwinden.



# Kapitel 6

## Fourierreihen

Jetzt, da wir eine Linie in der Kartenebene als Funktion im Parameterraum darstellen können (Kapitel 5), möchten wir diese Funktion unter Zuhilfenahme klar definierter Bausteine – wie in Abschnitt 1.2 gefordert – approximieren. Wie in diesem Kapitel gezeigt wird, handelt es sich im Falle von Fourierreihen bei diesen Bausteinen um die Cosinusgrundschiwingung und deren Oberschwingungen (“Harmonische”).

### 6.1 Konzept der Fourierreihe

Diese Einführung in die Theorie von Fourierreihen (6.1) basiert zu einem grossen Teil auf dem exzellenten Buch von Solymar (1988). Der Autor erklärt in einer leicht verständlichen Sprache und anhand vieler Beispiele das Konzept von Fourierreihen und beschreibt die nötigen Techniken zur Umsetzung in die Praxis.

Mathematiker, Ingenieure und Politiker nehmen noch heute in Anspruch, der “Vater” der Fourierreihe, *Jean Baptiste Joseph Fourier* (\*1768, †1830), sei einer aus ihrer Gilde gewesen. Zieht man lediglich in Betracht, dass Fourier an der bekannten Ingenieurschule *École Polytechnique* in Paris lehrte, könnte er durchaus als Ingenieur bezeichnet werden. Er führte jedoch den Titel eines Analysisprofessors, weshalb auch der Anspruch der Mathematiker durchaus legitim erscheint. Politiker schliesslich berufen sich auf die Zeit, als Fourier Gouverneur von Niederägypten war und später verschiedene administrative Posten unter Napoleon inne hatte. In der Tat handelt es sich bei der Fourieranalyse um eine sehr wichtige Errungenschaft, ohne die viele Ingenieurwissenschaften und viele Bereiche der angewandten Forschung kaum mehr auskommen könnten. Fourierreihen sind z.B. sehr nützlich für die Analyse von Differentialgleichungen, weil die Ableitungen der Basisfunktionen in einfacher Art und Weise in Relation zueinander gebracht werden können. Die *zweidimensionale* Fouriertransformation ist auch in der Fernerkundung von Bedeutung, um Filterungen im Frequenzraum vorzunehmen oder zur Charakterisierung von Geländetypen aufgrund ihrer Fourierrepräsentation (Mather 1987, 239).

#### 6.1.1 Die Behauptung

Zu Beginn tönt die Grundthese ziemlich gewagt. Es wird nämlich nichts weniger behauptet, als dass es möglich sei, eine beliebige periodische Funktion (z.B. jene in Abbildung 6.1 mit Periode  $2\pi$ ) aus der Grundschiwingung der Cosinusfunktion (Abb. 6.1b), der Grundschiwin-

gung der Sinusfunktion (Abb. 6.1c), der zweiten harmonischen<sup>1</sup> Cosinuskomponente (Abb. 6.1d), der zweiten harmonischen Sinuskomponente (Abb. 6.1e), der dritten harmonischen Cosinuskomponente (Abb. 6.1f), der dritten harmonischen Sinuskomponente (Abb. 6.1g) und so weiter bis zur Unendlichkeit, beliebig genau zu approximieren. Wir brauchen lediglich den richtigen Betrag jeder dieser Komponenten zu nehmen, diese aufzusummieren und schliesslich zu diesem Resultat eine vernünftig gewählte Konstante hinzuzufügen und schon würden wir die in Abb. 6.1a gezeigte Kurve erhalten. Der Grund, weshalb diese Behauptung so gewagt klingt, ist der, dass sie sich nicht mit dem gesunden Menschenverstand in Einklang bringen lässt. Es kann nämlich argumentiert werden, dass, um eine Funktion  $f(x)$  zwischen  $x_1$  und  $x_2$  zu approximieren (vgl. Abb. 6.2a), wohl ein kleiner Abschnitt einer Sinuskurve benötigt wird (in Abb. 6.2b durch eine starke Linie dargestellt), aber sicher nicht der Rest der Funktion (punktiert dargestellt), weil die Addition dieses Teils bloss die Approximation ausserhalb des Intervalls  $[x_1, x_2]$  verschlechtert. Es kann an dieser Stelle jedoch versichert werden, dass es tatsächlich möglich ist, durch sukzessives Aufaddieren von Oberschwingungen (Harmonischen) in den richtigen Proportionen, eine beliebige periodische Funktion mit beliebiger Genauigkeit zu approximieren. Der Beweis dieser Konvergenz soll hier nicht geliefert werden, er kann in jedem Lehrbuch über Fourierreihen nachgelesen werden (Hanna und Rowland 1990; Brown und Churchill 1993). Stattdessen soll die Behauptung im nächsten Abschnitt anhand eines Beispiels erklärt werden.

### 6.1.2 Ein Beispiel

Gegeben sei die in Abbildung 6.3 dargestellte periodische Funktion, welche durch ihre Fourierreihe approximiert werden soll. Es handelt sich dabei um eine sehr simple Funktion: sie weist den Wert 1 zwischen  $-\pi/4$  und  $\pi/4$  auf und ist Null ausserhalb dieses Bereichs. Die Periode ist  $2\pi$ . Ist es nun möglich, eine solche Funktion durch eine Fourierreihe zu approximieren? Ja, es ist möglich. Dies soll nun Schritt für Schritt gezeigt werden. Zuerst muss ein Periodenintervall dieser Funktion ausgewählt werden. Die Periode muss natürlich  $2\pi$  betragen, aber es muss nicht der Abschnitt der Funktion zwischen 0 und  $2\pi$  gewählt werden. Aus Symmetriegründen entscheiden wir uns für das Intervall zwischen  $-\pi$  und  $\pi$ , so dass wir eine *gerade Funktion* erhalten. Wir haben es mit einer geraden Funktion  $y = f(x)$  zu tun, wenn gilt:

$$(6.1) \quad f(a) = f(-a), \quad \forall a$$

Die Cosinusfunktion  $y = \cos x$  ist ein Beispiel für eine gerade Funktion. Mit anderen Worten: eine gerade Funktion liegt dann vor, wenn die positiven Funktionswerte an der  $y$ -Achse gespiegelt werden können, um die negativen Funktionswerte zu erhalten. Eine *ungerade Funktion*  $y = f(x)$  liegt dann vor, wenn gilt:

$$(6.2) \quad f(-a) = -f(a), \quad \forall a$$

Die Sinusfunktion  $y = \sin x$  ist ein Beispiel für eine ungerade Funktion. Um die negativen Funktionswerte zu erhalten, müssen die positiven Funktionswerte zuerst an der  $y$ -Achse und dann an der  $x$ -Achse gespiegelt werden. Da es sich im vorliegenden Beispiel nach der Wahl

---

<sup>1</sup>Die Fourieranalyse basiert darauf, eine periodische Funktion in ihre Grundschwingung (1. Harmonische) und deren Oberschwingungen (2., 3., ... Harmonische) zu zerlegen. Die Schwingungsdauer (Periodenlänge) der Oberschwingungen ergibt sich durch Halbieren, Dritteln, usw. derjenigen der Grundschwingung.

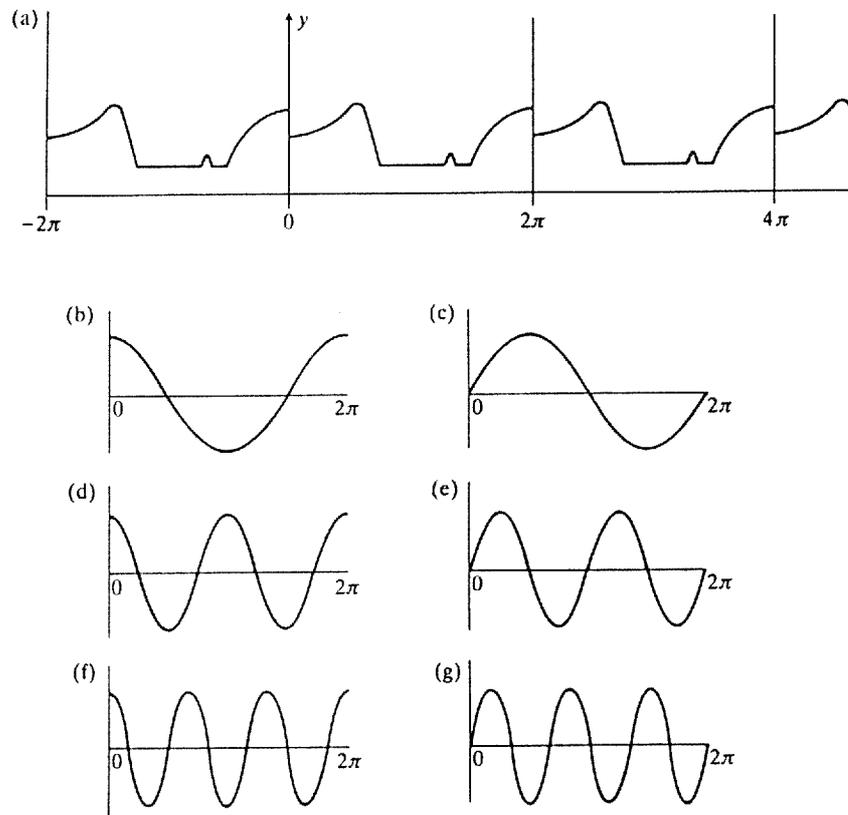


Abbildung 6.1: (a) Eine beliebige periodische Funktion mit Periode  $2\pi$ ; (b)-(g) Grundschwingung der Sinus- und Cosinusfunktion (Periode  $2\pi$ ) sowie deren 1. und 2. Harmonische (Periode  $\pi$ , bzw.  $2\pi/3$ ). Aus Solymar (1988, 7).

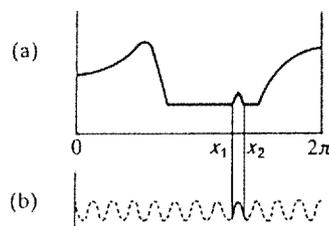


Abbildung 6.2: (a) Die periodische Funktion aus Abb. 6.1a. (b) Eine Approximationsfunktion im Intervall  $[x_1, x_2]$ . Aus Solymar (1988, 8).

des Intervalls  $[-\pi, \pi]$  um eine gerade Funktion handelt, liegt die Vermutung nahe, dass eine solche Funktion nur aus (ebenfalls geraden) Cosinuskomponenten aufgebaut sein wird und alle Sinuskomponenten Null sein werden. Diese Vermutung wird in Abschnitt 6.1.4 belegt. Die Fourierreihe kann dann geschrieben werden als:

$$(6.3) \quad f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kx$$

Die Funktion  $f(x)$  ist in Abbildung 6.4a dargestellt. Um die Konstante kümmern wir uns gegenwärtig noch nicht. Später wird klar, weshalb sie in dieser speziellen Form geschrieben wurde. Wie finden wir nun aber  $a_k$ , die Amplitude der  $k$ -ten Komponente? Auch darauf wird erst zu einem späteren Zeitpunkt eingegangen. Zuerst soll gezeigt werden, dass es tatsächlich möglich ist, eine periodische Funktion durch Addition sehr vieler individueller Komponenten zu approximieren. Für den Moment soll deshalb einfach angenommen werden, dass es uns gelungen ist, die Amplituden korrekt zu bestimmen. Die  $k$ -te Komponente kann geschrieben werden als

$$(6.4) \quad p_k = a_k \cos kx$$

und die Funktion  $f_n(x)$  wird definiert als

$$(6.5) \quad f_n(x) = \sum_{k=1}^n a_k \cos kx = \sum_{k=1}^n p_k.$$

Bei  $f_n(x)$  handelt es sich somit um die Approximationsfunktion, welche wir erhalten, wenn wir  $n$  Komponenten addieren. Betrachten wir nun  $f_1 = p_1$  (Abb. 6.4b). Dabei handelt es sich um die Cosinusgrundschwingung mit der Periode  $2\pi$ . Es muss zugegeben werden, dass sie der Kurve in Abb. 6.4a nicht sehr nahe kommt. Deshalb addieren wir die zweite Harmonische (mit einer Periode  $\pi$ ), welche in Abb. 6.4c dargestellt ist, dazu. Beim Resultat handelt es sich um  $f_2$ , welches in Abb. 6.4d zu sehen ist. Die Ähnlichkeit ist nach wie vor bescheiden. Deshalb addieren wir auch die dritte Harmonische (Abb. 6.4e) dazu, was in  $f_3$  resultiert. Ein Blick auf die Abbildungen 6.4b, d und f zeigt, dass die Mittelpartie der Funktion schmaler wird. Dies ist eigentlich nicht die Absicht, denn es soll ein rechteckiger Puls und keine Spitze approximiert werden. Aber der Schritt von  $f_2$  zu  $f_3$  sollte eher als ein "Anlaufholen" für den entscheidenden Sprung nach vorne gesehen werden. Sobald wir nämlich die fünfte Harmonische (Abb. 6.4g) addieren, wird der Mittelteil wieder breiter, wie dies aus Abbildung 6.4h ersichtlich ist. Wurde nun vergessen, die vierte Harmonische zu addieren? Nein, es wurde nicht. Die Symmetrieeigenschaften dieser periodischen Funktion sind derart, dass alle  $4n$  Harmonischen (wobei  $n$  eine ganze Zahl ist) eine Amplitude von Null aufweisen. Die Addition der sechsten Harmonischen (Abb. 6.4i) resultiert in  $f_6$ , dargestellt in Abb. 6.4j. Es sollte nun klar geworden sein, was diese höheren Harmonischen bewirken. Sie addieren und subtrahieren jeweils den richtigen Betrag an der richtigen Stelle der Funktion. Abbildung 6.4j erscheint bereits kantiger. Jetzt sollte es mit ein wenig Phantasie möglich sein, einzusehen, dass es durch die Addition von weiteren Komponenten machbar sein sollte, nahe an die Pulsfunktion in Abb. 6.4a heranzukommen. Die individuellen Komponenten werden nun nicht mehr abgebildet, sondern nur noch deren Summe. Mit 9 Harmonischen erhalten wir Abb. 6.4k, mit 13 Harmonischen Abb. 6.4l und mit 17 Harmonischen Abbildung 6.4m. Deutlich zeichnet sich nun ein Muster ab. Durch jede Addition

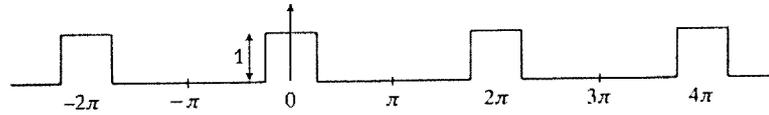


Abbildung 6.3: Eine Reihe rechteckiger Pulse. Aus Solymar (1988, 8).

von jeweils vier Harmonischen verbessert sich die Approximation. Was noch fehlt, ist die Konstante. Wird diese zur Kurve in Abb. 6.4m addiert, erhalten wir die Kurve in Abb. 6.4n, welche bereits eine gute Approximation der Ausgangsfunktion in Abb. 6.4a darstellt. Je mehr Harmonische wir addieren, desto besser wird die Approximation. In Abbildung 6.4o ist die Kurve abgebildet, wie sie durch die ersten 49 Harmonischen beschrieben wird. Es zeigt sich jedoch, dass in der Nähe des plötzlichen Übergangs von 1 nach 0 keine Verbesserung eintritt. Der Überschuss (*Overshoot*) auf der einen Seite und das Defizit (*Undershoot*) auf der anderen Seite machen keine Anstalten, zu verschwinden. Dies ist ein altbekanntes Problem von Fourierreihen, welches an Sprungstellen auftritt und als *Gibbs-Phänomen* bekannt ist. Mit geeigneten Filtern (z.B. einem sog. *Hamming Fenster*) ist es jedoch möglich, das Gibbs-Phänomen zu unterdrücken (Solymar 1988, 54ff, 114ff).

### 6.1.3 Die Formeln

An dieser Stelle soll auf die Herleitung der Formeln zur Berechnung der Fourierkoeffizienten z.B. in Solymar (1988). Wir beschränken uns hier auf die Wiedergabe der massgeblichen Formeln. Die Fourierreihe für den allgemeinen Fall einer Funktion mit *beliebiger* Periode  $T$  lässt sich schreiben als:

$$(6.6) \quad f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos \frac{2\pi kt}{T} + \sum_{k=1}^{\infty} b_k \sin \frac{2\pi kt}{T}$$

Die Cosinuskoeffizienten berechnen sich durch:

$$(6.7) \quad a_k = \frac{2}{T} \int_0^T f(t) \cos \frac{2\pi kt}{T} dt$$

Hier zeigt sich nun, weshalb die Konstante in Gleichung 6.6 in der Form  $a_0/2$  geschrieben wird. Weil Formel 6.7 auch für  $k = 0$  gilt, wird keine spezielle Formel zur Berechnung der Konstante benötigt. Die Sinuskoeffizienten berechnen sich analog:

$$(6.8) \quad b_k = \frac{2}{T} \int_0^T f(t) \sin \frac{2\pi kt}{T} dt$$

### 6.1.4 Berücksichtigung von Symmetrieeigenschaften

Die geschickte Wahl des Periodenintervalls im Beispiel in Abschnitt 6.1.2 führte zu einer geraden Funktion, mit dem Vorteil, dass sämtliche Sinuskoeffizienten der Fourierreihe einer solchen Funktion Null sind. Die Begründung dafür soll hier nachgeliefert werden. Die Definitionen für gerade und ungerade Funktionen wurden bereits in den Gleichungen 6.1 und

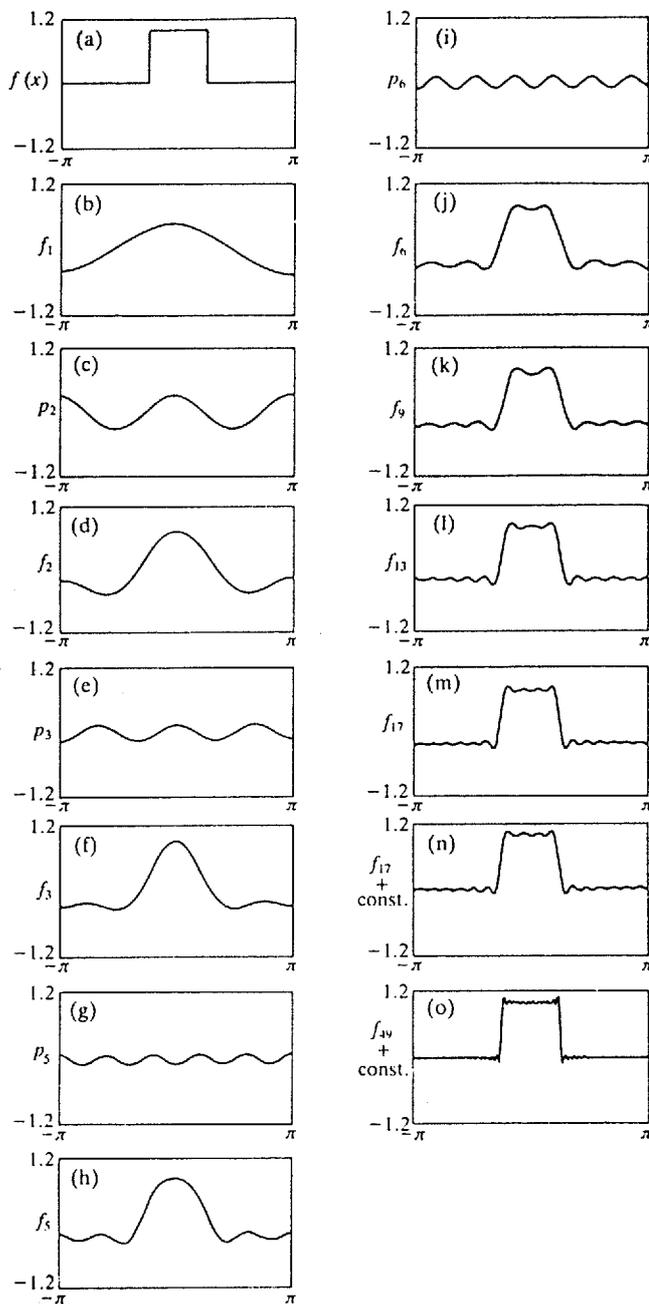


Abbildung 6.4: (a) Eine zwischen  $-\pi/2$  und  $\pi/2$  definierte Funktion ( $f(x) = 1$  für  $|x| < \pi/4$  und Null sonst), welche durch eine Fourierreihe approximiert werden soll. In den Abbildungen (b) bis (o) ist die Summe der ersten  $n$  Harmonischen  $f_n$ , ohne Berücksichtigung der Konstante, bzw. die  $n$ -te Harmonische  $p_n$  abgebildet. Aus Solymar (1988, 10).

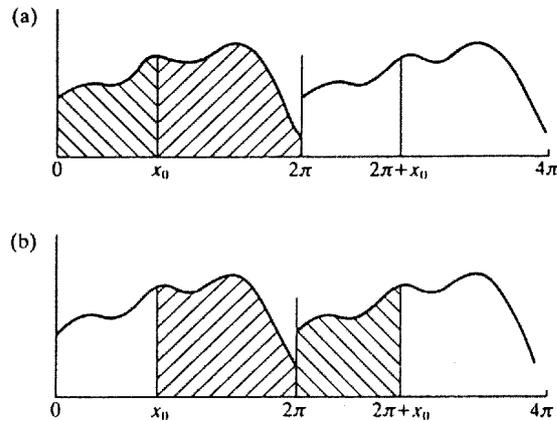


Abbildung 6.5: Die Integration einer periodischen Funktion für eine Periode ist unabhängig vom Startpunkt der Integration. Aus Solymar (1988, 20).

6.2 gegeben. Von Interesse ist aber auch, wie sich die Produkte von geraden und ungeraden Funktionen verhalten. Es gilt:

$$(6.9) \quad \textit{gerade} \times \textit{gerade} = \textit{gerade},$$

$$(6.10) \quad \textit{gerade} \times \textit{ungerade} = \textit{ungerade},$$

$$(6.11) \quad \textit{ungerade} \times \textit{ungerade} = \textit{gerade}.$$

Schliesslich sollen die Regeln erwähnt werden, wonach zwischen symmetrischen Grenzen  $c$  gilt:

$$(6.12) \quad \int_{-c}^c (\textit{ungerade Funktion}) dx = 0$$

$$(6.13) \quad \int_{-c}^c (\textit{gerade Funktion}) dx = 2 \int_0^c (\textit{gerade Funktion}) dx$$

Beide Gleichungen sind offensichtlich, wenn man sich die Fläche unter den Kurven vor Augen führt. Ebenfalls erwähnt werden soll das Theorem, wonach das Resultat der Integration einer periodischen Funktion für eine Periode unabhängig vom Startpunkt der Integration ist. Dies geht aus Abbildung 6.5 deutlich hervor, wo Integrationen von 0 bis  $2\pi$  und von  $x_0$  bis  $x_0 + 2\pi$  gezeigt werden. Ausgerüstet mit all diesen Theoremen können nun vereinfachte Formeln zur Berechnung der Fourierkoeffizienten hergeleitet werden, wenn  $f(t)$  entweder gerade oder ungerade ist. Wir benützen die Formeln 6.7 und 6.8, wählen aber Integrationsgrenzen von  $-T/2$  bis  $T/2$ , was nach dem obigen Theorem ja gestattet ist. Wenn  $f(t)$  gerade ist, haben wir es mit

$$(6.14) \quad a_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos \frac{2\pi kt}{T} dt$$

zu tun. Weil aber sowohl  $f(t)$  als auch  $\cos \frac{2\pi kt}{T}$  gerade Funktionen sind, vereinfacht sich Gleichung 6.14 zu

$$(6.15) \quad a_k = \frac{4}{T} \int_0^{T/2} f(t) \cos \frac{2\pi kt}{T} dt$$

Dies ist noch keine überwältigende Vereinfachung. Für  $b_k$  sieht die Sache allerdings bedeutend besser aus. Es zeigt sich nämlich, dass

$$(6.16) \quad b_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin \frac{2\pi kt}{T} dt = 0$$

wird, weil ja gilt *gerade*  $\times$  *ungerade* = *ungerade* und das Integral einer ungeraden Funktion zwischen symmetrischen Grenzen Null ist.

Wenn  $f(t)$  ungerade ist, führen ähnliche Überlegungen zu

$$(6.17) \quad a_k = 0 \quad \text{und} \quad b_k = \frac{4}{T} \int_0^{T/2} f(t) \sin \frac{2\pi kt}{T} dt$$

Dies bedeutet, dass wir durch die Wahl eines geeigneten Periodenintervalls zur Erzielung einer geraden oder ungeraden Funktion nur halb so viele Koeffizienten berechnen müssen – entweder die Sinus- oder Cosinuskoeffizienten, aber nie beide. Das heisst aber auch, dass sich die durch eine Fourierreihe repräsentierte periodische Funktion viel kompakter darstellen lässt.

