



Johannes Gutenberg-Universität Mainz

Fachbereich 08

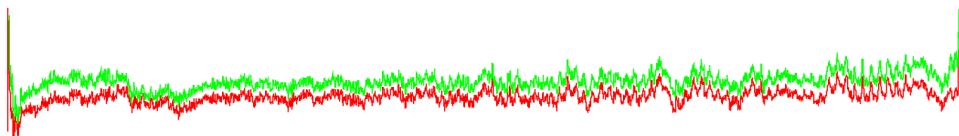
Institut für Informatik

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Mathematiker (Dipl.-Math.)

# Effizientes Alignment von Stahlband-Fingerprints



Mainz, den 15. Oktober 2008

## **Autor**

Mathias Kemeter  
kemeter@gmail.com

Nackstraße 45  
55118 Mainz

## **Betreuer**

Prof. Dr. Herbert Göttler  
goettler@uni-mainz.de

Prof. Dr. Elmar Schömer  
schoemer@uni-mainz.de



## Zusammenfassung

Bei einem Stahlband-Fingerprint handelt es sich um das Dickenprofil eines Stahlbandes. Dieses ist charakteristisch und einmalig, so dass sich das Band anhand seines Dickenschriebes identifizieren lässt. Ähnlich dem Abgleich des menschlichen Fingerabdrucks, sind jedoch mehrere Messungen des Dickenprofils von ein und demselben Band nicht identisch, da eine Messreihe Umwelteinflüssen und Messungenauigkeiten unterliegt.

Um zwei Dickenprofile desselben Stahlbandes zur Deckung zu bringen, ist es notwendig, diese horizontal zu skalieren und zu verschieben. In dieser Arbeit werden Algorithmen vorgestellt und verglichen, die solch ein optimales horizontales Alignment bestimmen. Der Fokus des Vergleichs liegt neben der Qualität des Ergebnisses vor allem auf der Laufzeit der Verfahren. Deshalb wird vom naiven algorithmischen Ansatz, dessen Ergebnis optimal ist, ausgegangen, um diesen bis zur Praxistauglichkeit zu beschleunigen.

Der aussichtsreichste Ansatz zur Beschleunigung arbeitet mit schneller Fouriertransformation um die Güte mehrerer Alignments effizient zu bestimmen. Dieser Algorithmus wird verwendet, um die verschiedenen Gütekriterien, auf die sich ein Alignment optimieren lässt, an zwei Anwendungsfällen zu evaluieren. Mit Hilfe eines Alignments soll die Länge des Verschnitts bestimmt werden, der zwischen zwei Messungen vom Stahlband abgetrennt wurde. Außerdem werden die Alignmentverfahren verwendet, um zu einer Dickenmessung, die im Kaltwalzwerk stattfindet, die Dickenmessung von dem gleichen Band aus dem Warmwalzwerk zu finden und das Stahlband somit zu identifizieren.

## Danksagung

Ich möchte mich an dieser Stelle bei allen Personen bedanken, die mich bei der Entstehung dieser Diplomarbeit unterstützt haben. Hier ist vor allem das Fingerprint-Projektteam des Instituts für Informatik an der Universität Mainz zu nennen, bei dem ich mich besonders für die dienstäglichen Treffen zum Ideenaustausch bedanken möchte:

*Prof. Dr. Herbert Göttler*

*Dipl.-Math. Thomas Gottron*

*Dipl.-Math. Constanze Lipowsky*

*Prof. Dr. Elmar Schömer*

Unentbehrliche Unterstützung habe ich auch von der iba AG bekommen, die mir für alle anwendungsbezogenen Fragestellungen zur Seite stand. Ich danke stellvertretend folgenden Personen:

*Dipl.-Ing. Horst Anhaus*

*Dipl.-Ing. Thomas George*

*Dipl.-Math. Detlef Maaß*

*Dipl.-Ing. Günter Sörgel*

Außerdem danke ich der voestalpine Stahl GmbH, die mir die Gelegenheit gegeben hat, die Verfahren in dieser Arbeit an realen Daten zu erproben. Hier möchte stellvertretend danken:

*Dr. Helmut Bachl*

Für das gnadenlose Aufspüren meiner Ausdrucks-, Komma-, Tipp- und Indexfehler, die ich vorher sorgsam versteckt habe, bedanke ich mich bei:

*Kerstin Fiedler*

*Sebastian Hoffmann*

*Sarah Kemeter*

Bei Kerstin möchte ich mich ganz besonders bedanken, da sie diese Arbeit nicht nur *mehrmals* probeglesen hat, sondern mir während meines gesamten Studiums und insbesondere in den letzten Monaten geholfen hat, das Gleichgewicht zu halten, auch wenn sie ihres dafür manchmal riskiert hat.

— **Danke!** —

## Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne unerlaubte Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Mainz, den 15. Oktober 2008

---

Mathias Kemeter

Diese Diplomarbeit ist in Zusammenarbeit mit den Firmen

iba AG		voestalpine Stahl GmbH
Königswarterstraße 44	&	voestalpine-Straße 3
D-90762 Fürth		A-4031 Linz

[www.iba-ag.com](http://www.iba-ag.com)

[www.voestalpine.com](http://www.voestalpine.com)

entstanden.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>11</b>
1.1	Der Walzprozess . . . . .	12
1.2	Problemstellung . . . . .	14
1.3	Aktueller Forschungsstand . . . . .	16
<b>2</b>	<b>Grundlagen</b>	<b>19</b>
2.1	Definition der Zeitreihe . . . . .	19
2.2	Interpolation von Zeitreihen . . . . .	20
2.3	Alignment von Zeitreihen . . . . .	23
2.4	Gütekriterien für Alignments . . . . .	24
2.4.1	Mittlerer quadratischer Fehler . . . . .	24
2.4.2	Korrelationskoeffizient . . . . .	27
2.4.3	Gleichläufigkeitskoeffizient . . . . .	28
2.5	Piecewise Aggregate Approximation . . . . .	30
2.6	Diskrete Fouriertransformation . . . . .	31
2.6.1	Einheitswurzeln . . . . .	32
<b>3</b>	<b>Datenaufbereitung</b>	<b>37</b>
<b>4</b>	<b>Alignmentverfahren</b>	<b>41</b>
4.1	Naives Alignment . . . . .	42
4.1.1	Offsetbestimmung . . . . .	43

4.1.2	Skalierungssuche . . . . .	44
4.2	Beschleunigung des naiven Alignments . . . . .	45
4.2.1	Fehlerberechnung mit oberer Grenze . . . . .	45
4.2.2	Fehlerberechnung mittels FFT . . . . .	47
4.2.3	Kombination mit Skalierungssuche . . . . .	57
4.2.4	Skalierungssuche durch Gitterheuristik . . . . .	57
4.2.5	Komprimierung der Daten . . . . .	60
4.2.6	Fazit . . . . .	60
4.3	Shotgunverfahren mit linearer Regression . . . . .	62
4.4	Vergleich der Ansätze . . . . .	63
<b>5</b>	<b>Berechnung von Kopf- und Fußschrott</b>	<b>67</b>
5.1	Datenbasis . . . . .	68
5.2	Schrottberechnung . . . . .	68
5.3	Ergebnisse . . . . .	69
<b>6</b>	<b>Zuordnung von Dickenschriften</b>	<b>71</b>
6.1	Naiver Algorithmus zur Zuordnung . . . . .	71
6.2	Beschleunigung durch Komprimierung . . . . .	72
6.3	Das Siebverfahren . . . . .	73
6.4	Einschränkung des Suchbereichs . . . . .	76
6.5	Ergebnisse . . . . .	76
6.5.1	Datenbasis . . . . .	77
6.5.2	Parameter . . . . .	78
6.5.3	Vergleich der Gütekriterien . . . . .	79
6.5.4	Vergleich der Verfahren . . . . .	81
<b>7</b>	<b>Fazit und Ausblick</b>	<b>85</b>



<b>A</b>	<b>Illustration des Produktionsprozesses</b>	<b>89</b>
<b>B</b>	<b>Daten</b>	<b>93</b>
B.1	Alignmentergebnisse für den Kopfschrott . . . . .	93
B.2	Ergebnisse des Siebverfahrens . . . . .	94
<b>C</b>	<b>Integration in den ibaAnalyzer</b>	<b>99</b>
	<b>Abbildungsverzeichnis</b>	<b>101</b>
	<b>Literaturverzeichnis</b>	<b>103</b>



# Kapitel 1

## Einleitung

In der modernen Industrie ist die Erhebung von Messdaten zur Prozess- und Qualitätskontrolle unerlässlich. Aufgrund immer genauer werdender Messsysteme sowie leistungsfähiger Computer, lassen sich aus den gewonnenen Messdaten zunehmend mehr Informationen über das jeweilige Produkt und den Produktionsprozess generieren. Diese Informationen tragen wesentlich zur Verbesserung der Qualität eines Produktes bei.

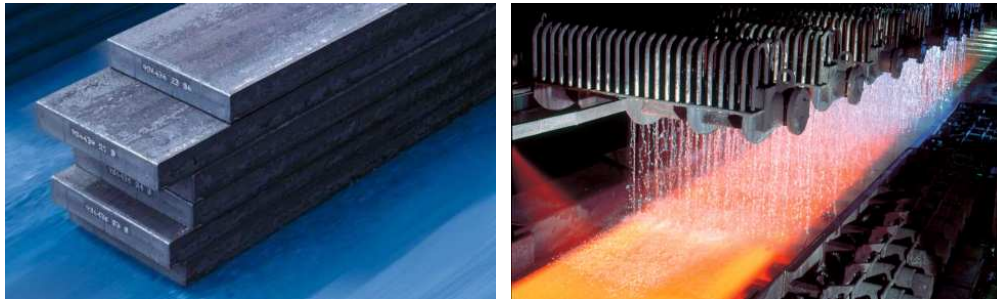
Diese Arbeit beschäftigt sich mit einem konkreten Beispiel aus der Stahlindustrie, nämlich der Produktion von Stahlbändern, die zu sogenannten Coils aufgewickelt werden (vgl. Abbildung 1.1). Alle vorgestellten Verfahren lassen sich jedoch nicht nur auf Messreihen von Stahlbändern, sondern auch auf allgemeine eindimensionale Messreihen anwenden.



*Abbildung 1.1: Ein Stahlcoil während des Aufhaspelns [vSGa]*

## 1.1 Der Walzprozess

Um ein Stahlband zu erzeugen, wird Erz im Hochofen zu Roheisen verschmolzen. Nach Beimengung von einigen Legierungszusätzen wird daraus Stahl, der in der Stranggießanlage zu einem langen Strang vergossen und in sogenannte Brammen (vgl. Abbildung 1.2) zerteilt wird. Diese Brammen sind bis zu  $250\text{ mm}$  dick und nur einige Meter lang. Um das für uns relevante Endprodukt, ein dünnes Stahlband, zu erhalten, muss die Bramme warm- und gegebenenfalls kaltgewalzt werden.



(a) Gegossene Brammen

(b) Warmgewalztes Fertigband

**Abbildung 1.2:** Der Stahl vor und nach dem Warmwalzen [vSGa]

Beim Warmwalzen wird die Bramme bei einer Temperatur von anfangs etwa  $1150\text{ }^{\circ}\text{C}$ , die gegen Ende des Walzvorgangs auf  $800\text{ }^{\circ}\text{C}$  abkühlt, auf eine Dicke von bis zu  $1,5\text{ mm}$  gewalzt, so dass das Fertigband nach dem Walzen eine Länge von über tausend Metern hat. Die Geschwindigkeit des Bandes während des Walzens hängt vor allem von der Enddicke ab. Ein typisches Band hat im letzten Abschnitt des Warmwalzwerks, der Warmfertigstraße, eine Geschwindigkeit von anfangs etwa  $10\text{ m/s}$ . Um die Temperatur des Stahlbandes nach dem Einfädeln in die Walzanlage möglichst konstant zu halten, wird das Band während des Walzvorgangs stetig beschleunigt, so dass gegen Ende eine Spitzengeschwindigkeit von bis zu  $20\text{ m/s}$  erreicht wird. Eine schematische Darstellung des Warmwalzwerks findet sich in Anhang A.

Ein Teil der warmgewalzten Coils wird nun direkt an den Kunden geliefert, während der andere Teil weiterbearbeitet wird. Zwischen dem Warmwalzen und der Weiterbearbeitung können einige Tage, aber auch mehrere Wochen vergehen, in denen das durchschnittlich 25 Tonnen schwere Coil zwischengelagert wird.

Vor dem nächsten Arbeitsschritt kann es zudem erforderlich sein, fehlerhafte Stücke oder Probetafeln aus dem Band zu schneiden und dieses somit zu teilen

(Bandteilung). Zur Inspektion der potentiell fehlerhaften Stellen, wird das Band abgewickelt und wieder aufgewickelt (Umwicklung), wodurch Anfang und Ende des Coils verdreht werden.

Da die voestalpine Stahl GmbH eine jährliche Produktion von etwa  $4\frac{1}{2}$  Millionen Tonnen Stahlblechen und somit etwa 180.000 Stahlcoils im Warmwalzwerk hat, müssen verschiedenste Maßnahmen ergriffen werden, um nach einer längeren Lagerzeit das richtige Coil zur Weiterbearbeitung zu finden. Laut [Bac08] sind dies bei voestalpine unter anderem

- automatische Nummerierung der Coils zur Identifikation,
- automatische Kranpositionsüberwachung in den Hallen,
- Speicherung der GPS-Koordinaten der Lagerplätze im Freilager und
- automatische Lagerverwaltung.



*Abbildung 1.3: Nummerierte Stahlcoils im Freilager [Bac08]*

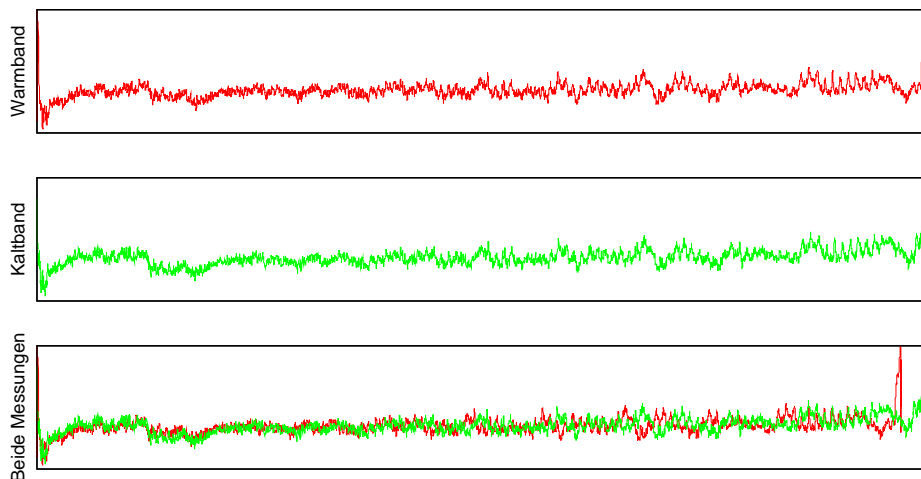
Nach der Zwischenlagerung beginnt typischerweise der Prozess des Kaltwalzens bzw. Beizens. Bei der voestalpine gibt es auch ein Walzwerk, in denen diese beiden Schritte in einer gekoppelten Beiz-Tandem-Anlage (vgl. Anhang A) vereint werden. Beim Kaltwalzen wird das Band bei Raumtemperatur auf eine Dicke von bis zu  $0,3\text{ mm}$  gewalzt. Die Sollstärke wird hier bis auf  $\frac{1}{100}\text{ mm}$  genau eingehalten. Die Durchschnittsgeschwindigkeit des Stahlbandes in den Kaltwalzanlagen liegt bei  $10\text{ m/s}$ . Je nach Anlage und Stahlart ist das Stahlband vor dem Walzprozess durch ein Beizebad gelaufen, um die Oberfläche zu reinigen.

Abschließend sei noch erwähnt, dass nicht alle Brammen das Warmwalzwerk passieren. Es gibt auch noch zahlreiche andere Werdegänge, wie beispielsweise die Grobblechstraße oder das Rohrwalzwerk. In Anhang A finden wir eine

Übersicht über die möglichen Werdegänge des Stahls in einem Hüttenwerk. Einer der größten Stränge darin ist das Warm- und Kaltwalzen. Diese Arbeit kann selbstverständlich nur oberflächlich die für uns nötigen Einblicke in den Produktionsprozess gewähren. Für tiefergehende Informationen zur Stahlerzeugung und -bearbeitung, lohnt ein Blick in [vSGb] oder [DFE<sup>+</sup>07].

## 1.2 Problemstellung

Wenn wir am Beispiel von Abbildung 1.4 eine Dickenmessung vom Ausgang der Warmbandstraße und eine Dickenmessung am Eingang des Kaltwalzwerks betrachten, so sehen wir in den beiden oberen Streifen, dass diese sich äußerst ähnlich sind, fast identisch aussehen. Das ist nicht überraschend, da es sich um zwei Messungen von ein und demselben Stahlband handelt.



**Abbildung 1.4:** Dickenprofile am Ausgang des Warmwalzwerks (oben) und Eingang des Kaltwalzwerks (Mitte) auf verschiedenen Ortsgittern und auf dem selben Ortsgitter (unten).

Visualisiert man die Messungen jedoch nicht getrennt voneinander, sondern in einem Graphen und auf demselben Ortsgitter, so erkennt man, dass die Profile keineswegs identisch sind. Da es sich um zwei verschiedene Messungen unter unterschiedlichen Umweltbedingungen, an verschiedenen Orten und mit unterschiedlichen Messgeräten handelt, gibt es viele mehr oder weniger große Abweichungen.

Am Anfang (Kopf) und Ende (Fuß) des Stahlbandes wird produktionsbedingt die Soll Dicke des Bandes beim Warmwalzen nur unzureichend genau erreicht. Des Weiteren enden Kopf und Fuß des Bandes keineswegs in einem geraden

Schnitt, sondern sind durch das Walzen deformiert, wie auch in Abbildung 1.5 zu sehen. Das Stahlband verhält sich ähnlich einem Teig, der ausgerollt wird,



*Abbildung 1.5: Ein warmgewalztes Coil, dessen Kopf und Fuß noch nicht abgeschrottet wurde. Das Bandende (rechts) ist stärker deformiert als der Bandanfang (links). [vSGa]*

denn dieser wird durch das Rollen nie eine rechteckige Form erhalten. Um ein „sauberes“ Band zu erhalten, wird also ein Teil des Stahlbandes an Kopf und Fuß abgetrennt, wodurch der Kaltbandmessung einige Meter im Vergleich zur Warmbandmessung fehlen.

Die fehlenden Messpunkte des Kaltbandes lassen sich mit einer Verschiebung der Messreihen gegeneinander ausgleichen. Im unteren Streifen von Abbildung 1.4 sehen wir jedoch, dass die Unterschiede der beiden Messungen eklatanter sind, als man zunächst vermuten könnte. Da die Dickenprofile hier auf dasselbe Ortsgitter aufgetragen sind, erkennt man, dass diese in horizontaler Richtung gegeneinander skaliert sind. Wir werden in Kapitel 3 erfahren, wodurch solche Abweichungen zustande kommen.

Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, das eine Verschiebung und Skalierung der Messreihen bestimmt, so dass diese möglichst deckungsgleich sind. Da dieses Verfahren in ein vorhandenes Softwareprodukt für die Analyse von Messreihen integriert werden soll (vgl. Anhang C), liegt der Fokus neben der Verlässlichkeit vor allem auf der Effizienz. In Kapitel 4 werden wir Verfahren kennenlernen, die diesen Anforderungen gerecht werden.

Eine mögliche Anwendung, deren Machbarkeit wir in Kapitel 5 untersuchen, ist die Berechnung der Länge von Kopf- und Fußschrott. Momentan wird die Größenordnung des Verschnitts entweder per Sichtschätzung der Mitarbeiter bestimmt oder es wird eine Hochrechnung aufgrund des Gewichts des Coils

vor und nach der Abtrennung des Schrottes gemacht. Die Genauigkeit dieser Schätzungen ist fehlerbehaftet und im Nachhinein nicht mehr auf Plausibilität überprüfbar. Eine Berechnung der verschrotteten Stahlmenge aufgrund der Dickenmesswerte ist eine günstige Alternative, da keine zusätzlichen Messeinrichtungen installiert werden müssen.

Trotz des hohen Aufwands, der betrieben wird, um eine sichere Materialverfolgung zwischen den verschiedenen Produktionsschritten zu gewährleisten (vgl. Abschnitt 1.1), kommt es gelegentlich vor, dass die Herkunft eines warmgewalzten Stahlbandes unbekannt ist. Solch ein Coil muss verschrottet werden, da die Herkunft und somit die Eigenschaften des Materials im Nachhinein nicht mehr bestimmt werden können.

Wir werden in Kapitel 6 Verfahren kennenlernen, die mit Hilfe der Algorithmen aus Kapitel 4 in der Lage sind, ein Dickenprofil dieses Coils, das zu einem früheren Zeitpunkt aufgezeichnet wurde, wiederzufinden. Das Dickenprofil soll das Stahlband also identifizieren, so wie ein Fingerabdruck einen Menschen identifiziert. Die Idee, das Dickenprofil eines Coils als sogenannten Stahlband-Fingerprint zu verwenden, wurde von der iba AG in [Anh07] patentiert.

### 1.3 Aktueller Forschungsstand

Im Vorfeld dieser Arbeit gab es bereits eine Kooperation zwischen der iba AG und der Universität Lübeck, aus der die Diplomarbeit „Genealogie von Stahlband-Fingerprints“ [Gno06] hervorgegangen ist.