### 6.1.5 Nicht-periodische Funktionen

Bis jetzt war lediglich von der Approximation periodischer Funktionen durch Fourierreihen die Rede. Bei der Parameterdarstellung einer kartographischen Linie handelt es sich jedoch nie um eine periodische Funktion. Dies ist jedoch halb so schlimm, wie anhand des Beispiels in Abbildung 6.6 gezeigt werden soll. Die in Abb. 6.6a dargestellte Funktion ist definiert durch

$$(6.18) \quad f(t) = -t^2 + 2t.$$

Im vorliegenden Beispiel ist es von keiner Bedeutung, welche Werte diese Funktion ausserhalb des Intervalls  $[0, 1]$  annimmt, z.B. weil das physikalische Problem, welches durch diese Kurve beschrieben wird, nur Werte zwischen 0 und 1 annehmen kann. Es ist offensichtlich, dass es sich bei  $f(t)$  um keine periodische Funktion handelt. Es hindert uns aber nichts daran, *anzunehmen*, wir hätten es mit einer periodischen Funktion zu tun, wie dies in Abbildung 6.6b dargestellt ist, und dann gemäss den Formeln 6.7 und 6.8 die Koeffizienten zu berechnen. Wenn die Funktion aber lediglich zwischen 0 und 1 von Interesse ist, dann können wir sie ausserhalb dieses Bereichs in beliebiger Weise fortsetzen. Es ist deshalb auch möglich,  $f(t)$  als eine ungerade Funktion aufzufassen, sie dementsprechend ins Intervall  $[-1, 0]$  zu erweitern, das Intervall  $[-1, 1]$  als Periode zu betrachten und daraus eine periodische Funktion zu konstruieren, wie dies in Abbildung 6.6c gezeigt wird. Es liegt auf der Hand, dass ebensogut eine gerade Funktion konstruiert werden kann. Die periodische Funktion in Abbildung 6.6d ist somit eine andere naheliegende Wahl. Zusammenfassend lässt sich die periodische Funktion in Abb. 6.6b mit *full-range* (weil der ursprüngliche Definitionsbereich (*range*) gleich der Periode der gewählten periodischen Funktion ist) und die Funktionen in Abb. 6.6c und d mit *half-range* (weil der ursprüngliche Definitionsbereich halb so gross ist wie die Periode der gewählten periodischen Funktion) umschreiben. Es stellt sich nun die Frage, welche der drei Möglichkeiten aus Abbildung 6.6 zu bevorzugen ist. Aus Abschnitt 6.1.4 wurde deutlich, dass sich die Fourierreihe einer geraden oder ungeraden Funktion einfacher berechnen und kompakter darstellen lässt. Einer *full-range*

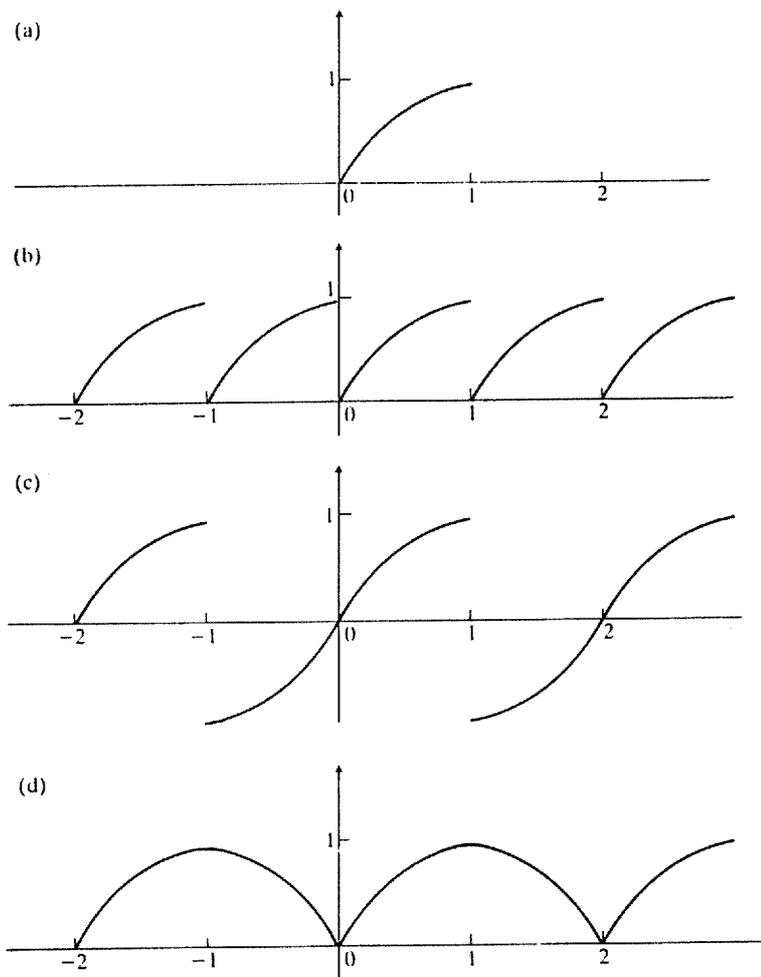


Abbildung 6.6: (a) Die Funktion  $f(t) = -t^2 + 2t$  definiert im Intervall  $[0, 1]$ . (b) Die periodische *full-range* Erweiterung der in (a) dargestellten Funktion. (c) Die ungerade periodische *half-range* Erweiterung der in (a) dargestellten Funktion. (d) Die gerade periodische *half-range* Erweiterung der in (a) dargestellten Funktion. Aus Solymar (1988, 39).

Funktion fehlen diese Eigenschaften, weshalb sie ausser Betracht fällt. Bleiben noch die Erweiterungen einer nicht-periodischen Funktion zu einer geraden bzw. ungeraden *half-range* Funktion. Eine ungerade Funktion hat den Nachteil, dass sie im allgemeinen eine Sprungstelle aufweist, im Beispiel aus Abb. 6.6c liegt diese bei  $t = 1$ . Die Frage ist, gegen welchen Wert die Fourierreihe an Diskontinuitätsstellen konvergiert. Es kann mit gutem Grund erwartet werden, dass sie gegen den Mittelwert der beiden Funktionswerte konvergiert. Dies ist auch tatsächlich der Fall (Solymar 1988, 19). Die Funktion  $f(t)$  konvergiert für alle  $t$  gegen den Wert

$$(6.19) \quad \frac{1}{2}[f(t+0) + f(t-0)],$$

mit

$$(6.20) \quad f(t+0) = \lim_{\substack{h \rightarrow 0 \\ h > 0}} f(t+h) \quad \text{und} \quad f(t-0) = \lim_{\substack{h \rightarrow 0 \\ h > 0}} f(t-h)$$

Die Approximation einer ungeraden Funktion wird somit an deren Anfangs- und Endpunkt im allgemeinen unbefriedigend ausfallen. Es kommt hinzu, dass infolge der Diskontinuität an dieser Stelle die gesamte Fourierreihe langsamer konvergiert. Das heisst mit anderen Worten, dass mehr Koeffizienten benötigt werden, um eine vorgegebene Approximationsgenauigkeit zu erreichen.

Aus diesen Überlegungen folgt, dass eine gerade *half-range* Funktion zu favorisieren ist, weil diese keine Diskontinuität aufweist. Bemerkenswert ist, dass es genügt, eine solche Funktion *anzunehmen*, es ist nicht nötig, sie wirklich zu konstruieren. Einzig die Periode  $T$  muss doppelt so lang wie der Definitionsbereich der ursprünglichen Funktion gewählt werden. Bedingt durch die Symmetrie einer geraden Funktion ist kein Integral zu berechnen, welches über  $T/2$  hinausgeht. Somit haben wir mit der ursprünglichen Funktion alle Funktionswerte zur Hand, welche zur Berechnung der Koeffizienten benötigt werden.

### 6.1.6 Zusammenfassung

Die Erkenntnisse aus Abschnitt 6.1 sind hier nochmals in einer schrittweisen Anleitung zur Approximation nicht-periodischer Funktionen durch Fourierreihen zusammengefasst:

1. Definition einer Periode  $T$ , welche doppelt so lang ist wie der Definitionsbereich der ursprünglichen Funktion  $f(t)$ .
2. Berechnung der Cosinuskoeffizienten:

$$a_k = \frac{4}{T} \int_0^{T/2} f(t) \cos \frac{2\pi kt}{T} dt, \quad k = 0, 1, 2, \dots, \infty$$

3. Approximation der Funktion  $f(t)$  durch deren Fourierreihe über dem Intervall  $[0, T/2]$  ( $\hat{=}$  ursprünglichen Definitionsbereich der Funktion), gegeben durch

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos \frac{2\pi kt}{T}$$

## 6.2 Fourierreihen parametrisierter Linien

In der Praxis kann natürlich nicht die unendliche Fourierreihe berechnet werden, sondern es ist nötig, sich auf die Berechnung einer bestimmten Anzahl von Koeffizienten zu beschränken. Eine solche endliche Fourierreihe wird mit  $f_n(t)$  bezeichnet, wobei  $n$  für den grössten Koeffizienten der Reihe steht. Eine solche Reihe beinhaltet die Koeffizienten  $a_0, \dots, a_k, \dots, a_n$ . Wie gross dieses  $n$  sein muss, um eine gegebene Funktion mit genügender Genauigkeit zu approximieren, hängt von der Komplexität der Funktion ab. Wie in Kapitel 5 erläutert wurde, bevorzugen wir den *t-alpha-Plot* zur Parametrisierung von kartographischen Linien. Bei der Funktion, deren Komplexität wir beurteilen müssen, handelt es sich folglich um eine *t-alpha*-Kurve. Für die Entwicklung in Fourierreihen handelt es sich dabei um wahrlich pathologische Kurven. Da sind zum einen die vielen Sprungstellen, welche uns jedesmal das *Gibbs-Phänomen* (vgl. Abschnitt 6.1.2) bescheren und zum anderen ausschliesslich *horizontale* Kurvensegmente. Wie bereits das Beispiel in Abschnitt 6.1.2 zeigte, werden zur Eliminierung der "Rippen" in horizontalen Segmenten sehr viele Koeffizienten benötigt. Dies ist umso bedauerlicher, da die restliche Form der Kurve bereits mit bedeutend weniger Koeffizienten gut angenähert werden konnte. Die vielen zusätzlichen Koeffizienten dienen also lediglich dazu, die Amplitude der "Rippen" zu verkleinern. Die Komplexität eines *t-alpha Plots* im Hinblick auf die Entwicklung in eine Fourierreihe ist demnach durch die Anzahl und die Amplitude der Sprungstellen gegeben. Diese hängt wiederum von der Angularität (vgl. 3.3) der Kurve in der Kartenebene ab. Wie bereits in Abschnitt 3.3 wird als Beispiel für eine Kurve mit niedriger Angularität die Linie *596ungen* (vgl. Abb. 3.6) gewählt. Linie *587ungen* andererseits steht stellvertretend für Linien mit einer hohen Angularität. In Abbildung 6.7 sind die ersten 500 Koeffizienten für diese beiden Linien abgebildet. Die verschieden schnelle Abnahme der Koeffizientenamplituden widerspiegelt die unterschiedliche Komplexität der Ausgangskurven. Aus einem solchen Diagramm kann abgelesen werden, wieviele Koeffizienten für eine gute Approximation in etwa benötigt werden. Es kann erwartet werden, dass für die Linie *596ungen* die ersten 150 bis 200 Koeffizienten genügen werden. Für grössere Koeffizienten bewegen sich die Amplituden lediglich noch im Tausendstelbereich, d.h. Harmonische dieser Ordnung leisten kaum mehr einen Beitrag. Anders sieht die Situation für die Linie *587ungen* aus. Zwischen dem 150. und dem 200. Koeffizienten sind die Amplituden nach wie vor signifikant und können nicht vernachlässigt werden. Erst ab  $a_{300}$  tendieren die Amplituden gegen Null.

Aber auch innerhalb der ersten 300 Koeffizienten finden sich solche mit sehr kleiner Amplitude. Es kann deshalb erwartet werden, dass – notabene mit derselben Anzahl Koeffizienten – eine bessere Approximation resultiert, wenn die Koeffizienten zuerst nach ihrer absoluten Amplitude absteigend sortiert werden und erst dann die ersten  $n$  Koeffizienten ausgewählt werden.

In den Abbildungen 6.8 und 6.9 (*596ungen*) bzw. 6.10 und 6.11 (*587ungen*) sind die Approximationen  $f_n$  der *t-alpha Plots* für verschiedene  $n$  dargestellt. Deutlich kommt zum Ausdruck, dass durch die Auswahl der  $n$  Koeffizienten mit der grössten Amplitude mehr höherfrequente Komponenten miteinbezogen werden als dies bei Wahl der *ersten*  $n$  Koeffizienten der Fall ist. Mit 500 Koeffizienten kann der *t-alpha Plot* der Linie *596ungen*, abgesehen vom *Gibbs-Phänomen*, welches deutlich zum Vorschein kommt, sehr gut approximiert werden. Die Halbierung von  $n$  auf 250 resultiert zwar in einem Approximationsqualitätsverlust, welcher sich jedoch nach Rücktransformation in die  $x, y$ -Ebene nicht manifestiert (Abb. 6.12). Wird die Approximationsqualität direkt in der  $x, y$ -Ebene beurteilt, kann auch eine

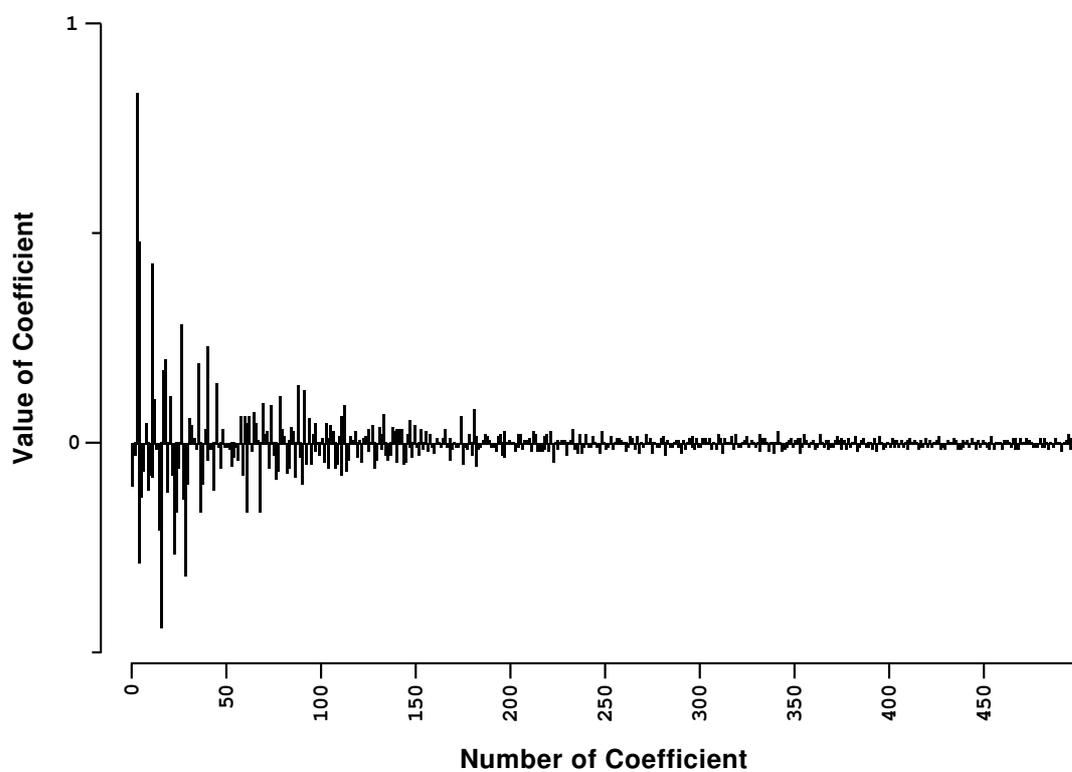
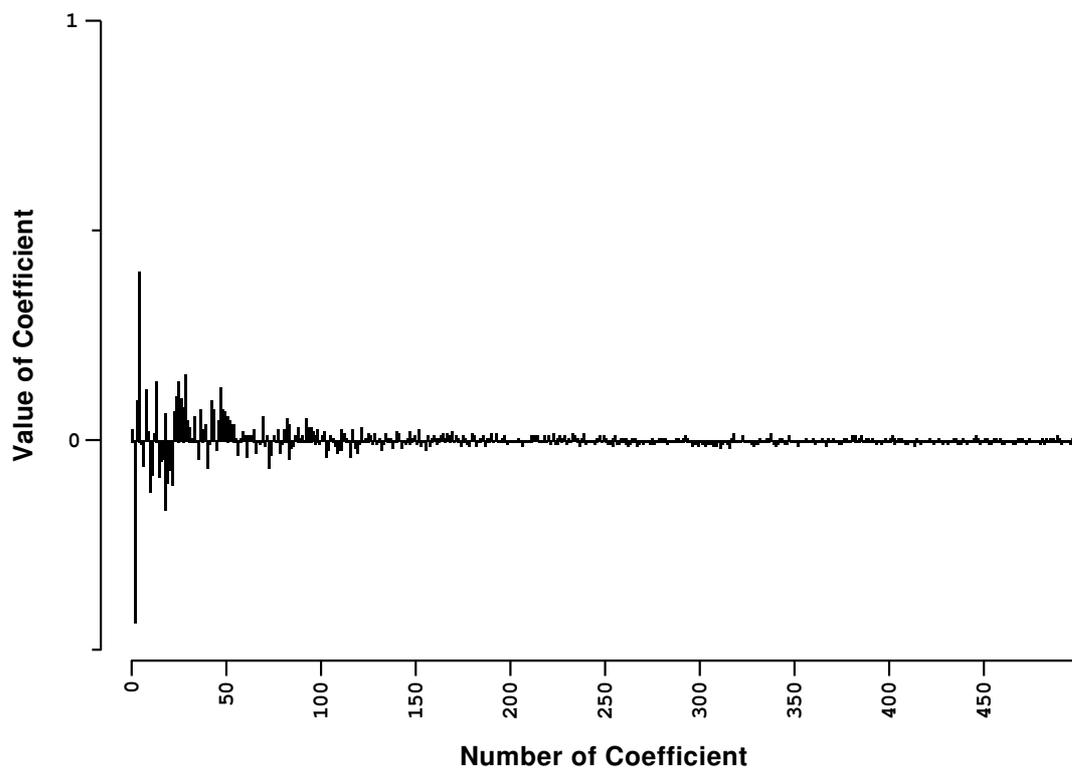


Abbildung 6.7: Die ersten 500 Fourierkoeffizienten  $a_k$  der  $t$ - $\alpha$  Plots der Linien 596ungen (oben) und 587ungen (unten).

Approximation mit lediglich 125 Koeffizienten noch als genügend bezeichnet werden. Ein Unterschied zur Approximation durch die 125 grössten Koeffizienten (vgl. Abb. 6.13) ist in der Kartenebene nicht auszumachen. Erst eine weitere Reduktion der Anzahl Koeffizienten bringt den Vorteil einer amplitudenbasierten Auswahl zum Vorschein. Bei der komplexeren Linie *587ungen* tritt dieser Vorteil schon früher zutage (vgl. Abbildungen 6.14 und 6.15). Es zeigt sich aber auch, dass für eine gute Approximation der Linie *587ungen* deutlich mehr Koeffizienten gebraucht werden als für die Linie *596ungen*. Während mit 500 Koeffizienten die Approximationsgenauigkeit noch im Bereich der Strichstärke der kartographischen Linie liegt, kann eine Rekonstruktion aus lediglich 125 Koeffizienten nicht mehr überzeugen.

Einen Überblick über die Approximationsqualität geben die Abbildungen 6.16, 6.17, 6.18, 6.19, wo die Linien in ihrer vollen Länge abgebildet sind. Es zeigt sich, dass die Rekonstruktion der Linie mit einer beschränkten Anzahl Koeffizienten zu einer Art “Generalisierung” der Linie führt. Fritsch und Lagrange (1995) haben sich eingehender mit der Filterung von Fourierreihen zur unmittelbaren Generalisierung von Linien auseinandergesetzt. Allerdings sind Fourierreihen, welche auf kurvilinear parametrisierten Linien beruhen, nicht für diesen Zweck geeignet, weil sie keine Linienverkürzung erlauben<sup>2</sup>. Die grossen Verschiebungen in obigen Abbildungen rühren denn auch von diesem Umstand her. Für unsere Zwecke ist diese unschöne Degradation der Fourierreihe jedoch nicht von Belang, da es uns hauptsächlich darum geht, die Linie in einer Komponentendarstellung repräsentieren zu können. Dies ist uns gelungen. Die Repräsentation eines *t-alpha Plots* – auf jener Stufe findet ja die eigentliche Approximation statt – durch eine Fourierreihe erfüllt zusätzlich folgende in Abschnitt 2.2 geforderte Kriterien:

**Fixe Länge:** Die Anzahl der Fourierkoeffizienten, welche zur Approximation einer Linie benötigt werden, hängt nicht von der Länge der Linie ab, sondern lediglich von deren Komplexität. Es hat sich gezeigt, dass mit 500 Koeffizienten sämtliche Linien des vorliegenden Datensatzes gut approximiert werden können.

**Effizienz:** Die Fourierkoeffizienten einer abgetasteten Linie, bzw. deren *t-alpha Plot*, lassen sich unter Berücksichtigung der Symmetrieeigenschaften einfach berechnen. Die Datenhaltung ist ebenfalls relativ effizient, es brauchen lediglich die 500 Koeffizienten, die Abtastschrittweite plus die Transformationsparameter gespeichert zu werden, was gegenüber Polylinien mit mehr als 256 Punkten (*x/y*-Paare) zu einer Komprimierung führt.

Der grosse Nachteil von Fourierreihen ist bislang noch nicht zur Sprache gekommen: die Repräsentation ist *nicht lokalisiert*. Ein Koeffizient  $a_k$  beschreibt immer die Energie einer bestimmen Frequenz für die ganze Linie. Es ist somit beinahe unmöglich, die Korrelationen zwischen den Fourierkoeffizienten einer ungeneralisierten und einer generalisierten Linie zu finden und zu beschreiben. In Kapitel 8 wird näher auf dieses Problem eingegangen. Es kann indes umgangen werden, wenn der *t-alpha Plot* einer anderen frequenzbasierten Transformation unterzogen wird, der *Wavelettransformation*, welche im nächsten Kapitel behandelt wird.

---

<sup>2</sup>Die Verkürzung könnte allenfalls bei der Rücktransformation in die *x, y*-Ebene berücksichtigt werden (siehe Kapitel 8).

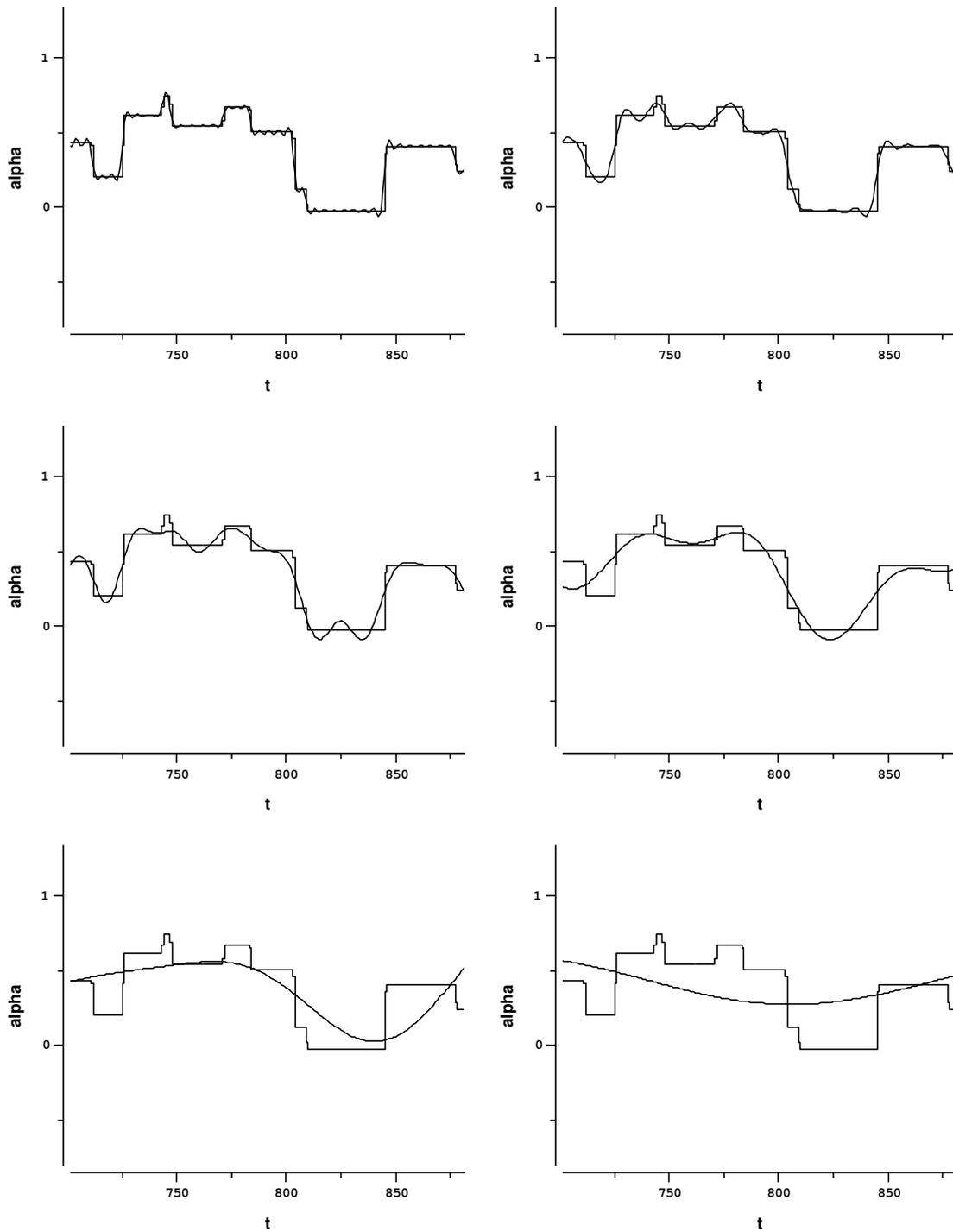


Abbildung 6.8: Ausschnitt aus der Approximation des  $t$ - $\alpha$  Plots der Linie 596ungen durch die (von oben links nach unten rechts) ersten 500, 250, 125, 50, 25, 10 Fourierkoeffizienten.

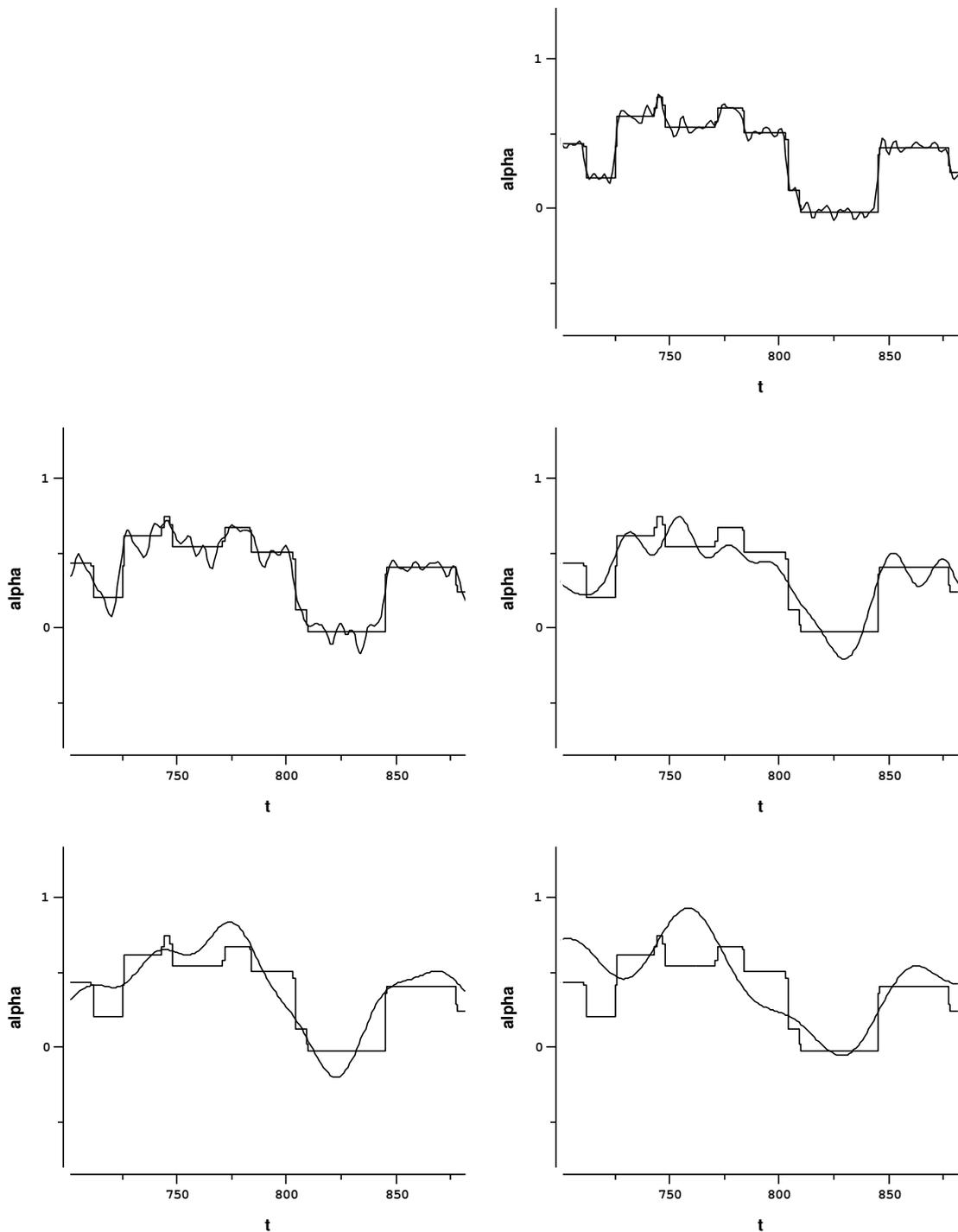


Abbildung 6.9: Ausschnitt aus der Approximation des  $t$ - $\alpha$  Plots der Linie 596ungen durch die (von oben links nach unten rechts) 250, 125, 50, 25, 10 Fourierkoeffizienten mit der grössten absoluten Amplitude.

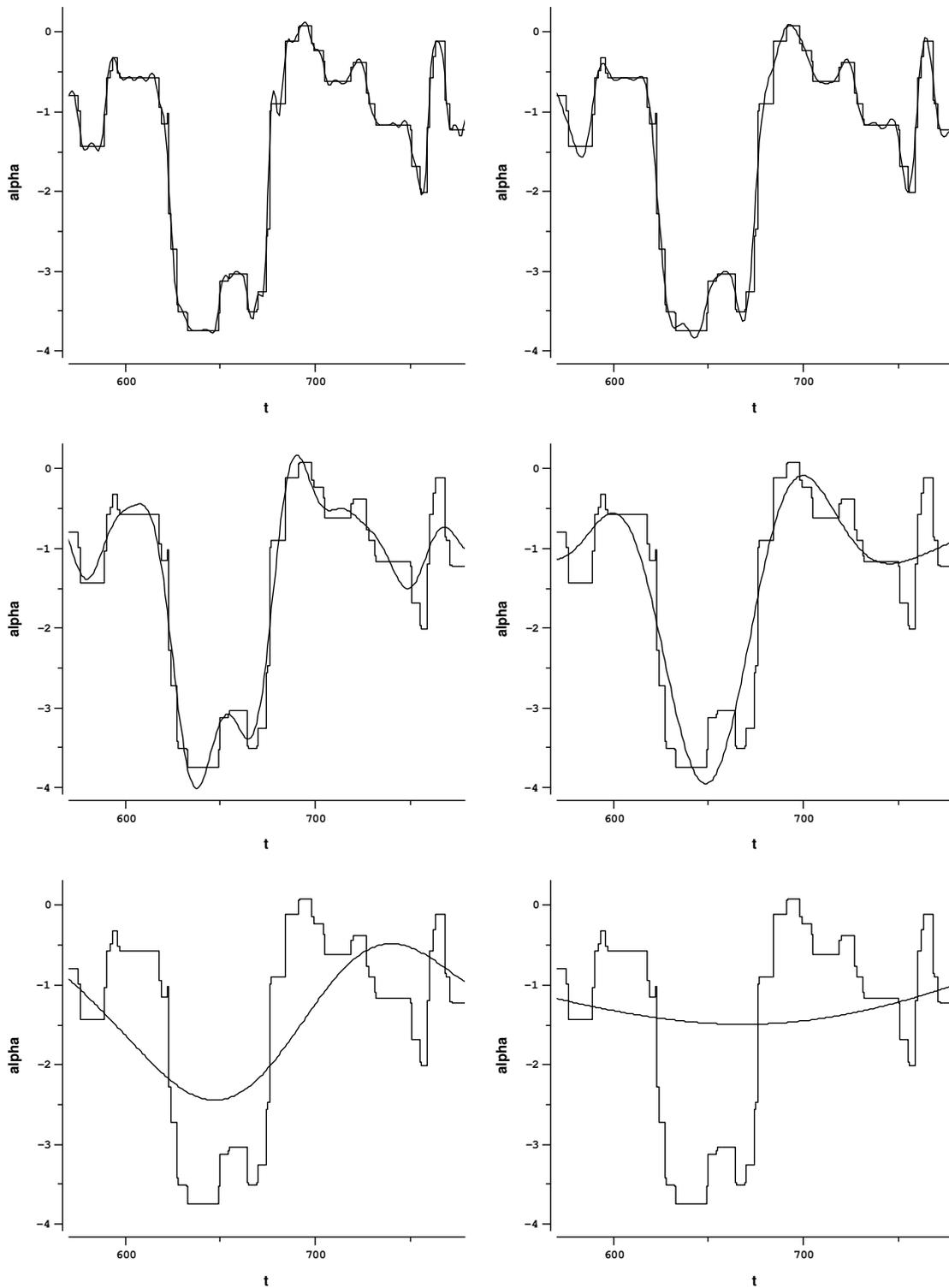


Abbildung 6.10: Ausschnitt aus der Approximation des  $t$ - $\alpha$  Plots der Linie 587ungen durch die (von oben links nach unten rechts) ersten 500, 250, 125, 50, 25 und 10 Fourierkoeffizienten.

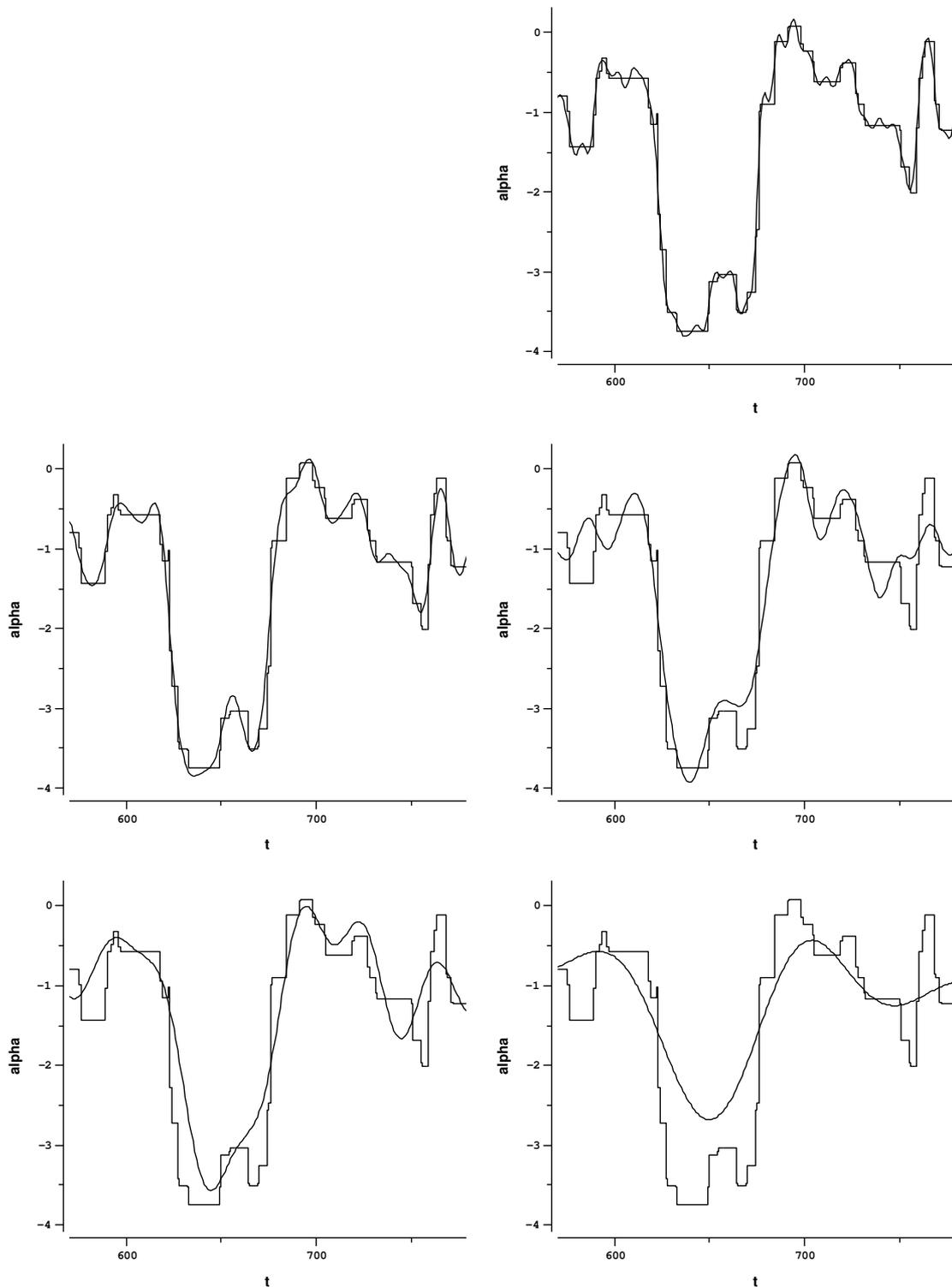


Abbildung 6.11: Ausschnitt aus der Approximation des  $t$ - $\alpha$  Plots der Linie 587ungen durch die (von oben links nach unten rechts) 250, 125, 50, 25, 10 Fourierkoeffizienten mit der grössten absoluten Amplitude.

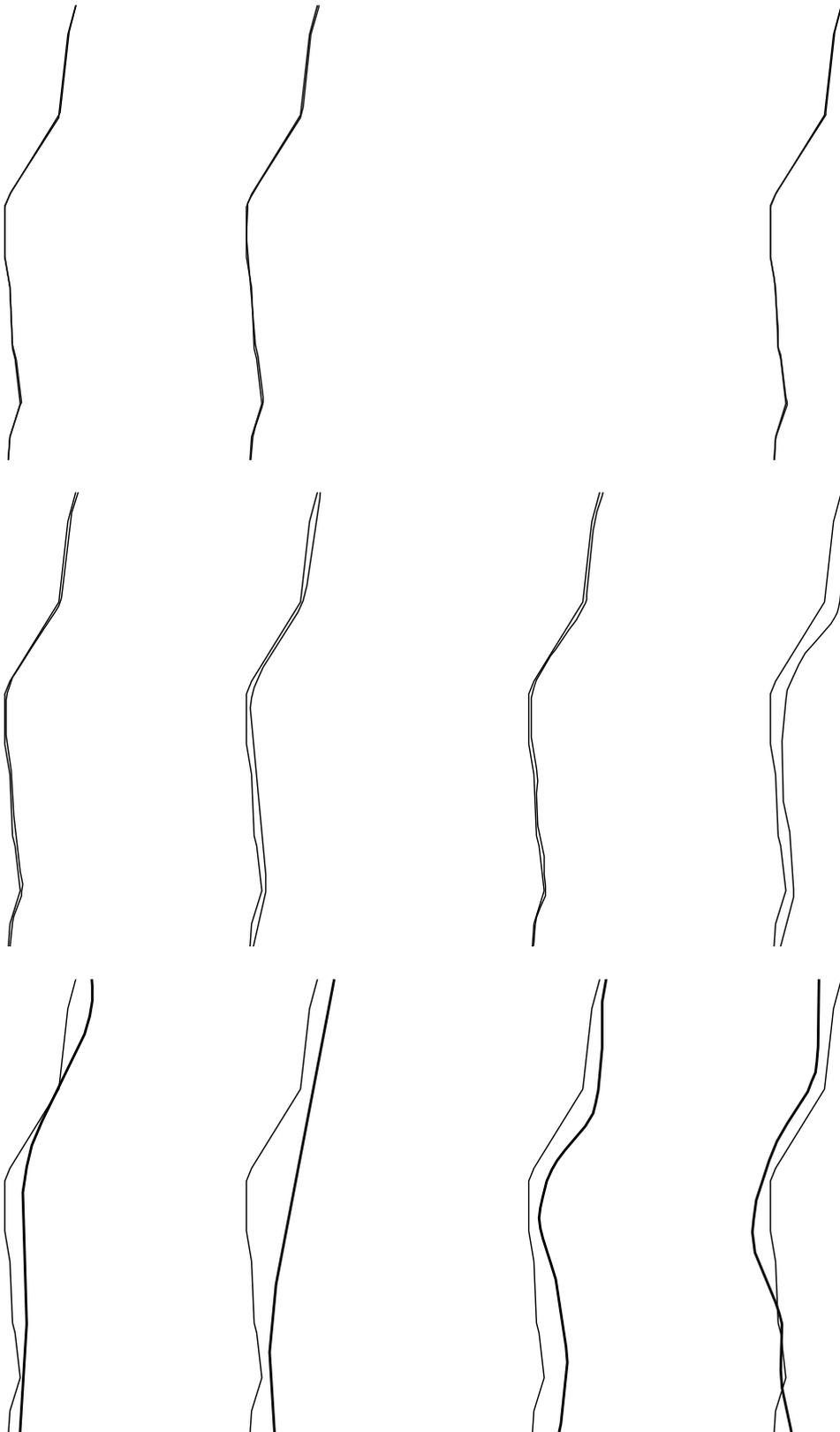


Abbildung 6.12: Rücktransformation der Approximationen aus 6.8 in die  $x, y$ -Ebene und Überlagerung mit der Ausgangslinie *596ungen*. Der Ausschnitt entspricht jenem des  $t$ - $\alpha$  Plots.

Abbildung 6.13: Rücktransformation der Approximationen aus 6.9 in die  $x, y$ -Ebene und Überlagerung mit der Ausgangslinie *596ungen*. Der Ausschnitt entspricht jenem des  $t$ - $\alpha$  Plots.

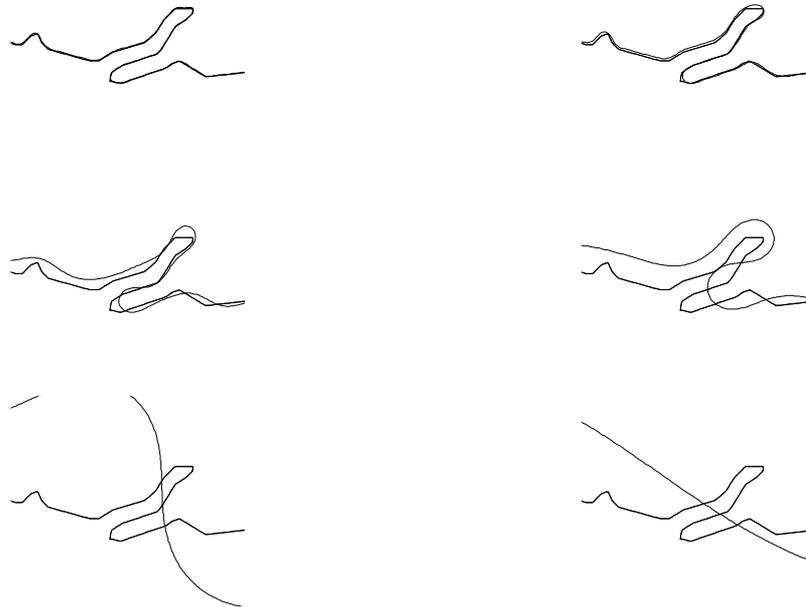


Abbildung 6.14: Rücktransformation der Approximationen aus 6.10 in die  $x, y$ -Ebene und Überlagerung mit der Ausgangslinie *587ungen*. Der Ausschnitt entspricht jenem des *t-alpha Plots*.

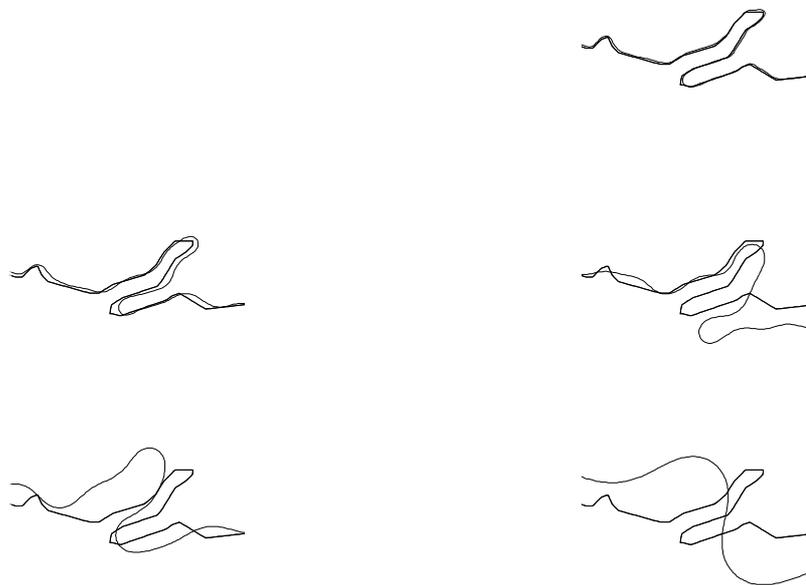


Abbildung 6.15: Rücktransformation der Approximationen aus 6.11 in die  $x, y$ -Ebene und Überlagerung mit der Ausgangslinie *587ungen*. Der Ausschnitt entspricht jenem des *t-alpha Plots*.

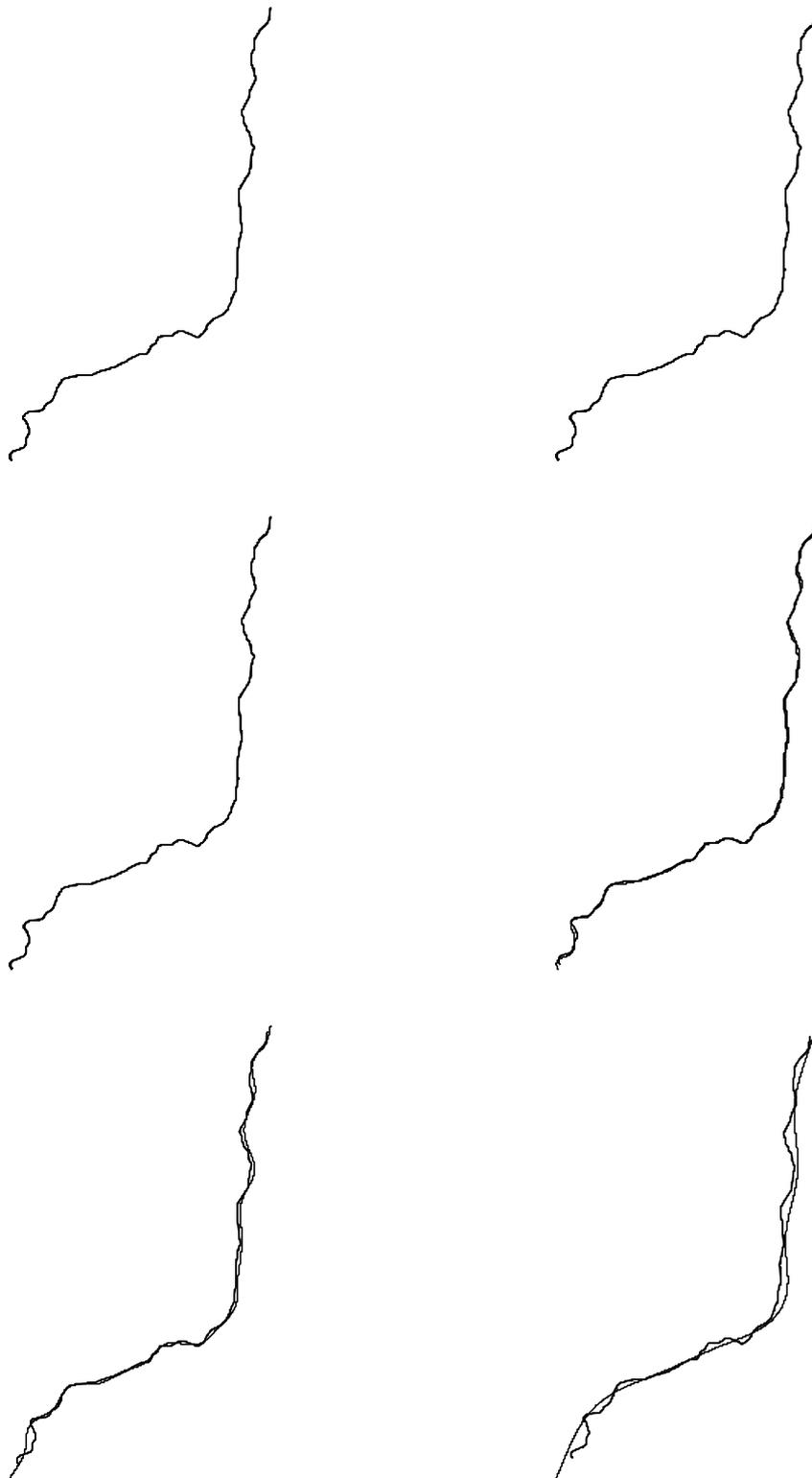


Abbildung 6.16: Approximation der Linie *596ungen* (via *t-alpha Plot*) durch die *ersten* 500, 250, 125, 50, 25, 10 Fourierkoeffizienten, überlagert mit der Ausgangsline.

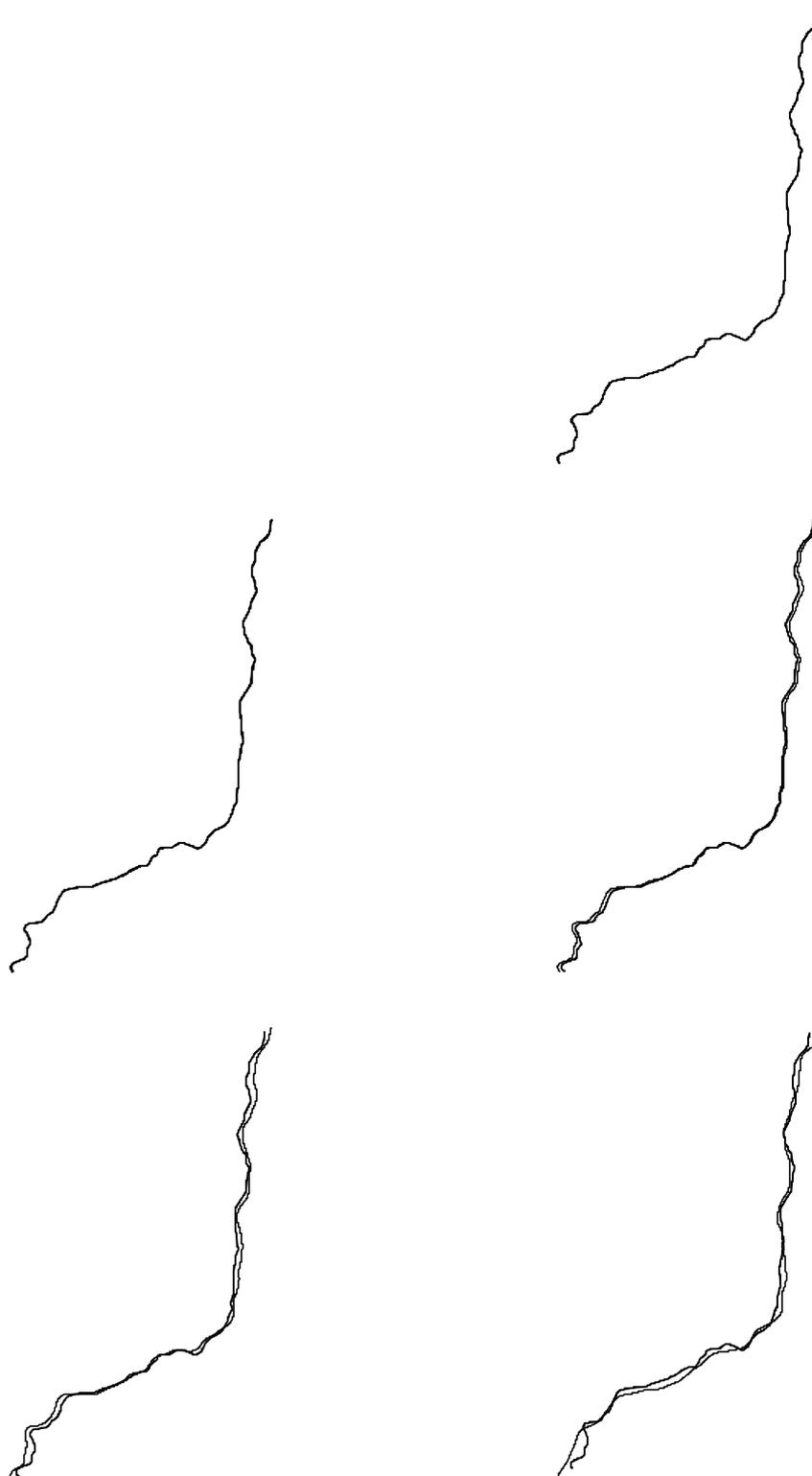


Abbildung 6.17: Approximation der Linie *596ungen* (via *t-alpha Plot*) durch die *grössten* 500, 250, 125, 50, 25, 10 Fourierkoeffizienten, überlagert mit der Ausgangslinie.