Der Autor, Harm Gnoyke, beschäftigte sich mit der Frage, wieviel Meter Dickenprofil von einem Stahlband gemessen werden müssen, um eine andere Messung desselben Stahlbandes in einer großen Menge von Dickenschrieben wiederzufinden. Um diese Frage zu beantworten, wurden verschiedene Abstandsmaße vorgestellt, mit denen sich Dickenschriebe vergleichen lassen. Auf Basis dieser Abstandsmaße wurden einige Ansätze zur Nearest-Neighbour-Suche besprochen und die Treffgenauigkeit für möglichst kurze Suchmuster evaluiert. Um den Suchprozess zu beschleunigen, verwendete er die auch in dieser Arbeit aufgegriffene Piecewise Aggregate Approximation (siehe Abschnitt 2.5).

Harm Gnoyke kam zu dem Schluss, dass „ein zufällig gewählter Ausschnitt von ca. 12 m eines Bandes genügt, um es mit einer Sicherheit von ungefähr 90% zu identifizieren“ (Zitat aus [Gno06]), wenn man den euklidischen Abstand zum Vergleich der Messungen verwendet.

In dieser Arbeit wurde das Problem der gegeneinander skalierten Messreihen



zwar kurz erwähnt, jedoch wurden keine Verfahren vorgestellt, die mit diesem Problem effizient umgehen können, da die Bestimmung eines Skalierungsfaktors dort nicht von Belang war. Einerseits hat Herr Gnoyke möglichst kurze Abschnitte eines Bandes gesucht, so dass die verschiedene horizontale Skalierung der Messungen nicht ins Gewicht fiel, und andererseits arbeitete er mit einzelnen, künstlich verrauschten Messungen. In diesem künstlichen Rauschen wurden weder horizontale Skalierungsunterschiede noch produktionsbedingte Messungenauigkeiten berücksichtigt.

Das Alignment von Datenreihen ist auch jenseits der Stahlindustrie ein Thema mit großer praktischer Bedeutung. Das Alignmentproblem ordnet sich unter dem Oberbegriff des Data-Mining ein, das sich mit der Informationsgenerierung aus Daten jeglicher Art beschäftigt und somit auch mit der Mustererkennung in Daten.

Alignmentalgorithmen werden beispielsweise zum Vergleich von DNA-Sequenzen benötigt. Die Adaptierung der dort verwendeten Verfahren, insbesondere des Shotgunverfahrens (vgl. [VAM<sup>+</sup>01]) und Dynamic Time Warping (vgl. [BC94],[RK04]), auf unsere Problemstellung wird in [GGK<sup>+</sup>08] näher untersucht und kommt auch in den Abschnitten 4.3 und 4.4 dieser Arbeit kurz zur Sprache.

Neben vielen weiteren interessanten Anwendungen zur Mustererkennung, unter anderem in der Bild- und Tonverarbeitung, hat die Verwendung von Alignmentverfahren in der Jahrringforschung große Bedeutung für diese Arbeit gehabt. Auch wenn die Problemstellungen von denen in dieser Arbeit etwas abweicht, konnte ein Ansatz zum Vergleich von Datenreihen für unsere Anwendung übernommen werden. Über die spezielle Anwendung in der Jahrringforschung werden wir in Abschnitt 2.4.3 mehr erfahren.



# Kapitel 2

## Grundlagen

Dieses Kapitel soll einige einfache mathematische Grundlagen vermitteln, die wir zum Verständnis dieser Arbeit benötigen. Hierzu lernen wir erst einmal grundlegende Definitionen zur abstrakten Darstellung unseres Problems kennen. Diese Definitionen verwenden wir anschließend, um einen Überblick über verschiedene Kriterien zu geben, unter denen ein Alignment „optimal“ sein kann.

### 2.1 Definition der Zeitreihe

Die Dickenschriebe von Stahlbändern bestehen aus diskreten Dickenmessungen, die zu bestimmten Zeitpunkten aufgenommen wurden. Somit können wir diese Dickenschriebe als Zeitreihen auffassen.

**Definition 2.1 (2d-Zeitreihe)**

*Eine 2d-Zeitreihe ist ein Tupel  $(t, d) \in \mathbb{R}^n \times \mathbb{R}^n$ , wobei  $d$  eine Folge von Messwerten ist und  $t$  eine Folge von Zeit- bzw. Ortswerten. Für  $i = 0, \dots, n - 1$  gilt:  $d_i$  wurde zur Zeit (bzw. am Ort)  $t_i$  aufgezeichnet.*

Die in Walzwerken verwendeten Messgeräte sind so eingerichtet, dass Messungen zyklisch, also in festen Zeitintervallen, stattfinden (beispielsweise im 10ms-Takt). Da uns später nur die zeitliche oder örtliche Distanz zwischen den Messwerten interessieren wird, ist die Zeit- bzw. Ortskomponente der 2d-Zeitreihe überflüssig. Obwohl wir diese Komponente fortan weitestgehend ignorieren, sollten wir nicht vergessen, dass Messreihen naturgemäß zweidimensionale Zeitreihen sind.

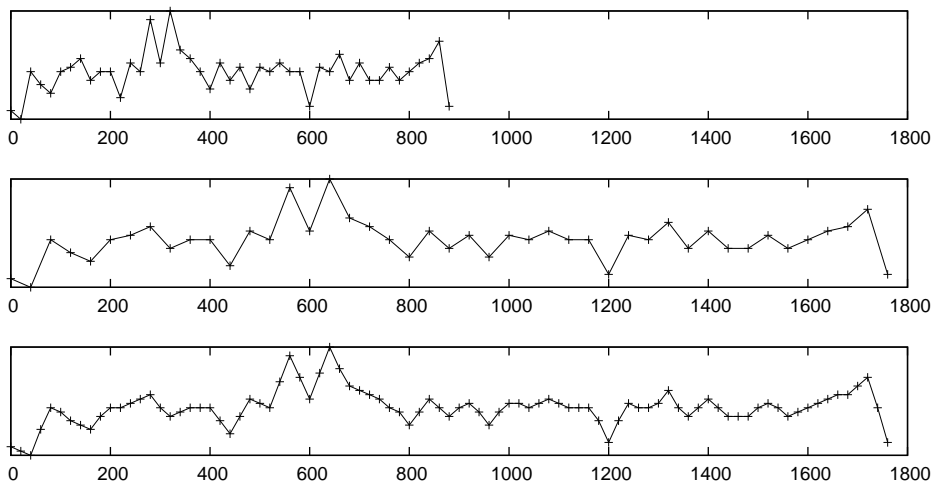
**Definition 2.2 (1d-Zeitreihe)**

Eine 1d-Zeitreihe ist ein Vektor  $d \in \mathbb{R}^n$  dessen Komponenten  $d_i \in \mathbb{R} \quad i = 0, \dots, n - 1$  äquidistant verteilte Messwerte sind.

Da wir uns im Folgenden, wie bereits erwähnt, fast ausschließlich mit 1d-Zeitreihen beschäftigen, verwenden wir vereinfachend den Begriff *Zeitreihe* und meinen stets die 1d-Zeitreihe.

**2.2 Interpolation von Zeitreihen**

Um zwei Dickenprofile von ein und demselben Stahlband zur Deckung zu bringen, ist es aufgrund von Messungenauigkeiten notwendig, diese in horizontale Richtung zu skalieren. In Abbildung 2.1 sehen wir beispielhaft einen der Übersichtlichkeit halber vergrößerten Dickenschrieb eines Stahlbandes, der im mittleren Teil der Abbildung mit dem Faktor 2 in horizontale Richtung skaliert wurde. Während der Abstand zwischen zwei aufeinanderfolgenden Messpunk-



**Abbildung 2.1:** Originaldickenschrieb (oben) mit Faktor 2 in  $x$ -Richtung skaliert (Mitte) und auf die Ausgangsaufösung transformiert (unten)

ten des Originalschriebes 20 m beträgt, ist der Abstand der Messpunkte bei dem skalierten Pendant 40 m.

Für den Vergleich zweier Zeitreihen mit einem Gütekriterium aus Abschnitt 2.4 müssen sich jedoch beide Zeitreihen auf das gleiche Ortsgitter beziehen. Um das Ortsgitter der Originaldaten beizubehalten, muss am Beispiel der skalierten Messreihe in Abbildung 2.1 zwischen je zwei Messpunkten ein neuer hinzugefügt werden, so dass das Ortsgitter auch nach der Skalierung wieder eine Auflösung

von 20  $m$  pro Messpunkt hat (siehe Abbildung 2.1 unten).

Da die neuen Messpunkte unbekannt sind, müssen wir die Messreihe durch eine durchgängige Kurve interpolieren, um sinnvolle Abschätzungen für diese Punkte zu erhalten. Am einfachsten ist es, eine Messreihe durch eine Treppenfunktion zu interpolieren (siehe Abbildung 2.2). Da wir bei unseren Messdaten stets von einem äquidistanten Ortsgitter ausgehen, genügt es folgende Definitionen für  $1d$ -Zeitreihen zu betrachten.

**Definition 2.3 (Interpolation durch eine Treppenfunktion)**

Eine Zeitreihe  $d \in \mathbb{R}^n$  lässt sich durch die Treppenfunktion

$$f(x) := d_{\lceil x-0,5 \rceil}$$

mit  $x \in [0, n - 1]$  interpolieren.

Der Wert  $x$  wird auf den nächsten ganzzahligen Index gerundet. Somit bedeutet diese Definition nichts anderes, als dass wir einen neuen Punkt durch den nächstgelegenen vorhandenen Messpunkt approximieren.

Es widerspricht unserer Intuition, die diskreten Daten aus Abbildung 2.2 als Treppenfunktion zu interpolieren. Eine stückweise lineare Interpolation der einzelnen Messpunkte, wie sie bereits in Abbildung 2.1 zur besseren Visualisierung verwendet wurde, erscheint realistischer für das kontinuierliche Dickenprofil eines Stahlbandes.

**Definition 2.4 (Lineare Interpolation)**

Eine Zeitreihe  $d \in \mathbb{R}^n$  lässt sich durch die stückweise lineare Funktion

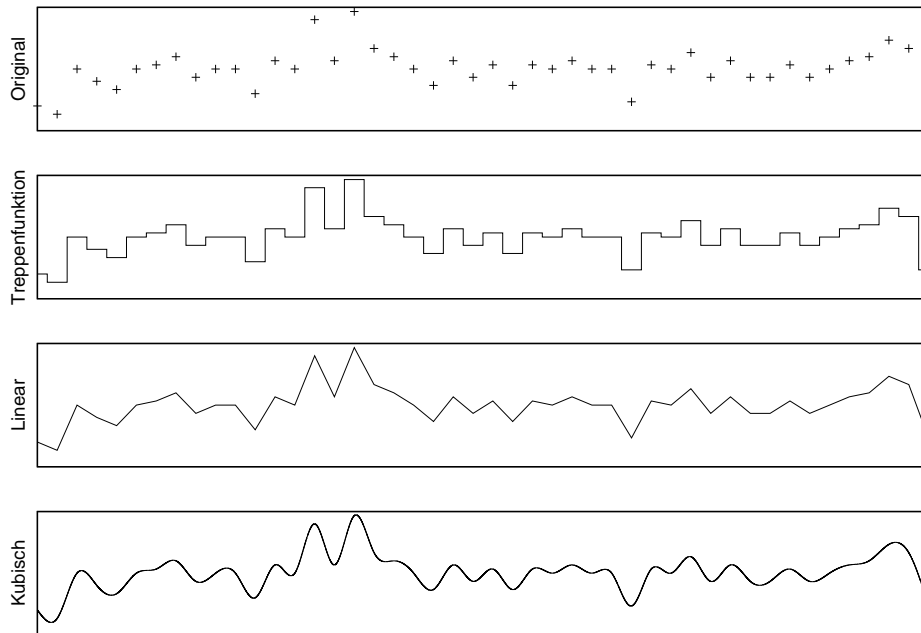
$$f(x) := (\lceil x \rceil - x) \cdot d_{\lceil x \rceil} + (x - \lfloor x \rfloor) \cdot d_{\lfloor x \rfloor}$$

mit  $x \in [0, n - 1]$  interpolieren.

Wenn wir eine diskrete Zeitreihe durch eine kontinuierliche Kurve interpolieren, so können wir diese in  $x$ -Richtung skalieren. Durch eine anschließende Diskretisierung der skalierten Interpolationsfunktion, können wir diese wieder als Zeitreihe auf ein beliebiges Ortsgitter transformieren.

**Definition 2.5 (Skalierung einer Zeitreihe in  $x$ -Richtung)**

Sei  $d \in \mathbb{R}^n$  eine Zeitreihe,  $f_d$  eine beliebige Interpolationsfunktion von  $d$  und  $s \in \mathbb{R}$ . Dann ist die mit dem Faktor  $s$  skalierte Zeitreihe  $d' \in \mathbb{R}^{\lceil sn \rceil}$  definiert



**Abbildung 2.2:** Vergleich verschiedener Interpolationsfunktionen

durch

$$d'_i := f_d \left( \frac{i}{s} \right)$$

für  $i = 0, \dots, \lceil sn \rceil - 2$  und

$$d'_{\lceil sn \rceil - 1} := f_d(n)$$

für den letzten Datenpunkt.

Bei der stückweise linearen Interpolationsfunktionen handelt es sich um eine Spline-Interpolation vom Grad 1 (vgl. auch [HB06]). Die Idee, einen höheren Grad für eine möglicherweise realistischere Interpolation der Messdaten zu verwenden, ist naheliegend. In Abbildung 2.2 sehen wir beispielsweise einen kubischen Spline, der deutlich glatter ist als die Kurve, die durch lineare Interpolation entstanden ist. Es hat sich jedoch herausgestellt, dass aufgrund des hohen Messtakts und den relativ kleinen Skalierungsabweichungen kein Unterschied in den späteren Alignmentergebnissen bei Verwendung der vorgestellten Interpolationsmethoden zu vermerken ist. Somit werden auch höhergradige Spline-Interpolationen keine Veränderung des Ergebnisses mehr bewirken, so dass der Aufwand ihrer Implementierung nicht gerechtfertigt ist.

## 2.3 Alignment von Zeitreihen

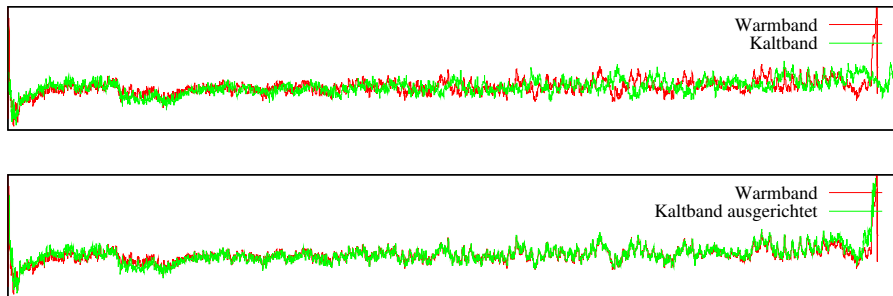
Da das Stahlband im Produktionsprozess zwischen Warm- und Kaltbandstraße an Anfang und Ende beschnitten wird, reicht eine Skalierung der Messungen gegeneinander nicht aus, um sie, wie in Abbildung 2.3 zu sehen, zur Deckung zu bringen. Die Messungen müssen zusätzlich noch gegeneinander verschoben werden. Beide Faktoren zusammen, Skalierung und Verschiebung, werden als Alignment bezüglich der  $x$ -Achse oder auch horizontales Alignment bezeichnet.

### Definition 2.6 (Horizontales Alignment)

Ein horizontales Alignment zweier Zeitreihen ist die Ausrichtung dieser Zeitreihen zueinander. Unter Ausrichtung versteht man eine Verschiebung (Offset) und Skalierung entlang der  $x$ -Achse einer Zeitreihe relativ zu der anderen.

Man kann ein Alignment also als Quadrupel  $A \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{Z} \times \mathbb{R}$  auffassen, wobei die ersten beiden Komponenten den Zeitreihen entsprechen und die dritte und vierte Komponente Offset und Skalierung angeben.

In Abbildung 2.3 sehen wir ein Alignment, das mit einem Skalierungsfaktor von 0,968622 und einer Verschiebung der Kaltbandmessung um 20 Messpunkte nach „rechts“ erreicht wurde. Wie wir an diesem Beispiel exemplarisch sehen,



**Abbildung 2.3:** Warm- und Kaltbandmessung im Original (oben) und nach optimalen Alignment (unten)

ist die Skalierung in  $y$ -Richtung weitgehend identisch. Eine Verschiebung der beiden Messreihen in  $y$ -Richtung existiert nur dann, wenn es sich bei der einen Datenreihe um eine Messung der absoluten Dicke und bei der anderen um eine Messung der Dickenabweichung handelt. Um die beiden Messkurven so übereinander zu legen, wie in Abbildung 2.3 unten zu sehen ist, müssen wir entweder die Sollstärke des Stahlbandes, die selbstverständlich bekannt ist, auf die Dickenabweichung addieren oder die beiden Messreihen um ihren Mittelwert bereinigen (siehe Abschnitt 2.4.1).

Wir müssen in dieser Arbeit also nur das Alignment bezüglich der  $x$ -Achse betrachten und nur für dessen Optimierung einen algorithmischen Aufwand leisten. Daher nennen wir das horizontale Alignment an vielen Stellen vereinfachend nur *Alignment*.

## 2.4 Gütekriterien für Alignments

Um ein möglichst gutes Alignment zu erzielen, bzw. dessen Güte erst einmal zu bewerten, benötigen wir geeignete Abstands- bzw. Ähnlichkeitsmaße zwischen Zeitreihen. Im Folgenden werden wir drei Abstandsmaße kennenlernen, mit denen sich unter gewissen Voraussetzungen eine Aussage über die Ähnlichkeit von Zeitreihen treffen lässt.

Bei den Zeitreihen, die wir mit den Gütekriterien vergleichen, handelt es sich nicht etwa um die kompletten Messreihen, sondern lediglich den Ausschnitt beider Messreihen im Alignment, der sich gegenseitig überlappt. So ist gewährleistet, dass wir nur Zeitreihen gleicher Länge miteinander vergleichen.

### 2.4.1 Mittlerer quadratischer Fehler

Der mittlere quadratische Fehler<sup>1</sup> ist ein Abstandsmaß zwischen Zeitreihen, das eng verwandt mit dem euklidischen Abstand ist.

**Definition 2.7 (Mittlerer quadratischer Fehler)**

*Der mittlere quadratische Fehler (MQF) zwischen zwei gleichlangen Zeitreihen  $x, y \in \mathbb{R}^n$  ist*

$$\text{MQF}(x, y) := \frac{1}{n} \cdot \sum_{i=0}^{n-1} (x_i - y_i)^2$$

Der mittlere quadratische Fehler ist eine Maßzahl für die Fläche zwischen zwei Messkurven. Das heißt je größer der mittlere quadratische Fehler ist, umso größer ist auch die Fläche zwischen den Kurven. Somit ist anschaulich klar, dass ein Alignment mit einem niedrigen MQF zwei Messkurven gut zur Deckung bringt.

Die Abbildung  $\text{MQF} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  ist jedoch keine Metrik im  $\mathbb{R}^n$ , da sie die Dreiecksungleichung nicht erfüllt. Seien  $x, y, z \in \mathbb{R}$  drei  $1d$ -Zeitreihen der Länge

---

<sup>1</sup>engl.: mean squared error (MSE)



$n = 1$  mit  $x := 1, y := 2, z := 3$ . Dann ist

$$\text{MQF}(x, z) = 4 \not\leq 2 = \text{MQF}(x, y) + \text{MQF}(y, z)$$

Um eine Metrik (vgl. beispielsweise [For77]) zu erhalten und somit die Dreiecksungleichung nutzen zu können, müssen wir den MQF leicht modifizieren.

**Lemma 2.8 (Metrikeigenschaften)**

Die Abbildung  $\sqrt{\text{MQF}} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}, (x, y) \mapsto \sqrt{\text{MQF}(x, y)}$  ist eine Metrik im  $\mathbb{R}^n$ .

**Beweis** Seien  $x, y, z \in \mathbb{R}^n$ . Wie wir sehen ist

$$\sqrt{\text{MQF}}(x, y) = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - y_i)^2} = \sqrt{\frac{1}{n}} \cdot \sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2} = \frac{d_{\text{eukl}}(x, y)}{\sqrt{n}}$$

wobei

$$d_{\text{eukl}}(x, y) = \sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2}$$

der euklidische Abstand im  $\mathbb{R}^n$  ist. Da wir wissen, dass der euklidische Abstand die Axiome einer Metrik erfüllt, fällt es uns nicht mehr schwer diese auch für  $\sqrt{\text{MQF}}$  nachzuweisen.