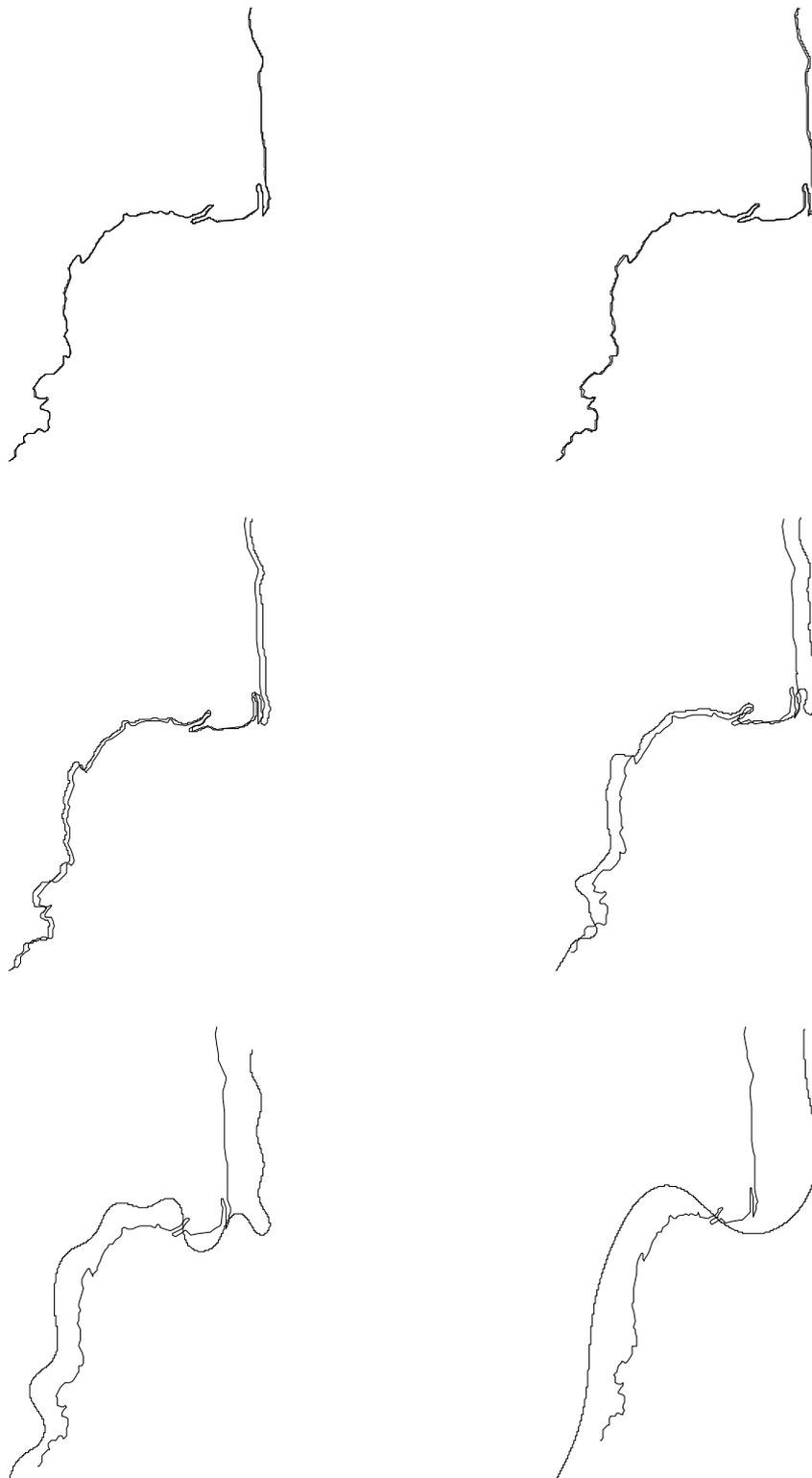


Abbildung 6.18: Approximation der Linie *587ungen* (via *t-alpha Plot*) durch die *ersten* 500, 250, 125, 50, 25, 10 Fourierkoeffizienten, überlagert mit der Ausgangsline.



Abbildung 6.19: Approximation der Linie *587ungen* (via *t-alpha Plot*) durch die *grössten* 500, 250, 125, 50, 25, 10 Fourierkoeffizienten, überlagert mit der Ausgangslinie.



# Kapitel 7

## Wavelettransformation

Fourierreihen weisen einen gravierenden Nachteil auf: man erhält keine Information darüber, *wann* eine bestimmte Frequenz aufgetreten ist<sup>1</sup>, sondern nur, *dass* eine bestimmte Frequenz aufgetreten ist. Es wird dabei auch von *mangelnder Lokalisierung* im Frequenzraum gesprochen. Häufig ist aber eine lokale Frequenzzerlegung einer Funktion  $f$  angestrebt, ähnlich der Notation in der Musik, wo dem Musiker oder der Musikerin gesagt wird, wann (= Zeitinformation) welche Note (= Frequenzinformation) gespielt werden soll (Daubechies 1993). Die sogenannte *Short-Time-Fouriertransformation (STFT)* versuchte dies mit einem Zeitfenster, welches über das Signal gelegt wird, zu erreichen. Eine solche Repräsentation weist zwei Parameter auf, *Frequenz* und *Zeitpunkt*. Die *STFT* hat aber den Nachteil, dass sich die Fenstergrösse dem Verhalten eines Signals nicht anpassen lässt – bei hohen Frequenzen wäre aber ein kleines Zeitfenster, bei niedrigen Frequenzen ein grosses Zeitfenster wünschenswert (Zeller 1994).

Wavelettransformationen (*WT*), von denen in diesem Kapitel die Rede sein soll, weisen nun jedoch genau diese Eigenschaft auf. Gemäss Wilson (1993) sind Wavelets erstmals Mitte der 80er Jahre durch die Arbeit von Grossman und Morlet (1984) in der Literatur aufgetaucht. Der französische Mathematiker Yves Meyer (Meyer et al. 1987) fasste die Entwicklungen auf diesem Gebiet schliesslich in einer einheitlichen Theorie – der Wavelettheorie – zusammen. Stephan Mallat (1989) von der New York University und Ingrid Daubechies (1988, 1990, 1993) von den Bell Labs entwickelten die Theorie weiter und stellten Zusammenhänge zur Signalverarbeitung her. Die Wavelettheorie ist jedoch alles andere als leicht verdaulich: “Wavelets pur sind ein hochgezüchtetes Produkt der Mathematik und man benötigt einen gewaltigen Apparat, um alle Zusammenhänge zu erläutern (Zeller 1994).” Ein Grossteil der Forschung in diesem Bereich beschäftigt sich damit, neue Waveletbasisfunktionen zu finden und zu beweisen, dass sie den mathematischen Ansprüchen genügen.

### 7.1 Konzept der Wavelettransformation

Während bei der Fouriertransformation noch zwei Funktionen (Cosinus und Sinus) als Grundbausteine des Baukastens für periodische Signale dienen, ist dies bei der Wavelettransformation noch eine einzige ausgezeichnete Funktion, aus der sich alle Signale auf eine bestimmte Weise zusammensetzen lassen. Diese Funktion wird *Basis-* oder *mother*

---

<sup>1</sup>Wenn der diskrete kurvilineare Distanzparameter  $t$  als Zeit aufgefasst wird.

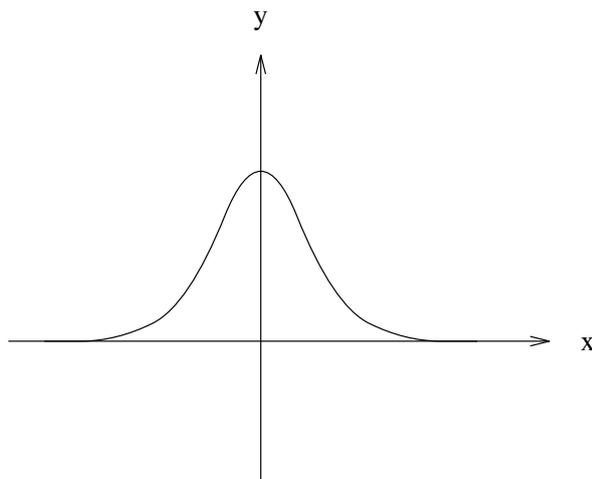


Abbildung 7.1: Eine “kleine” Funktion, welche aber das “Wellen”-Kriterium nicht erfüllt und somit auch keine “kleine Welle” (wavelet) sein kann.

*wavelet*  $\psi$  genannt<sup>2</sup>. Die Signale, die mit Hilfe eines “Waveletbaukastens” entstehen, sind nicht mehr periodisch – auch zur Fouriertransformation von nicht-periodischen Signalen wird die Periode schliesslich künstlich erzeugt (vgl. 6.1.5). Um alle Signale beliebiger Dauer und endlicher Energie aus Varianten einer einzigen Funktion zusammensetzen zu können, genügt es nun aber nicht mehr, nur gestauchte und gedehnte Varianten (Dilatationen) einer Basisfunktion zu benutzen. Dieses Ziel erreicht man erst dann, wenn sich die ausgewählten Basisfunktionen auch beliebig verschieben (Translationen) lassen. Allen Basisfunktionen ist gemeinsam, dass der jeweilige Baukasten, den sie erzeugen, aus Dilatationen und Translationen ihres *mother wavelets*  $\psi$  bestehen (Zeller 1994). Angesichts der grossen Anzahl der dadurch entstehenden Basisfunktionen kann Strichartz (1994) dem (etablierten) Namen *mother wavelet* allerdings nicht viel abgewinnen: “In the French literature it [die Funktion, welche die Dilatationen und Translationen beschreibt] is sometimes called “le père” and  $\psi$  is called “la mère”, but this shows a scandalous misunderstanding of human reproduction; in fact, the generation of wavelets more closely resembles the reproductive life style of amoebas.”

Wodurch zeichnet sich eine solche “kleine Welle” (engl. *wavelet*) nun aus? Wavelets müssen einerseits oszillieren<sup>3</sup> (daher die “Welle”) und andererseits Amplituden aufweisen, welche sowohl in negativer wie positiver Richtung rasch gegen Null streben (deshalb “klein”). Eine Sinusfunktion erfüllt das Kriterium der Welle, konvergiert aber nicht gegen Null und kann somit kein Basiswavelet sein. Die Funktion in Abbildung 7.1 andererseits konvergiert wohl beidseitig gegen Null, oszilliert aber nicht. Beide Bedingungen erfüllen hingegen die Funktionen in Abbildung 7.2. Dabei handelt es sich um Wavelets, welche im *Wavelet Packet Laboratory* (Coifman et al. 1992; WPL 1994), einer Software zur explorativen Waveletanalyse, implementiert sind.

Es existieren verschiedene Arten von Wavelettransformationen. Der allgemeinste Fall ist

<sup>2</sup>Dieses  $\psi$  steht in keinem Bezug zu jenem des *psi-s Plots*.  $\psi$  wurde deshalb gewählt, weil es der gängigen Notation für Basiswavelets entspricht.

<sup>3</sup>Die Forderung nach Oszillation leitet sich aus dem Erfordernis ab, dass das Integral über einem Wavelet Null ergeben muss.

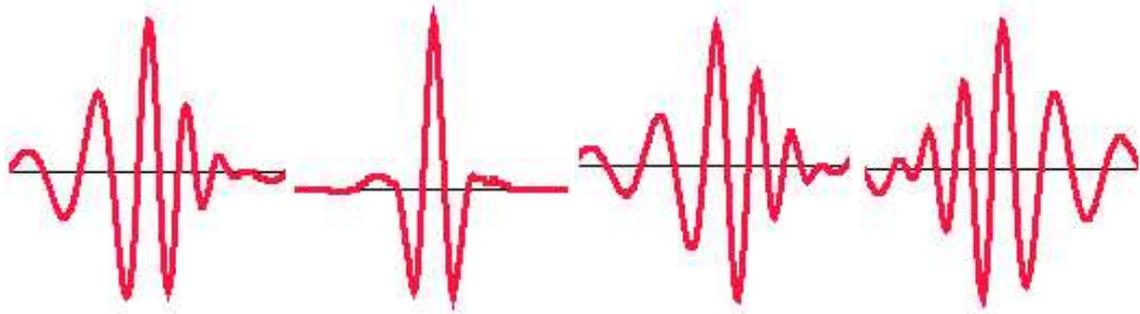


Abbildung 7.2: Basiswavelets aus dem *Wavelet Packet Laboratory*. Vlnr: Daubechies Wavelet, *Coiflet*, Beylkin Wavelet, Vaidyanathan Wavelet. Aus: WPL (1994).

die *kontinuierliche Wavelettransformation* (Daubechies 1993), welche es den Parametern, welche Dilatation und Translation des Basiswavelets steuern, erlaubt, jeden beliebigen reellen Wert anzunehmen. Führt man sich vor Augen, dass dies bedeutet, dass ein mit einem bestimmten Faktor skaliertes Basiswavelet nun in sehr kleinen Schritten entlang dem Signal verschoben wird, wird klar, dass der “Waveletbaukasten” aus sehr vielen formidentischen, aber sich überlappenden Funktionen besteht. Die Folge ist eine Fülle hochgradig redundanter Waveletkoeffizienten. Abhilfe schafft die *diskrete Wavelettransformation* mit einer *orthonormalen Waveletbasis* (Daubechies 1993) wie sie im nächsten Abschnitt behandelt wird.

### 7.1.1 Orthonormale Haar-Waveletbasis

#### Erstellung der Waveletbasis

Um eine orthonormale Waveletbasis zu erhalten, ist es einerseits nötig, nur gewisse Dilatationen und Translationen des Basiswavelets zuzulassen. In den meisten Anwendungen kommt eine Basisfunktion zum Zug, welche in Zweierpotenzen skaliert wird (sog. *dyadische Skalierung*, Wilson 1993):

$$(7.1) \quad \psi_{m,n}(x) = 2^{-m/2} \psi(2^{-m}x - n)$$

Die Parameter  $m$  und  $n$ , welche Dilatation und Translation des Basiswavelets bestimmen, können zudem nur noch ganzzahlige Werte annehmen. Andererseits muss ein Basiswavelet  $\psi$  gefunden werden, welches auch wirklich eine orthonormale Basis bildet. Das älteste Beispiel für eine solche Funktion ist die sogenannte *Haar-Funktion* (Haar 1910):

$$(7.2) \quad \psi(x) = \begin{cases} 1 & 0 \leq x < 1/2, \\ -1 & 1/2 \leq x < 1, \\ 0 & \text{sonst.} \end{cases}$$

Dabei handelt es sich um eine denkbar einfache Funktion, wie Abbildung 7.3 (Mitte) zeigt. Daraus wird auch ersichtlich, dass es sich bei Basiswavelets nicht um stetige Funktionen

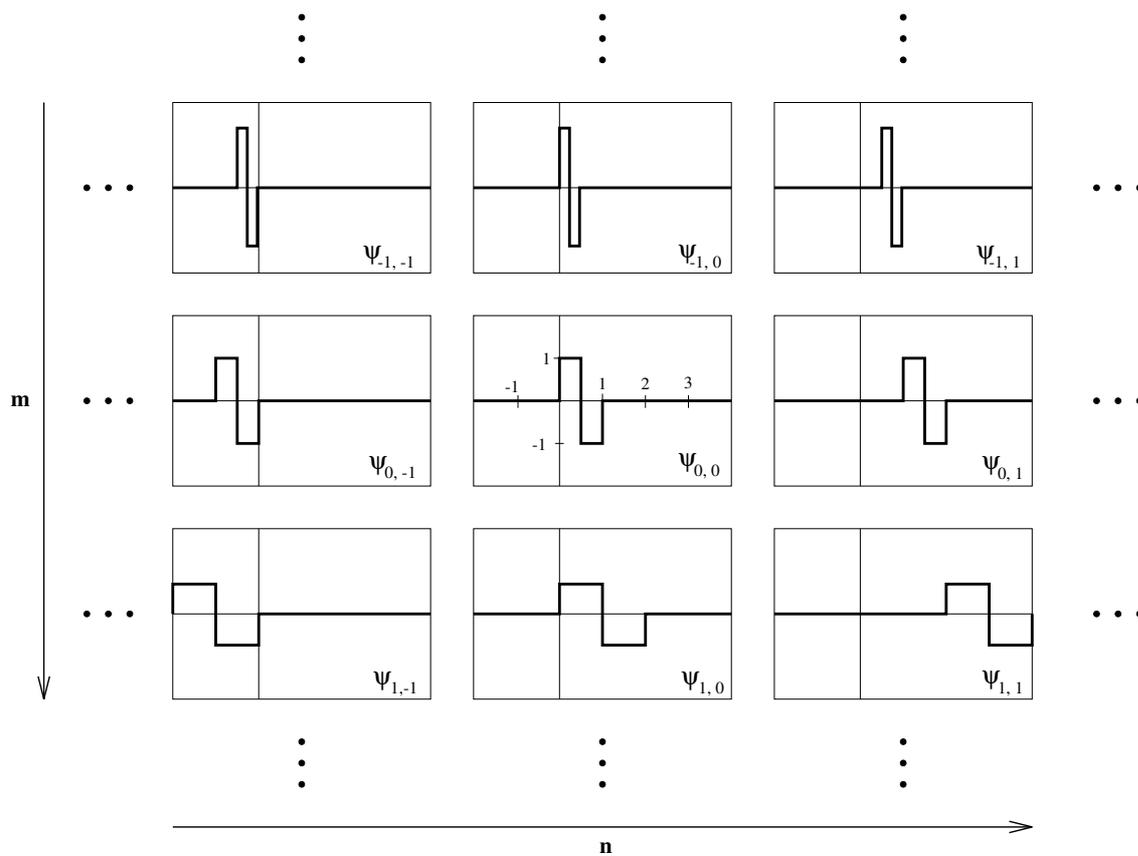


Abbildung 7.3: Das Haar-Basiswavelet (Mitte) und verschobene (in Zeilenrichtung) und skalierte (in Spaltenrichtung) Versionen  $\psi_{m,n}$  davon.

handeln muss. Nun, da wir eine konkrete Basisfunktion zur Hand haben, kann auch Gleichung 7.1 veranschaulicht werden. Rund um das Basiswavelet in Abbildung 7.3 sind skalierte und verschobene Versionen desselben für verschiedene Werte von  $m$  (Dilatation) und  $n$  (Translation) angeordnet. Daraus geht auch hervor, welche Funktion der Normalisierungsterm  $2^{-m/2}$  in Gleichung 7.1 erfüllt: er stellt sicher, dass die *Energie*  $E$

$$(7.3) \quad E(f) = \int f(x)^2 dx$$

eines jeden Wavelets 1 beträgt. Die Energie eines Haar-Wavelets  $\psi_{m,n}(x)$  berechnet sich demzufolge aus

$$(7.4) \quad E(\psi_{m,n}) = \int_{2^m n}^{2^m(n+1)} \psi_{m,n}(x) dx$$

Die Integrationsgrenzen definieren die sogenannte *Unterstützung* (engl. *support*) des Wavelets. Ausserhalb dieses Intervalls ist der Funktionswert eines Wavelets Null.

Der vollständige Beweis, dass eine Haar-Basis tatsächlich eine orthonormale Waveletbasis definiert, soll hier nicht wiedergegeben werden. Er kann bei Bedarf in Daubechies

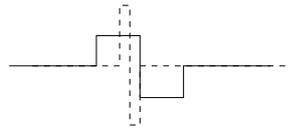


Abbildung 7.4: Zwei Haar-Wavelets; der *Support* des “schmaleren” Wavelets liegt vollständig in einem Intervall, auf welchem das “breitere” Wavelet konstant ist. Nach Daubechies (1993).

(1993) nachgelesen werden. Es soll hier lediglich gezeigt werden, dass die  $\psi_{m,n}$  orthonormal zueinander sind, was dem ersten Teil des Beweises entspricht. *Normalität* ist, wie oben beschrieben, dadurch gegeben, dass die Wavelets alle eine Energie von 1 aufweisen. Bleibt noch zu zeigen, dass die Wavelets einer Haar-Familie *orthogonal* sind. Zwei Funktionen  $f(x)$  und  $g(x)$  sind orthogonal, wenn ihr *inneres Produkt*  $\int f(x)g(x) dx$  Null ist. Dass dies für Haar-Wavelets gilt, ist einfach zu zeigen. Weil die Unterstützung eines Wavelets  $\psi_{m,n}$  nur das Intervall  $[2^m n, 2^m(n+1)]$  umfasst, folgt, dass sich zwei Haar-Wavelets mit derselben Skalierung (mit demselben  $m$ -Wert) nie überlappen. Deren inneres Produkt ist somit Null. Überlappender *Support* ist nur dann möglich, wenn es sich um Wavelets verschiedener Grösse handelt, wie in Abbildung 7.4 gezeigt. Es ist jedoch leicht zu ersehen, dass der *Support* von  $\psi_{m,n}$ , wenn gilt  $m < m'$ , vollständig in einem Bereich liegt, in dem  $\psi_{m',n'}$  konstant ist. Daraus folgt, dass das innere Produkt von  $\psi_{m,n}$  und  $\psi_{m',n'}$  proportional zum Integral von  $\psi$  selbst ist, welches jedoch Null ist (Daubechies 1993).

Die Basisfunktionen unseres “Haar-Waveletbaukastens” hätten wir somit bestimmt. Durch geschickte Kombination dieser Funktionen sollte es nun möglich sein, ein beliebiges Signal zu approximieren. Bleibt die Frage, *wieviele* dieser Funktionen zur Approximation des Signals beisteuern soll. Davon handelt der nächste Abschnitt.

### Berechnung der Waveletkoeffizienten

Bei den gesuchten Beträgen, welche die einzelnen Waveletfunktionen gewichten, handelt es sich um die *Waveletkoeffizienten*  $W_{m,n}(f)$  der zu transformierenden Funktion  $f(t)$ . Ein solcher Koeffizient für eine gegebene Skalierung  $m$  und eine Translation  $n$  sagt etwas darüber aus, wie gut das Signal  $f(t)$  mit dem transformierten Basiswavelet  $\psi_{m,n}$  übereinstimmt; ist das Signal  $f(t)$  gleich  $\psi_{m,n}$ , dann wird der Koeffizient einen grossen Wert aufweisen. Der Waveletkoeffizient repräsentiert somit den “Korrelationsgrad” zwischen zwei Funktionen für eine bestimmte Skalierung und Translation. Es ist naheliegend, zur Berechnung der Waveletkoeffizienten wieder auf das innere Produkt der beiden Funktionen  $f(t)$  und  $\psi_{m,n}(t)$  zurückzugreifen:

$$(7.5) \quad W_{m,n}(f) = \int f(t)\psi_{m,n}(t) dt$$

Die Menge aller Waveletkoeffizienten  $W_{m,n}(f)$  definiert die *Wavelettransformation* der Funktion  $f(t)$  in bezug auf das Basiswavelet  $\psi$ . Im Unterschied zu anderen Transformationen ergibt eine Wavelettransformation erst einen Sinn, wenn eine andere Funktion mit ihr assoziiert wird, nämlich diejenige des *mother wavelets*  $\psi$ . Ein Signal  $f(t)$  hat unendlich viele “Wavelettransformationen”, weil viele verschiedene Basiswavelets existieren (für jedes *mother wavelet* existiert eine eigene *WT*).

Die Rekonstruktion des ursprünglichen Signals  $f(t)$  aus einer Wavelettransformation gestaltet sich einfach. Es handelt sich um die mit den Waveletkoeffizienten gewichtete Summe der skalierten und verschobenen Versionen des Basiswavelets  $\psi$ :

$$(7.6) \quad f(t) = \sum_{m,n \in \mathbb{Z}} W_{m,n}(f) \psi_{m,n}(t)$$

### Zusammenfassung

Hier sind die Formeln zur Berechnung der Waveletkoeffizienten  $W_{m,n}(f)$  einer Funktion  $f(t)$  basierend auf einer Haar-Basis noch einmal zusammengefasst:

$$(7.7) \quad W_{m,n}(f) = \int_{2^{m_n}}^{2^{m(n+1)}} f(t) \psi_{m,n}(t) dt$$

mit

$$(7.8) \quad \psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m}t - n)$$

und

$$(7.9) \quad \psi(x) = \begin{cases} 1 & 0 \leq x < 1/2, \\ -1 & 1/2 \leq x < 1, \\ 0 & \text{sonst.} \end{cases}$$

Fourier- und Wavelettransformationen lassen sich einfach miteinander in Bezug setzen. Fouriermodelle repräsentieren Funktionen als gewichtete Summe von Sinus- und Cosinusfunktionen verschiedener Frequenz. Das Gewicht jeder einzelnen Frequenz ist gegeben durch die Fourierkoeffizienten  $a_k$  und  $b_k$  (vgl. Kapitel 6). Analog dazu repräsentieren Waveletmodelle beliebige Funktionen durch eine gewichtete Summe von skalierten und verschobenen *mother wavelets*. Das *mother wavelet* der Wavelettransformation ersetzt die Sinus- und Cosinusfunktion, Dilatation und Translation ersetzen die Frequenzveränderungen und die zweidimensionale Oberfläche der Waveletkoeffizienten ersetzt die eindimensionalen Fourierkoeffizienten (Young 1993).

## 7.2 Wavelettransformation parametrisierter Linien

Die Haar-Funktion wird in der Literatur jeweils bloss als einfaches Beispiel zur Konstruktion einer orthonormalen Waveletbasis beigezogen. Im gleichen Atemzug wird jeweils auf die Nachteile der Haar-Funktion hingewiesen. Kritisiert wird in erster Linie die relativ schlechte Lokalisierung infolge der stückweise konstanten Funktionsabschnitte sowie die Diskontinuität der Funktion (Daubechies 1993). Unter der Voraussetzung, dass die Ursprungslinie nur mit Schrittweiten  $2^k$  ( $k \in \mathbb{N}_0$ ) abgetastet wird, erweist sich eine Haar-Basis für eine Wavelettransformation eines *t-alpha Plots* aber als geradezu ideal. Mit Segmentlängen von 1, 2, 4, ... stellt die Sprungstelle eines Haar-Wavelets kein Problem dar, weil diese dann u.U. genau mit einer Diskontinuität des *t-alpha Plots* zusammenfällt. Die mangelnde Lokalisierung kommt nicht zum Tragen, weil wir es mit einem abgetasteten Signal zu tun haben. Somit bestimmt die Auflösung des *t-alpha Plots* die maximal mögliche Lokalisierung und nicht die Form des Basiswavelets. Eigene Experimente mit dem *Wavelet Packet Laboratory*

(Coifman et al. 1992; WPL 1994) haben ergeben, dass die Wavelettransformation eines *t-alpha Plots* mit einer Haar-Basis mindestens so gut ausfällt wie mit einem anderen, stetigen *mother wavelet*, wie sie z.B. in Abbildung 7.2 dargestellt sind.