- *Definitheit:*

$$\sqrt{\text{MQF}}(x, y) = 0 \Leftrightarrow \frac{d_{\text{eukl}}(x, y)}{\sqrt{n}} = 0 \Leftrightarrow d_{\text{eukl}}(x, y) = 0 \Leftrightarrow x = y$$

- *Symmetrie:*

$$\sqrt{\text{MQF}}(x, y) = \frac{d_{\text{eukl}}(x, y)}{\sqrt{n}} = \frac{d_{\text{eukl}}(y, x)}{\sqrt{n}} = \sqrt{\text{MQF}}(y, x)$$

- *Dreiecksungleichung:*

$$\begin{aligned} \sqrt{\text{MQF}}(x, y) &= \frac{d_{\text{eukl}}(x, y)}{\sqrt{n}} \leq \frac{d_{\text{eukl}}(x, z) + d_{\text{eukl}}(z, y)}{\sqrt{n}} \\ &= \sqrt{\text{MQF}}(x, z) + \sqrt{\text{MQF}}(z, y) \end{aligned}$$

$\Rightarrow \sqrt{\text{MQF}}$  ist eine Metrik im  $\mathbb{R}^n$ . □

Wir könnten uns nun fragen, weshalb wir nicht einfach den euklidischen Abstand als Gütekriterium verwenden. Dies wäre ungeschickt, da wir das Gütekriterium über Alignments mit verschiedenen Skalierungsfaktoren optimieren. Da der euklidische Abstand nicht über die Länge der Zeitreihen gemittelt, hätten zwei Zeitreihen, von denen eine sehr klein skaliert ist, also der Überlappungsbereich beider Messreihen klein ist, von vorne herein einen relativ niedrigen euklidischen Abstand. Das Alignment würde somit durch den euklidischen Abstand unverhältnismäßig gut bewertet werden.

Da wir den mittleren quadratischen Fehler verwenden wollen, um eine Aussage über die Ähnlichkeit von Zeitreihen zu treffen, wäre es wünschenswert, wenn zwei zueinander in vertikale Richtung verschobene, aber ansonsten gleiche Zeitreihen, mit einem niedrigen Fehler bewertet würden. Um dies zu erreichen, sollten wir die Zeitreihen vor der Berechnung des MQF um ihren Mittelwert bereinigen, das heißt wir subtrahieren von jedem einzelnen Messpunkt den Mittelwert der gesamten Zeitreihe.

**Definition 2.9 (Mittelwertbereinigter MQF)**

Der mittelwertbereinigte MQF zwischen zwei gleichlangen Zeitreihen  $x, y \in \mathbb{R}^n$  ist

$$\text{MQF}_m(x, y) := \text{MQF}(x - \mathbf{1} \cdot \bar{x}, y - \mathbf{1} \cdot \bar{y}) = \frac{1}{n} \cdot \sum_{i=0}^{n-1} ((x_i - \bar{x}) - (y_i - \bar{y}))^2$$

Dabei sind  $\bar{x} = \frac{1}{n} \cdot \sum_{i=0}^{n-1} x_i$  bzw.  $\bar{y} = \frac{1}{n} \cdot \sum_{i=0}^{n-1} y_i$  die Mittelwerte von  $x$  bzw.  $y$  und  $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^n$  der Einsvektor. Analog zu Lemma 2.8 definieren wir

$$\sqrt{\text{MQF}_m}(x, y) := \sqrt{\text{MQF}(x - \mathbf{1} \cdot \bar{x}, y - \mathbf{1} \cdot \bar{y})}$$

um die Dreiecksungleichung zu erhalten.

Für die Abbildung  $\sqrt{\text{MQF}_m}$  gilt zwar die Dreiecksungleichung und Symmetrie, was vollkommen analog zu Lemma 2.8 gezeigt werden kann, allerdings handelt es sich hierbei nicht um eine Metrik, da die Definitheitsbedingung verletzt wird. So haben beispielsweise alle konstanten Zeitreihen einen  $\sqrt{\text{MQF}_m}$  von 0, woraus jedoch nicht folgt, dass diese auch gleich sind. Es gilt vielmehr, dass

$$\sqrt{\text{MQF}_m}(x, y) = 0 \quad \Leftrightarrow \quad x = y + \mathbf{1} \cdot b$$

für ein  $b \in \mathbb{R}$ .

### 2.4.2 Korrelationskoeffizient

Ein Maß, das unter anderem in der Bild- oder Tonverarbeitung verwendet wird, um die Ähnlichkeit von Zeitreihen zu messen, ist der Korrelationskoeffizient.

**Definition 2.10 (Korrelationskoeffizient)**

Der (empirische) Korrelationskoeffizient zweier Zeitreihen  $x, y \in \mathbb{R}^n$  ist

$$\text{Kor}(x, y) := \frac{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \bar{y})^2}}$$

Obwohl nach [Pfa63] erhebliche Zweifel daran bestehen, ob der Korrelationskoeffizient tatsächlich als Maß für die Ähnlichkeit von Zeitreihen dienen kann, werden wir in Kapitel 6 empirisch belegen, dass er für unsere Zwecke durchaus geeignet ist.

Sehen wir uns also einige Eigenschaften des Korrelationskoeffizienten an, die für uns von Interesse sind.

**Lemma 2.11 (Eigenschaften des Korrelationskoeffizienten)**

Seien  $x, y \in \mathbb{R}^n$ . Es gilt

1.  $\text{Kor}(x, y) = \text{Kor}(y, x)$
2.  $\text{Kor}(x, -y) = -\text{Kor}(x, y)$
3.  $\text{Kor}(x, x) = 1$
4.  $\text{Kor}(x, y) \in [-1, 1]$

**Beweis**    Seien  $x, y \in \mathbb{R}^n$

**zu 1)** Klar nach Definition 2.10.

**zu 2)** Für den Mittelwert gilt:  $\overline{-y} = -\bar{y}$

$$\begin{aligned} \Rightarrow \text{Kor}(x, -y) &= \frac{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})(-y_i + \bar{y})}{\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (-y_i + \bar{y})^2}} \\ &= \frac{-\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \bar{y})^2}} \\ &= -\text{Kor}(x, y) \end{aligned}$$

zu 3)

$$\begin{aligned} \text{Kor}(x, x) &= \frac{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})(x_i - \bar{x})}{\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}} \\ &= \frac{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}{\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}^2} = \frac{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}{\left| \frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2 \right|} \\ &= 1 \quad , \text{ da } (x_i - \bar{x})^2 > 0 \quad \forall i \end{aligned}$$

Mit 2) folgt insbesondere:  $\text{Kor}(x, -x) = -1$

zu 4) Sei  $\langle \cdot, \cdot \rangle$  das Standardskalarprodukt und  $\|\cdot\|$  die induzierte Norm im  $\mathbb{R}^n$ . Dann folgt mit der Cauchy-Schwarzschen Ungleichung

$$\begin{aligned} \frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})(y_i - \bar{y}) &= \frac{\langle x - \bar{x}, y - \bar{y} \rangle}{n} \\ &\leq \frac{\|x - \bar{x}\| \cdot \|y - \bar{y}\|}{n} \\ &= \frac{1}{\sqrt{n}} \cdot \sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \frac{1}{\sqrt{n}} \cdot \sqrt{\sum_{i=0}^{n-1} (y_i - \bar{y})^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \bar{y})^2} \end{aligned}$$

$$\Rightarrow \text{Kor}(x, y) \leq 1$$

Angenommen  $\exists x', y' \in \mathbb{R}^n$ , so dass  $\text{Kor}(x', y') < -1$ : Dann folgt mit 2), dass  $\text{Kor}(x, -y) = -\text{Kor}(x, y) > 1$ .

Dies ist ein Widerspruch zu  $\text{Kor}(x, y) \leq 1 \quad \forall x, y \in \mathbb{R}^n$ .

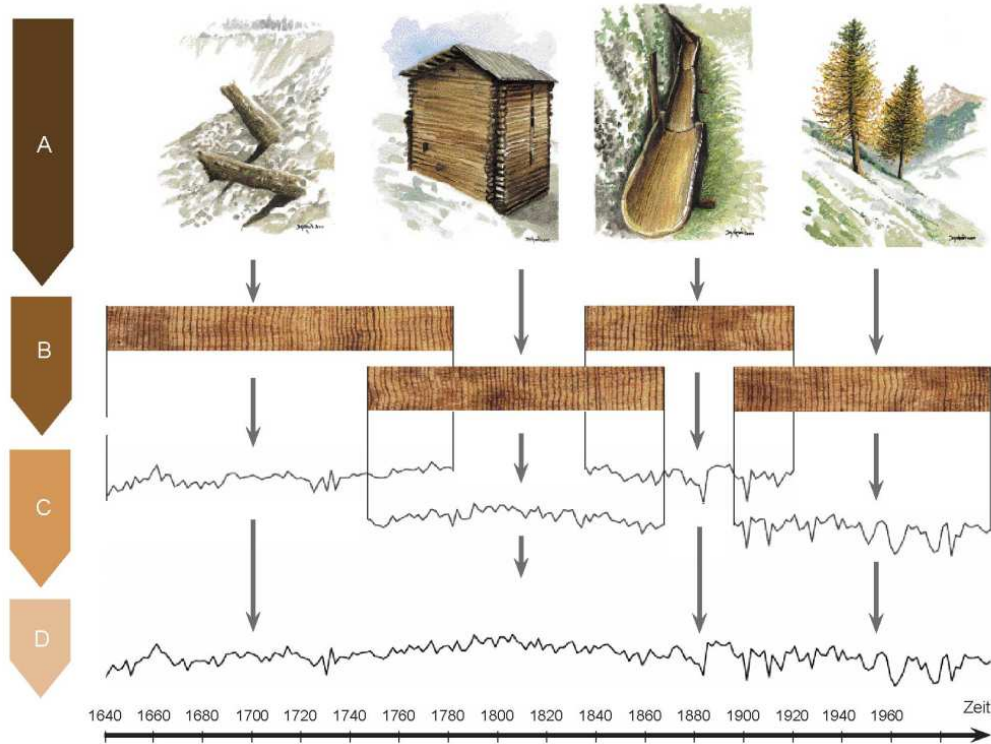
$$\Rightarrow \text{Kor}(x, y) \in [-1, 1] \quad \square$$

### 2.4.3 Gleichläufigkeitskoeffizient

Der Gleichläufigkeitskoeffizient ist ein Ähnlichkeitsmaß für Zeitreihen, das in der Dendrochronologie erfolgreich zur Bestimmung von optimalen Alignments verwendet wird.

Die Dendrochronologie bzw. Jahrringforschung beschäftigt sich damit das Alter von Bäumen oder Holz, das zum Beispiel bei einer Ausgrabung gefunden wird, anhand der Struktur ihrer Jahresringe zu bestimmen. Hierzu muss eine Sequenz von einer langen Zeitreihe, einer sogenannten Chronologie, gefunden

werden, die einer kurzen Vorgabereihe, die aus den verschiedenen Dicken der Jahresringe eines Fundes entsteht, möglichst ähnlich ist. Abbildung 2.4 zeigt die Entstehung einer solchen Chronologie durch Alignments einzelner Fundstücke untereinander.



**Abbildung 2.4:** Entstehung einer Chronologie (in Abschnitt D) durch Anwendung von Crossdating (in Abschnitt C) [Dey00]

Um die Ähnlichkeit zweier Zeitreihen zu bestimmen, wurde der Gleichläufigkeitskoeffizient 1943 von Bruno Huber in [Hub43] vorgeschlagen und beispielsweise von Carola Wenk in „Algorithmen für das Crossdating von Jahresringen“ [Wen97] in Alignmentalgorithmen verwendet.

**Definition 2.12 (Gleichläufigkeitskoeffizient)**

Seien  $x, y \in \mathbb{R}^n$  zwei Zeitreihen. Der Gleichläufigkeitskoeffizient (GLK) ist

$$\text{GLK}(x, y) := \frac{1}{n-1} \cdot \sum_{i=0}^{n-2} \chi(\text{sign}(x_{i+1} - x_i) = \text{sign}(y_{i+1} - y_i))$$

wobei

$$\chi(a = b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$$

Anschaulich gibt der GLK an, welcher Prozentsatz der beiden Zeitreihen in dieselbe Richtung verläuft. Ob dies ein adäquates Maß für Ähnlichkeit ist, werden wir in späteren Kapiteln evaluieren. Die Besonderheit dieses Gütekriteriums ist, dass jeglicher  $y$ -Versatz einzelner Messpunkte ignoriert wird. So haben zum Beispiel alle monoton steigenden (fallenden) Zeitreihen einen GLK von 1 mit jeder anderen monoton steigenden (fallenden) Zeitreihe. Das heißt mit dem GLK lässt sich nicht bestimmen, welche dieser Zeitreihen sich am ähnlichsten sind.

## 2.5 Piecewise Aggregate Approximation

Wir werden später sehen, dass für die Zuordnung von Dickenschriften ein Alignment mit geringer Genauigkeit ausreicht. Es bietet sich daher an, die vorliegenden Daten zu komprimieren, um eine höhere Geschwindigkeit auf Kosten der Genauigkeit zu erreichen.

Mit der Piecewise Aggregate Approximation (PAA), welche beispielsweise in [Gno06] beschrieben wird, wird eine Zeitreihe durch stückweise Mittelwertbildung komprimiert.

### Definition 2.13 (Piecewise Aggregate Approximation)

Sei  $x \in \mathbb{R}^n$  eine Zeitreihe der Länge  $n$ . Dann ist  $\text{PAA}_k(x)$  eine Zeitreihe der Länge  $\lceil \frac{n}{k} \rceil$ , deren Komponenten

$$(\text{PAA}_k(x))_j := \frac{1}{k} \cdot \sum_{i=kj}^{k(j+1)-1} x_i \quad , j = 0, \dots, \lceil \frac{n}{k} \rceil - 2$$

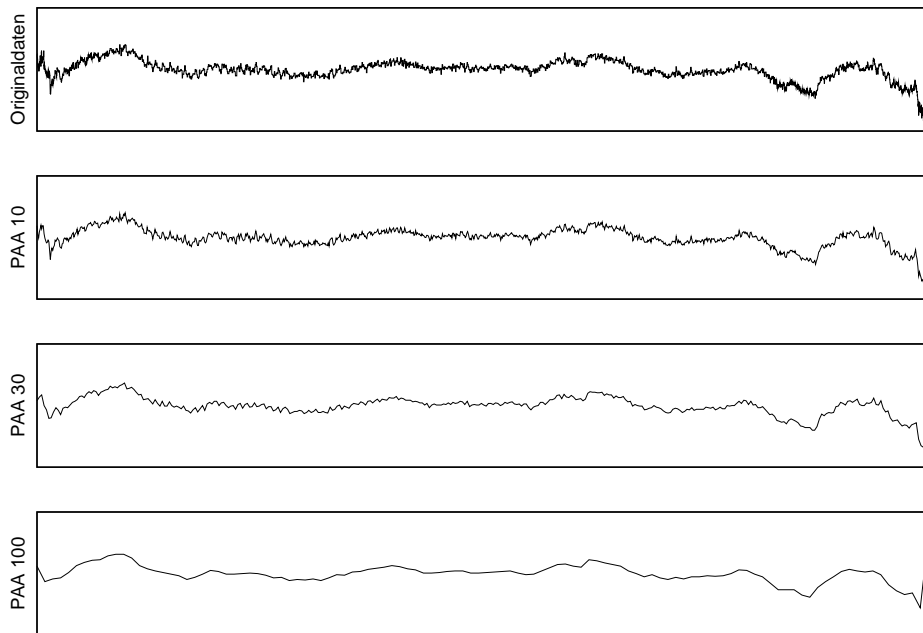
und

$$(\text{PAA}_k(x))_{\lceil \frac{n}{k} \rceil - 1} := \frac{1}{n - (k \cdot (\lceil \frac{n}{k} \rceil - 1) - 1)} \cdot \sum_{i=k \cdot (\lceil \frac{n}{k} \rceil - 1)}^{n-1} x_i$$

sind.

Wie wir in Abbildung 2.5 sehen, bleibt die Charakteristik eines Dickenscribes auch bei relativ hohen Komprimierungsstufen erhalten.

Neben der Komprimierung der Daten, ist ein weiterer Vorteil dieser Methode, dass sie für große  $k$  auch in der Lage ist, kurzzeitige Messstörungen und Ausreißer aus den Daten zu filtern.



*Abbildung 2.5: Vergleich verschiedener Komprimierungsstufen*

## 2.6 Diskrete Fouriertransformation

Die diskrete Fouriertransformation (DFT) wird in der Signalverarbeitung verwendet, um Messreihen zu komprimieren und deren Schwingungen zu analysieren. Wir werden sie in Kapitel 4.2 von einem abstrakteren Standpunkt aus betrachten und mit Hilfe einiger Fouriertransformationen die Berechnungen in unserem Alignmentverfahren beschleunigen. Aus diesem Grund werden wir uns hier nur die Definitionen ansehen, ohne auf die praktische Bedeutung der Fouriertransformation näher einzugehen.

### **Definition 2.14 (Diskrete Fouriertransformation)**

Sei  $x \in \mathbb{C}^n$  ein komplexer Vektor. Die Abbildung  $\text{DFT} : \mathbb{C}^n \rightarrow \mathbb{C}^n$  ist definiert durch

$$\text{DFT}(x)_k := \sum_{j=0}^{n-1} x_j \cdot e^{-2\pi i \cdot \frac{jk}{n}} \quad \text{mit } k = 0, \dots, n-1$$

und wird *diskrete Fouriertransformation* genannt.

Der Vektor  $\hat{x} := \text{DFT}(x)$  ist die *Fouriertransformierte* von  $x$ , während dessen Komponenten  $\hat{x}_i$  als *Fourierkoeffizienten* bezeichnet werden.

**Lemma 2.15 (Inverse DFT)**

Sei  $\hat{x} \in \mathbb{C}^n$ . Die Abbildung  $\text{iDFT} : \mathbb{C}^n \rightarrow \mathbb{C}^n$  mit

$$\text{iDFT}(\hat{x})_k = \frac{1}{n} \sum_{j=0}^{n-1} \hat{x}_j \cdot e^{2\pi i \cdot \frac{jk}{n}} \quad \text{für } k = 0, \dots, n-1$$

ist die Inverse der Abbildung DFT aus Definition 2.14. Es gilt

$$\text{DFT} \circ \text{iDFT} = \text{iDFT} \circ \text{DFT} = \text{id}$$

Bevor wir dieses Lemma beweisen können, müssen wir noch den Begriff der Einheitswurzel definieren und einige für uns wichtige Eigenschaften herausstellen.

**2.6.1 Einheitswurzeln**

Für den Beweis von Lemma 2.15 reicht es, die Einheitswurzeln im Körper  $\mathbb{C}$  zu betrachten. Da wir Einheitswurzeln nur für diesen Beweis benötigen, werden wir versuchen die Ausführungen möglichst kurz zu halten.

**Definition 2.16 ( $n$ -te Einheitswurzel in  $\mathbb{C}$ )**

$z \in \mathbb{C}$  wird  $n$ -te Einheitswurzel genannt, falls  $z^n = 1$  gilt.

**Lemma 2.17 (Einheitswurzeln in  $\mathbb{C}$ )**

Für ein  $n \in \mathbb{N}$  existieren genau  $n$  verschiedene  $n$ -te Einheitswurzeln  $\zeta_1, \dots, \zeta_n \in \mathbb{C}$ . Diese haben die Form  $\zeta_k = e^{\frac{2\pi i k}{n}}$  für  $k = 1, \dots, n$ .

$\zeta := \zeta_1$  wird primitive  $n$ -te Einheitswurzel genannt. Somit gilt offensichtlich  $\zeta_k = \zeta^k$ , weswegen wir diese Notation äquivalent verwenden.

**Beweis** Laut dem Fundamentalsatz der Algebra für reelle Polynome (vgl. beispielsweise [Rem92]) besitzt das Polynom  $z^n - 1$  genau  $n$  Nullstellen (mit Vielfachheiten) in  $\mathbb{C}$ . Es existieren also  $n$  - nicht unbedingt verschiedene - Werte  $z \in \mathbb{C}$ , so dass  $z^n = 1$  gilt.

Um die Behauptung zu zeigen, genügt es also zu zeigen, dass  $\zeta_1, \dots, \zeta_n$  genau diese Nullstellen sind. Mit

$$\zeta_k^n = \left( e^{\frac{2\pi i k}{n}} \right)^n = (e^{2\pi i})^k = (\cos(2\pi) + i \sin(2\pi))^k = (1 + i \cdot 0)^k = 1$$

folgt, dass

$$\zeta_k^n - 1 = 0 \quad \text{für } k = 1, \dots, n$$

□



Wie wir sehen, scheint es fast unlogisch  $k$  auf den Bereich von 1 bis  $n$  einzuschränken, da der Beweis für jedes beliebige  $k \in \mathbb{Z}$  funktioniert. Im folgenden Lemma werden wir jedoch sehen, dass wir die Einheitswurzeln, die beispielsweise aus einem  $k \geq n$  resultieren, bereits definiert haben.

**Lemma 2.18 (Periodizität)**

Sei  $n \in \mathbb{N}$  und  $\zeta \in \mathbb{C}$  die primitive  $n$ -te Einheitswurzel. Dann gilt für alle  $k = 1, \dots, n$  und  $p \in \mathbb{Z}$

$$\zeta^{k+pn} = \zeta^k$$

**Beweis** Durch einfaches Nachrechnen sehen wir, dass

$$\zeta^{k+pn} = \zeta^k \cdot \zeta^{pn} = \zeta^k \cdot (\zeta^n)^p = \zeta^k \cdot 1^p = \zeta^k \quad \square$$

Bevor wir nun endlich zum Beweis von Lemma 2.15 kommen, benötigen wir noch zwei Summenformeln, die für Einheitswurzeln gelten.