Im Gegensatz zur Fourierreihe, wo die bestmögliche Approximation theoretisch nur mit unendlich vielen Koeffizienten erreicht werden kann, ist die Anzahl der Waveletkoeffizienten zur Erzielung der bestmöglichen Approximation endlich und determinierbar:

$m_{min}$ : Das schmalste Wavelet ist abhängig von der Abtastschrittweite  $\Delta t = 2^k$  der Ursprungslinie, welche die Segmentlänge des *t-alpha Plots* bestimmt. Das *Supportintervall*  $2^{m_{min}}$  des schmalsten Wavelets muss doppelt so breit sein wie das kürzeste Liniensegment (vgl. Abb. 7.5), also  $2^{k+1}$ . Daraus folgt:

$$(7.10) \quad m_{min} = k + 1$$

$m_{max}$ : Das breiteste Wavelet ist abhängig von der Länge des *t-alpha Plots*  $t_{max} + 1$  und somit von der Länge der Ausgangslinie. Das *Supportintervall*  $2^{m_{max}}$  des breitesten Wavelets muss so gross sein, dass der ganze *t-alpha Plot* Platz darin findet (vgl. Abb. 7.6), also

$$(7.11) \quad m_{max} = \text{ceil}(\text{lb}(t_{max} + 1)),$$

wobei die C Funktion  $\text{ceil}(x)$  (für engl. *ceiling*, Decke) den kleinsten Ganzzahlwert nicht kleiner als  $x$  zurückgibt (Kernighan und Ritchie 1988). Die Koeffizienten  $m'$  mit  $m' > m_{max}$  werden in der Konstante  $c$  zusammengefasst (Berechnung siehe 7.2.1).

$n_{min}$ : Startposition der Translationen ist der Anfangspunkt des *t-alpha Plots*, somit

$$(7.12) \quad n_{min} = 0$$

$n_{max}$ : Die Anzahl der Translationen ist abhängig von der Länge des *t-alpha Plots*  $t_{max} + 1$  und vom Betrag des Supports der Wavelets mit einer bestimmten Skalierung  $m$  (vgl. Abb. 7.7):

$$(7.13) \quad n_{max}(m) = \text{ceil}\left(\frac{t_{max} + 1}{2^m}\right) - 1$$

### 7.2.1 Berechnung der Verschiebungskonstante $c$

Für Wavelets  $\psi_{m',0}$  mit  $m' > m_{max}$  liegt die gesamte *t-alpha*-Funktion im *positiven* Supportbereich des Wavelets. Mit zunehmendem  $m'$  wird der Beitrag des betreffenden Koeffizienten bedingt durch die Energienormalisierung (vgl. Gleichung 7.1) immer kleiner. Bei der Rekonstruktion der Funktion durch die inverse Wavelettransformation (vgl. Gleichung 7.6) tragen diese Koeffizienten  $W_{m',0}$  dazu bei, dass sich der Graph der Funktion von "unten" her asymptotisch seiner ursprünglichen Lage nähert. Der Beitrag aller  $W_{m',0}$  besteht somit darin, die Kurve durch eine vertikale Translation an ihre eigentliche Lage zu verschieben. Diese Koeffizienten müssen nun aber nicht alle berechnet und aufsummiert werden (es wären zudem unendlich viele), sondern diese Summe lässt sich, da es sich um eine geometrische

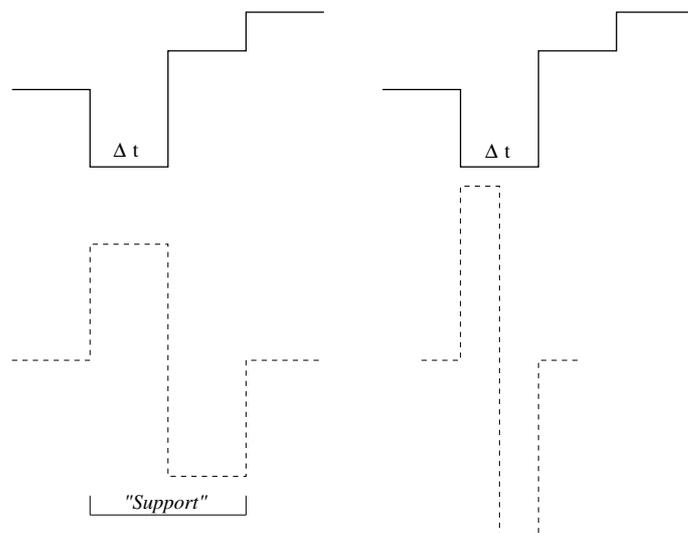


Abbildung 7.5: Eine  $t$ - $\alpha$  Kurve (fett) und ein Haar-Wavelet  $\psi_{m,n}$  (strichliert). Links: Das "schmalste" Wavelet, dessen Koeffizient bei einer Abtastschrittweite  $\Delta t$  nicht Null wird. Dessen *Support* entspricht  $2\Delta t$ . Rechts: Das nächstkleinere Wavelet ist immer vollständig in einem Intervall des  $t$ - $\alpha$  Plots enthalten, wo dieser konstant ist. Ein solcher Waveletkoeffizient ist immer Null, weil das innere Produkt zwischen  $\psi_{m,n}$  und  $\alpha(t)$  Null ist.

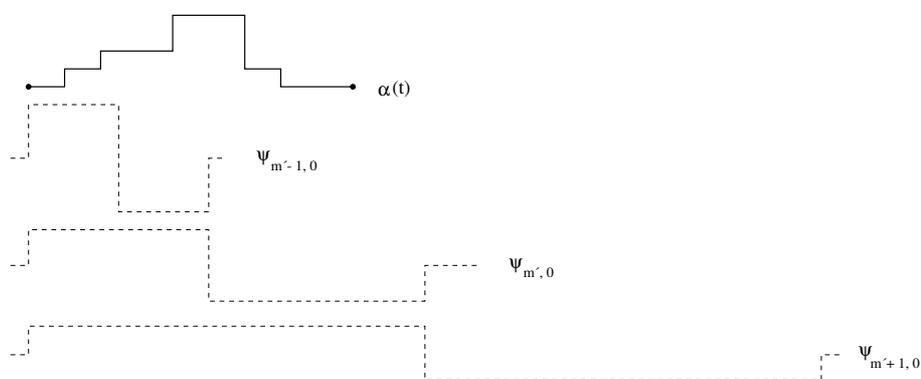


Abbildung 7.6: Ein vollständiger  $t$ - $\alpha$  Plot einer fiktiven Linie (oben) und drei Wavelets mit grossem  $m'$  (unten). Bei  $\psi_{m'-1,0}$  liegt die Funktion nicht vollständig innerhalb des *Supports* des Wavelets. Bei  $\psi_{m',0}$  ist dies der Fall, und die Kurve liegt nicht ausschliesslich im positiven *Support* des Wavelets. Bei  $\psi_{m'+1,0}$  liegt die Kurve vollständig im positiven *Support* des Wavelets. Der Koeffizient für dieses und jedes Wavelet mit noch grösserem  $m$  entspricht dem Integral über der Kurve, gewichtet mit dem Energienormalisierungsfaktor. Diese unendliche Summe der Waveletkoeffizienten mit  $m > m'$  lässt sich einfach berechnen. Es ist somit nicht nötig, diese einzeln zu bestimmen. Daraus folgt, dass  $m_{max}$  gleich  $m'$  sein sollte.

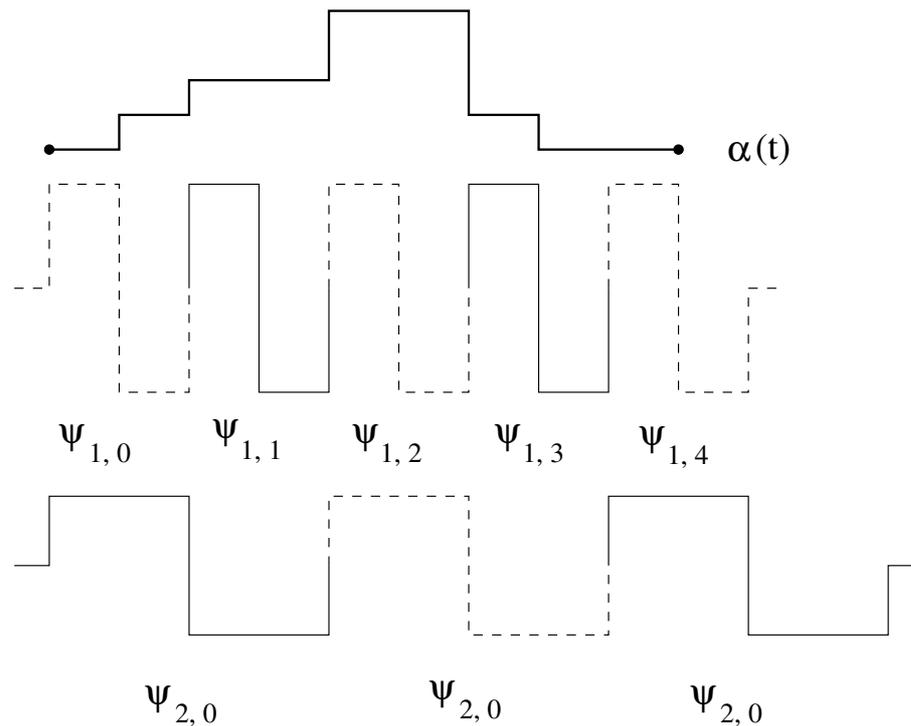


Abbildung 7.7: Die maximale Anzahl an Translationen ( $n_{max}(m)$ ) hängt von der Skalierung  $m$  ab. Sie muss so gewählt werden, dass das betreffende Wavelet  $\psi_{m,n}$  entlang der ganzen Funktion verschoben werden kann.

Reihe handelt, mit einer Formel berechnen. Die Berechnung der  $W_{m',0}$  für  $m' > m_{max}$  nach Formel 7.7 vereinfacht sich zu

$$(7.14) \quad W_{m',0}(f) = 2^{-m'/2} \int f(t) dt$$

weil die ganze Funktion  $f(t)$  im positiven Support des betreffenden Wavelets liegt und sich  $\psi_{m',0}$  (Gl. 7.8) reduziert zu  $2^{-m'/2}$ . Bei der Rekonstruktion nach Gleichung 7.6 schlägt ein solches  $W_{m',0}$  folgendermassen zu Buche:

$$(7.15) \quad W_{m',0}(f)\psi_{m',0}(t) = 2^{-m'/2} \int f(t) dt 2^{-m'/2} = 2^{-m'} \int f(t) dt$$

Die gesuchte Translationskonstante  $c$  ergibt sich aus der Summe

$$(7.16) \quad c = \sum_{\substack{m' > m_{max} \\ m \in \mathbb{Z}}} W_{m',0}(f)\psi_{m',0}(t) = 2^{-m} \int f(t) dt = \int f(t) dt \sum_{\substack{m' > m_{max} \\ m \in \mathbb{Z}}} 2^{-m}$$

Beim Term  $\sum 2^{-m}$  handelt es sich um eine unendliche geometrische Reihe. Deren Summe  $s$  lässt sich allgemein bestimmen aus

$$(7.17) \quad s = \lim_{n \rightarrow \infty} s_n = \frac{a_1}{1 - q} \quad |q| < 1, q = \frac{a_{n+1}}{a_n}, q \neq 0, q \neq 1$$

$a_1 = \text{Anfangswert}$

Faktor  $q$  und Anfangswert  $a_1$  sind in unserem Fall offensichtlich

$$(7.18) \quad q = \frac{2^{-(m+1)}}{2^{-m}} = \frac{1}{2}, \quad a_1 = 2^{-(m_{max}+1)}$$

und die Summe  $s$  demnach

$$(7.19) \quad s = \sum_{m'=m_{max}+1}^{\infty} 2^{-m'} = \frac{2^{-(m_{max}+1)}}{1/2} = 2^{-m_{max}}$$

Gleichung 7.19 eingesetzt in 7.16 ergibt schliesslich die gesuchte Verschiebungskonstante  $c$ :

$$(7.20) \quad c = 2^{-m_{max}} \int f(t) dt$$

handelt. Nun haben wir alle nötigen Formeln zur Hand, um die unendliche Doppelsumme der inversen Wavelettransformation aus Gleichung 7.6 in eine endliche zu verwandeln:

$$(7.21) \quad f(t) = c + \sum_{m=m_{min}}^{m_{max}} \sum_{n=n_{min}}^{n_{max}(m)} W_{m,n}(f) \psi_{m,n}(t)$$

## 7.2.2 Bestimmung der Anzahl Koeffizienten

Wieviele Koeffizienten umfasst nun eine Wavelettransformation einer Linie? Am einfachsten lässt sich diese Anzahl bestimmen, wenn sowohl die Linienlänge als auch die Schrittweite ein Vielfaches von 2 ist. Die Länge kann dann als  $2^l$ , die Abtastschrittweite als  $2^k$  geschrieben werden. Die Anzahl Koeffizienten  $W_N$  ergibt sich dann aus

$$(7.22) \quad W_N = 2^{l-k+1} - 1$$

In der Praxis handelt es sich bei der Länge einer parametrisierten Linie natürlich kaum um ein Vielfaches von 2. Um die Anzahl der Koeffizienten exakt zu bestimmen, muss die Anzahl der Summanden in Gleichung 7.21 ermittelt werden. Eine sehr gute Näherung liefert allerdings

$$(7.23) \quad W_N \approx 2n_{max}(m_{min}),$$

wie aus Abbildung 7.8 klar werden sollte. Für die "Abtastung" des gesamten *t-alpha Plots* mit Länge 20 werden mit dem "schmalsten" Wavelet  $\psi_{0,1}$  exakt 10 Schritte benötigt. Wird das Wavelet weiterverschoben, resultieren nur noch Koeffizienten mit dem Wert Null. Das nächstbreitere Wavelet benötigt noch halb so viele Schritte (weil es ja doppelt so breit ist), nämlich noch fünf. Nun lässt sich die Anzahl der Schritte nicht mehr ganzzahlig teilen, weshalb das Wavelet  $\psi_{3,n}$  drei Schritte benötigt. Die beiden nächstbreiteren Wavelets brauchen noch zwei, bzw. einen Schritt um die ganze Funktion zu überstreichen. Bei der Anzahl benötigter Wavelets handelt es sich folglich um:

$$(7.24) \quad W_N = 10 + \text{ceil}\left(\frac{10}{2}\right) + \text{ceil}\left(\frac{10}{4}\right) + \text{ceil}\left(\frac{10}{8}\right) + \text{ceil}\left(\frac{10}{16}\right) = 21$$

Von der Rundung durch den ceil-Operator abgesehen handelt es sich dabei um die Summe einer geometrischen Reihe. Gleichung 7.23 basiert denn auch auf dieser Annahme. Die Approximation von  $W_N$  durch 7.23 ergibt 20 anstatt der korrekten 21 Koeffizienten.

Dies bedeutet, dass die Anzahl Koeffizienten in etwa gleich gross ist wie die Anzahl Segmente im *t-alpha Plot*. Die Wavelettransformation weist somit eine *lineare Komplexität* in Abhängigkeit der Linienlänge auf.

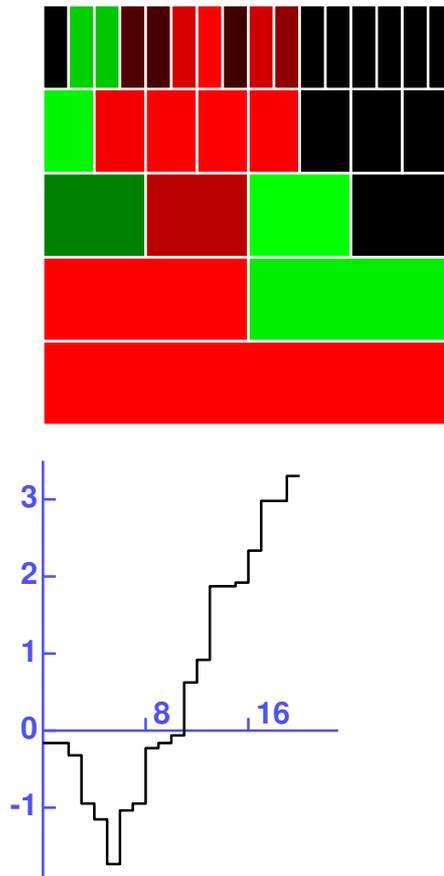


Abbildung 7.8: Darstellung eines *t-alpha Plots* einer mit Schrittweite 1 abgetasteten Linie (unten) und der dazugehörigen Waveletkoeffizienten (oben). Die Breite eines Rechtecks entspricht dem *Support* des fraglichen Wavelets. Links oben befindet sich der Waveletkoeffizient  $W_{1,0}$ , rechts daneben  $W_{1,1}$ , dann  $W_{1,2}$ , usw. Auf der zweiten Zeile folgt ganz links aussen  $W_{2,0}$ , auf der dritten Zeile  $W_{3,0}$  und zuunterst schliesslich  $W_{5,0}$ . Positive Werte sind grün dargestellt, negative rot und Nullwerte schwarz, mit Übergangsfarben für Zwischenwerte.

### 7.2.3 Inverse Wavelettransformation

Die inverse Wavelettransformation eines *t-alpha Plots* gemäss Gleichung 7.21 resultiert in einer perfekten Rekonstruktion der  $\alpha(t)$  Funktion und somit auch der Ausgangslinie. Anders als bei der inversen Fouriertransformation ist bei der Wavelettransformation klar (a) wieviele Koeffizienten berechnet werden müssen und (b) wieviele Koeffizienten zur Rekonstruktion benötigt werden, um die “bestmögliche” Approximation des Ausgangssignals zu erzielen<sup>4</sup>. Im Falle der Wavelettransformation handelt es sich bei (a) und (b) um dieselben Koeffizienten. In Kapitel 6 wurden zur Bestimmung der nötigen Anzahl Koeffizienten die Abbildungen 6.8 bis 6.19 beigezogen. Für die inverse Wavelettransformation wäre das entsprechende Vorgehen nicht nötig. Trotzdem sollen Rekonstruktionen, welche lediglich auf einem Teil der Waveletkoeffizienten basieren, auch hier wiedergegeben werden (allerdings nurmehr für Linie *587ungen*). Zum einen, um zu zeigen, welches Degradationsverhalten eine Wavelettransformation aufweist, zum anderen aber auch, um auf das Potential von Filteralgorithmen hinzuweisen, wie sie Fritsch und Lagrange (1995) vorgeschlagen haben.

Dem “Abschneiden” von Fourierkoeffizienten (Rekonstruktion aus den ersten  $n$  Koeffizienten) entspricht bei der Wavelettransformation das Weglassen einer ganzen Klasse von Koeffizienten. Während durch das Weglassen eines Fourierkoeffizienten die entsprechende Frequenz global für die ganze zu rekonstruierende Funktion eliminiert wird (da keine Lokalisierung vorhanden ist), führt ein analoges Vorgehen bei einer Wavelettransformation lediglich zur *lokalen* Unterdrückung der “Frequenz” (im Bereich des *Supports* des jeweiligen Wavelets). Um globale Veränderungen zu erzielen, müssen demnach alle Waveletkoeffizienten für eine bestimmte Dilatation  $m$  eliminiert werden (entsprechend einer Zeile in der Darstellung der Waveletkoeffizienten in Abbildung 7.8). Wir schreiben solche Klassen von Waveletkoeffizienten fortan als  $W_{1,n}$ ,  $W_{2,n}$ , usw. Werden verschiedene Klassen zusammengefasst, so wird dies als  $W_{1\dots 11,n}$ ,  $W_{2\dots 11,n}$ , usw. geschrieben. In den Abbildungen 7.9, 7.11 und 7.13 sind solche partielle Rekonstruktionen dargestellt.

Wie bei der inversen Fouriertransformation ist es aber auch möglich, die Waveletkoeffizienten zuerst absteigend nach ihrer absoluten Amplitude zu sortieren und erst dann die ersten  $n$  Koeffizienten für die Rekonstruktion auszuwählen<sup>5</sup>. Die Abbildungen 7.10, 7.12 und 7.14 zeigen die Resultate dieser Rekonstruktionsmethode.

Im Vergleich zur Fourierreihe fällt sofort die Angularität der Approximation auf. Dies ist aber nur auf den ersten Blick ein Nachteil, denn es ist bedeutend einfacher, eine kantige Kurve in einem Nachbearbeitungsschritt zu glätten (z.B. mit einem Gauss’schen Filter) als umgekehrt. Es zeigt sich auch, dass die vermeintlich ineffiziente Waveletrepräsentation, was die Anzahl Koeffizienten betrifft, ohne grossen Qualitätsverlust komprimiert werden kann. Auch mit der Hälfte der für eine optimale Approximation nötigen Koeffizienten – 753 statt 1505 – kann eine beinahe perfekte Übereinstimmung mit der Ausgangskurve erzielt werden. Eine amplitudenbasierte Rekonstruktion kommt u.U. sogar mit lediglich 25% der Koeffizienten aus.

Mit der Wavelettransformation wurde somit eine Repräsentation gefunden, welche die in Abschnitt 1.2 geforderte lokalisierte Komponentendarstellung einer Linie erlaubt. Allerdings kann das Kriterium der fixen Länge, welches die Fourierrepräsentation noch erfüllte, bedingt durch die Lokalisierung nicht mehr erfüllt werden. Die Effizienz der Berechnung konnte allerdings gegenüber der Fourierreihe noch gesteigert werden, weil sich sämtliche Iterationen

<sup>4</sup>Im Falle der Fourierreihe wären dies unendlich viele Koeffizienten.

<sup>5</sup>Dieses  $n$  steht in keiner Beziehung zum Translationsindex  $n$  eines Wavelets, bzw. dessen Koeffizienten.

auf das *Supportintervall* des gerade verarbeiteten Wavelets beschränken lassen. Dies gilt sowohl für die Transformation als auch für deren Inverse. Wie bereits im Abschnitt 7.2.2 gezeigt wurde, weisen die Algorithmen zur Berechnung der Wavelettransformation zudem lediglich eine *lineare* Komplexität auf.

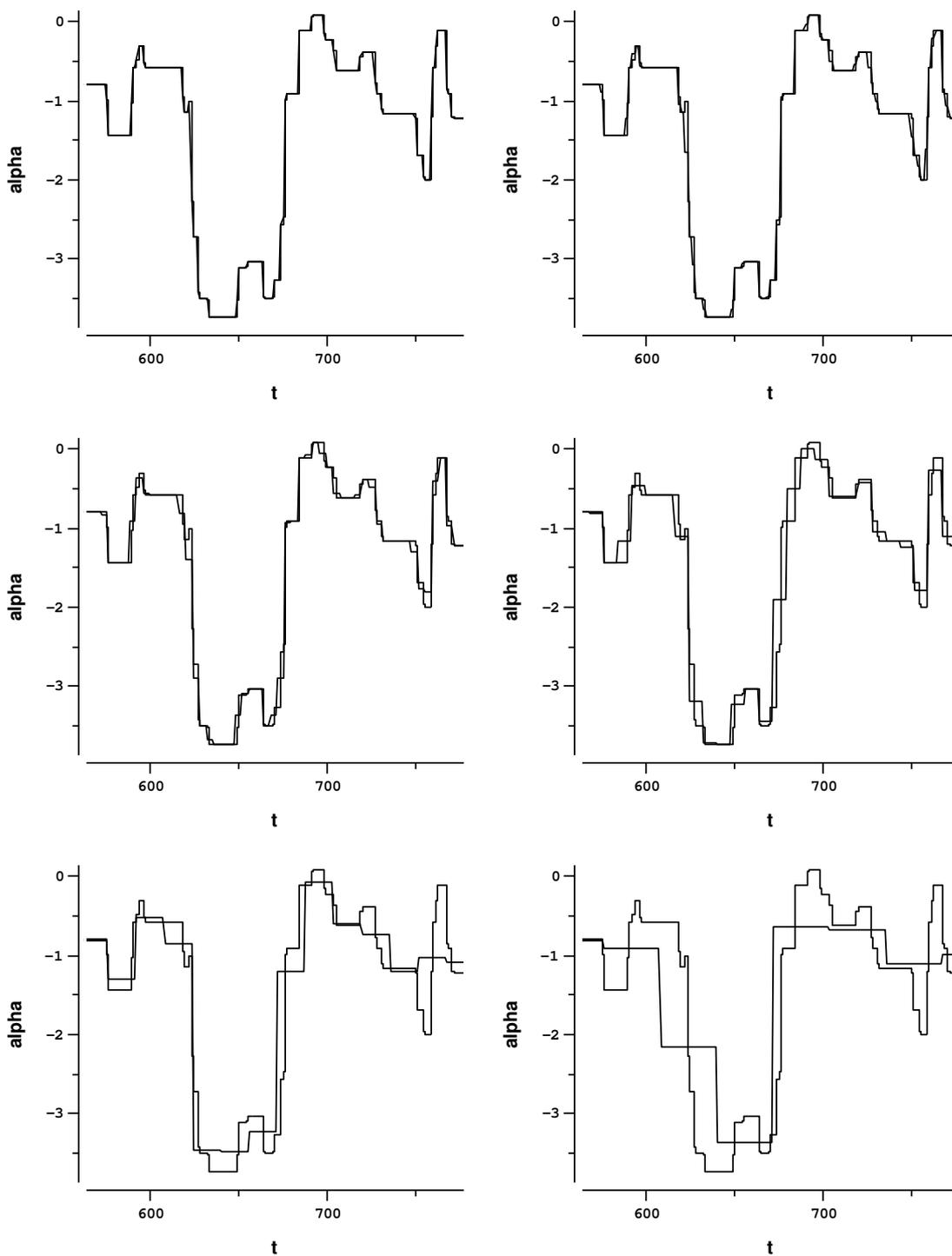


Abbildung 7.9: Ausschnitt aus der Approximation des  $t$ - $\alpha$  Plots der Linie 587ungen (vgl. Abb. 3.3) durch die (von oben links nach unten rechts) Waveletkoeffizienten  $W_{1\dots 11,n}$  (1505),  $W_{2\dots 11,n}$  (753),  $W_{3\dots 11,n}$  (377),  $W_{4\dots 11,n}$  (189),  $W_{5\dots 11,n}$  (95) und  $W_{6\dots 11,n}$  (48). In Klammern stehen die Anzahl Koeffizienten.

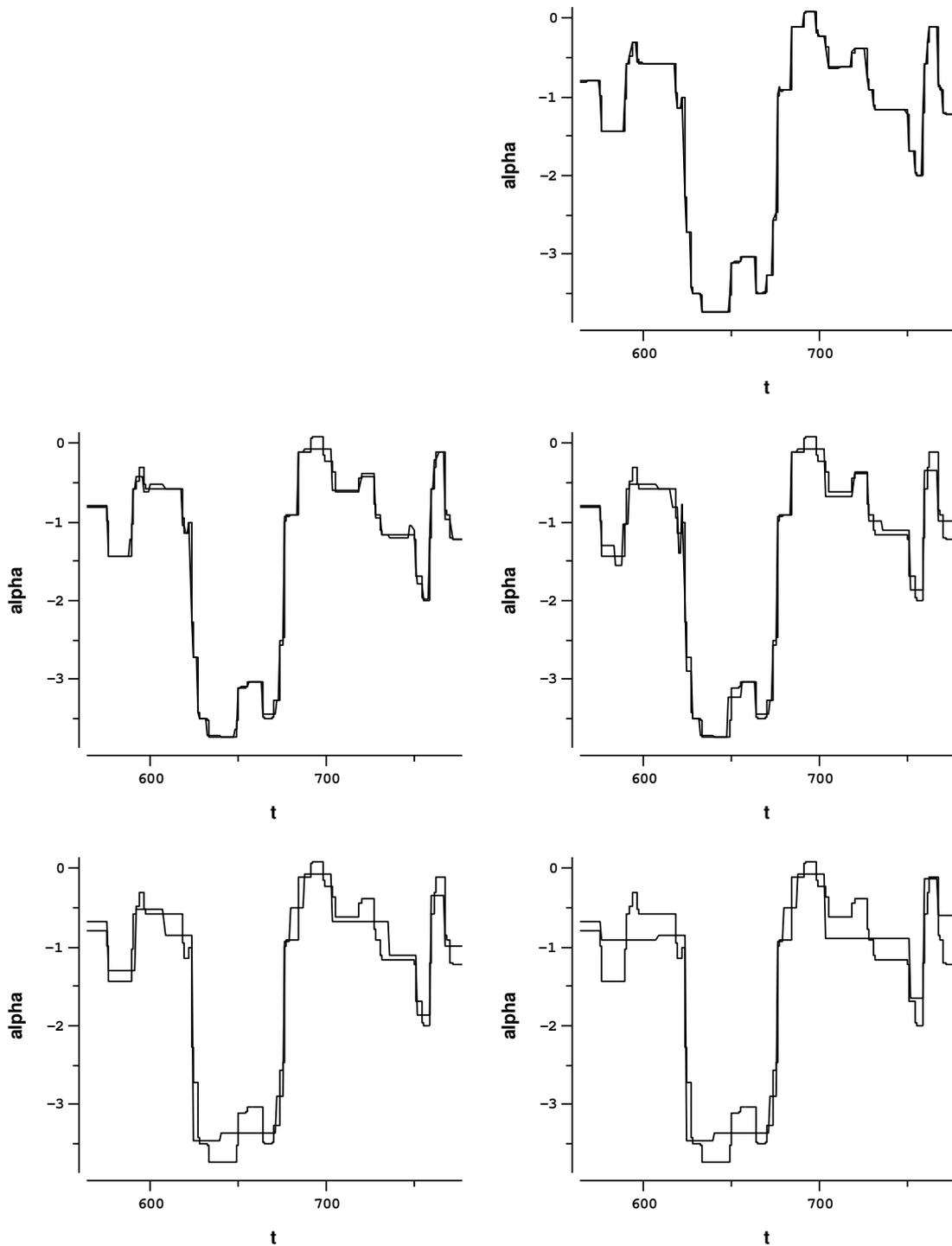


Abbildung 7.10: Ausschnitt aus der Approximation des  $t$ - $\alpha$  Plots der Linie 587ungen durch die (von oben links nach unten rechts) 50% (753), 25% (377), 12% (189), 6% (95), 3% (48) der Waveletkoeffizienten mit der grössten absoluten Amplitude. In Klammern stehen die Anzahl Koeffizienten.

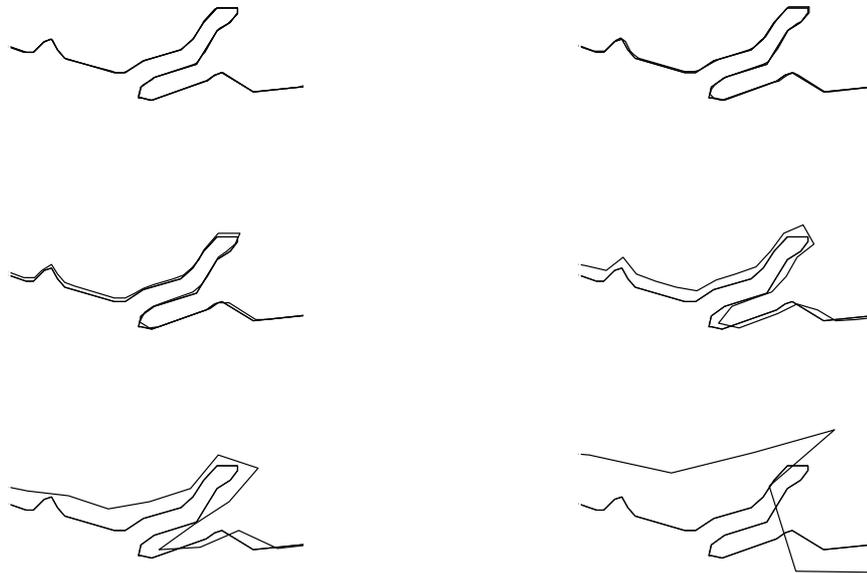


Abbildung 7.11: Rücktransformation der Approximationen aus 7.9 in die  $x, y$ -Ebene und Überlagerung mit der Ausgangslinie *587ungen*. Der Ausschnitt entspricht jenem des *t-alpha Plots*.

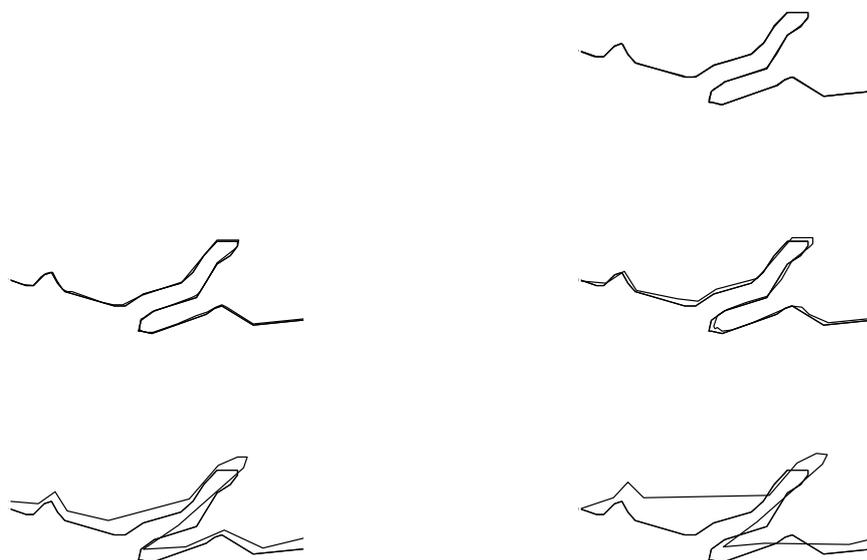


Abbildung 7.12: Rücktransformation der Approximationen aus 7.10 in die  $x, y$ -Ebene und Überlagerung mit der Ausgangslinie *587ungen*. Der Ausschnitt entspricht jenem des *t-alpha Plots*.

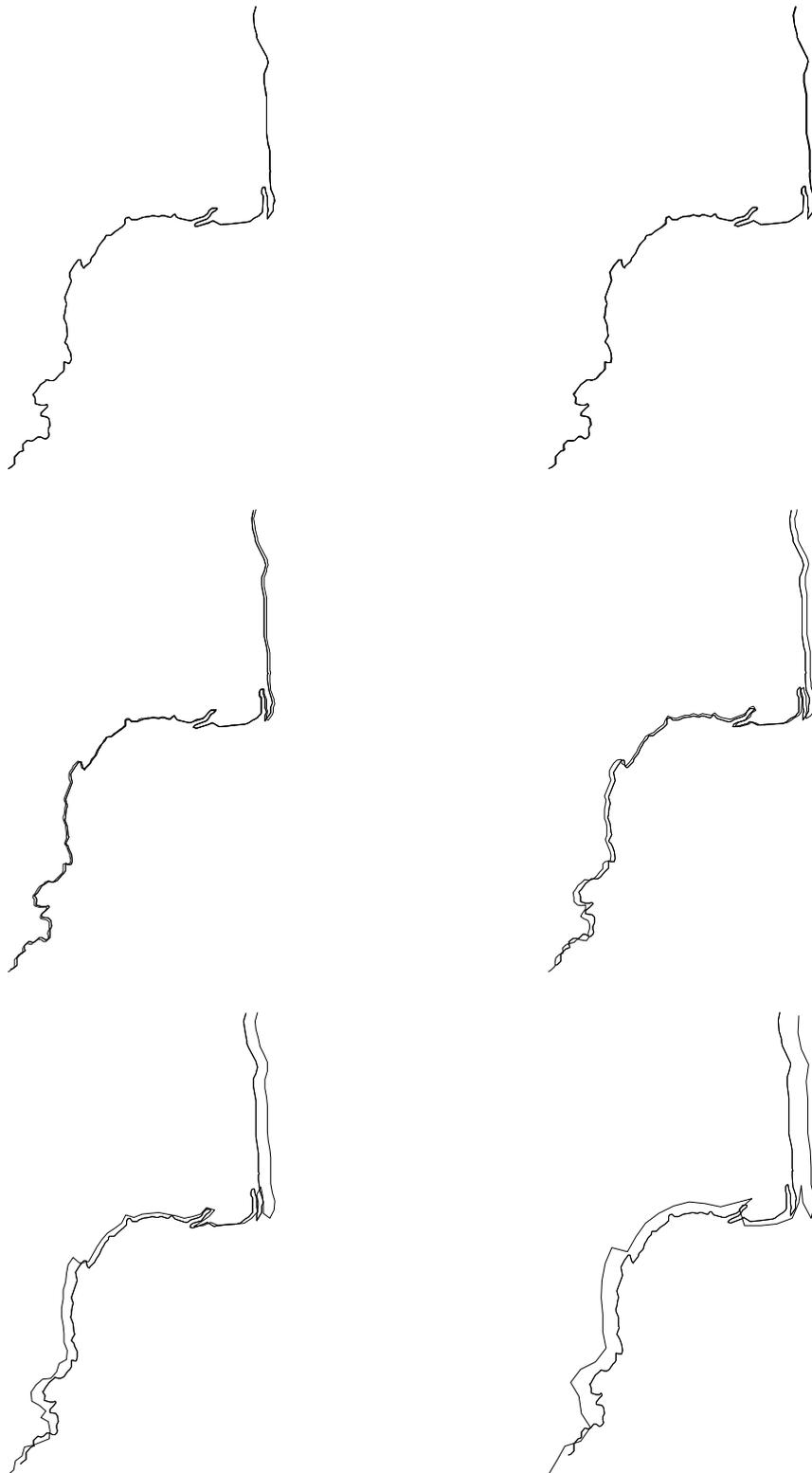


Abbildung 7.13: Approximation der Linie *587ungen* (via *t-alpha Plot*) durch die Waveletkoeffizienten (von oben links nach unten rechts)  $W_{1\dots 11,n}$  (1505),  $W_{2\dots 11,n}$  (753),  $W_{3\dots 11,n}$  (377),  $W_{4\dots 11,n}$  (189),  $W_{5\dots 11,n}$  (95) und  $W_{6\dots 11,n}$  (48), überlagert mit der Ausgangslinie. In Klammern stehen die Anzahl Koeffizienten.

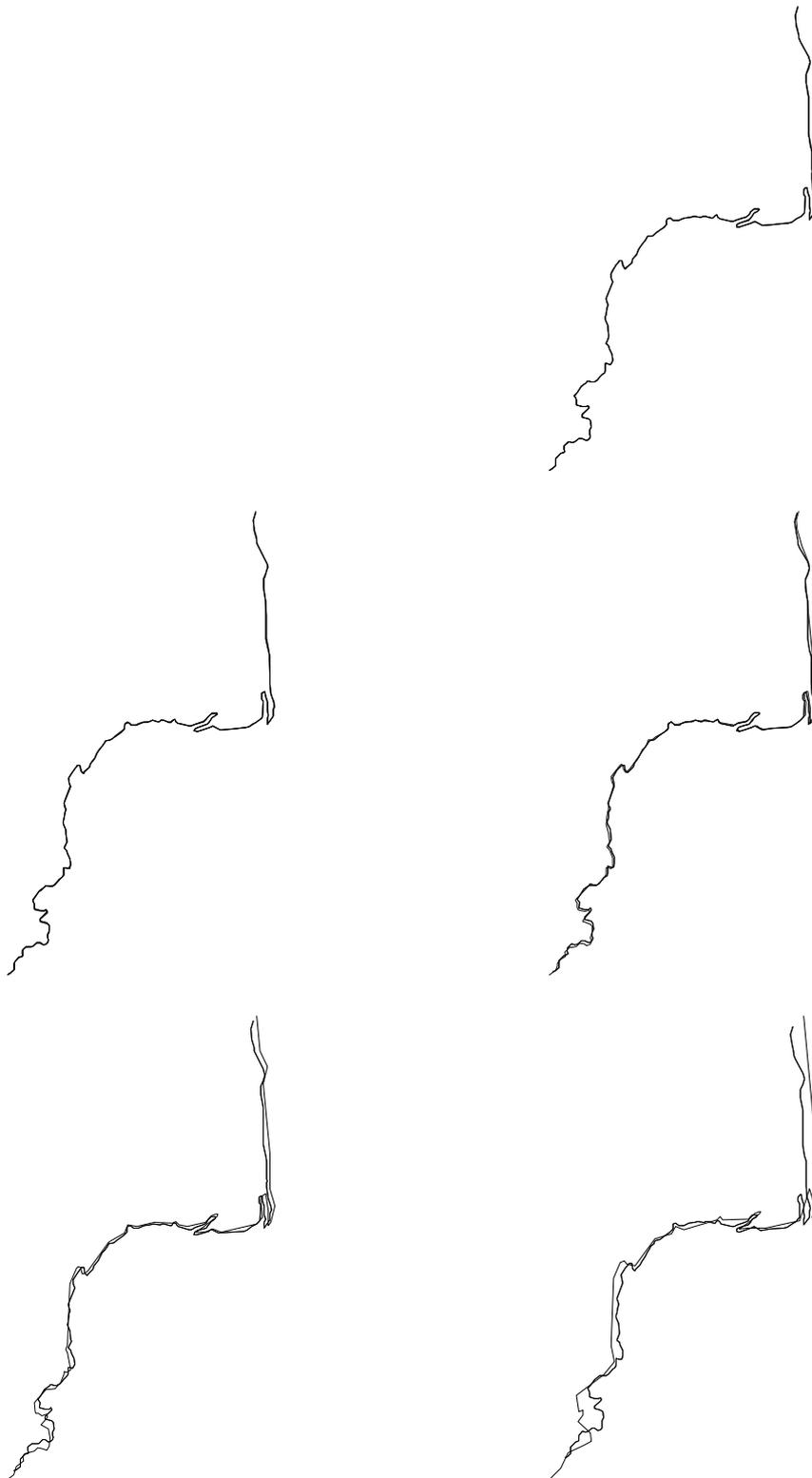


Abbildung 7.14: Approximation der Linie *587ungen* (via *t-alpha Plot*) durch die (von oben links nach unten rechts) 50% (753), 25% (377), 12% (189), 6% (95), 3% (48) der Waveletkoeffizienten mit der grössten absoluten Amplitude, überlagert mit der Ausgangslinie. In Klammern stehen die Anzahl Koeffizienten.

## Kapitel 8

# Untersuchung von Linienpaaren

Im Verlauf dieser Arbeit wurden bislang lediglich *einzelne* Linien transformiert, zuerst in den Parameterraum, dann in den Frequenz- und Waveletraum. Im Hinblick auf die Wissensextraktion aus manuell generalisierten Linien ist jedoch auch von Interesse, wie die Beschreibungen der ungeneralisierten und der generalisierten Version einer Linie korrelieren. Von einer “guten” Repräsentation im Sinne von Abschnitt 2.2 wird erwartet, dass sie das *Stabilitätskriterium* erfüllt. Zwei ähnliche Kurven sollten demnach auch ähnliche Beschreibungen aufweisen. Sie sollten sich lediglich an Stellen unterscheiden, wo auch tatsächlich Abweichungen zwischen den Kurven auftreten. In diesem Kapitel wird vorwiegend auf Probleme hingewiesen, welche auftreten, wenn *Linienpaare* analysiert werden sollen.

Wie aus Abbildung 8.1 ersichtlich ist, steht es um die Stabilität des *t-alpha Plots* schlecht bestellt. Die *t-alpha*-Repräsentation teilt dieses Schicksal mit allen anderen kurvilinearen Parametrisierungen. Dadurch, dass der Bezug zum Ortsraum einzig und alleine durch den Anfangspunkt der Parametrisierung gegeben ist, pflanzen sich *lokal* auftretende Verkürzungen, wie sie für die Generalisierung typisch sind (z.B. infolge Eliminierung einer Kleinform), über die ganze Länge der Linie fort. Es ist offensichtlich, dass die Repräsentation nicht a priori instabil ist, die beiden *t-alpha Plots* weisen sehr wohl eine ähnliche Gestalt auf. Das grosse Problem ist die fehlende Lokalisierung. Abhilfe könnte eine Dehnung der kürzeren, generalisierten Linie auf die Länge der ungeneralisierten Linie schaffen. Eine einfache lineare Dehnung, wie sie in Abbildung 8.2 dargestellt ist, genügt allerdings nicht. Die Korrelation verbessert sich lediglich an den beiden Enden der Kurve, dazwischen ist keine Verbesserung auszumachen. Mehr Erfolg dürfte eine nicht-lineare Dehnung (bzw. eine stückweise lineare) versprechen, so dass die entsprechenden lokalen Extrema der beiden *t-alpha Plots* übereinander zu liegen kommen.

Angenommen, es wird eine Dehnungsmethode gefunden, welche es erlaubt, die *t-alpha*-Repräsentationen eines Linienpaars zu synchronisieren, dann könnten in einem zweiten Schritt die Wavelettransformationen der beiden Linien berechnet werden. Die Residuen<sup>1</sup> der beiden Transformationen (Koeffizienten) beschreiben dann aber erst, wie aus der Wavelettransformation der ungeneralisierten Kurve jene der *gestreckten* generalisierten Kurve erhalten werden kann. Das bedeutet, dass z.B. ein Neuronales Netz neben der Bestimmung der neuen Koeffizientenwerte auch gleich noch die Stauchungsfaktoren bestimmen müsste, was wenig erfolgversprechend scheint.

---

<sup>1</sup>Es könnte sich auch um multiplikative Relationen handeln. Für die folgenden Überlegungen ist die Art der Übergangstransformation unerheblich.

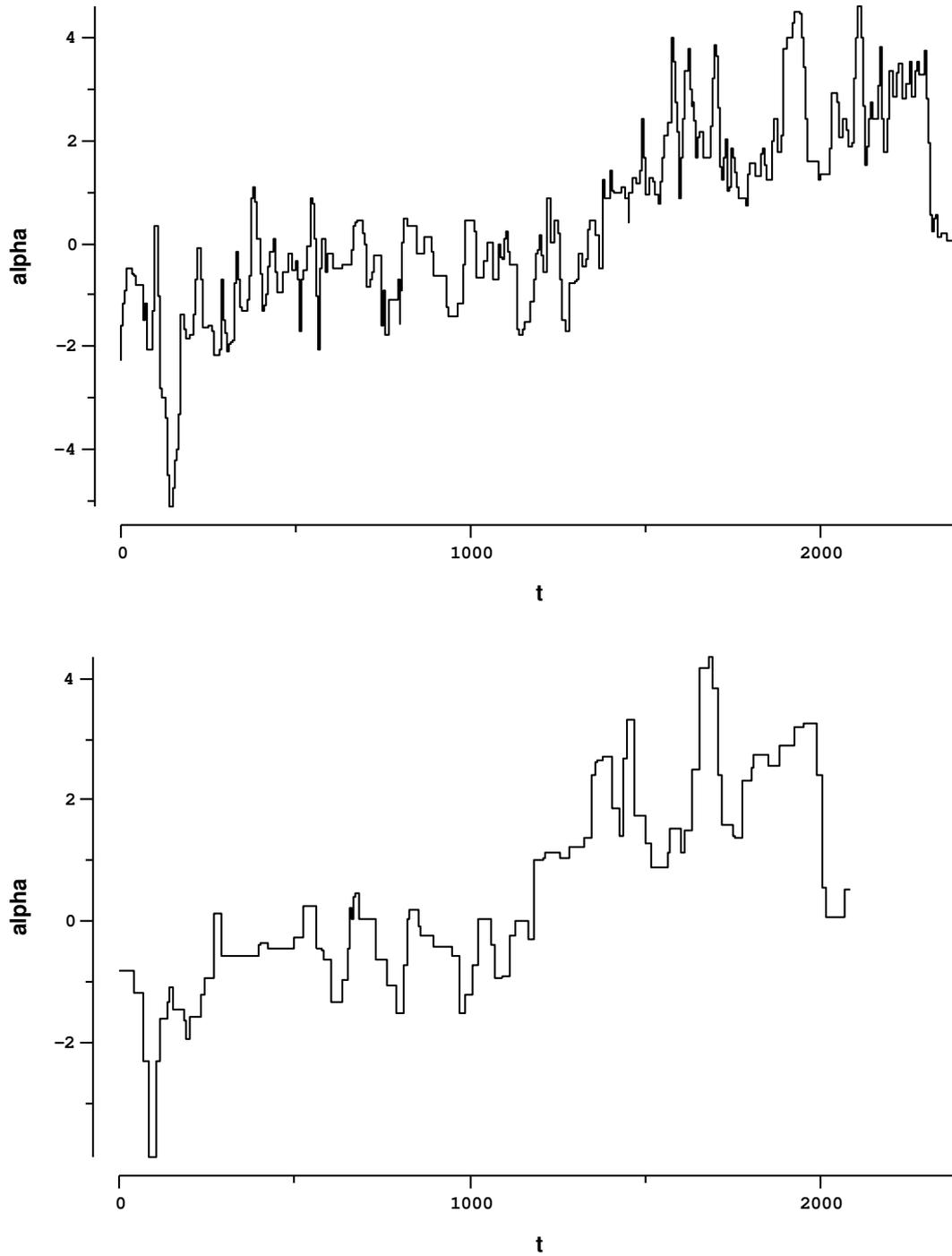


Abbildung 8.1: Die  $t$ - $\alpha$  Plots der Linien 501ungen (oben) und 501gen (unten).

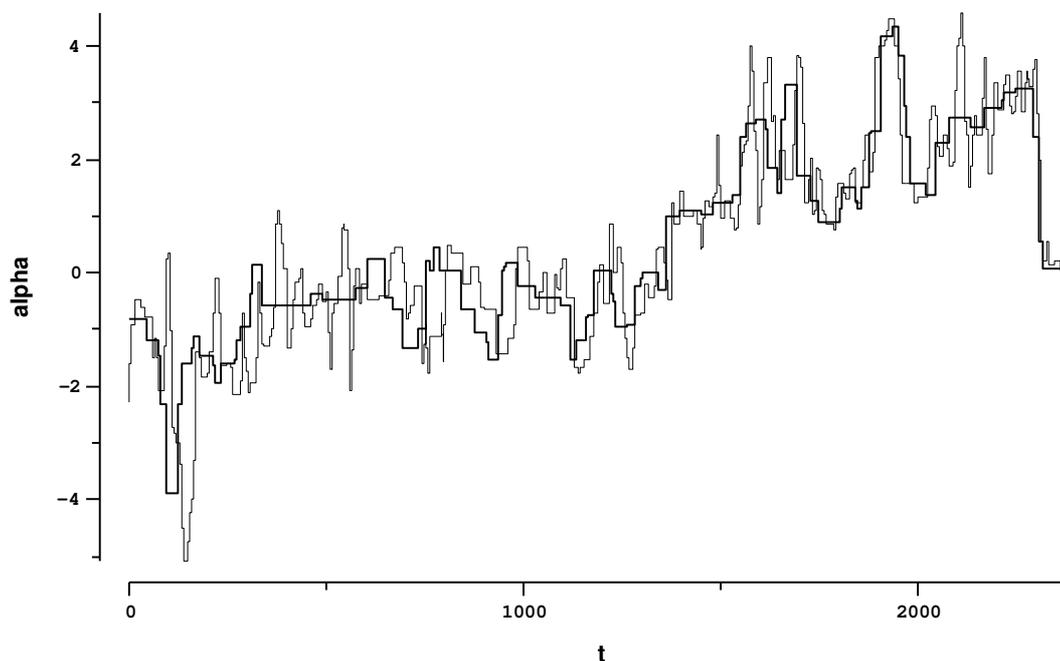


Abbildung 8.2: Linear gedehnte Version des  $t$ - $\alpha$  Plots der Linie 501gen.

Unter diesem Gesichtspunkt betrachtet erscheint die vielgepriesene Lokalisierung der Wavelettransformation plötzlich als Nachteil. Die Fouriertransformation andererseits ist resistent gegen variierende Linienlängen. Unabhängig von der Länge der einzelnen Linie eines Paares können mit jeweils gleich vielen (und genügend vielen) Koeffizienten beide Linien rekonstruiert werden. Wie zu befürchten, hat auch dieser Ansatz einen Pferdefuss. Von der Fouriertransformation kann nämlich mit Blick auf Abbildung 8.4 behauptet werden, dass sie nicht stabil ist. Dargestellt sind die ersten 100 Fourierkoeffizienten des Linienpaares 488, welches in Abbildung 8.3 wiedergegeben ist. Die beiden Linien können ohne weiteres als sehr ähnlich bezeichnet werden, abgesehen von einer Krümmung im Mittelteil der Ausgangskurve, welche in der manuell generalisierten Version unterdrückt wurde. Wäre das Stabilitätskriterium erfüllt, würde man erwarten, dass sich die Fouriertransformationen dieser beiden Linien über weite Strecken sehr ähnlich sind. Dies ist offensichtlich nicht der Fall.