**Lemma 2.19 (Summenformeln)**

Für  $n > 1$  und die primitive  $n$ -te Einheitswurzel  $\zeta$  gilt

$$\sum_{k=0}^{n-1} \zeta^k = 0$$

und

$$\sum_{k=0}^{n-1} \overline{\zeta^{jk}} \cdot \zeta^{lk} = n \cdot \delta_{j \bmod n, l \bmod n} = \begin{cases} n & , \text{ falls } j \bmod n = l \bmod n \\ 0 & , \text{ sonst} \end{cases}$$

**Beweis** Die erste Summe ist eine Partialsumme der geometrischen Reihe. Somit ist klar, dass

$$\sum_{k=0}^{n-1} \zeta^k = \frac{\zeta^n - 1}{\zeta - 1} = \frac{1 - 1}{\zeta - 1} = 0 \quad \text{für } n > 1.$$

Analog dazu sehen wir, dass  $\sum_{k=0}^{n-1} (\zeta_j)^k = 0$  für jede beliebige  $n$ -te Einheitswurzel  $\zeta_j \neq 1$ . Mit dieser Erkenntnis lässt sich auch die Korrektheit der zweiten Summe zeigen. Betrachten wir also

$$f(j, l) := \sum_{k=0}^{n-1} \overline{\zeta^{jk}} \cdot \zeta^{lk}$$

Falls  $j \bmod n = l \bmod n$  ist, so gilt wegen der Periodizität der Einheitswurzeln

$$f(j, l) = \sum_{k=0}^{n-1} \overline{\zeta^{jk}} \cdot \zeta^{lk} = \sum_{k=0}^{n-1} e^{-\frac{2\pi i}{n}(jk) \bmod n} \cdot e^{\frac{2\pi i}{n}(lk) \bmod n}$$

und wegen  $j \bmod n = l \bmod n \Leftrightarrow (jk) \bmod n = (lk) \bmod n$  ergibt sich

$$f(j, l) = \sum_{k=0}^{n-1} e^0 = n$$

Für den Fall, dass  $j \bmod n \neq l \bmod n$  ist, sehen wir

$$f(j, l) = \sum_{k=0}^{n-1} e^{-\frac{2\pi i}{n}jk} \cdot e^{\frac{2\pi i}{n}lk} = \sum_{k=0}^{n-1} e^{\frac{2\pi i}{n}(l-j)k}$$

Wegen der Periodizität ist  $e^{\frac{2\pi i}{n}(l-j)} = e^{\frac{2\pi i}{n}(l-j) \bmod n}$  für alle  $j, l \in \mathbb{Z}$  eine  $n$ -te Einheitswurzel  $\neq 1$ , da  $(l-j) \bmod n \neq 0$ . Somit gilt wegen der ersten Summe

$$f(j, l) = \sum_{k=0}^{n-1} \left( e^{\frac{2\pi i}{n}(l-j)} \right)^k = 0$$

für  $n > 1$ . Also ist  $f(j, l) = n \cdot \delta_{j \bmod n, l \bmod n}$  □

Kommen wir nun zurück zum eigentlichen Thema des Kapitels, der diskreten Fouriertransformation. Hier steht noch immer der Beweis von Lemma 2.15 aus, der uns nach dem Einschub über Einheitswurzeln in  $\mathbb{C}$  deutlich leichter fällt.

**Beweis (von Lemma 2.15)** Es ist zu zeigen, dass  $\text{iDFT}(\text{DFT}(x))_k = x_k$  für alle  $x \in \mathbb{R}^n$  und  $k = 0, \dots, n-1$ .

$$\begin{aligned} \text{iDFT}(\text{DFT}(x))_k &= \frac{1}{n} \sum_{j=0}^{n-1} \text{DFT}(x)_j \cdot e^{2\pi i \cdot \frac{jk}{n}} \\ &= \frac{1}{n} \sum_{j=0}^{n-1} \sum_{l=0}^{n-1} x_l \cdot e^{-\frac{2\pi i}{n}lj} \cdot e^{\frac{2\pi i}{n}jk} \end{aligned}$$

Nach dem Vertauschen der Summen und Einsetzen der  $n$ -ten Einheitswurzeln ergibt sich

$$\text{iDFT}(\text{DFT}(x))_k = \frac{1}{n} \sum_{l=0}^{n-1} x_l \sum_{j=0}^{n-1} \overline{\zeta^{lj}} \cdot \zeta^{kj}$$

Nach Lemma 2.19 ist dies gerade

$$\frac{1}{n} \sum_{l=0}^{n-1} x_l \cdot n\delta_{l,k}$$

*Es folgt also letztendlich*

$$\text{iDFT}(\text{DFT}(x))_k = \frac{1}{n} \sum_{l=0}^{n-1} x_l \cdot n\delta_{l,k} = n \cdot \frac{1}{n} \cdot x_k = x_k \quad .$$

*Somit gilt auch*

$$\begin{aligned} \text{iDFT}(\text{DFT}(x)) &= x \\ \Leftrightarrow \text{DFT}(\text{iDFT}(\text{DFT}(x))) &= \text{DFT}(x) \\ \Leftrightarrow \text{DFT}(\text{iDFT}(x)) &= x \quad \square \end{aligned}$$

Für uns besteht ein wesentlicher Vorteil in der Verwendung der DFT darin, dass  $\text{DFT}(x)$  für  $x \in \mathbb{C}^n$  durch sehr effiziente Algorithmen berechnet werden kann. Der bekannteste und erste Algorithmus dieser Art wurde von Cooley und Tukey (vgl. [CT65] bzw. [But05]) im Jahr 1965 vorgestellt. Dieser Algorithmus, auch Radix-2-Algorithmus genannt, nutzt das Divide-and-Conquer Prinzip, um den Vektor  $\text{DFT}(x)$  in  $\mathcal{O}(n \log n)$  zu berechnen, während der naive Ansatz in  $\mathcal{O}(n^2)$  liegt.

Obwohl wir die DFT für komplexe Vektoren (bzw. Zeitreihen) definiert haben, werden wir sie nur auf reelle Zeitreihen anwenden. Für diesen Spezialfall existieren effizientere Algorithmen als der Radix-2-Algorithmus. Einen für unsere Zwecke geeigneten Algorithmus findet man zum Beispiel in [PTVF02, Kapitel 12.3]. Der dort vorgestellte Algorithmus ist darauf spezialisiert, zwei reelle Folgen gleichzeitig zu transformieren. Dies entspricht genau unserer Anwendung in Kapitel 4.2.



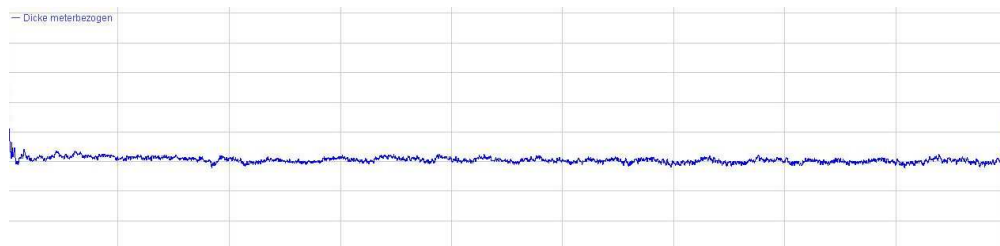
## Kapitel 3

# Datenaufbereitung

Die Dickenmesswerte, welche im Walzwerk aufgezeichnet werden, sind zeitbezogen. Das heißt das Messgerät misst die Dicke des Stahlbandes in einem gewissen Zeittakt (beispielsweise alle  $10\text{ ms}$ ). Da die Geschwindigkeit des Stahlbandes, während es durch die Walzanlage läuft, nicht konstant ist (siehe Abbildung 3.1(a)), ist der Dickenschrieb gegenüber einer längenbezogenen Darstellung ungleichmäßig skaliert. Um auf diese Messwerte ein Alignmentverfahren anwenden zu können, das einen Skalierungsfaktor für das gesamte Band sucht, müssen die Daten vor dem Alignment so aufbereitet werden, dass die Messpunkte örtlich äquidistant aufgelöst sind.



(a) Zeitbezogene Daten von Dicke (blau) und Geschwindigkeit (rot) aus der Warmfertigstraße



(b) Meterbezogener Dickenschrieb

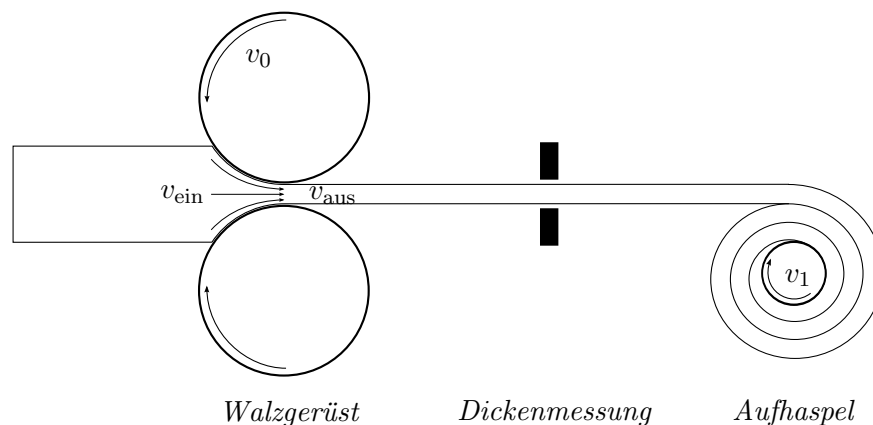
**Abbildung 3.1:** Aufbereitung eines meterbezogenen Stahlbanddickenprofils

Da an verschiedenen Stellen im Walzprozess auch Geschwindigkeiten gemessen werden, können wir mit Hilfe der Geschwindigkeitsmessungen die zeitbezogene Dickenmessung auf ein Ortsgitter transformieren (siehe Abbildung 3.1(b)). In der Praxis gibt es jedoch zwei Stolpersteine, die zu beachten sind:

- Die Messgeräte messen kontinuierlich über alle Bänder hinweg.
- Die Geschwindigkeitsmessungen finden weder direkt am Band noch in unmittelbarer Nähe des Dickenmessgerätes statt.

Wie man in Abbildung 3.1(a) sieht, beginnt am rechten Rand das Dickenprofil des nächsten Stahlbandes. In einer Messdatei können also Messdaten von mehreren verschiedenen Stahlbändern abgelegt sein, da die Messgeräte ununterbrochen messen. Um die Daten eines einzelnen Stahlbandes zu extrahieren, muss ein Binärsignal herangezogen werden, das angibt, ob sich zu einem gewissen Zeitpunkt Walzgut in der Dickenmessung befunden hat oder nicht. Somit lässt sich ein Band gezielt aus der Messdatei extrahieren, indem man einen Block aus den Messdaten ausschneidet, an dem das Binärsignal `true` ist (siehe Balken am unteren Rand von Abbildung 3.1(a)).

Das größere Problem bei der Transformation der Messungen auf ein Ortsgitter ist jedoch, dass die tatsächliche Bandgeschwindigkeit im Produktionsprozess nicht gemessen wird. Die Geschwindigkeit des Bandes kann jedoch aus der Um-drehungsgeschwindigkeit  $v_0$  der Walzen berechnet werden (siehe Abbildung 3.2). Vor allem bei der starken Dickenreduktion im Warmwalzwerk ist dies jedoch nur



**Abbildung 3.2:** Dicken- und Geschwindigkeitsmessung im Warmwalzwerk

eine Approximation an die tatsächliche Bandgeschwindigkeit. Die Verformung des Stahlbandes beim Walzen hat einen Materialfluss zur Folge, der sich vor allem auf die Länge des Bandes auswirkt. Durch die sogenannte Materialvoreilung

---

hat das Stahlband, wie in Abbildung 3.2 zu sehen, eine Austrittsgeschwindigkeit  $v_{\text{aus}}$  aus den Walzen, die höher als die Eintrittsgeschwindigkeit  $v_{\text{ein}}$  ist. Für die Walzengeschwindigkeit  $v_0$  gilt

$$v_{\text{ein}} \leq v_0 \leq v_{\text{aus}} \quad .$$

Wie wir in Abbildung 3.2 sehen, ist  $v_{\text{aus}}$  die tatsächliche Bandgeschwindigkeit im Dickenmessgerät. Das heißt mit der Walzengeschwindigkeit  $v_0$  schätzen wir die Bandgeschwindigkeit in der Dickenmessung nach unten ab.

Statt der Walzengeschwindigkeit  $v_0$  können wir auch die Geschwindigkeit des Aufhaspels  $v_1$  heranziehen, um die Bandgeschwindigkeit zu bestimmen. Durch das Aufwickeln des Stahlbandes, nimmt der Durchmesser und somit auch der Umfang der Aufhaspel jedoch stetig zu (vgl. auch Abbildung 1.1). Das heißt, wir benötigen zusätzlich zur Geschwindigkeit noch den aktuellen Durchmesser der Aufhaspel. Tatsächlich ist dieser in den Geschwindigkeitssignalen aus dem Walzwerk schon mit berücksichtigt, so dass die gelieferte Aufhaspelgeschwindigkeit in etwa der Bandgeschwindigkeit entspricht. Diese Abschätzung ist jedoch ungenauer als die Abschätzung der Bandgeschwindigkeit durch die Walzengeschwindigkeit, da es durch die Messung des Haspeldurchmessers mehr Störfaktoren gibt.

Ein weiteres Problem dieser Approximationen ist der Ort, an dem die Geschwindigkeit gemessen wird. Die Walzengeschwindigkeit wird vor und die Aufhaspelgeschwindigkeit nach der Dickenmessung gemessen. Wenn wir uns nun für eine der beiden Geschwindigkeiten zur Approximation entscheiden, so haben wir immer einen gewissen Abschnitt, in dem die Bandgeschwindigkeit vollkommen unbekannt ist. Am Beispiel der Walzengeschwindigkeit wäre dies der Abschnitt, an dem das Ende des Bandes das letzte Walzgerüst verlässt, aber sich das Band noch in der Dickenmessung befindet. Wenn wir davon ausgehen, dass Walzen und Dickenmessgerät  $10 \text{ m}$  voneinander entfernt sind, hätten wir einen Abschnitt von  $10 \text{ m}$  im Band, von dem wir keine Geschwindigkeit beim Durchlaufen der Dickenmessung kennen.

Die genaueste Transformation auf ein Ortsgitter lässt sich mit einer Kombination der beiden gemessenen Geschwindigkeiten erreichen. Da die Walzengeschwindigkeit etwas genauer ist, sollte sie für den größten Teil der Messreihe als Bandgeschwindigkeit angenommen werden. Je nachdem wie groß der Abstand von den Walzen zur Dickemessung in der jeweiligen Anlage ist, muss für die letzten Meter der Messreihe die Aufhaspelgeschwindigkeit herangezogen werden.

Durch die Voreilung, die wir nun im größten Teil der Messung nicht berücksichtigen, wird die auf ein Ortsgitter transformierte Messreihe des Warmbandes im Verhältnis zur Messreihe des Kaltbandes um einige Meter zu klein skaliert sein.

Die Transformation der Kaltbandmessung ist unkomplizierter, da die Dickenmessung am Eingang des Walzwerks und somit noch vor dem ersten Walzgerüst stattfindet. Im Regelfall sind hier Messrollen installiert, die die Geschwindigkeit des Bandes verlässlich messen. Selbst wenn diese Messrollen nicht existieren, so wäre die Approximation der Bandgeschwindigkeit durch die Walzengeschwindigkeit deutlich exakter als dies beim Warmwalzen der Fall ist, da die Voreilung aufgrund der eher geringen Dickenreduktion nur eine untergeordnete Rolle spielt.



## Kapitel 4

# Alignmentverfahren

In diesem Kapitel setzen wir uns mit Verfahren zur Bestimmung von Alignments, die möglichst optimal bezüglich eines Gütekriteriums sind, auseinander und vergleichen diese abschließend miteinander. Wir gehen dabei vom naiven Verfahren aus und werden dieses sukzessiv verbessern und insbesondere beschleunigen.

Bei allen Verfahren muss zu Beginn, also mit Übergabe der Parameter, spezifiziert werden, welche Messreihe die auszurichtende ist. Die andere Messreihe (sog. Fixedseries) wird dann im ganzen Algorithmus unverändert gelassen, während die auszurichtende Messreihe (sog. Alignseries) horizontal skaliert und verschoben wird.

Der Einfachheit halber wollen wir davon ausgehen, dass die auszurichtende Messreihe ein Teil der festen Messreihe ist, also insbesondere kürzer ist, wenn sie optimal ausgerichtet wurde (siehe Abbildung 4.1). Alle Verfahren lassen sich analog zu den Pseudocodes auch ohne diese Einschränkung implementieren, was allerdings auf den vorliegenden Daten nicht notwendig ist, da wir jeweils zwei Messungen von ein und dem selben Stahlband bevor und nachdem ein Teil abgeschnitten wurde vergleichen. Wir wissen also, dass die Kaltbandmessung komplett in der Warmbandmessung wiederzufinden sein muss.

Die hier vorgestellten Alignmentverfahren optimieren das Alignment bezüglich eines der Gütekriterien aus Abschnitt 2.4. Da die Auswertung dieser Gütekriterien im Wesentlichen der Berechnung einer Summe entspricht, wollen wir uns nur exemplarisch die Berechnung des MQF ansehen.

### **Algorithmus 4.1 (Berechnung des MQF mit Offset)**

*Eingabe:*

- Messreihen  $F = F_0, \dots, F_{n-1}$  und  $A = A_0, \dots, A_{m-1}$

- Verschiebung  $v$  von  $A$  in  $F$

*Ausgabe:*

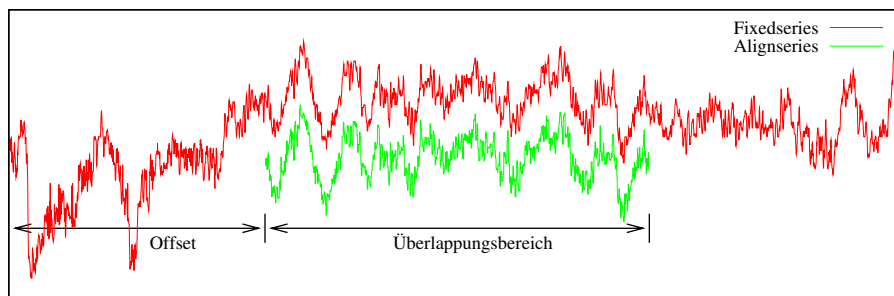
Der MQF von  $F$  und  $A$  im Überlappungsbereich mit der Verschiebung  $v$ .

```

1: function MSE( $F, A, v$ )
2:    $error = 0$ 
3:    $begin = \max(0, -v)$ 
4:    $end = \min(n - v, m)$ 
5:   for  $i = begin, \dots, end - 1$  do
6:      $error = error + (F_{v+i} - A_i)^2$ 
7:   return  $error / (end - begin)$ 

```

Wie wir sehen, wird das Gütekriterium (in diesem Fall der MQF) lediglich im Überlappungsbereich (siehe Abbildung 4.1) der beiden Messreihen ausgewertet.



**Abbildung 4.1:** Überlappungsbereich zwischen zwei Messreihen

### **Laufzeit (Algorithmus 4.1)**

*Aus*

$$end - begin = \min(n - v, m) - \max(0, -v) \leq \min(n, m)$$

ergibt sich, dass die Schleife aus Zeile 5 laut Zeile 3 und 4 maximal  $\min(n, m)$ -mal ausgeführt wird. Mit der Einschränkung, dass  $A$  kürzer als  $F$  ist, ist die Laufzeit also in  $\mathcal{O}(m)$ .

## **4.1 Naives Alignment**

Die kanonische Herangehensweise, ein optimales Alignment zu finden, ist, sich alle möglichen Alignments anzusehen und das Gütekriterium auszuwerten. Das

Alignment mit dem besten Gütekriterium ist dann das gesuchte.

#### 4.1.1 Offsetbestimmung

Wir wollen erst einmal eine Verschiebung der Messreihen gegeneinander finden, die das Gütekriterium optimiert. Der naive Algorithmus hierzu besteht lediglich aus der Auswertung des Gütekriteriums für jedes Offset.

##### Algorithmus 4.2 (Naive Bestimmung des Offsets)

*Eingabe:*

Messreihen  $F = F_0, \dots, F_{n-1}$  und  $A = A_0, \dots, A_{m-1}$  mit  $m \leq n$

*Ausgabe:*

Verschiebung von  $A$  gegen  $F$ , so dass der MQF der Überlappung minimal wird.

```
1: function CALCULATEOFFSET( $F, A$ )
2:    $mse = \infty$ 
3:   for all Offsets do
4:     if  $MSE(F, A, i) < mse$  then           // siehe Algorithmus 4.1
5:       Merke aktuelles  $i$  als offset
6:       Aktualisiere  $mse$ 
7:   return ( $offset, mse$ )           // Verschiebung und Fehler zurückgeben
```

##### Laufzeit (Algorithmus 4.2)

Da jeder Messpunkt in  $F$  ein mögliches Offset darstellt, wird die Schleife aus der 3. Zeile  $(n - m)$ -mal<sup>1</sup> ausgeführt. Zusammen mit der Laufzeit von Algorithmus 4.1 ergibt sich eine Gesamtlaufzeit in  $\mathcal{O}((n - m) \cdot m) \subset \mathcal{O}(n \cdot m)$ .

Natürlich ist es möglich, den Korrelationskoeffizienten aus Definition 2.10 oder den Gleichläufigkeitskoeffizienten aus Definition 2.12 zu maximieren statt den mittleren quadratischen Fehler zu minimieren. Hierzu muss Zeile 4 entsprechend abgeändert werden. Die Laufzeit wird dadurch nicht beeinflusst.

Wir können nun also ein Offset bestimmen, so dass das Gütekriterium zwischen den Messreihen  $F$  und  $A$  optimal wird. Leider ist dieses Offset alleine noch nicht ausreichend, um die Datensätze zufriedenstellend zur Deckung zu bringen. Durch den Materialfluss (siehe Kapitel 3) beim Walzen, der insbesondere bei der starken Dickenreduktion in der Warmwalzanlage ins Gewicht fällt, sind die Geschwindigkeitsmessungen nicht verlässlich. Das heißt, die Datensätze sind verschieden skaliert.