Anders die Waveletkoeffizienten, wie sie aus Abbildung 8.5 hervorgehen. Abgesehen von einer leichten Positionsverschiebung, bedingt durch eine ebenfalls geringe Linienverkürzung, ist, mit Ausnahme des Bereichs zwischen  $t = [376, \dots, 465]$ , eine relativ gute Korrelation auszumachen. Positive Koeffizienten sind grün, negative rot und solche mit Wert Null schwarz dargestellt. Es ist zu beachten, dass die Farbwerte nicht über Zeilen hinweg verglichen werden können; dem jeweils grössten Wert einer Zeile  $m$  wird 100 % Grün (d.h. RGB x00ff00), dem kleinsten 100% Rot (RGB xff0000) zugewiesen. Diese Skalierung wurde nötig, weil die Koeffizientenamplituden mit aufsteigendem  $m$  zunehmen. Eine Skalierung über alle Zeilen hätte zur Folge, dass die Koeffizienten im oberen Bereich der Darstellung grösstenteils schwarz dargestellt würden.

Es ist ebenfalls ersichtlich, dass, wie schon aus dem  $t$ - $\alpha$  Plot, auch aus dessen Wave-

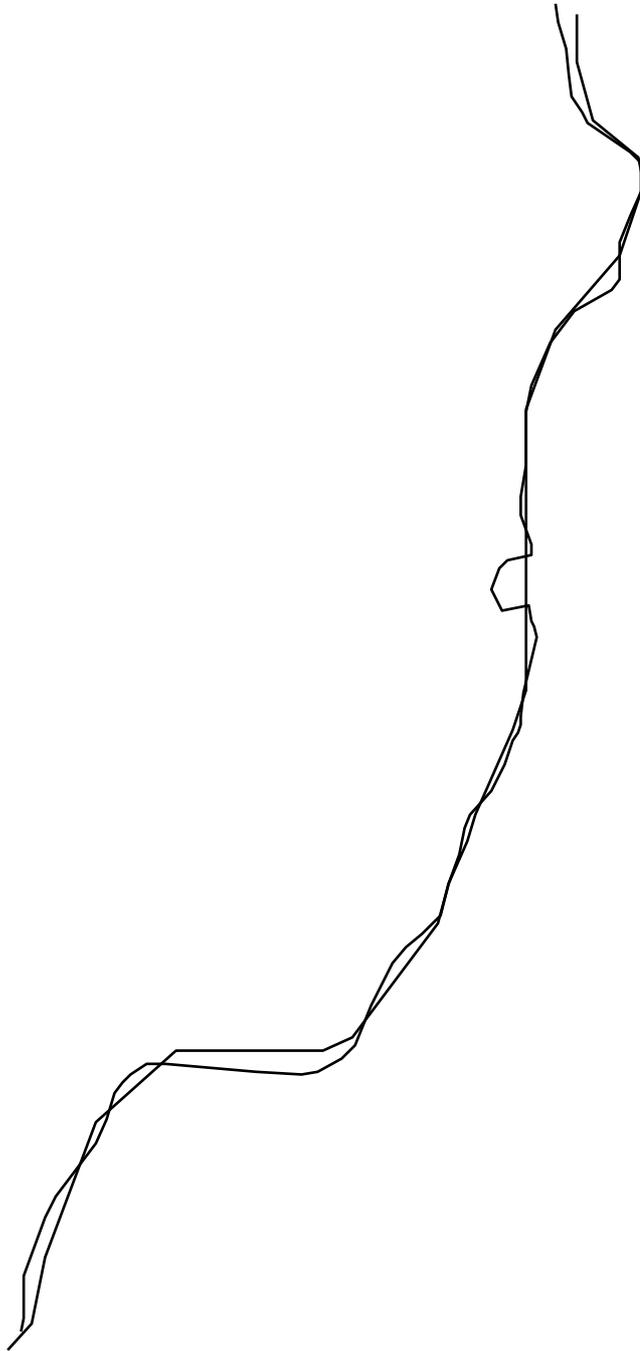


Abbildung 8.3: Die Linie *488ungen* und die manuell generalisierte Linie *488gen*.

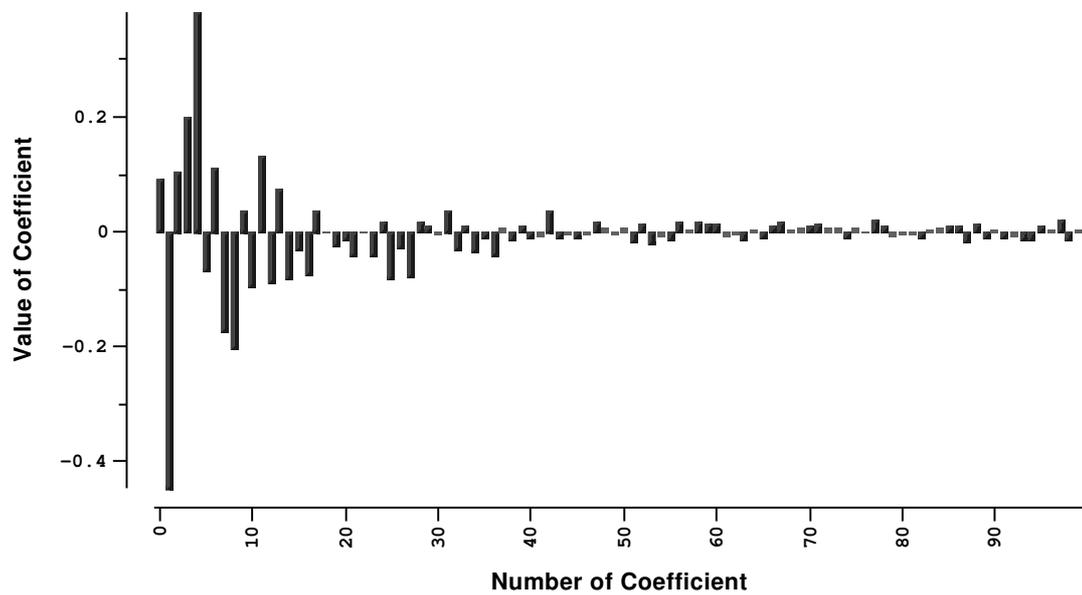
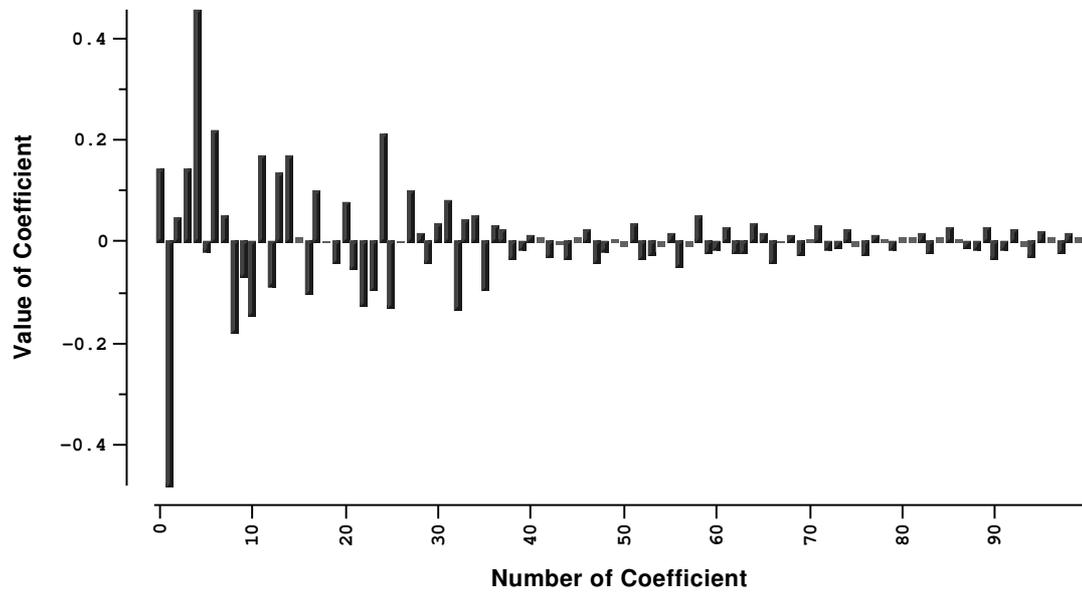


Abbildung 8.4: Die ersten 100 Fourierkoeffizienten der Linien *488ungen* (oben) und *488gen* (unten).

lettrtransformation Formeigenschaften der Kurve abgelesen werden können. Findet innerhalb des *Supports* eines Wavelets eine *Abnahme* des  $\alpha$ -Wertes statt (d.h. die Kurve beschreibt eine *Rechtskurve* in der  $x, y$ -Ebene), dann resultiert dies in einem *positiven* (d.h. grün dargestellten) Waveletkoeffizienten. Die Begründung dafür ist einfach zu geben: eine Abnahme des  $\alpha$ -Wertes bedeutet, dass der Graph des *t-alpha Plots* links “höher” ist als rechts. Genau diese Form weist aber auch unser Basis-(Haar)-Wavelet auf. Das innere Produkt (vgl. Gl. 7.5) zwischen dem Wavelet und dem betreffenden Abschnitt des *t-alpha Plots* ist somit positiv (“hohe Korrelation”). Das Analoge gilt, mit umgekehrten Vorzeichen, für eine Zunahme von  $\alpha$ ; die Koeffizienten werden negativ und somit rot dargestellt. Kommt ein Wavelet vollständig in einen Bereich zu liegen, in dem *t-alpha* konstant ist, wird das innere Produkt, und somit auch der Koeffizient, Null. Diese hohe negative Korrelation der Waveletkoeffizienten mit  $\Delta\alpha$  hat zur Folge, dass auch aus der Wavelettransformation die Wendepunkte, welche die Kurve in der  $x, y$ -Ebene aufweist, abgelesen werden können: sie liegen im Bereich, wo die Koeffizienten für ein gegebenes  $m$  ihr Vorzeichen wechseln. Die Wendepunkte können natürlich am genauesten in der Waveletklasse  $m_{min}$  lokalisiert werden.

Nun soll noch kurz auf die Koeffizienten zwischen  $t = [376, \dots, 465]$  eingegangen werden. In diesem Bereich liegt offensichtlich keine Korrelation zwischen den Koeffizienten der ungeneralisierten und generalisierten Linie vor. Das ist aber auch gut so: beim gewählten Ausschnitt handelt es sich nämlich genau um jene Stelle, wo die beiden Kurven im Ortsraum die grössten Unterschiede aufweisen. Wie es sich für eine stabile Repräsentation gehört, unterscheiden sich an dieser Stelle auch die beiden Wavelettransformationen am stärksten. Abbildung 8.6 zeigt, wie sich die rekonstruierte Linie *488ungen* im Ortsraum verhält, wenn – analog zur Wavelettransformation von *488gen* – den Koeffizienten in den obersten 6 Zeilen ( $m_{1\dots5}$ ), deren *Support* vollständig im Intervall  $t = [376, \dots, 465]$  liegt, der Wert Null zugewiesen wird. Die Kurve wird, wie erhofft, eliminiert. An diesem Beispiel konnte somit gezeigt werden,

- dass aufgrund einer Analyse der Residuen zwischen den Wavelettransformationen eines Linienpaars entschieden werden kann, wo eingegriffen werden muss
- dass ein lokaler Eingriff tatsächlich möglich ist.

Der letzte Punkt bedarf einer Erläuterung. Von “lokal” kann nämlich nicht direkt die Rede sein, weil sich die rekonstruierte Kurve in ihrer Gesamtheit von der Ursprungslinie wegbewegt hat. Die Ursache liegt wiederum in der bereits mehrmals erwähnten Unzulänglichkeit des *t-alpha Plots* begründet: es ist unmöglich, durch die Modifikation eines *t-alpha Plots* eine Linienverkürzung zu erreichen. Durch die Streckung<sup>2</sup> einer Kurve, wie in obigem Beispiel geschehen, wird der ganze restliche Teil der Kurve verschoben. Bleibt noch zu erklären, weshalb die rekonstruierte Kurve dann auf der ganzen Länge vom Original abweicht und nicht erst am Ort des “chirurgischen Eingriffs” einsetzt. Dem ist so, weil eine schwerpunktbasierte Rücktransformation zum Zuge kommt, wie sie in Abschnitt 5.4.3 beschrieben wurde. Obwohl sich die Kurve nun an keiner Stelle mehr mit dem Original deckt, ist eine schwerpunktbasierte Rücktransformation einer fixpunktbasierten vorzuziehen (vgl. dazu auch das Beispiel der Uhr im Vorwort).

---

<sup>2</sup>Das Ziel wäre eigentlich eine *Verkürzung*, welche durch eine Veränderung der *t-alpha* Werte jedoch nie erzielt werden kann, weil lediglich die Richtungswinkel der Abtastsegmente verändert werden können, jedoch nicht deren Anzahl.

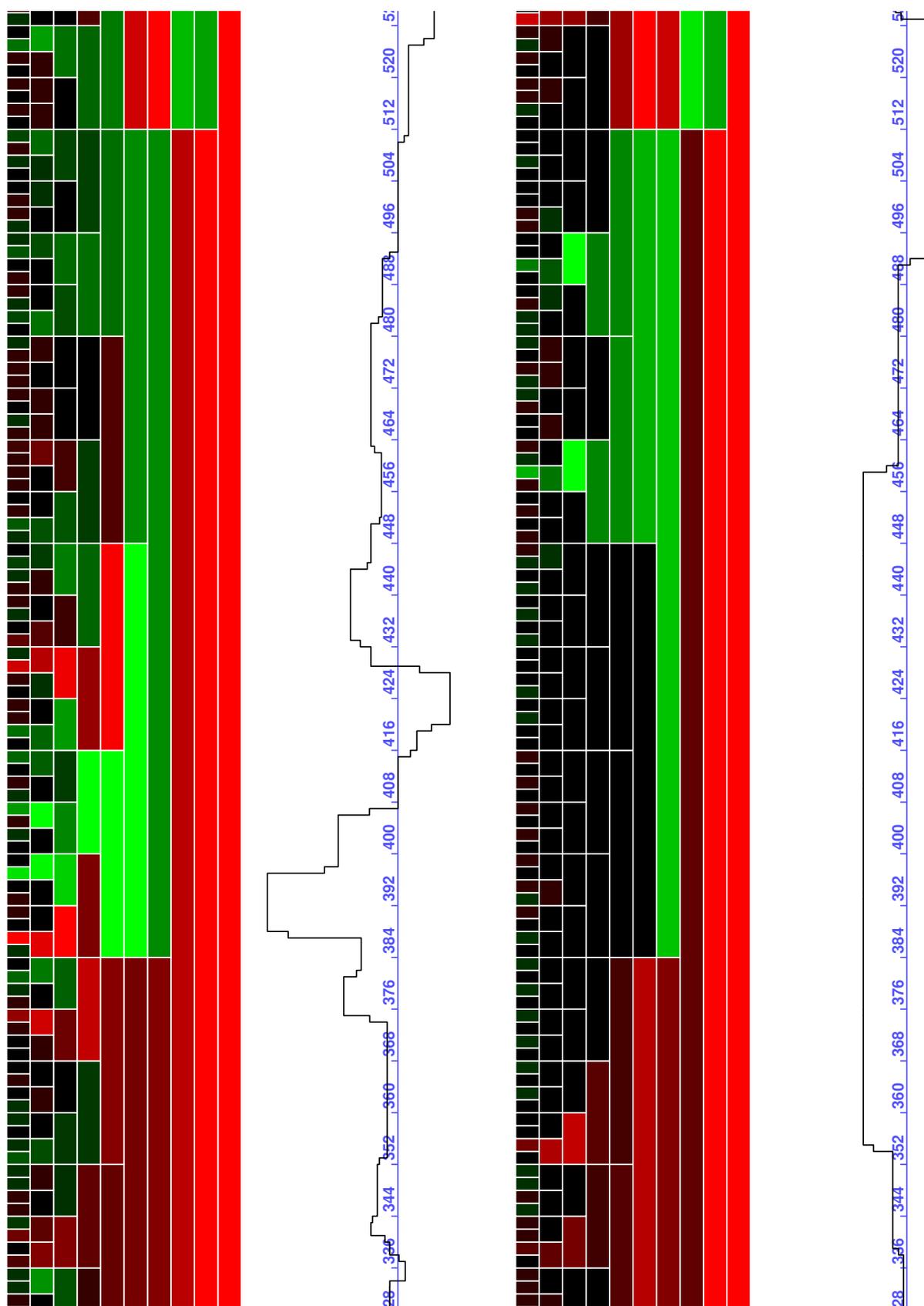


Abbildung 8.5: Teil der Waveletkoeffizienten der Linien *488ungen* (oben) und *488gen* (unten). Positive Werte sind grün dargestellt, negative rot und Nullwerte schwarz, mit Übergangsfarben für Zwischenwerte.

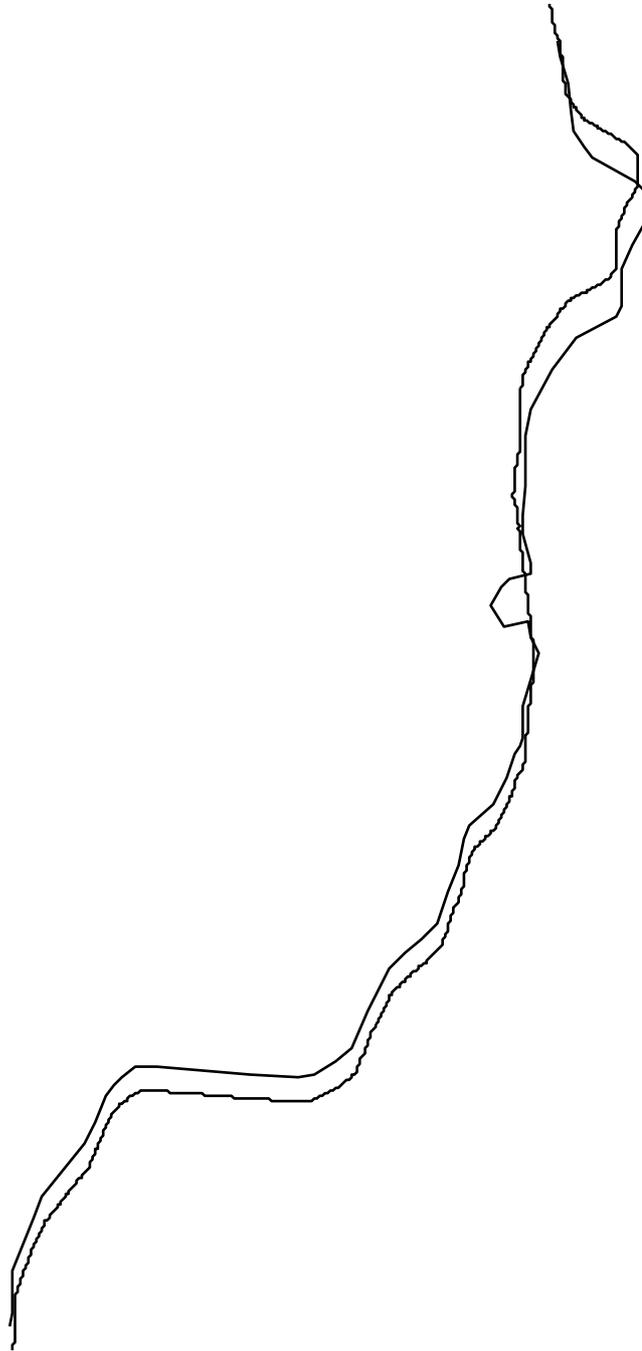


Abbildung 8.6: Die Linie 488ungen nach einem lokalen Eingriff in die Wavelettransformation. Die Kurve im Mittelteil konnte zum Verschwinden gebracht werden.

## Kapitel 9

# Schlussfolgerungen und Ausblick

### 9.1 Was wurde erreicht?

Im Rahmen dieser Arbeit wurde Schritt für Schritt gezeigt, wie eine Polylinie aus dem Ortsraum ( $x, y$ -Ebene) in den Frequenzraum (Fourierraum, Waveletraum) überführt werden kann. Die Gliederung des ersten Teils der Arbeit in die Kapitel *Abtastung*, *Normalisierung* und *Parametrisierung* entspricht der Reihenfolge der Vorverarbeitungsschritte, denen eine Linie unterzogen werden muss, um sie schliesslich in eine frequenzbasierte Repräsentation zu transformieren.

Es zeigte sich, dass bereits an die Parametrisierungsmethode hohe Anforderungen gestellt werden müssen, insbesondere was die Berechnung von Formmassen betrifft. Der *Psi-s Plot* genügte diesem Kriterium, bedingt durch die Diskontinuität bei der Richtungswinkelbestimmung, nicht. In Anlehnung an *Kurvaturelemente*, wie sie für *Chain Codes* berechnet werden (vgl. 5.4.1), konnte eine Formel gefunden werden, welche die künstlichen Sprungstellen des *Psi-s Plots* eliminiert. Bei der resultierenden Funktion, welche wir *t-alpha* genannt haben, handelt es sich um eine *kumulative* Darstellung der relativen Richtungsänderungen der Kurve. Die lokalen Extrema eines *t-alpha Plots* entsprechen den Wendepunkten der Kurve in der  $x, y$ -Ebene, aufgrund derer sehr einfach Formmasse der Kurve bestimmt werden können. Sprungstellen in einem *t-alpha Plot* stellen nun keine Artefakte mehr dar, wie dies im *Psi-s Plot* noch der Fall war, im Gegenteil, sie können nun auf solche hinweisen. Deutlich sichtbare Ausreisser in einer *t-alpha*-Repräsentation weisen nämlich vielfach auf Digitalisierfehler hin, welche in der  $x, y$ -Ebene von blossen Auge kaum zu entdecken sind. Durch die Vergrösserung der Abtastschrittweite können solche kleinsten Knitterungen eliminiert werden. In diesem Beispiel kommt zum Vorschein, dass die verschiedenen Repräsentationsräume nicht isoliert betrachtet werden sollten und Rückkopplungen jederzeit möglich sein müssen. Diesem Umstand wurde in der entwickelten Softwareplattform *REPTILE* (siehe auch Anhang A) Rechnung getragen. In *REPTILE* muss diese Rückpropagierung durch den Benutzer oder die Benutzerin ausgelöst werden (Anpassung der Schrittweite). Es ist jedoch denkbar, dass ein System einen durch einen Repräsentationswechsel entstandenen "Erkenntnisgewinn" auswertet und die nötigen Anpassungen automatisch vornimmt. Im Hinblick auf das von Weibel (1991) propagierte Konzept der *Amplified Intelligence* wäre es jedoch vorzuziehen, wenn das System lediglich auf diesen Umstand hinweisen würde. Nur so behält der Benutzer die volle Kontrolle über den Prozess.

Basierend auf einer *t-alpha*-Repräsentation wurden in der Folge die Fouriertransforma-

tion einer Linie und deren Haar-Wavelettransformation berechnet. Es zeigte sich, dass sich eine Kurve durch beide Transformationen, genügend Koeffizienten vorausgesetzt, praktisch fehlerfrei rekonstruieren lässt. Unterschiede traten jedoch im Degradationsverhalten zutage. Rekonstruktionen mit sukzessive abnehmender Anzahl Koeffizienten führen bei der Fourierreihe zu sehr glatten, geschwungenen Kurven, während die aus Wavelets rekonstruierten Kurven kantige Formen annehmen. Dieser Unterschied liegt natürlich in den verschiedenartigen Basisfunktionen begründet – die stetige Cosinusfunktion einerseits, die stückweise konstante Haar-Funktion andererseits. Das vermeintlich schlechte Degradationsverhalten beider Transformationen *im Ortsraum* – störend sind in erster Linie die grossen Verschiebungen bezogen auf die Ausgangskurve – rührt einzig und alleine von der Unmöglichkeit her, eine *t-alpha*-parametrisierte Linie zu verkürzen. Das *tatsächliche* Degradationsverhalten der beiden spektralen Transformationen lässt sich denn auch nur im Parameterraum beurteilen. Mit Blick auf die Abbildungen 6.8, 6.10 und 7.9 kann dieses durchaus als gut bezeichnet werden.

## 9.2 Ausblick auf zukünftige Entwicklungen

Probleme, welche in dieser Arbeit nicht gelöst werden konnten, betreffen Linienverkürzungen im *t-alpha*-Raum und zwar in zweifacher Hinsicht:

**Transformation nach *t-alpha*:** Unterschiedliche Längen von generalisierter und ungeneralisierter Linie führen nach einer kurvilinearen Parametrisierung zu asynchronen *t-alpha Plots*.

**Rücktransformation aus *t-alpha*:** Durch eine Manipulation der  $\alpha$ -Werte lassen sich lediglich Streckungen der Kurve erzielen, aber keine Verkürzungen.

Beide Defizite rühren von der fehlenden Verankerung des *t-alpha Plots* im Ortsraum und dem ausschliesslich rekursiven Aufbau der Repräsentation her. Es müssen Mittel und Wege gefunden werden, diese Klippen zu umschiffen. Priorität hat dabei die Suche nach einer Methode zur Linienverkürzung bei der Rücktransformation aus dem Parameterraum (entspricht dem zweiten Punkt in obiger Liste). Ohne eine solche zur Hand zu haben, führen lokale Eingriffe in die Wavelettransformation (interaktiv oder mit Hilfe speziell entwickelter Algorithmen), zu unbefriedigenden Resultaten. Erst in zweiter Linie sollte versucht werden, durch Ausgleichsrechnungen zu synchronen *t-alpha Plots* eines Linienpaars zu gelangen, weil Überlagerungen erst aktuell werden, wenn es um die Wissensextraktion aus bestehenden Linienpaaren geht (beispielsweise zur Verwendung künstlicher neuronaler Netze).

Gelingt es, das Linienverkürzungsproblem in den Griff zu bekommen, haben wir mit der Wavelettransformation eine sehr potente Repräsentation zur Hand. Da es sich bei der WT um eine Mischung aus *multiscale Representation* (vgl. CSSI in Abschnitt 2.3.2) und klassischer spektraler Repräsentation handelt, ergibt sich eine sehr hohe Informationsdichte:

- Information über das Verhalten des Signals in verschiedenen Massstäben.
- Deutung von Vorzeichenwechseln als Wendepunkte.
- Lokalisierte Information über die Bausteine, aus welchen das Signal aufgebaut ist.