---

<sup>1</sup>Bei den letzten  $m$  Offsets liegt  $A$  nicht mehr in  $F$ .

### 4.1.2 Skalierungssuche

Um den kanonischen Ansatz weiter zu verfolgen, müssen wir uns nun alle möglichen Skalierungen ansehen und wieder jeweils das Gütekriterium auswerten. Allerdings gibt es potentiell unendlich viele verschiedene Skalierungsfaktoren. Die Tatsache, dass der auszurichtende Datensatz komplett im festen Datensatz wiederzufinden ist, schränkt den Suchbereich für den Skalierungsfaktor zwar ein, jedoch gibt es dazwischen immer noch unendlich viele Möglichkeiten zur Wahl des Faktors.

Da die vorliegenden Daten diskret sind, muss glücklicherweise nicht *jeder* Skalierungsfaktor durchprobiert werden. Es macht wenig Sinn, die auszurichtende Messreihe so zu skalieren, dass sie nachher  $\frac{1}{2}$  Messpunkt größer oder kleiner ist als vorher. Wir wollen uns darauf einigen, dass ausgehend von einem Skalierungsfaktor  $s_i$  der kleinstmögliche folgende Skalierungsfaktor  $s_{i+1}$  den Datensatz exakt um einen Messpunkt größer macht als  $s_i$ .

Sei also  $m$  die Länge des auszurichtenden Datensatzes, dann ist

$$s_{i+1} \cdot m = s_i \cdot m + 1 \quad \Leftrightarrow \quad s_{i+1} = s_i + \frac{1}{m} \quad .$$

Auf diese Weise können wir im folgenden Algorithmus über *alle* Skalierungsfaktoren iterieren.

#### Algorithmus 4.3 (Naives Alignment)

*Eingabe:*

- Messreihen  $F = F_0, \dots, F_{n-1}$  und  $A = A_0, \dots, A_{m-1}$
- Intervall  $[from, to]$  mit  $to \cdot m \leq n$ , in dem der Skalierungsfaktor gesucht wird

*Ausgabe:*

Horizontale Verschiebung und Skalierung von  $A$  gegen  $F$ , so dass der MQF der Überlappung minimal wird

```

1: function ALIGNNAIV( $F, A, from, to$ )
2:    $mse = \infty$ 
3:   for  $i = from \cdot m, \dots, to \cdot m$  do
4:     Skaliere  $A$  so, dass es Länge  $i$  hat           // siehe Abschnitt 2.2
5:      $result = \text{CALCULATEOFFSET}(F, S)$            // siehe Algorithmus 4.2
6:     if  $result.mse < mse$  then
7:       Merke  $offset$  und  $scale$ 
8:       Aktualisiere  $mse$ 
9:   return ( $offset, scale, mse$ )

```

**Laufzeit (Algorithmus 4.3)**

Die Schleife aus Zeile 3 wird  $(to \cdot m - from \cdot m)$ -mal ausgeführt. Das Skalieren von  $A$  in Zeile 4 liegt in  $\mathcal{O}(m)$  und fällt somit in der asymptotischen Laufzeitbetrachtung nicht ins Gewicht. Mit dem Aufruf von Algorithmus 4.2 ergibt sich also eine Gesamtlaufzeit in  $\mathcal{O}((to - from) \cdot m \cdot m \cdot n) \subset \mathcal{O}(n \cdot m^2)$ , da die Größe des Suchintervalls als konstanter Faktor zu betrachten ist.

Auch hier wurde der mittlere quadratische Fehler nur beispielhaft eingesetzt. Je nachdem welches Gütekriterium Algorithmus 4.2 zurückgibt, kann auch der Korrelations- oder Gleichläufigkeitskoeffizient maximiert werden.

## 4.2 Beschleunigung des naiven Alignments

Der Vorteil des naiven Alignmentverfahrens aus Abschnitt 4.1 ist, dass das gefundene Alignment *garantiert* optimal bezüglich des Gütekriteriums ist. Der schwerwiegende Nachteil ist die Performanz, da alle möglichen Alignments berechnet werden müssen. Wir werden in diesem Abschnitt die Performanz des naiven Alignmentverfahrens deutlich steigern ohne den Vorteil der Optimalität gänzlich aufzugeben.

Dazu wollen wir erst den Algorithmus zur Offsetbestimmung aus Abschnitt 4.1.1 und dann die Skalierungssuche aus Abschnitt 4.1.2 näher betrachten.

### 4.2.1 Fehlerberechnung mit oberer Grenze

Im Laufe von Algorithmus 4.2 wird der Wert der Variablen  $mse$  in Zeile 6 immer wieder aktualisiert, falls ein neues Offset in einem kleineren Fehler, als dem bisher gefundenen, resultiert. Man kann nun die Tatsache, dass die Werte, die  $mse$  trägt, über die Iterationen der Schleife in Zeile 3 monoton fallend sind, ausnutzen, um den Algorithmus zu beschleunigen.

Da bei der Berechnung des MQF in Algorithmus 4.1 der Wert der Variablen  $error$  ebenfalls im Verlauf der Schleife monoton steigend ist, kann man diese bereits abbrechen, wenn  $\frac{error}{end - begin}$  größer als der bisher kleinste gefundene Fehler aus Algorithmus 4.2 ist.

Wir benötigen also einen Algorithmus, der den MQF berechnet, falls dieser kleiner als eine obere Schranke ist. Falls der MQF größer als diese Schranke ist, so interessiert uns nicht der genaue Wert des Fehlers, sondern eben nur die Tatsache, dass er größer als die gegebene Schranke ist.

**Algorithmus 4.4 (Berechnung des MQF mit oberer Grenze)**

*Eingabe:*

- Messreihen  $F = F_0, \dots, F_{n-1}$  und  $A = A_0, \dots, A_{m-1}$
- Verschiebung  $v$  von  $A$  in  $F$
- Obere Schranke für die Fehlerberechnung

*Ausgabe:*

Der MQF von  $F$  und  $A$  mit der Verschiebung  $v$ , falls dieser kleiner als  $bound$  ist.

```

1: function MSE( $F, A, v, bound$ )
2:   Initialisierung wie in Algorithmus 4.1
3:   for  $i = begin, \dots, end$  do
4:      $error = error + (F_{v+i} - A_i)^2$ 
5:     if  $error/(end - begin) \geq bound$  then
6:       return  $\infty$ 
7:   return  $error/(end - begin)$ 

```

Dieser Algorithmus zur Fehlerberechnung kann nun in Algorithmus 4.2 verwendet werden. Als obere Schranke muss hier der Wert der Variablen  $mse$  übergeben werden. Die Modifikation hat keine Auswirkungen auf die theoretische Laufzeit des Verfahrens, da diese im schlimmsten Fall gleich ist. Allerdings lässt sich in der Praxis ein spürbarer Performanzgewinn feststellen (siehe Abschnitt 4.2.6).

Die Berechnung des Gleichläufigkeitskoeffizienten lässt sich vollkommen analog zur Fehlerberechnung abbrechen. Die Idee ist jedoch nicht ohne Weiteres auf den Korrelationskoeffizienten übertragbar.

**Abbruch der Berechnung des Korrelationskoeffizienten**

Um das Verfahren auf den Korrelationskoeffizienten eines Alignments, also von zwei ausgerichteten Messreihen  $F \in \mathbb{R}^n$  und  $A \in \mathbb{R}^m$ , zu übertragen, müssen wir diesen frühzeitig nach oben abschätzen.

Sei  $F' \in \mathbb{R}^m$  der Ausschnitt von  $F$ , der im Überlappungsbereich zwischen  $F$  und  $A$  liegt. Wir wollen mit  $Kor_k(F', A)$  den Korrelationskoeffizienten der ersten  $k$  Messpunkte von  $F'$  und  $A$  bezeichnen, also

$$Kor_k(F', A) := \frac{\frac{1}{k} \sum_{i=0}^{k-1} (F'_i - \bar{F}') (A_i - \bar{A})}{\sqrt{\frac{1}{k} \sum_{i=0}^{k-1} (F'_i - \bar{F}')^2} \cdot \sqrt{\frac{1}{k} \sum_{i=0}^{k-1} (A_i - \bar{A})^2}} .$$

Betrachten wir diese Definition, wie bereits im Beweis von Lemma 2.11, mit

dem Standardskalarprodukt und der dadurch induzierten Norm, so ergibt sich, dass

$$\text{Kor}_k(F', A) = \frac{\langle F' - \bar{F}', A - \bar{A} \rangle}{\|F' - \bar{F}'\| \cdot \|A - \bar{A}\|}$$

ist. Dies entspricht genau dem Kosinus des Winkels zwischen den Vektoren  $F' - \bar{F}'$  und  $A - \bar{A}$  (vgl. beispielsweise [Bär96]).

Mit dieser geometrischen Interpretation lässt sich anschaulich nachvollziehen, weshalb wir keine sinnvolle obere Schranke für  $\text{Kor}_m(F', A) = \text{Kor}(F', A)$  angeben können. Stellen wir uns zwei beliebige Vektoren  $x, y \in \mathbb{R}^2$  vor, so können wir den Winkel zwischen diesen durch Hinzunahme einer weiteren Dimension, also  $k \rightarrow k+1$ , beliebig minimieren und damit den Kosinus bzw. die Korrelation maximieren. Es gilt also

$$\begin{aligned} \angle\left(\begin{bmatrix} x \\ x_2 \end{bmatrix}, \begin{bmatrix} y \\ y_2 \end{bmatrix}\right) &\longrightarrow 0 \quad \text{für } x_2, y_2 \longrightarrow \pm\infty \\ \Rightarrow \text{Kor}\left(\begin{bmatrix} x \\ x_2 \end{bmatrix}, \begin{bmatrix} y \\ y_2 \end{bmatrix}\right) &\longrightarrow 1 \quad \text{für } x_2, y_2 \longrightarrow \pm\infty \quad . \end{aligned}$$

Um eine Abschätzung der Korrelation zu erlauben, müssten wir eine weitere, durchaus realistische, Bedingung an die Dickenprofile stellen, nämlich, dass die Dicke nicht gegen  $\pm\infty$  geht. Falls wir ein Intervall angeben können, innerhalb dessen sich die Dicke bewegt, so können wir zu jedem „Zeitpunkt“  $k$  der Korrelationsberechnung eine Abschätzung des Koeffizienten nach oben angeben, indem wir annehmen, dass beide Messreihen ab dem Messpunkt  $x_k$  bzw.  $y_k$  gleichermaßen zwischen den Grenzen des Intervalls oszillieren.

Um das Intervall, in dem sich die Dicke bewegt, sinnvoll zu bestimmen, könnten wir entweder die maximalen Toleranzgrenzen des Walzprozesses oder einfach das Minimum und Maximum des vorliegenden Dickenprofils heranziehen. Minimum und Maximum müssten dann selbstverständlich vorher ermittelt werden, wodurch die Rechenzeit des Algorithmus höher als mit dem mittleren quadratischen Fehler wäre.

Wir werden in Abschnitt 4.2.6 bzw. 6.5.3 sehen, dass weder die Ergebnisse des Verfahrens noch die des Korrelationskoeffizienten als Gütekriterium diesen Aufwand rechtfertigen.

#### 4.2.2 Fehlerberechnung mittels FFT

Wir verfolgen nun einen anderen Ansatz, das naive Alignment zu beschleunigen. Hierzu wollen wir uns einen groben Überblick über die Idee hinter Algorithmus

4.5 verschaffen und dessen Korrektheit erst später beweisen.

Der mittlere quadratische Fehler im Überlappungsbereich zwischen zwei Messreihen  $x$  und  $y$ , die um  $t$  Messpunkte gegeneinander verschoben sind, lautet

$$\frac{1}{m} \sum_{i=0}^{m-1} (x_{i+t} - y_i)^2 \quad ,$$

wobei wir davon ausgehen, dass der Überlappungsbereich stets die Größe  $m$  hat. Das heißt, dass  $y$  durch die Verschiebung  $t$  nicht über  $x$  hinausragt, was unserer Annahme entspricht, dass die auszurichtende Messreihe innerhalb der festen Messreihe aufzufinden ist.

Durch Anwenden der zweiten binomischen Formel können wir die Summe aufspalten in

$$\frac{1}{m} \left( \sum_{i=0}^{m-1} (x_{i+t})^2 - 2 \cdot \sum_{i=0}^{m-1} x_{i+t} y_i + \sum_{i=0}^{m-1} (y_i)^2 \right) \quad .$$

Die mittlere Summe erinnert an die Kreuzkorrelation von  $x$  und  $y$  im Überlappungsbereich. Im Allgemeinen ist die diskrete, reelle Kreuzkorrelation zwischen zwei Zeitreihen  $f, g \in \mathbb{R}^n$  definiert als

$$(f \star g)_t = \frac{1}{n} \sum_{i=0}^{n-1} f_i \cdot g_{(i+t) \bmod n}$$

für  $t = 0, \dots, n - 1$ .

Die Kreuzkorrelation ist eng verwandt mit der Faltung, die sich im reellen Fall nur durch den Index in  $g$  von der Kreuzkorrelation unterscheidet. Sie ist definiert als

$$(f \otimes g)_t = \frac{1}{n} \sum_{i=0}^{n-1} f_i \cdot g_{(t-i) \bmod n}$$

für  $t = 0, \dots, n - 1$ . Wie wir sofort sehen, liegt der Aufwand für die naive Auswertung dieser Summe für alle  $t$  in  $\mathcal{O}(n^2)$ . Sie kann jedoch aufgrund des Faltungssatzes, der besagt, dass

$$f \otimes g = \text{iDFT}(\text{DFT}(f) \cdot \text{DFT}(g)) \quad ,$$

in  $\mathcal{O}(n \log n)$  berechnet werden (vgl. [PTVF02]). Diesen theoretischen Laufzeitvorteil wollen wir nun in Algorithmus 4.5 in ähnlicher Weise auf die Berechnung des mittleren quadratischen Fehlers übertragen.

Die genaue Bedeutung von Faltung und Kreuzkorrelation soll hier nicht weiter vertieft werden. Verständlich aufbereitete Informationen dazu finden sich in [But05].



**Algorithmus 4.5 (Bestimmung des Offsets durch schnelle Faltung)***Eingabe:*Messreihen  $F = F_0, \dots, F_{n-1}$  und  $A = A_0, \dots, A_{m-1}$  mit  $m \leq n$ *Ausgabe:*Verschiebung von  $A$  gegen  $F$ , so dass der MQF der Überlappung minimal wird

```
1: function CONVOLUTIONOFFSET( $F, A$ )
2:   Sei  $n = \text{LENGTH}(F)$  und  $m = \text{LENGTH}(A)$ 
3:   Fülle  $A$  mit Nullen auf die Länge von  $F$  auf
4:    $\mathcal{F} = \text{FFT}(F)$ 
5:    $\mathcal{A} = \text{FFT}(A)$ 
6:    $\mathcal{A}^*_k = \overline{\mathcal{A}_k} \quad \forall k$  // punktweise komplexe Konjugation
7:    $\mathcal{H} = \mathcal{F} \cdot \mathcal{A}^*$  // punktweise Multiplikation
8:    $H = \text{INVERSEFFT}(\mathcal{H})$ 
9:    $f, a = 0$ 
10:  for  $l = 0, \dots, m - 1$  do
11:     $f = f + F_l^2$ 
12:     $a = a + A_l^2$ 
13:   $mse = \infty$ 
14:  for  $t = 0, \dots, n - m$  do
15:    if  $\frac{1}{m} \cdot (f - 2H_t + a) < mse$  then
16:      Merke aktuelles  $t$  als offset
17:      Aktualisiere  $mse$ 
18:       $f = f - F_t^2 + F_{t+m}^2$ 
19:  return (offset,  $mse$ )
```

**Laufzeit (Algorithmus 4.5)**

Wie in Abschnitt 2.6 beschrieben, liegt eine Berechnung der DFT mit einem FFT-Algorithmus (beispielsweise aus der freien FFTW-Bibliothek [FJ05]) in  $\mathcal{O}(n \log n)$ . Dies trifft sowohl auf die Fouriertransformation von  $F$  als auch auf die von  $A$  zu, da  $A$  in Zeile 3 mit Nullen auf die Länge von  $F$  aufgefüllt wird. Die Schleifen in Zeile 10 bzw. 14 liegen in  $\mathcal{O}(m)$  bzw.  $\mathcal{O}(n)$  und fallen somit in der asymptotischen Laufzeit nicht ins Gewicht. Die Gesamtlaufzeit liegt also in  $\mathcal{O}(n \log n)$ .

Die Bestimmung des Offsets mittels schneller Fouriertransformation ist asymptotisch schneller als der naive Ansatz aus Algorithmus 4.2, falls  $m > \log n$ . Dies ist in den meisten Anwendungen (so auch in unserer) der Fall. Allerdings ist

noch nicht ganz klar, dass der Algorithmus tatsächlich das gewünschte Ergebnis liefert. Dies wollen wir mit dem nächsten Lemma beweisen.

**Lemma 4.1 (Korrektheit von Algorithmus 4.5)**

*Algorithmus 4.5 gibt ein Offset von  $A$  in  $F$  zurück, so dass der MQF im Überlappungsbereich minimal wird.*

**Beweis** Das Kernstück von Algorithmus 4.5 ist Zeile 15. Hier wird der mittlere quadratische Fehler berechnet. Um zu zeigen, dass dem so ist, betrachten wir erst den Inhalt des Feldes  $H$  und anschließend den Inhalt der Variablen  $f$  und  $a$ , um dann zu zeigen, dass

$$\frac{1}{m} \cdot (f - 2H_t + a)$$

dem mittleren quadratischen Fehler im Überlappungsbereich zwischen  $F$  und  $A$  mit einem Offset von  $t$  entspricht.

Beginnen wir also mit Feld  $H$ . Laut Zeile 8 ist

$$H_t = (\text{iDFT}(\mathcal{H}))_t ,$$

was sich nach Zeile 4-7 schreiben lässt als

$$\begin{aligned} H_t &= \frac{1}{n} \sum_{j=0}^{n-1} \mathcal{F}_j \cdot \overline{\mathcal{A}_j} \cdot e^{\frac{2\pi i}{n} jt} \\ &= \frac{1}{n} \sum_{j=0}^{n-1} \left[ \sum_{k=0}^{n-1} F_k \cdot e^{-\frac{2\pi i}{n} kj} \right] \cdot \left[ \sum_{l=0}^{n-1} A_l \cdot e^{-\frac{2\pi i}{n} lj} \right] \cdot e^{\frac{2\pi i}{n} jt} . \end{aligned}$$

Nach dem Vertauschen der Summen ist

$$\begin{aligned} H_t &= \frac{1}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} F_k A_l \cdot \sum_{j=0}^{n-1} e^{-\frac{2\pi i}{n} kj} e^{\frac{2\pi i}{n} lj} e^{\frac{2\pi i}{n} jt} \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} F_k A_l \cdot \sum_{j=0}^{n-1} e^{\frac{2\pi i}{n} kj} \cdot e^{\frac{2\pi i}{n} (l+t)j} . \end{aligned}$$

Mit Lemma 2.19 können wir die innere Summe umschreiben. Da  $k \in [0, n-1]$  und somit  $k \bmod n = k$ , folgt

$$\begin{aligned} H_t &= \frac{1}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} F_k A_l \cdot n \delta_{k, (l+t) \bmod n} \\ &= n \cdot \frac{1}{n} \cdot \sum_{l=0}^{n-1} F_{(l+t) \bmod n} A_l \\ &= \sum_{l=0}^{n-1} F_{(l+t) \bmod n} A_l . \end{aligned}$$

Ein Teil der Summanden ist jedoch 0, da  $A$  in Zeile 3 mit Nullen auf die Länge von  $F$  aufgefüllt wurde, das heißt es ist  $A_i = 0$  für  $i = m, \dots, n-1$ . Somit lässt sich die Summe verkürzen zu

$$H_t = \sum_{l=0}^{m-1} F_{(l+t) \bmod n} A_l \quad .$$

Im Algorithmus werden nicht alle Elemente des Feldes  $H$  verwendet. Die Schleife in Zeile 14 läuft nur von 0 bis  $n-m$ , obwohl  $H$  die Länge  $n$  hat. In Zeile 15 gilt also  $0 \leq t \leq n-m$ . Somit ergibt sich in obiger Summe, dass

$$l+t \leq (m-1) + (n-m) = n-1$$

und somit

$$(l+t) \bmod n = l+t \quad .$$

Für den im Algorithmus genutzten Indexbereich von  $H$ , also  $t = 0, \dots, n-m$ , gilt

$$H_t = \sum_{l=0}^{m-1} F_{l+t} A_l \quad .$$

Kommen wir nun zu den Variablen  $f$  und  $a$ . Nach Ablauf der Schleife in Zeile 10 ist

$$a = \sum_{l=0}^{m-1} (A_l)^2 \quad \text{und} \quad f = \sum_{l=0}^{m-1} (F_l)^2 \quad .$$

Mit  $a$  brauchen wir uns nicht weiter zu beschäftigen, da es im Laufe des Algorithmus nicht mehr verändert wird. Es ist also nur noch zu klären, was mit  $f$  in Zeile 18 passiert.

Der Einfachheit halber wollen wir den Wert der Variablen  $f$  in Zeile 15 im Schleifendurchlauf  $t$  mit  $f_t$  bezeichnen. Aus

$$f_0 = \sum_{l=0}^{m-1} (F_l)^2$$

ergibt sich mit Zeile 18 induktiv, dass

$$f_t = \sum_{l=t}^{m-1+t} (F_l)^2 = \sum_{l=0}^{m-1} (F_{l+t})^2 \quad .$$

Diese Erkenntnisse setzen wir nun in die Formel aus Zeile 15 ein und sehen,

dass im  $t$ -ten Schleifendurchlauf

$$\begin{aligned}
 \frac{1}{m} \cdot (f - 2H_t + a) &= \frac{1}{m} \cdot (f_t - 2H_t + a) \\
 &= \frac{1}{m} \cdot \left( \sum_{l=0}^{m-1} (F_{l+t})^2 - 2 \cdot \sum_{l=0}^{m-1} F_{l+t} A_l + \sum_{l=0}^{m-1} (A_l)^2 \right) \\
 &= \frac{1}{m} \cdot \sum_{l=0}^{m-1} ((F_{l+t})^2 - 2 \cdot F_{l+t} A_l + (A_l)^2) \\
 &= \frac{1}{m} \cdot \sum_{l=0}^{m-1} (F_{l+t} - A_l)^2
 \end{aligned}$$

ist. Dies entspricht genau dem mittleren quadratischen Fehler im Überlappungsbereich zwischen  $F$  und  $A$  mit einem Offset von  $t$ .

Da wir uns im Laufe der Schleife den kleinsten MQF und das zugehörige Offset merken, ist die Korrektheit des Algorithmus gezeigt.  $\square$

Die meisten Bibliotheken für FFT-Algorithmen berechnen die inverse DFT ohne den Faktor  $\frac{1}{n}$  aus Definition 2.15. Wenn solch eine Bibliothek verwendet wird, muss der Faktor in der Formel aus Zeile 15 noch hinzugefügt werden. Sie lautet dann also

$$\frac{1}{m} \cdot \left( f - 2 \cdot \frac{H_t}{n} + a \right) .$$

Wir haben nun einen Algorithmus gefunden, dessen theoretische Laufzeit die bisher vorgestellten Algorithmen übertrifft und der nichtsdestotrotz das optimale Alignment bezüglich des MQF liefert. Es ist jedoch noch offen, ob sich dieser theoretische Laufzeitvorteil auch in der Praxis merklich auswirkt. Diesen Aspekt werden wir in Abschnitt 4.2.6 untersuchen.

Vorher wollen wir noch klären, ob sich der gleiche Ansatz auch für die schnelle Berechnung der anderen Gütekriterien, also den Korrelations- und Gleichläufigkeitskoeffizienten, verwenden lässt.

### Analoge Berechnung des Korrelationskoeffizienten

Der Korrelationskoeffizient im Überlappungsbereich von  $F$  und  $A$  mit einem Offset von  $t$  ist

$$\text{Kor}(F, A) = \frac{\frac{1}{m} \sum_{i=0}^{m-1} (F_{i+t} - \bar{F}_{[t,t+m]})(A_i - \bar{A})}{\sqrt{\frac{1}{m} \sum_{i=0}^{m-1} (F_{i+t} - \bar{F}_{[t,t+m]})^2} \cdot \sqrt{\frac{1}{m} \sum_{i=0}^{m-1} (A_i - \bar{A})^2}} , \quad (4.1)$$

wobei  $\bar{F}_{[t,t+m]}$  der Mittelwert von  $F$  im Indexbereich  $t, \dots, t+m$  ist. Wenn wir  $F$  und  $A$  vor Eingabe in Algorithmus 4.5 um ihren Mittelwert bereinigen, also  $F = F - \bar{F}$  und  $A = A - \bar{A}$  setzen, lässt sich Gleichung (4.1) im Wesentlichen schreiben als

$$\frac{\frac{1}{m}H_t}{\sqrt{\frac{1}{m}f_t} \cdot \sqrt{\frac{1}{m}a}} = \frac{H_t}{\sqrt{f_t} \cdot a}$$

mit den Bezeichnungen aus dem Beweis von Lemma 4.1. Diese Formel können wir in Zeile 15 maximieren statt den MQF zu minimieren, da die notwendigen Werte  $H_t, f_t$  und  $a$  bereits berechnet sind.

Zu beachten ist jedoch die Einschränkung „im Wesentlichen“ im letzten Absatz. Wenn wir obige Formel nutzen, um den Korrelationskoeffizienten zu berechnen, gehen wir davon aus, dass  $\bar{F}_{[t,t+m]} = \bar{F}$  ist, was natürlich nicht korrekt ist. In der Praxis unterscheiden sich die stückweisen Mittelwerte von  $F$  bei ausreichender Länge des Überlappungsbereiches jedoch nur unwesentlich von dem gesamten Mittelwert, da Stahlbänder mit sehr hoher Genauigkeit gewalzt werden.

Zusammenfassend können wir feststellen, dass diese Berechnung zwar nicht ganz korrekt ist, aber der Fehler sich nicht merklich auf unsere Anwendung auswirkt.

### Analoge Berechnung des Gleichläufigkeitskoeffizienten

Auch bei der Berechnung des Gleichläufigkeitskoeffizienten müssen wir einen kleinen Kompromiss eingehen, der sich aber wiederum nicht auf die Praxis auswirkt. Dieser Kompromiss besteht darin, dass wir Abschnitte, in denen beide Messkurven konstant sind (also keine Steigung haben), in der Berechnung des GLK als gegenläufig betrachten.

In der Praxis kommt eine Folge von exakt gleichen Messwerten nur dann vor, wenn das Messgerät auf eine niedrige Auflösung eingestellt ist und die Messkurve dadurch stufig wird, dass Punkte interpoliert werden. Mehrere *echte* gleiche Messwerte in Folge gibt es aufgrund der hohen Genauigkeit der Messgeräte nicht. Wenn wir also zwei stufige Kurven vergleichen wollen, können wir beide Kurven so interpolieren, dass die Auflösung der höheren der beiden Einzelaufösungen entspricht. Dies führt dazu, dass eine der beiden Kurven keine konstanten Abschnitte mehr besitzt.

Somit gibt es also insbesondere auch keine Verschiebung, so dass Abschnitte existieren, in denen beide Messkurven „gleichzeitig“ konstant sind. Dadurch fällt der Kompromiss, den wir nun eingehen, nicht ins Gewicht. Gehen wir also davon aus, dass diese Abschnitte nicht existieren.

Es ist

$$\text{GLK}(F, A) = \frac{1}{m-1} \cdot \sum_{i=0}^{m-2} \chi(F'_{i+t} = A'_i) \quad (4.2)$$

der Gleichläufigkeitskoeffizient im Überlappungsbereich von  $F$  und  $A$  mit einem Offset von  $t$ . Hierbei sind  $F'$  und  $A'$  definiert durch

$$F'_i := \text{sign}(F_{i+1} - F_i) \quad \text{und} \quad A'_i := \text{sign}(A_{i+1} - A_i) \quad .$$

Nach unserer Annahme ist o.B.d.A.  $F'_i \neq 0$  für  $i = 0, \dots, n-1$ . Betrachten wir die Summe

$$\frac{1}{m-1} \sum_{i=0}^{m-2} \frac{(F'_{i+t} + A'_i)^2}{4} \quad , \quad (4.3)$$

so sehen wir, dass diese fast dem GLK entspricht. Da  $F', A' \in \{-1, 0, 1\}^{n-1}$ , gilt

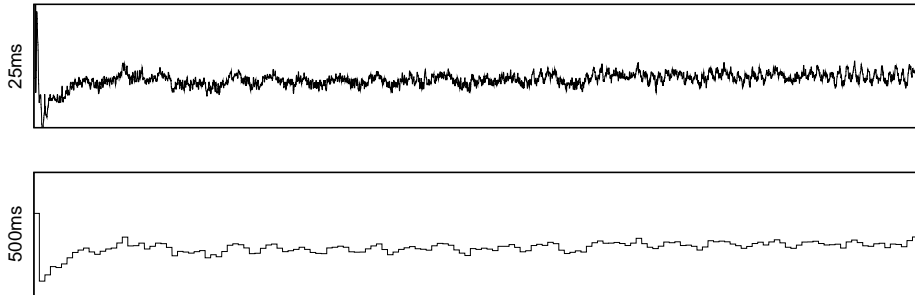
$$\begin{aligned} F'_{i+t} \neq A'_i \text{ und keines von beiden } 0 &\Leftrightarrow \frac{(F'_{i+t} + A'_i)^2}{4} = 0 \\ F'_{i+t} \neq A'_i \text{ und eines von beiden } 0 &\Leftrightarrow \frac{(F'_{i+t} + A'_i)^2}{4} = \frac{1}{4} \\ F'_{i+t} = A'_i \text{ und } F'_{i+t} \neq 0 &\Leftrightarrow \frac{(F'_{i+t} + A'_i)^2}{4} = 1 \quad . \end{aligned}$$

Wir machen also mit der Summe aus Gleichung (4.3) insofern einen Fehler, dass wir Abschnitte, in denen  $A$  konstant ist, zu einem Viertel als gleichläufig betrachten.

In der eigentlichen Definition des GLK in Gleichung (4.2) werden Abschnitte, in denen eine der beiden Messkurven konstant ist und die andere nicht, als nicht gleichläufig betrachtet. Wie bereits erwähnt, entstehen konstante Abschnitte in den Messkurven im Allgemeinen durch eine zu niedrige Auflösung der Messgeräte.

Falls das Messsystem, das die Messwerte verarbeitet, auf eine höhere Auflösung als das Messgerät selbst eingestellt ist, so muss das Messsystem Werte interpolieren, um die ihm vorgegebene Auflösung einzuhalten. Zur Interpolation wird hier häufig der letzte gemessene Wert verwendet. Eine lineare Interpolation kommt beispielsweise nicht in Frage, da dem Messsystem zum Interpolationszeitpunkt der nächste Messwert noch nicht bekannt ist. Aus dieser Konfiguration entsteht eine Messkurve, wie sie in Abbildung 4.2 übertrieben dargestellt ist. In diesem Beispiel wäre die Auflösung des Messsystems  $5 \text{ ms}$  und die Auflösung des Messgerätes  $25 \text{ ms}$  (oben) bzw.  $500 \text{ ms}$  (unten).

Weiterhin sehen wir an diesem Beispiel, dass es wenig Sinn macht, konstante Abschnitte als nicht gleichläufig mit steigenden oder fallenden Abschnitten zu



**Abbildung 4.2:** Dickenschrieb mit einer Auflösung von 25 ms (oben) und einer Auflösung von 500 ms (unten). Beide auf einem 5 ms-Zeitgitter.

betrachten, da wir über diese Abschnitte nichts wissen. Sie sind lediglich vom Messsystem interpoliert. Somit weichen wir nun von der ursprünglichen Definition des Gleichläufigkeitskoeffizienten ab und wollen versuchen, die konstanten Abschnitte als gleichläufig mit fallenden *und* steigenden Abschnitten zu bewerten. Für jeden Punkt in solch einem Abschnitt zählen wir drei Viertel zu wenig zu der Summe, da wir ihn nur zu einem Viertel als gleichläufig betrachtet haben. Mit der ersten binomischen Formel können wir unseren modifizierten GLK als

$$\frac{1}{4(m-1)} \left( \sum_{i=0}^{m-2} (F'_{i+t})^2 + 2 \cdot \sum_{i=0}^{m-2} F'_{i+t} A'_i + \sum_{i=0}^{m-2} (A'_i)^2 \right)$$

schreiben. Da  $A'_i \in \{-1, 0, 1\}$ , ist

$$(m-1) - \sum_{i=0}^{m-2} (A'_i)^2$$

genau die Anzahl der Nullen in  $A'$ . Das bedeutet, Gleichung (4.3) sollte

$$\begin{aligned} & \frac{1}{m-1} \left( \sum_{i=0}^{m-2} \frac{(F'_{i+t} + A'_i)^2}{4} + \frac{3}{4} \cdot (m-1 - \sum_{i=0}^{m-2} (A'_i)^2) \right) \\ &= \frac{1}{4(m-1)} \left( \sum_{i=0}^{m-2} (F'_{i+t})^2 + 2 \cdot \sum_{i=0}^{m-2} F'_{i+t} A'_i - 2 \cdot \sum_{i=0}^{m-2} (A'_i)^2 \right) + \frac{3}{4} \end{aligned} \quad (4.4)$$

lauten, wenn wir konstante Abschnitte in einer Kurve als gleichläufig betrachten wollen.

Wir können nun also ein Gütekriterium  $GLK'$  mit Formel (4.4) definieren, das mit der genannten Modifikation weitgehend dem  $GLK$  entspricht und diesen sogar noch geeigneter für unsere Anwendung macht.

Wenn wir als Eingabe für Algorithmus 4.5 nun  $F'$  und  $A'$  statt  $F$  und  $A$  verwen-

den<sup>2</sup>, können wir  $GLK'$  mit den Bezeichnungen aus dem Beweis von Lemma 4.1 schreiben als

$$GLK'(F, A) = \frac{1}{4(m-1)} (f_t + 2H_t - 2a) + \frac{3}{4}$$

Mit dieser Formel können wir in Zeile 15 des Algorithmus den leicht modifizierten Gleichläufigkeitskoeffizienten maximieren statt den mittleren quadratischen Fehler zu minimieren.

### Aufhebung der Offset-Einschränkung

Wie am Anfang des Kapitels bereits erwähnt wurde, ist die Einschränkung, dass sich das Kaltband komplett im Warmband wiederfinden lässt, künstlich auferlegt, um die Pseudocodes übersichtlicher zu halten. Wir wollen kurz beispielhaft skizzieren, wie man diese Einschränkung in Algorithmus 4.5 aufhebt, da dies dort nicht vollkommen trivial ist.

Wir wollen einer Empfehlung aus [But05] nachkommen, die lautet „Viele Nullen tun gut!“. Um zum Beispiel negative Offsets zuzulassen, was bedeutet, dass der Beginn der Alignseries vor den Anfang der Fixedseries geschoben werden darf, müssen wir ein sogenanntes Zero-padding durchführen (vgl. auch [But05]).

Nehmen wir an, wir möchten zulassen, dass das Kaltband 5 Messpunkte vor der Warmbandmessung beginnt, was natürlich in der Praxis Unsinn ist, so müssen wir in Algorithmus 4.5 direkt vor Zeile 3 dem Feld  $F$  fünf Nullen am Anfang hinzufügen. Der erste Messwert der Fixedseries findet sich nun also in  $F_5$ . Dafür können wir später im Feld  $H_0$  die Korrelationssumme für ein Offset von  $-5$  abrufen.

Nun müssen wir noch beachten, dass es 5 Offsets gibt, in denen in Zeile 15 nicht die komplette quadratische Summe über das Feld  $A$  zur Berechnung des mittleren quadratischen Fehler herangezogen werden darf. Bei einem Offset von  $-5$  gehen beispielsweise die ersten 5 Punkte der Alignseries nicht mit in die Fehlerberechnung ein. Die quadratische Summe über  $A$  muss also in diesem Fall analog zur quadratischen Summe über  $F$  berechnet werden, indem gegebenenfalls in der Schleife in Zeile 14 ein Wert zu der Summe addiert oder, falls die Alignseries hinter die Fixedseries geschoben wird, subtrahiert wird.

Analog müssen Nullen an das Ende der Fixedseries angefügt werden, falls die Alignseries auch hinter der Fixedseries aufhören kann. Bei dieser Implemen-

---

<sup>2</sup>Vorsicht:  $A'$  darf nicht durch lineare Interpolation skaliert werden, sonder *muss* als Treppenfunktion interpoliert werden!



tierung muss die Anzahl an Punkten, die die Alignseries über die Fixedseries hinausragen darf (der sogenannte Overlap), in jedem Fall vorgegeben werden.

In unserer Anwendung kann solch ein Overlap ausschließlich aufgrund fehlerhafter Geschwindigkeitssignale oder Dickenprofile, die ungenau aus dem Messstrom ausgeschnitten wurden, entstehen (vgl. Kapitel 3).

### 4.2.3 Kombination mit Skalierungssuche

In den Abschnitten 4.2.1 und 4.2.2 haben wir lediglich Beschleunigungen der Offsetbestimmung betrachtet. Wenn wir die beiden vorgestellten Algorithmen nun in Algorithmus 4.3 verwenden wollen, um zusätzlich noch einen Skalierungsfaktor zu bestimmen, so finden einige unnötige Mehrfachberechnungen statt, die hier kurz erwähnt werden sollen.

#### Fehlerberechnung mittels FFT

In Algorithmus 4.3 wird die Offsetbestimmung für jede mögliche Länge der Messreihe  $A$  durchgeführt. Die Messreihe  $F$  verändert sich durch diese Iteration nicht. Wenn wir also Algorithmus 4.5 in Algorithmus 4.3 verwenden, so müssen wir die Fouriertransformierte von  $F$  nicht in jeder Iteration von 4.3 neu berechnen, sondern nur einmal zu Beginn des Algorithmus.

#### Fehlerberechnung mit oberer Grenze

Wenn wir Algorithmus 4.4 zur Offsetbestimmung in Algorithmus 4.3 verwenden, so sollten wir diesem als obere Grenze für die Fehlerberechnung nicht nur den kleinsten gefundenen Fehler innerhalb der Offsetbestimmung übergeben, sondern den global kleinsten Fehler, den wir bis zu der jeweiligen Iteration der Schleife über die Skalierungsfaktoren gefunden haben.

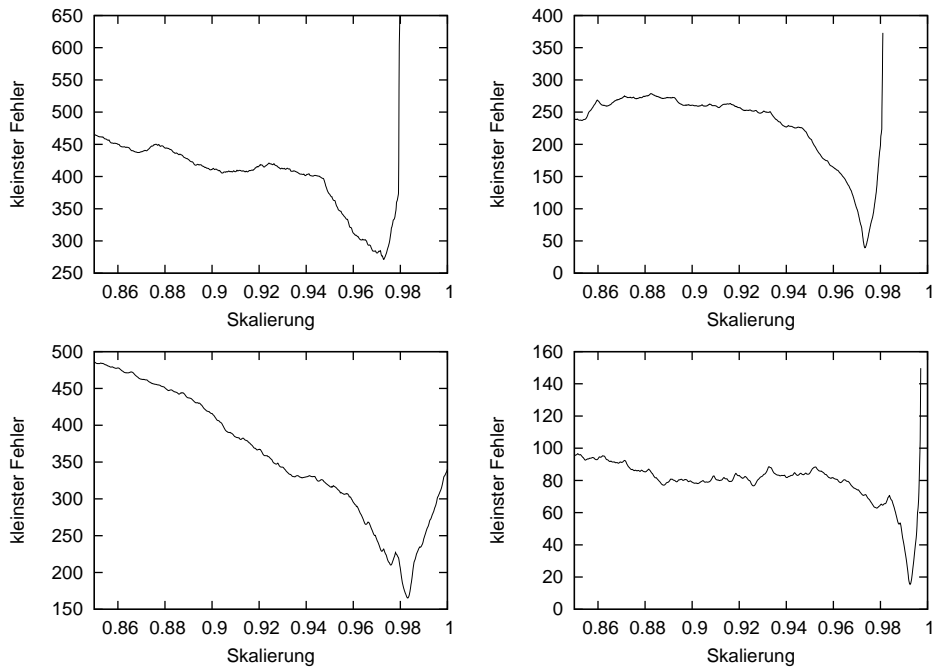
In den folgenden Pseudocodes werden wir die Doppelberechnungen ignorieren, um die Algorithmen übersichtlicher zu halten.

### 4.2.4 Skalierungssuche durch Gitterheuristik

Bis hierher haben wir nur Beschleunigungsansätze betrachtet, die ein beweisbar optimales Ergebnis für das Alignment bezüglich eines Gütekriteriums zugelassen haben. Wir wollen nun auch die Suche des Skalierungsfaktors drastisch

beschleunigen, verlieren dabei allerdings die Garantie auf ein optimales Ergebnis.

In Abbildung 4.3 sehen wir am Beispiel von vier Messreihen den kleinsten Fehler, der für einen festen Skalierungsfaktor durch Verschieben der Messreihen zu erlangen ist, aufgetragen. Es fällt bei allen Kurven auf, dass das globale Minimum eindeutig ist bzw. deutlich sichtbaren Senke liegt. Das globale Minimum ist also, wie zu erwarten, kein Peak, sondern die Kurve nähert sich diesem mehr oder weniger langsam an.



**Abbildung 4.3:** Skalierungssuche bei vier verschiedenen Dickenschreiben von jeweils Kalt- und Warmbandmessung. Für jeden Skalierungsfaktor ist der kleinste erreichbare Fehler aufgetragen.

Algorithmus 4.3 tastet mit der Schleife in Zeile 3 jeden einzelnen Punkt dieser Kurve ab. Wir wollen den Algorithmus nun so modifizieren, dass die Fehler für verschiedene Skalierungen im ersten Schritt nur auf einem groben Gitter ausgewertet werden und dann in der Nachbarschaft des besten Treffers das Gitter verfeinert wird. Mit Nachbarschaft bezeichnen wir hier den Bereich um den besten Treffer, der noch nicht abgetastet wurde.