In einem ersten Schritt sollten einfache Filteralgorithmen zur Linienvereinfachung entwickelt werden. Aufgrund der hohen Informationsdichte der WT könnte allenfalls auch komplexeren Algorithmen, etwa zur Implementierung von speziellen Generalisierungsoperatoren wie z.B. der Karikatur, Erfolg beschieden sein. Als Fernziel könnte schliesslich versucht werden, bestehende, wavelettransformierte Linienpaare für das Training von Neuronalen Netzen beizuziehen. Dabei ist allerdings der Auswahl der Trainingsdaten grösste Aufmerksamkeit zu schenken. Im Verlaufe dieser Arbeit hat sich nämlich gezeigt, dass manuell generalisierte Linien nicht a priori eine hohe Generalisierungsqualität garantieren, bzw. diese Qualität zumindest nicht immer ersichtlich war, weil die Linienpaare nicht in ihrem Kontext beurteilt wurden (vgl. dazu auch Ehrliholzer 1996).



# Literaturverzeichnis

- Bahrenberg, Gerhard, E. Giese und J. Nipper (1990). *Statistische Methoden in der Geographie*, Bd. 1, Univariate und bivariate Statistik d. Reihe *Teubner Studienbücher der Geographie*. Teubner Stuttgart, 3. Aufl.
- Ballard, Dana H. und C. M. Brown (1982). *Computer Vision*. Prentice-Hall, Englewood Cliffs.
- Brown, James Ward und R. V. Churchill (1993). *Fourier Series and Boundary Value Problems*. McGraw-Hill, New York, 5. Aufl.
- Buttenfield, Barbara P. (1985). *Treatment of the Cartographic Line*. *Cartographica*, 22(2):1–26.
- Coifman, Ronald R., Y. Meyer und V. Wickerhauser (1992). *Wavelet Analysis and Signal Processing*. In: Ruskai, Mary Beth, G. Beylkin, I. Daubechies, Y. Meyer, R. Coifman, S. Mallat und L. Raphael, Hrsg.: *Wavelets and their Applications*, S. 153–178, Boston. Jones and Bartlett.
- Daubechies, Ingrid (1988). *Orthonormal Bases of Compactly Supported Wavelets*. *Communications on Pure and Applied Mathematics*, 41:909–996.
- Daubechies, Ingrid (1990). *The Wavelet Transform, Time-Frequency Localization and Fourier Analysis*. *IEEE Trans. Inform. Theory*, 36:961–1004.
- Daubechies, Ingrid (1993). *Wavelet Transforms and Orthonormal Wavelet Bases*. In: Daubechies, Ingrid, Hrsg.: *Different Perspectives on Wavelets*, Bd. 47 d. Reihe *Proceedings of Symposia in Applied Mathematics*, S. 1–33, Providence, Rhode Island. American Mathematical Society.
- Douglas, David H. und T. K. Peucker (1973). *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Character*. *The Canadian Cartographer*, 10(2):112–123.
- Eccles, M.J., M. McQueen und D. Rosen (1977). *Analysis of the Digitized Boundaries of Planar Objects*. *Pattern Recognition*, 9(1):31–41.
- Ehrliholzer, Regula (1996). *Methoden für die Bewertung der Qualität von Generalisierungslösungen*. Diplomarbeit, Geographisches Institut der Universität Zürich.
- Foley, James D., A. van Dam, S. K. Feiner und J. F. Hughes (1991). *Computer Graphics; Principles and Practice*. Addison-Wesley Systems Programming Series. Addison-Wesley, 2. Aufl.

- Freeman, H. (1974). *Computer Processing of Line-Drawing Images*. Computing Surveys, 6(1).
- Freeman, H. (1978). *Shape Description Via the Use of Critical Points*. Pattern Recognition, 10:159–166.
- Fritsch, Emmanuel (1994). *Recherche d'outils et de représentations pour la généralisation*. stage de DEA effectué au COGIT-IGN et au Geographisches Institut de l'Université de Zürich.
- Fritsch, Emmanuel und J. P. Lagrange (1995). *Spectral Representations of Linear Features for Generalisation*. In: Frank, Andrew U. und W. Kuhn, Hrsg.: *Spatial Information Theory; A Theoretical Basis for GIS. Proceedings of COSIT '95, Semmering, Austria*, Lecture Notes in Computer Science, S. 157–171. Springer.
- Goodchild, M. F. und D. M. Mark (1987). *The Fractal Nature of Geographic Phenomena*. Annals of the Association of American Geographers, 77(2):265–278.
- Grossman, A. und J. Morlet (1984). *Decomposition of Hardy functions into square integrable wavelets of constant shape*. SIAM Journal of Applied Mathematics, 15:723–736.
- Haar, A. (1910). *Zur Theorie der orthogonalen Funktionssysteme*. Math. Ann., 69:331–371.
- Hanna, J. Ray und J. H. Rowland (1990). *Fourier Series, Transforms, and Boundary Value Problems*. Wiley, New York, 2. Aufl.
- Kernighan, Brian W. und D. M. Ritchie (1988). *The C Programming Language*. Prentice Hall, New Jersey, 2. Aufl.
- Laboratoire COGIT, Direction technique, Service de la recherche (1992). *IGN's research on generalization*. Technischer Bericht Institut Géographique National (IGN).
- Lang, T. (1969). *Rules For the Robot Draughtsmen*. The Geographical Magazine, 42(1):50–51.
- Laurini, Robert und D. Thompson (1992). *Fundamentals of Spatial Information Systems*. The A.P.I.C. Series. Academic Press.
- Mallat, S. (1989). *Multiresolution Approximation and Wavelets*. Trans. Am. Math. Soc., 315:69–88.
- Mather, Paul M. (1987). *Computer Processing of Remotely-Sensed Images*. John Wiley & Sons, Chichester.
- McMaster, Robert B. und K. S. Shea (1992). *Generalization in Digital Cartography*. Association of American Geographers, Washington D.C.
- Meng, L. (1993). *Application of neural network in cartographic pattern recognition*. In: Mesenburg, Peter, Hrsg.: *Proceedings of the 16th International Cartographic Conference*, Bd. 1, S. 192–210, Köln. International Cartographic Association, German Society of Cartography.

- Meyer, Y., S. Jaffard und O. Rioul (1987). *L'analyse par ondelettes*. Pour la science, S. 28–37.
- Mokhtarian, Farzin (1994). *CSSI Quellcode in C*. Der Programmcode ist *nicht public domain*, wurde mir für diese Arbeit jedoch freundlicherweise vom Autor zur Verfügung gestellt.
- Mokhtarian, Farzin und A. K. Mackworth (1992). *A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(8).
- O'Neill, Michael P. und D. M. Mark (1987). *The Psi-s Plot: A Useful Representation for Digital Cartographic Lines*. In: *Proceedings AutoCarto 8*, S. 231–240, Baltimore, Maryland.
- Ousterhout, John K. (1994). *Tcl and the Tk toolkit*. Addison-Wesley Professional Computing Series. Addison-Wesley, Reading, MA.
- Pavlidis, T. (1978). *A Review of Algorithms for Shape Analysis*. Computer Graphics and Image Processing, 7:243–258.
- Plazanet, Corinne (1995). *Measurement, Characterization and Classification for Automated Line Feature Generalization*. In: *Proceedings of ACSM/ASPRS*, Bd. 4 d. Reihe *Auto Carto 12*, S. 59–68, Charlotte, North Carolina.
- Reichenbacher, Tumasch (1995). *Eine Methode für den Wissenserwerb in der kartographischen Generalisierung durch Interaktionsaufzeichnung und induktives Lernen*. Diplomarbeit, Geographisches Institut der Universität Zürich.
- Reumann, K. und A. P. M. Witkam (1974). *Optimizing Curve Segmentation in Computer Graphics*. In: *International Computing Symposium*, S. 467–472, Amsterdam. North-Holland Publishing Company.
- SGK, Schweizerische Gesellschaft für Kartographie, Hrsg. (1980). *Kartographische Generalisierung, Topographische Karten*. Nr. 1 in *Kartographische Schriftenreihe*. 2. Aufl.
- Shea, K. Stuart und R. B. McMaster (1989). *Cartographic Generalization in a Digital Environment: When and How To Generalize*. In: *Proceedings of the Auto-Carto 9*, S. 56–67, Baltimore, Maryland.
- Solymar, Laszlo (1988). *Lectures on Fourier Series*. Oxford University Press.
- Stansfield, J. L. (1980). *Conclusions from the commodity expert project*. AI Memo 601, Mass. Inst. Technol. AI Lab, Cambridge, MA.
- Strichartz, Robert S. (1994). *Construction of orthonormal wavelets*. In: Benedetto, John J. und M. W. Frazier, Hrsg.: *Wavelets; Mathematics and Applications*, Studies in Advanced Mathematics, Kap. 1, S. 23–50. CRC Press, Boca Raton, Florida.
- Thapa, K. (1988). *Automatic Line Generalization Using Zero-Crossings*. Photogrammetric Engineering and Remote Sensing, 54(4):511–517.

- Visvalingam, M. und J. D. Whyatt (1992). *Line Generalisation by Repeated Elimination of the Smallest Area*. Discussion Paper 10, Cartographic Information Systems Research Group, The University of Hull.
- Weibel, Robert (1991). *Amplified Intelligence and Rule-Based Systems*. In: Buttenfield, Barbara P. und R. B. McMaster, Hrsg.: *Map Generalization – Making Rules for Knowledge Representation*, S. 172–186. Longman, London.
- Weibel, Robert, S. Keller und T. Reichenbacher (1995). *Overcoming the Knowledge Acquisition Bottleneck in Map Generalization: The Role of Interactive Systems and Computational Intelligence*. In: Frank, Andrew U. und W. Kuhn, Hrsg.: *Spatial Information Theory; A Theoretical Basis for GIS. Proceedings of COSIT '95, Semmering, Austria*, Lecture Notes in Computer Science, S. 139–156. Springer.
- Wilson, R. (1993). *Wavelets?*. In: *Colloquium on Applications of Wavelet Transforms in Image Processing*, S. 1–6, Savoy Place, London. Professional Group E4 (Image Processing and Vision), The Institution of Electrical Engineers (IEE).
- Winston, Patrick Henry (1984). *Artificial Intelligence*. Addison-Wesley, Reading, 2. Aufl.
- Witkin, Andrew P. (1983). *Scale-Space Filtering*. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Bd. 2, S. 1019–1023, Karlsruhe, Germany.
- WPL (1994). *Wavelet Packet Laboratory (Xwpl) Version 1.3*. Software erhältlich via anonymous ftp vom Server pascal.math.yale.edu.
- Young, Randy K. (1993). *Wavelet Theory and its Applications*. Kluwer international series in engineering and computer science; SECS 189. Kluwer Academic Publishers, Norwell, Massachusetts.
- Zeller, Meinrad (1994). *Flinkes Wellenspiel*. c't, magazin für computertechnik, 11:258–261.

## Anhang A

# Die Softwareplattform *REPTILE*

Zu Experimentierzwecken wurde in dieser Arbeit eine Softwareplattform namens *REPTILE* (**RE**presentation **T**ool and **I**nteractive **L**ine-editing **E**nvironment) entwickelt. Das Diagramm in Abbildung A.1 vermittelt einen Eindruck von dessen Funktionsumfang. *REPTILE* wurde weitgehend in der Skriptsprache *Tcl* (*“tool command language”*), bzw. dessen grafischer Erweiterung *Tk* (*“X Window Toolkit”*) implementiert (Ousterhout 1994). Der gesamte algorithmische Teil wurde jedoch aus Effizienzgründen in der Programmiersprache **C** (Kernighan und Ritchie 1988) implementiert. Folgende **C**-Funktionen wurden entwickelt:

**SampleCmd**: Abtastung einer Linie

**NormalizeCmd**: Normalisierung einer Linie

**ParamCmd**: Parametrisierung einer Linie (*Psi-s*, *t-alpha*, *t/x* oder *t/y*)

**ReconstructCmd**: Rücktransformation der parametrisierten Linie in die *x, y*-Ebene

**InvNormalizeCmd**: Rücktransformation einer normalisierten Linie an ihre ursprüngliche Position in der *x, y*-Ebene

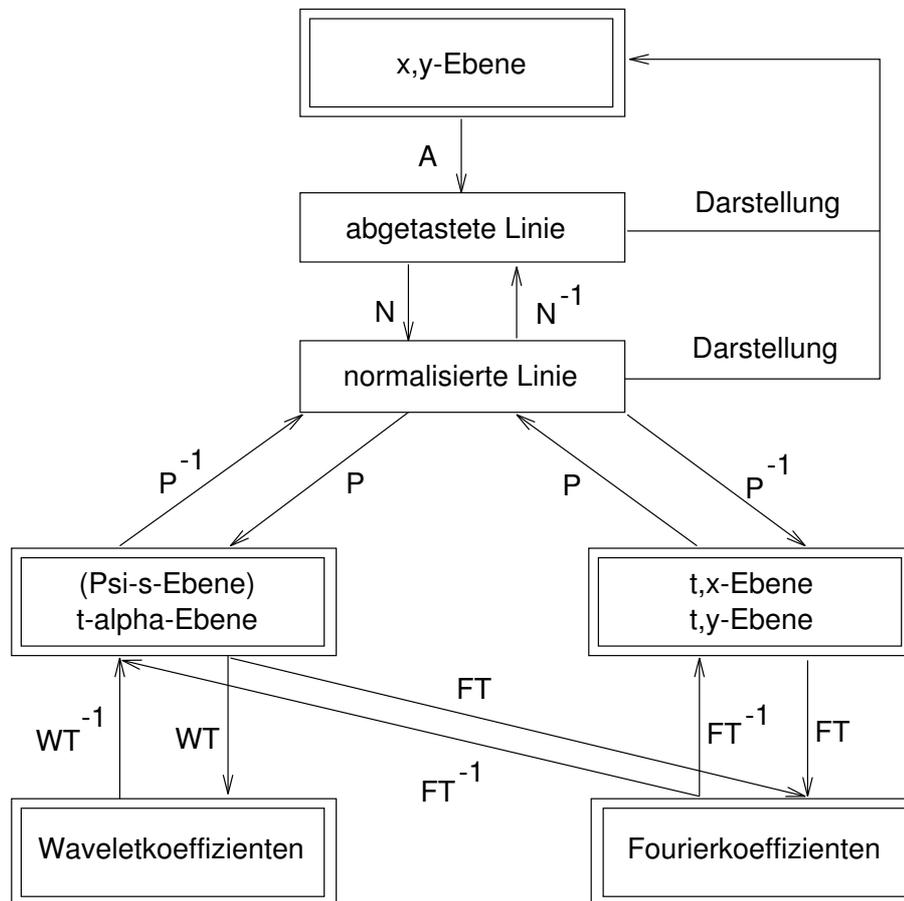
**ToFourierCmd**: Berechnet die Fouriertransformation einer *t-alpha*-, *t/x*-, bzw. *t/y*-parametrisierten Linie

**FromFourierCmd**: Inverse Fouriertransformation

**ToWaveletsCmd**: Berechnet die Wavelettransformation einer *t-alpha*-parametrisierten Linie

**FromWaveletsCmd**: Inverse Wavelettransformation

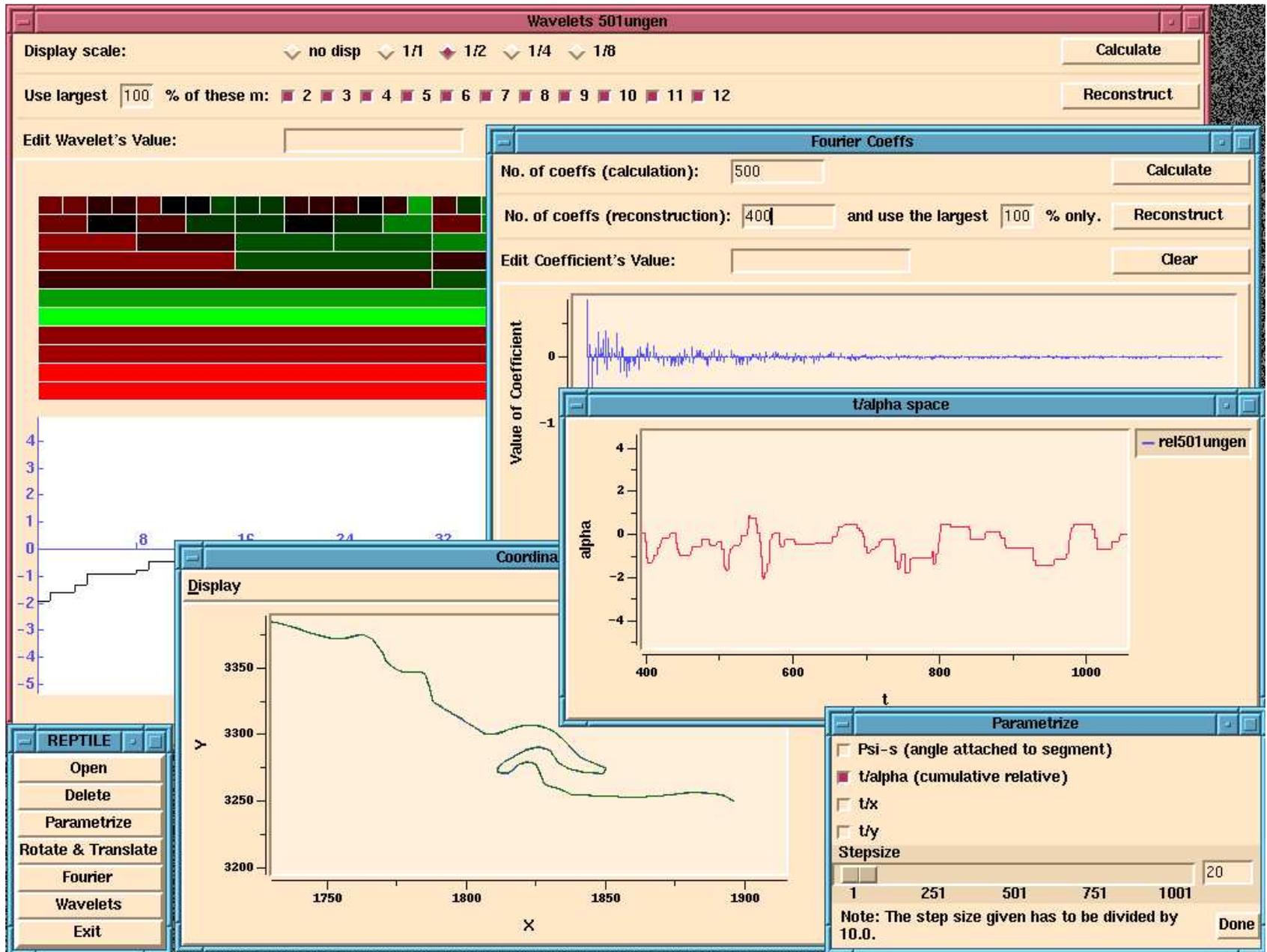
Die Offenheit von *Tcl* – und die Tatsache, dass es sich bei *Tcl* um eine **C**-Bibliothek handelt – erlaubte es auf einfache Weise, diese **C**-Funktionen in die Applikation einzubinden. Bezüglich der explorativen Analyse kommt dem Visualisierungsteil eine grosse Bedeutung zu. Jede Repräsentationsebene wird in einem separaten Fenster dargestellt. Diese Fenster sind jedoch miteinander verknüpft, so dass eine vom Benutzer initiierte Veränderung (z.B. der Abtastschrittweite) zur Neuberechnung und Aufdatierung der jeweiligen Linie in allen Repräsentationsebenen führt. Abbildung A.2 zeigt, wie sich die *REPTILE*-Arbeitsumgebung auf dem Bildschirm präsentiert.



A: Abtastung  
 N: Normalisierung  
 P: Parametrisierung  
 FT: Fouriertransformation  
 WT: Wavelettransformation

Abbildung A.1: Das *REPTILE* Funktionsdiagramm. Ein Kästchen mit Doppelrahmen entspricht einem *REPTILE*-Fenster.

Abbildung A.2: Eine REPTILE-Arbeitsession, wie sie sich auf dem Bildschirm präsentiert.





# Anhang B

## Daten

### B.1 Datenquelle

Die Rohdaten für diese Arbeit wurden freundlicherweise vom *Institut Géographique National (IGN)* in Paris, dem französischen Äquivalent zum Bundesamt für Landestopographie, zur Verfügung gestellt. Es handelt sich dabei um zwei Strassennetze in verschiedenem Massstab, welchen folgende Datenquellen zugrunde liegen:

**Ausgangsmassstab:** digitale Version des Kartenblattes 52 *Grenoble-Valance* der *Série Verte* im Massstab 1:100'000. Dieser Datensatz wurde aber bereits aus der *BDCARTO*, einer am *IGN* aufgesetzten Datenbank im Massstab 1:50'000, abgeleitet (Laboratoire COGIT 1992).

**Zielmassstab:** manuell generalisierte und nachträglich digitalisierte Version des obigen Kartenblattes im Massstab 1:250'000. Unklar ist, auf welcher Basis die Generalisierung beruht (*Série Verte* oder *BDCARTO*).

### B.2 Linienpaare

Aus den beiden Strassennetzen, welche im Datenaustauschformat *DXF* vorlagen, wurden zuerst *ARC/INFO-Coverages* generiert. Die Zuordnung der manuell generalisierten Strassenlinien zu ihren ungeneralisierten Pendanten wurde soweit als möglich automatisiert. Zu diesem Zweck wurde eine *AML* (ein *ARC/INFO-Skript*) geschrieben, welche auf der *A/I*-Funktion *matchcover* beruht. Trotzdem musste viel Zeit in die manuelle Nachbearbeitung gesteckt werden, in erster Linie, weil die zum Teil grossen Verdrängungseffekte eine automatische Zuordnung oft nicht mehr erlaubten.

Nach der Eliminierung von sehr kurzen Linienelementen resultierten schliesslich 628 Linienpaare. Die Koordinaten jeder Linie wurden mit dem *A/I*-Befehl *ungenerate* in eine eigene *ASCII*-Datei ausgegeben. Liniennummer (*nnn*) und Art wurden im Dateinamen gespeichert:

**Ungeneralisiert:** *nnn*ungen

**Generalisiert:** *nnn*gen

Zur Veranschaulichung sind in Abbildung B.1 einige der 628 Linienpaare wiedergegeben.

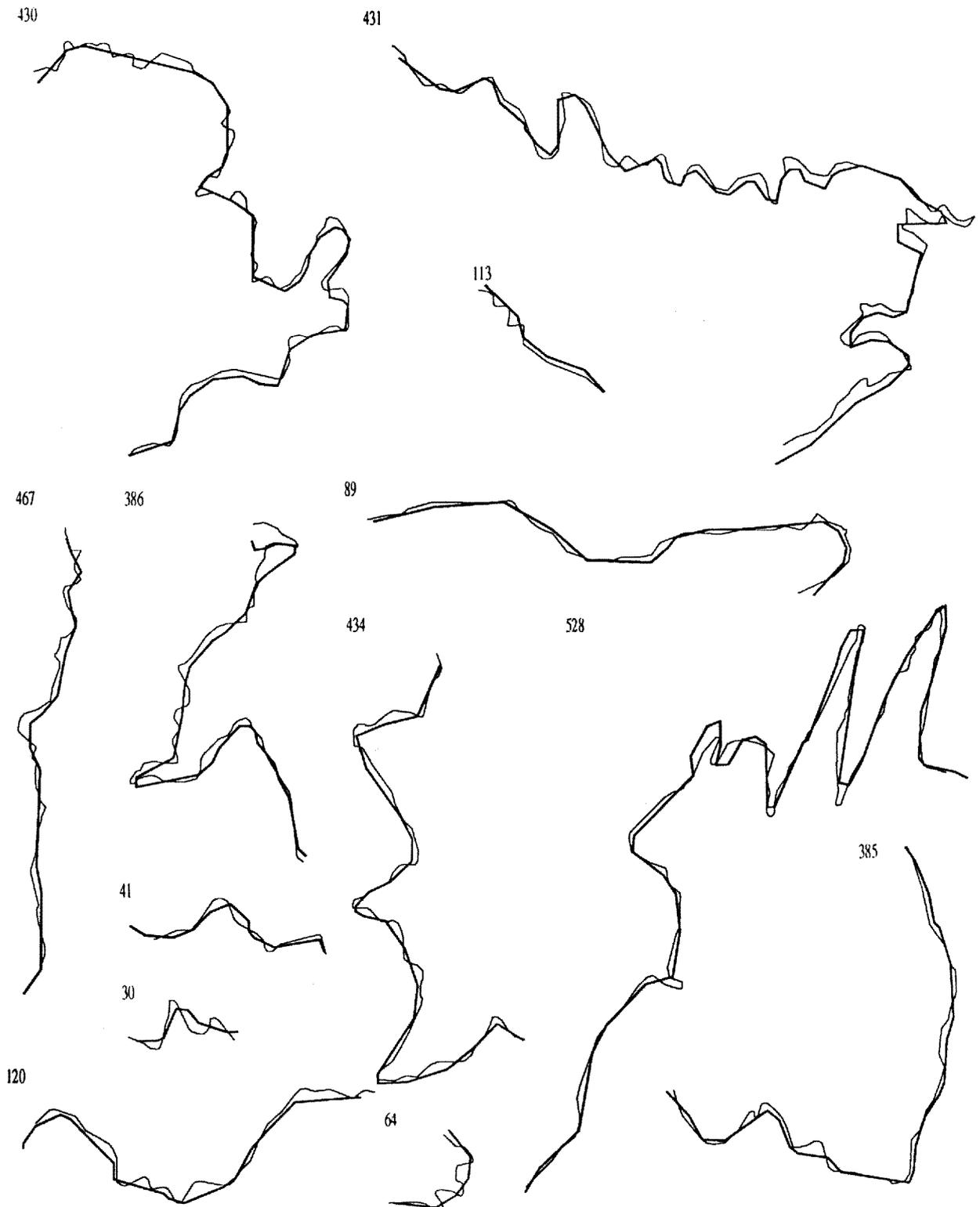


Abbildung B.1: Einige Beispiele aus dem 628 Linienpaare umfassenden Experimentierdatensatz. Ungeneralisierte Linien sind dünn dargestellt, ihre manuell generalisierten Pendanten fett.