#### Algorithmus 4.6 (Alignment mit Gitterheuristik)

*Eingabe:*

- Messreihen  $F = F_0, \dots, F_{n-1}$  und  $A = A_0, \dots, A_{m-1}$
- Intervall  $[from, to]$  mit  $to \cdot m \leq n$ , in dem die Skalierung gesucht wird

- Schrittweite  $step$  für die Schleife in Zeile 3

*Ausgabe:*

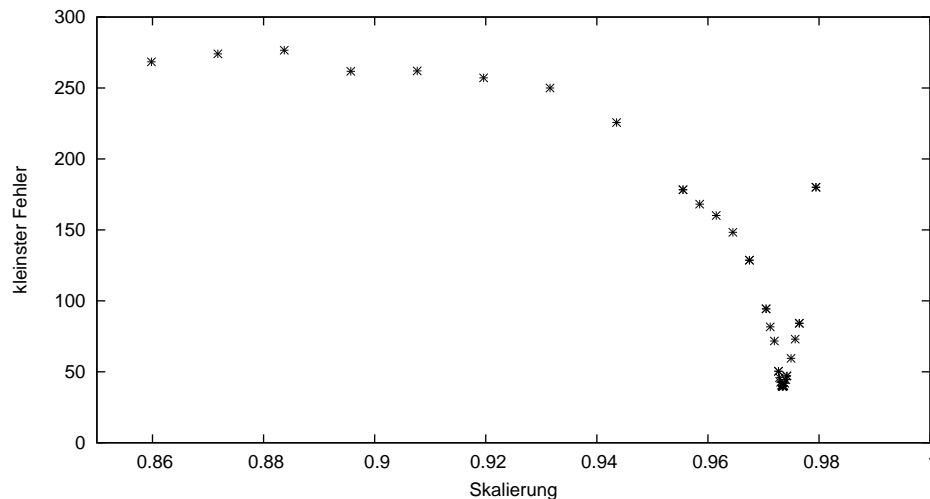
Verschiebung und Skalierung von  $A$  gegen  $F$ , so dass der MQF der Überlappung auf dem abgetasteten Gitter minimal wird.

```

1: function ALIGNGRID( $F, A, from, to, step$ )
2:    $mse = \infty$ 
3:   for  $i = from \cdot m, from \cdot m + 1 \cdot step, from \cdot m + 2 \cdot step, \dots, to \cdot m$  do
4:     Skaliere  $A$  so, dass es Länge  $i$  hat
5:      $result = \text{CALCULATEOFFSET}(F, S)$  // z.B. Algorithmus 4.5
6:     if  $result.mse < mse$  then
7:       Merke  $offset$  und  $scale$ 
8:       Aktualisiere  $mse$ 
9:   if  $step > 1$  then
10:    return ALIGNGRID( $F, A, scale - \frac{step}{m}, scale + \frac{step}{m}, step \text{ div } 4$ )
11:  else
12:    return ( $offset, scale, mse$ )

```

Die Verfeinerung der Schrittweite zur Skalierungssuche in Zeile 10 ist natürlich willkürlich und kann prinzipiell frei gewählt werden. Eine Division der alten Schrittweite durch 4 hat sich jedoch als vernünftige Wahl bewährt.



**Abbildung 4.4:** Fehlerauswertungen der Skalierungssuche mit Gitterheuristik am Beispiel der Messreihe rechts oben aus Abbildung 4.3

In Abbildung 4.4 sehen wir, wie das Auswertungsgitter um das globale Minimum herum immer feiner wird. Für die Abbildung wurde als Startschrittweite 128 gewählt. In diesem Beispiel wurden nur 48 statt 1418 Auswertungen benötigt,

um *exakt* dasselbe Ergebnis zu erhalten.

Selbstverständlich ist es möglich mit diesem Verfahren ein vollkommen falsches Alignment zu erhalten, falls das globale Minimum nicht in der Nachbarschaft des besten Treffers bei großer Schrittweite ist. Bei der Wahl der Startschrittweite muss deshalb ein Kompromiss zwischen Geschwindigkeit und Verlässlichkeit gemacht werden, denn je kleiner die Schrittweite ist, desto geringer ist die Wahrscheinlichkeit, dass das Gitter an der falschen Stelle verfeinert wird.

Wenn das Alignment automatisch und ohne visuelle Kontrolle durchgeführt wird, sollte die Schrittweite kleiner gewählt werden, um die Verlässlichkeit zu gewährleisten. In vielen Anwendungsfällen sitzt der Benutzer jedoch vor dem Computer und sieht sich das Ergebnis des Alignments an. Hier kann die Schrittweite vom Benutzer zunächst sehr groß gewählt werden und bei offensichtlichem Misserfolg verkleinert werden.

Im Allgemeinen lässt sich sagen, dass eine Startschrittweite  $< 100$  bei den vorliegenden Daten unkritisch ist und in allen bisher geprüften Fällen zum selben Ergebnis wie ein Alignment mit Startschrittweite 1 geführt hat (siehe auch Abschnitt 4.2.6).

### 4.2.5 Komprimierung der Daten

Es gibt Anwendungsfälle, in denen ein ungenaues Alignment genügt, um die jeweils gewünschte Aussage zu treffen (siehe Kapitel 6). In diesen speziellen Anwendungen können die vorliegenden Messreihen komprimiert werden, um das Alignment zu beschleunigen. Wie wir in Kapitel 6 sehen werden, eignet sich die in Abschnitt 2.5 vorgestellte PAA zur Komprimierung.

Bereits in Abbildung 2.5 haben wir gesehen, dass die Charakteristik der Messreihen trotz der Komprimierung erhalten bleibt. Es ist klar, dass durch die Kompression eine deutliche Laufzeitersparnis entsteht, denn statt  $n$  Messpunkten müssen nach einer Kompression mit Faktor  $p$  nur noch  $\left\lceil \frac{n}{p} \right\rceil$  Messpunkte verarbeitet werden.

### 4.2.6 Fazit

Wir haben in diesem Abschnitt Ansätze zur Beschleunigung des naiven Alignmentalgorithmus kennengelernt und uns den theoretischen Laufzeitgewinn angesehen. Dieser sagt jedoch nichts über den Laufzeitgewinn in der Praxis aus. Deswegen wollen wir uns abschließend die absoluten Laufzeiten der verschiedenen Algorithmen ansehen und diese miteinander vergleichen. Die Algorithmen

wurden mit den Verbesserungen aus Abschnitt 4.2.3 in der Programmiersprache Java implementiert.

Alle Zeitangaben in Tabelle 4.1 sind gemittelt über 23 Alignments bei einer Skalierungssuche im Intervall  $[0, 7; 1, 1]$  und einer Anzahl von etwa 9.000 Messpunkten pro Coil. Sie sollen keine Aussage über die bestmögliche Geschwindigkeit der Algorithmen liefern, sondern lediglich einen Vergleich der Beschleunigungen untereinander ermöglichen.<sup>3</sup>

Algorithmus	∅ Dauer
Naives Alignment (Alg. 4.3)	ca. 8 Minuten
Abbruch der Fehlerberechnung (Alg. 4.3 mit Alg. 4.4)	ca. $6\frac{1}{2}$ Minuten
Fehlerberechnung mittels FFT (Alg. 4.3 mit Alg. 4.5)	ca. 42 Sekunden

***Tabelle 4.1:** Absolute Laufzeiten der Alignmentverfahren*

Wie wir in der Tabelle sehen, bringt der frühzeitige Abbruch der Fehlerberechnung einen gewissen Laufzeitvorteil gegenüber dem naiven Alignment. Dieser Laufzeitvorteil ist jedoch geringfügig im Vergleich zur Laufzeit des Alignments mit Fehlerberechnung durch FFT. Da die Laufzeit des Alignmentalgorithmus mit frühzeitigem Abbruch der Fehlerberechnung enorm von den vorliegenden Daten abhängt, gibt es viele Messreihen, bei denen praktisch kein Laufzeitgewinn im Vergleich zum naiven Alignment vorhanden ist und einige bei denen der Laufzeitgewinn größer ist, als obiger Durchschnittswert vermuten lässt.

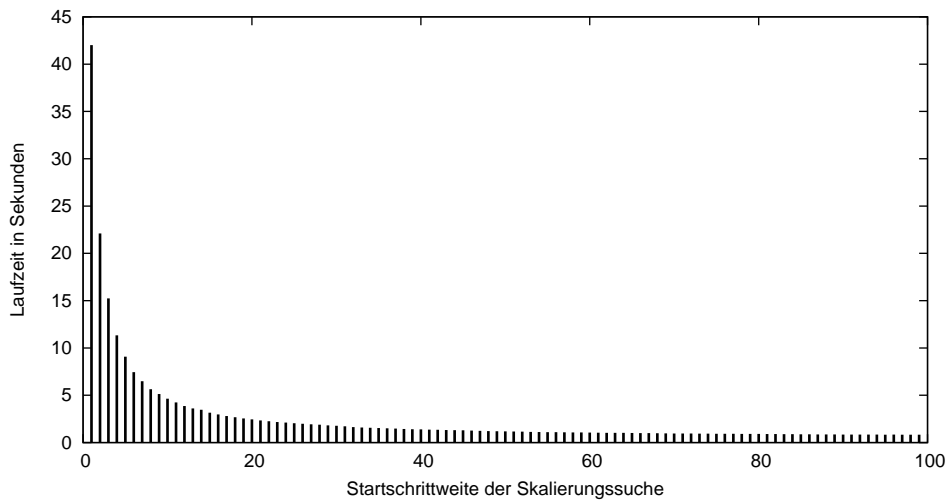
Aufgrund der vorliegenden Ergebnisse betrachten wir im Folgenden das naive Alignment 4.3 mit der Fehlerberechnung durch FFT aus Algorithmus 4.5 als Alignmentalgorithmus der Wahl und werden nur dessen Laufzeiten in Kombination mit der Gitterheuristik aus Abschnitt 4.2.4 betrachten.

In Abbildung 4.5 sehen wir die absoluten Laufzeiten von Algorithmus 4.6 mit der Fehlerberechnung durch FFT für alle Startschrittweiten zwischen 1 und 100. Die Laufzeiten wurden wiederum über 23 Alignments von verschiedenen Messreihen gemittelt. In dieser Versuchsreihe wurde eine gemittelte Laufzeit von weniger als einer Sekunde ab einer Startschrittweite von 67 erreicht.

Der Laufzeitgewinn des Algorithmus ist unwichtig, wenn dadurch das Ergebnis verfälscht wird. Deshalb wurde während der Auswertungen überprüft, ob für jede Startschrittweite exakt dasselbe Offset und dieselbe Skalierung wie beim Alignment mit Startschrittweite 1 berechnet wird. Dies war für jedes der 23 Alignments bei allen Startschrittweiten der Fall.

---

<sup>3</sup> Referenzrechner: Lenovo ThinkPad T60; Intel Core 2 1,83GHz; 3,0GB RAM  
Implementierung: Java



**Abbildung 4.5:** Absolute Laufzeit von Algorithmus 4.6 mit Offsetbestimmung mittels FFT für verschiedene Startschrittweiten der Gitterheuristik

Das erste falsche, also vom Originalalignment abweichende, Alignment wurde in dieser Versuchsreihe mit einer Startschrittweite von 294 berechnet. Um das Risiko eines falschen Alignments nahezu auszuschließen, hat sich eine Startschrittweite von weniger als 1% der Sequenzlänge bewährt. Da die Startschrittweite nicht direkt von der Länge der Messreihe abhängt, ist dies jedoch nur als Faustregel zu verstehen. Bei den vorliegenden Daten bedeutet das, dass die Startschrittweite  $< 90$  gewählt werden sollte.

Die Ergebnisse legen nahe, dass eine Kombination aus Algorithmus 4.6 mit 4.5 am praktikabelsten ist. Wir werden diese Empfehlung in Kapitel 6 untermauern, wo dieses Alignmentverfahren zur Zuordnung von Dickenschrieben verwendet wird.

### 4.3 Shotgunverfahren mit linearer Regression

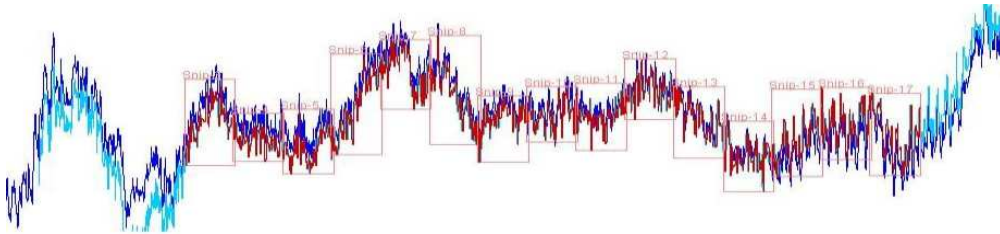
Neben dem beschleunigten naiven Alignment gibt es noch weitere Ansätze Stahlband-Fingerprints auszurichten. Das Shotgunverfahren mit linearer Regression wurde inspiriert vom Shotgunverfahren zum Alignment von DNA-Sequenzen und der Tatsache, dass in [Gno06] festgestellt wurde, dass ein relativ kleiner Abschnitt eines Stahlband-Dickenprofils reicht, um das Stahlband mit hoher Sicherheit zu identifizieren.

Das Shotgunverfahren arbeitet schnipselbasiert, was bedeutet, dass ein Dickenprofil (die Alignseries) in kleine Schnipsel unterteilt und aus den Einzelalign-

ments dieser Schnipsel ein globales Alignment berechnet wird.

Dies geschieht, indem eine Schnipsellänge von beispielsweise 500 Messpunkten gewählt wird. Anschließend wird die Alignseries in Schnipsel dieser Größe geteilt. Mit Hilfe eines Offset-Aligners, also einem Alignmentverfahren, das nur eine Verschiebung und keine Skalierung bestimmt, wird dann die Position jedes einzelnen Schnipsels in der Fixedseries bestimmt. Da die Schnipsel relativ kurz sind, fällt die horizontale Skalierung nicht ins Gewicht.

Mit den Positionen der einzelnen Schnipsel lässt sich nun mit wenig Rechenaufwand ein globales Alignment berechnen, indem eine Regressionsgerade durch die gefundenen Schnipselpositionen bestimmt wird. Die Steigung der Regressionsgerade entspricht hierbei dem gesuchten Skalierungsfaktor, während der y-Achsenabschnitt das Offset wiedergibt.



**Abbildung 4.6:** *Stahlband-Dickenprofile, die mit dem Shotgunverfahren ausgerichtet wurden. Die verwendeten Schnipsel und deren gefundene Positionen sind jeweils markiert.*

Die Schnipsel neigen dazu sich gegenseitig zu überlappen, so dass durch die Regressionsanalyse ein Skalierungsfaktor  $< 1$  für das Kaltband berechnet wird. Für einen größeren Skalierungsfaktor müsste jeder Schnipsel einen Abstand zu seinem Vorgänger aufweisen.

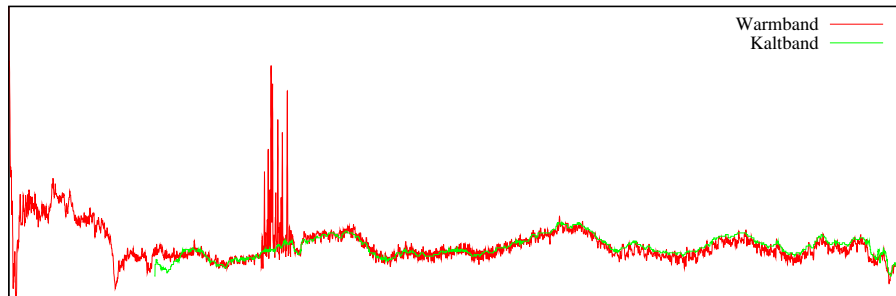
In der Praxis kommt es vor, dass Schnipsel falsch positioniert werden, da sich beispielsweise an der Stelle des Schnipsels eine Messstörung befindet. Modifikationen, die solche Falschpositionierungen bzw. deren Auswirkungen mindern, werden in [GGK<sup>+</sup>08] untersucht. Uns soll es an dieser Stelle genügen, die Idee des Verfahrens zu kennen, um sie mit der Herangehensweise aus Abschnitt 4.2 zu vergleichen.

## 4.4 Vergleich der Ansätze

Vergleicht man die Verfahren aus Abschnitt 4.2, die alle auf dem naiven Alignment beruhen, mit dem Shotgunverfahren, so fällt sofort auf, dass das Shot-

gunverfahren kein global optimales Alignment liefert. Das heißt, die einzelnen Schnipsel werden zwar möglichst optimal positioniert, es kann jedoch nicht garantiert werden, dass diese Positionierung auch für die Ausrichtung der kompletten Messreihen optimal ist.

Tatsächlich sind die Alignmentergebnisse des Shotgunverfahrens jedoch sehr ähnlich zu denen, die global optimal sind, solange die Daten keine großen Messstörungen enthalten. Falls Störungen auftreten, werden einige Schnipsel falsch positioniert, was das Gesamtergebnis stark beeinflusst und sogar komplett verfälschen kann. Die Verfahren, die auf der naiven Herangehensweise basieren, verhalten sich bei Messstörungen zwar prinzipiell ähnlich, jedoch sind diese Verfahren etwas toleranter, da eine lokale Störung durch die globale Passgenauigkeit des Alignments ausgeglichen werden kann.



**Abbildung 4.7:** Alignment mit einer etwa 40 m breiten Messstörung in der Warmbandmessung.

In Abbildung 4.7 sehen wir zwei ausgerichtete Dickenprofile, wobei die Warmbandmessung eine ca. 40 m breite Messstörung hat. Diese kann zum Beispiel durch Ablagerungen (Zunder oder Ähnliches) auf dem Warmband entstanden sein. Die Breite der Störung entspricht in etwa der Schnipsellänge eines Schnipsels beim Shotgunverfahren, wodurch dieses erhebliche Probleme hat, ein Alignment zu finden, das die beiden Kurven zur Deckung bringt. Die Ausrichtung auf Basis des naiven Algorithmus, die auch für die Abbildung genutzt wurde, funktioniert jedoch, da der Rest der Kurve so gleichläufig ist, dass die Störung im Gütekriterium kompensiert werden kann.

Das Shotgunverfahren kann so modifiziert werden, dass es schlecht passende Schnipsel nicht in die Berechnung des globalen Alignments einbezieht. Dann kommt es jedoch vor, dass sehr viele Schnipsel aus der Berechnung ausgeschlossen werden und sich das Alignment nur auf wenige Daten stützt. Das Verfahren wird also instabiler.

Gegenüber dem naiven Alignmentverfahren 4.3, das in  $\mathcal{O}(nm^2)$  arbeitet, hat das



Shotgunverfahren eine deutlich bessere Laufzeit. Um ein globales Alignment zu bestimmen, werden lediglich  $\approx \frac{10000}{500} = 20$  lokale Offsetbestimmungen benötigt, was mit Algorithmus 4.5 in  $\mathcal{O}(n \log n)$  liegt. Die Regressionsanalyse benötigt nur noch einen vernachlässigbaren Mehraufwand, der abhängig von der Anzahl der Schnipsel und somit auch von  $n$  ist. Obwohl die Anzahl der Schnipsel und somit die Anzahl der benötigten Offsetbestimmungen genau genommen auch von der Länge der Messreihen, also von  $n$ , abhängt, dürfte die asymptotische Laufzeit des Shotgunverfahrens die hier vorgestellten Alignmentalgorithmen übertreffen.

Betrachtet man die absolute Laufzeit, so benötigt das Shotgunverfahren für ein Alignment nur wenige Sekunden und ist deutlich schneller als die Verfahren aus den letzten Abschnitten, die eine Optimalität des Ergebnisses garantieren. Es benötigt jedoch länger zur Berechnung des Alignments als die Verfahren, die mit einer Skalierungssuche auf einem groben Gitter arbeiten.

In der Praxis sind die Verfahren mit Gitterheuristik vor allem robuster, aber auch schneller, als das Shotgunverfahren. In [GGK<sup>+</sup>08] werden einige Modifikationen des Shotgunverfahrens untersucht, die zu einer höheren Robustheit führen, aber auch die Laufzeit negativ beeinflussen.

Der naive Ansatz ist vor allem für den Anwendungsfall in Kapitel 6, der Zuordnung von Dickenschrieben, deutlich besser geeignet als der Schnipselansatz des Shotgunverfahrens. Mit dem Ansatz des Shotgunverfahrens ist es jedoch prinzipiell möglich, einzelne Schnipsel aus der Betrachtung auszuschließen und somit Alignments zwischen Bändern zu erlauben, deren Dickenschriebe Lücken aufweisen.

Denkbar ist solch ein Fall, falls aus einem Warmband nach genauerer Untersuchung eine sogenannte Fehlerstrecke geschnitten wird. Hier wird das Band nach dem Heraustrennen eines Teils gegebenenfalls wieder zusammenschweißt, so dass dem im Kaltwalzwerk angefertigten Dickenschrieb ein Abschnitt mitten im Band fehlt.



## Kapitel 5

# Berechnung von Kopf- und Fußschrott

Eine mögliche Anwendung der vorgestellten Alignmentverfahren ist die Berechnung der Länge von Kopf- und Fußschrott zwischen Warm- und Kaltband. In Abschnitt 1.2 wurde bereits erwähnt, dass das Stahlband aufgrund der durch das Walzen deformierten Randbereiche nach dem Warmwalzen um einige Meter beschnitten wird. Dies ist jedoch nicht der einzige Grund, weshalb es zu einer Abschrottung der Randbereiche kommen kann. Aufgrund von Transportschäden, die vorrangig in den äußeren Windungen, also dem Kopf des Kaltbandes, eines Coils auftreten, ist es auch möglich, dass hier deutlich mehr als nur einige Meter abgeschrottet werden (vgl. Abbildung 5.1).



*Abbildung 5.1: Ein warmgewalztes Stahlcoil mit deformierten Außenwindungen aufgrund eines Transportschadens [vSGa]*

Wenn nun ein Dickenprofil des Stahlbandes vor der Abschrottung und ein Dickenprofil nach der Abschrottung existiert, so sollte sich die Länge des abgeschrotteten Materials mit einem Alignment der beiden Dickenprofile berechnen lassen. Diese beiden Messungen entsprechen in der Praxis der Dickenmessung am Ausgang des Warmfertigstraße und der Dickenmessung am Eingang des Kaltwalzwerks.

## 5.1 Datenbasis

Um die Genauigkeit der Kopf- und Fußschrottberechnungen mittels Alignmentverfahren festzustellen, hat die voestalpine Stahl GmbH Datensätze von 23 Stahlbändern bereitgestellt, die jeweils eine Dickenmessung am Ausgang der Warmbandstraße und eine Dickenmessung am Eingang der Beize (bzw. Schubbeize) enthalten. Die Messungen haben eine durchschnittliche Anzahl von etwa 9.000 Messpunkten. Die zugrundeliegenden Bänder waren also bis zu 1 *km* lang. Zu den Datensätzen wurden zusätzlich Schätzungen für die Länge des abgeschrotteten Materials am Bandkopf geliefert, um die Richtigkeit der berechneten Ergebnisse einschätzen zu können.

## 5.2 Schrottberechnung

Um die Länge des Schrotts am Anfang eines Datensatzes zu berechnen, wird ein optimales Alignment der beiden Messreihen mit einem der vorgestellten Alignmentverfahren berechnet. Da dem Algorithmus die längere Messreihe, also das Warmband, als Fixedseries und die beschnittene Messreihe, hier die Beizemessung, als Alignseries übergeben wird, wird die Beizemessung verschoben und skaliert.

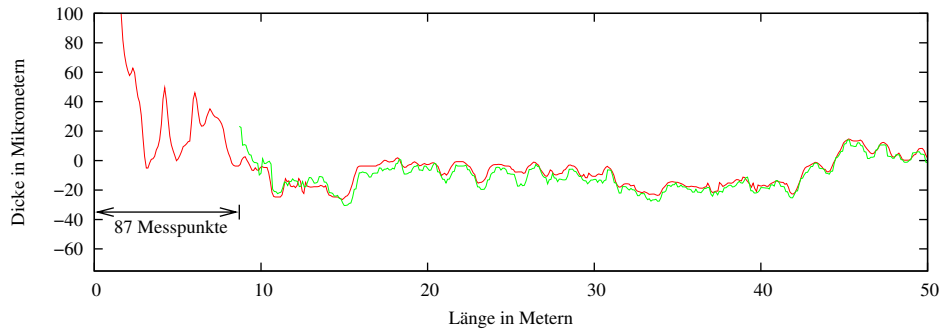
Aus Kapitel 3 wissen wir, dass die Kaltband- bzw. Beizemessung weniger fehlerbehaftet ist, da wir hier ein genaues Geschwindigkeitssignal kennen. Da die Ortsauflösung der Beizemessung somit deutlich verlässlicher ist, sollten wir die Warmbandmessung anstatt der Beizemessung skalieren.

Wir müssen also das Offset  $v_a$  und die Skalierung  $s_a$ , die der Alignmentalgorithmus berechnet hat, von der Alignseries auf die Fixedseries umrechnen. Das Offset  $v_f$  und die Skalierung  $s_f$  bezogen auf die Warmbandmessung lauten dann also

$$s_f = \frac{1}{s_a} \quad \text{bzw.} \quad v_f = -s_f \cdot v_a \quad . \quad (5.1)$$

Wenden wir nun  $s_f$  und  $v_f$  auf die Warmbandmessung an, so sehen wir, dass

die Warmbandmessung einen Overlap gegenüber der Beizemessung hat. Die links bzw. rechts überstehenden Messpunkte sollten dem Fuß- bzw. Kopfschrott entsprechen.



**Abbildung 5.2:** Visualisierung des Fußschrott zwischen einer Warmbandmessung (rot) und Beizemessung (grün)

In Abbildung 5.2 sehen wir eine Warmband- und eine Beizemessung. Der Ausschnitt zeigt den Fuß der Beizemessung. Bei den überstehenden Messpunkten des Warmbandes handelt es sich also um den Fußschrott.

Der Alignmentalgorithmus hat für die Beizemessung als Alignseries ein Offset von 87 Messpunkten und einen Skalierungsfaktor von 0,99 berechnet. Mit Hilfe der Gleichungen 5.1 können wir dieses Ergebnis auf die Warmbandmessung beziehen. Somit ergibt sich ein Fußschrott von  $\frac{1}{0,99} \cdot 87 \approx 88$  Messpunkten. Dies entspricht 8,8 m, da die Daten eine Auflösung von 10 cm pro Messpunkt haben.

### 5.3 Ergebnisse

Da wir noch evaluieren sollten, welches Gütekriterium sich am besten für das Alignment eignet, befindet sich in Anhang B.1 eine Tabelle mit den Schätzungen der voestalpine Stahl GmbH und einer Kopfschrottberechnung mit jedem Gütekriterium.

Es fällt auf, dass die Alignments mit den verschiedenen Gütekriterien nicht bedeutend voneinander abweichen. Leider weichen die berechneten Werte deutlich von den Schätzungen ab. Im Mittel beträgt diese Abweichung in etwa 70 cm.

Da alle 23 Alignments optisch passgenau wirken (siehe auch Abbildung 5.2), lässt sich die hohe Abweichung nur mit einem Zusammenspiel der folgenden Faktoren begründen.

1. Beim Ausschneiden der Messreihe aus der kontinuierlichen Messung wur-

den Fehler gemacht, da Anfang und Ende der eigentlichen Messung insbesondere beim Warmband nicht exakt zu bestimmen sind. So kann die Messreihe länger bzw. kürzer als das tatsächliche Band werden (vgl. Kapitel 3).

2. Die Annahme, dass die Daten nur linear gegeneinander skaliert sind, stimmt nicht für das komplette Band. Gerade am Anfang und Ende einer Messung kann es, aufgrund des dort sehr unzuverlässigen Geschwindigkeitssignals, zu größeren Ungenauigkeiten kommen, die auch eine Veränderung der Skalierung in diesem Bereich zur Folge haben können.
3. Die gelieferten Referenzwerte sind zu ungenau. Die Motivation zur Kopf- und Fußschrottberechnung mittels eines Alignmentverfahrens ist gerade der Mangel an Messmöglichkeiten. Tatsächlich sind die Schätzungen nicht immer plausibel, wie auch von voestalpine selbst bestätigt wurde.

Zusammenfassend müssen wir erkennen, dass die Berechnung von Kopf- und Fußschrott nicht so einfach ist, wie es sich zu Beginn dargestellt hat. Obwohl das Alignment im sogenannten Filetbereich des Dickenprofils, also des Profils ohne Kopf und Fuß, äußerst exakt zu sein scheint, ergeben sich aus den genannten Gründen gerade im Randbereich Probleme, die nur durch eine verbesserte Datenqualität auszuräumen sind. Aufgrund der extremen Bedingungen im Produktionsprozess, ist jedoch ein erheblicher Aufwand nötig, um die benötigte Messdatenqualität zu erreichen.

Die Frage der Genauigkeit lässt sich mit den uns vorliegenden Daten also nicht klären. Die Tatsache, dass es sich bei den Referenzwerten um Schätzungen handelt, kann uns hoffen lassen, dass die Schrottberechnung mit Hilfe der Alignmentverfahren genauer ist als die errechnete Durchschnittsabweichung von 70 *cm* befürchten lässt. Wir haben jedoch gesehen, dass die drei vorgestellten Gütekriterien sinnvolle Ergebnisse geliefert haben und sich somit auf gleiche Weise für das Alignment von Dickenschrieben eignen.

## Kapitel 6

# Zuordnung von Dickenschrieben

Bisher sind wir stets davon ausgegangen, dass wir zu einem gegebenen Dickenschrieb aus dem Kaltwalzwerk das entsprechende Pendant aus dem Warmwalzwerk kennen. Es gibt jedoch Fälle, in denen das Materialtracking fehlerhaft war und eben dieses Pendant nicht bekannt ist (siehe Abschnitt 1.2).

Die Gütekriterien, die in den letzten Kapiteln verwendet wurden, bewerten die Ähnlichkeit zwischen zwei Dickenschrieben. Die Idee, mit ihrer Hilfe eine Warmbandmessung zu einer gegebenen Kaltbandmessung wiederzufinden, ist also naheliegend. Da solch eine Suchanfrage jedoch über eine große Menge an Datensätzen geschieht, bleibt zu ermitteln, ob die Treffgenauigkeit und Performanz unserer Verfahren für diese Aufgabenstellung ausreichend ist.

### 6.1 Naiver Algorithmus zur Zuordnung

Ähnlich wie in Kapitel 4 wollen wir von einem naiven algorithmischen Ansatz ausgehen und diesen dann beschleunigen. Algorithmus 6.1 zeigt eine sehr rechenintensive Herangehensweise an die Problemstellung. Hier wird der zu suchende Dickenschrieb einfach mit allen in Frage kommenden Dickenschrieben mittels eines Alignments verglichen. Der mittlere quadratische Fehler wird hier wieder exemplarisch verwendet.

#### **Algorithmus 6.1 (Naiver Algorithmus zur Zuordnung)**

*Eingabe:*

- Referenzmessung  $S$
- Datenpool  $DB$  von Messungen, in denen  $S$  gesucht wird

*Ausgabe:*

Die *ID* der ähnlichsten Messung in *DB*

```
1: function FIND(S, DB)
2:   Wähle sinnvolle Parameter für Alignmentalgorithmus 4.6.
3:   mse =  $\infty$ 
4:   for all Messreihen M in DB do
5:     result = ALIGNGRID(M, S, from, to, step)           // Algorithmus 4.6
6:     if result.mse < mse then
7:       Merke ID von M
8:       Aktualisiere mse
9:   return ID
```

Statt das ähnlichste Band zurückzugeben, kann auch eine Rangliste der ähnlichsten Bänder zurückgegeben werden. Es hat sich jedoch herausgestellt, dass das Gegenstück zu dem gesuchten Dickenscrieb entweder auf Platz eins dieser Rangliste steht oder weit abgeschlagen ist.

Wir werden in Abschnitt 6.4 sehen, dass eine Anzahl von etwa 1.000 Bändern eine realistische Größe für den Datenpool der in Frage kommenden Bänder ist. Des Weiteren wissen wir aus Abschnitt 4.2.6, dass ein Alignment mit höchster Genauigkeit und den dort verwendeten Parametern etwa 42 Sekunden benötigt. Rein rechnerisch würde eine Bandsuche so mindestens 42.000 Sekunden  $\approx 11,5$  Stunden dauern. Selbst wenn wir die Gitterheuristik aus Abschnitt 4.2.4 verwenden und eine Alignmentzeit von einer Sekunde annehmen, blieben noch mindestens 16 Minuten.

Für die Bandsuche bzw. Zuordnung besteht also offensichtlich die Notwendigkeit den naiven Algorithmus zu beschleunigen. Dies werden wir in den nächsten Abschnitten tun und abschließend die benötigte Zeit und die Erfolgsquote experimentell feststellen.

## 6.2 Beschleunigung durch Komprimierung

Durch eine Komprimierung sämtlicher Dickenscriebe mittels PAA, kann die Zeit, die für ein einzelnes Alignment benötigt wird, vermindert werden. Um über den gesamten Zuordnungsalgorithmus 6.1 den größtmöglichen Laufzeitvorteil zu erhalten, sollten die PAA-Darstellungen der Messreihen bereits bekannt bzw. vorberechnet sein und beispielsweise in einer Datenbank vorliegen.

Die PAA-Darstellungen könnten parallel zum Aufzeichnen der Daten berechnet



werden, so dass zu jedem gespeicherten Datensatz auch eine PAA-Darstellung mit der gewünschten Komprimierungsstufe vorliegt. Der erhöhte Speicherplatzbedarf, der durch die redundante Speicherung des Originaldickenschriebs und des komprimierten Datensatzes entsteht, ist selbst bei Speicherung mehrerer PAA-Darstellungen pro Coil unproblematisch, da für jedes Coil ohnehin über tausend Messsignale zumindest kurzfristig gespeichert werden.

Wenn wir nun Algorithmus 6.1 mit PAA-komprimierten Daten aufrufen, erhalten wir zwar deutlich schneller ein Ergebnis, allerdings haben wir keine Garantie, dass dieses Ergebnis dasselbe ist wie bei einer Zuordnung auf unkomprimierten Daten. Im nächsten Abschnitt werden wir den Laufzeitvorteil der Komprimierung zumindest teilweise beibehalten und trotzdem ein optimales Ergebnis erhalten.

### 6.3 Das Siebverfahren

Das Siebverfahren wird von dem Ziel motiviert, eine möglichst optimale Zuordnung von Stahlband-Dickenschrieben in möglichst kurzer Zeit zu erhalten. Wie in Abschnitt 6.2 beschrieben, ist die Komprimierung der Messreihen eine geeignete Strategie die Suchzeit zu verkürzen, wohingegen wir den unkomprimierten Vergleich der naiven Zuordnung aus Abschnitt 6.1 benötigen, um die Optimalität des Ergebnisses zu garantieren.

Nachdem ein Vergleich zwischen zwei komprimierten Datensätzen stattgefunden hat, wollen wir eine Aussage über den mittleren quadratischen Fehler zwischen den unkomprimierten Darstellungen der Dickenschriebe treffen. Dazu sollten wir den Fehler kennen, der zwischen der Original- und der PAA-Darstellung eines Datensatzes besteht.

Wenn also  $M \in \mathbb{R}^n$  eine Messreihe und  $M^{(p)} := \text{PAA}_p(M)$  deren PAA-Darstellung mit einem beliebigen Kompressionsfaktor  $p \in \mathbb{N}$  ist, so wollen wir  $\text{MQF}(M, M^{(p)})$  berechnen. Da  $M^{(p)} \in \mathbb{R}^{\lfloor \frac{n}{p} \rfloor}$ , ist der mittlere quadratische Fehler zwischen  $M$  und  $M^{(p)}$  jedoch nicht definiert.

Da jeder Datenpunkt in  $M^{(p)}$  aus der Mittelung von  $p$  Datenpunkten in  $M$  besteht, existiert eine Zuordnung von  $M$  zu den Datenpunkten von  $M^{(p)}$ , mit deren Hilfe wir den mittleren quadratischen Fehler zwischen einer Messreihe und deren PAA-Darstellung definieren können. Danach ist

$$\text{MQF}(M, M^{(p)}) := \frac{1}{n} \sum_{i=0}^{n-1} (M_i - M_{\lfloor \frac{i}{p} \rfloor}^{(p)})^2 \quad . \quad (6.1)$$

Ein Punkt aus  $M$  wird also mit genau dem Punkt aus  $M^{(p)}$  verglichen, zu dessen Wert er beigetragen hat.

Wir benötigen im Folgenden die Dreiecksungleichung, also müssen wir als Gütekriterium  $\sqrt{\text{MQF}}$  aus Abschnitt 2.4.1 anstelle des mittleren quadratischen Fehlers verwenden. Für eine bessere Übersicht und um den Metrikeigenschaften Ausdruck zu verleihen, verwenden wir nun die Notation  $|x - y|$  anstelle von  $\sqrt{\text{MQF}}(x, y)$ .

Sei nun  $S \in \mathbb{R}^n$  eine weitere Messreihe, so können wir den Fehler zwischen  $M$  und  $S$  mit zweimaliger Anwendung der Dreiecksungleichung nach oben abschätzen durch

$$\begin{aligned} |M - S| &\leq |M - M^{(p)}| + |M^{(p)} - S| \\ &\leq |M - M^{(p)}| + |M^{(p)} - S^{(p)}| + |S^{(p)} - S| \end{aligned} \quad (6.2)$$

Analog dazu schätzen wir den Fehler zwischen  $M^{(p)}$  und  $S^{(p)}$  durch

$$\begin{aligned} |M^{(p)} - S^{(p)}| &\leq |M^{(p)} - M| + |M - S^{(p)}| \\ &\leq |M^{(p)} - M| + |M - S| + |S - S^{(p)}| \end{aligned} \quad (6.3)$$

ab. Nun erhalten wir durch Auflösen von (6.3) nach  $|M - S|$  eine untere Schranke für den Fehler zwischen  $M$  und  $S$ :

$$|M - S| \geq |M^{(p)} - S^{(p)}| - |M^{(p)} - M| - |S - S^{(p)}| \quad (6.4)$$

Wegen den Gleichungen (6.2), (6.4) und der Symmetrie von  $\sqrt{\text{MQF}}$  wissen wir, dass

$$|M - S| \in \left[ |M^{(p)} - S^{(p)}| - \Delta, |M^{(p)} - S^{(p)}| + \Delta \right] \quad (6.5)$$

mit

$$\Delta := |M - M^{(p)}| + |S - S^{(p)}| \quad (6.6)$$

ist. Diese Abschätzung wollen wir nun verwenden, um unseren Suchalgorithmus 6.1 zu beschleunigen und die Optimalität des Ergebnisses dennoch beizubehalten. Hierzu suchen wir die Referenzmessung  $S$  zum Anfang mit einer relativ hohen Komprimierungsstufe, um mit Hilfe von Abschätzung (6.5) möglichst viele Kandidaten im Datenpool auszuschließen. Anschließend fahren wir mit den übrigen Datensätzen und einer niedrigeren Komprimierungsstufe, die natürlich auch rechenintensiver ist, fort.

**Algorithmus 6.2 (Siebverfahren zur Zuordnung)**

*Eingabe:*

- Referenzmessung  $S$
- Datenpool  $DB$  mit Messungen und PAA-Darstellungen

- Länge  $k$  der Ergebnisliste

*Ausgabe:*

Die  $k$  ähnlichsten Messungen zu  $S$  in  $DB$

```

1: function FINDSIEVE( $S, DB, k$ )
2:   Wähle sinnvolle Parameter für Alignmentalgorithmus 4.6.
3:    $max = \infty$ 
4:   for all PAA-Stufen  $p$  do                                     // Absteigend sortiert
5:      $Bucket = \emptyset$ 
6:     for all Messreihen  $M \in DB$  do
7:        $step$  an PAA-Stufe anpassen                               // z.B. 1% der Datenlänge
8:        $result = \text{ALIGNGRID}(M^{(p)}, S^{(p)}, from, to, step)$  // Algorithmus 4.6
9:        $abw = |M^{(p)} - M| + |S^{(p)} - S|$                        //  $\Delta$  aus 6.6
10:       $Int = [\sqrt{result.mse - abw}, \sqrt{result.mse + abw}]$  // Abschätzung 6.5
11:      if  $Bucket.SIZE < k \vee Int.lower \leq max$  then
12:         $Bucket.ADD(M, Int)$ 
13:        if  $Int.upper < max$  then
14:          Setze  $max = k$ -t kleinste obere Schranke aus  $Bucket$ 
15:          Entferne alle Elemente  $E$  mit  $E.lower > max$  aus  $Bucket$ 
16:         $DB = Bucket$ 
17:   return  $DB$ 

```

Welche „Siebebenen“, also PAA-Stufen, von der Schleife in Zeile 4 durchlaufen werden, hängt stark von den zugrunde liegenden Daten ab und muss aus Erfahrungswerten bestimmt werden. Bei der Wahl dieser Abstufungen verhält es sich wie mit einem echten Sieb: Einerseits sollte mit einer Siebebene eine adäquate Anzahl von Datensätzen ausgesiebt werden, andererseits hat ein größeres Sieb eine höhere Durchflussgeschwindigkeit.

In den Zeilen 11 bis 15 werden alle Elemente, die sozusagen durch das Sieb gelaufen sind, zurück in den „Eimer“ getan, dessen Inhalt weiter gesiebt wird bis nach der letzten Siebstufe nur noch  $k$  Elemente übrig sind.

Da die Berechnung des Fehlers zwischen einem Datensatz der Länge  $n$  und seiner PAA-Darstellung, sowie die Berechnung dieser Darstellung selbst, in  $\mathcal{O}(n)$  liegen, kann der Algorithmus nur eine schnellere Laufzeit haben, wenn diese beiden Faktoren bereits vorberechnet sind und im Datenpool bzw. der Datenbank verfügbar sind.

Da wir für Algorithmus 6.2 die Dreiecksungleichung benötigen, kommt von den hier verwendeten Gütekriterien nur der mittlere quadratische Fehler bzw. dessen

Wurzel als Suchkriterium in Frage.

In Anhang B.2 können wir uns einen Eindruck davon verschaffen, wie effektiv mit verschiedenen PAA-Stufen gesiebt werden kann. Die Ergebnisse in Abschnitt 6.5.4 werden zeigen, dass Abschätzung (6.5) scharf genug ist, um das Siebverfahren deutlich effizienter als die naive Herangehensweise zu machen.

### 6.4 Einschränkung des Suchbereichs

In Kapitel 1.1 haben wir gelesen, dass die voestalpine Stahl GmbH im Warmwalzwerk eine Jahresproduktion von etwa 180.000 Stahlcoils hat. Wenn wir von einer Vorhaltezeit der Daten von nur einem Monat ausgehen, so müssen wir eine Kaltbandmessung in etwa 15.000 Warmbandmessungen suchen. In Abschnitt 6.1 wurde jedoch bereits angedeutet, dass eine Auswahl von etwa 1.000 Messungen realistisch sei.

Diese stark verkleinerte Auswahl kommt durch diverse Kenngrößen der Stahlbänder zustande. So kann eine Vorauswahl für den Datenpool, in dem das Kaltband gesucht wird, zum Beispiel über

- die Dicke,
- die Breite,
- die ungefähre Länge,
- die Materialbeschaffenheit und
- den Produktionszeitraum

getroffen werden. Der Datenpool, der nach dieser Vorauswahl entsteht, kann noch weiter verkleinert werden, indem alle Daten von Coils, die bereits im Kaltwalzwerk weiterbearbeitet wurden, herausgestrichen werden, so dass die Größe des Datenpools maximal der Anzahl der gelagerten Coils entspricht.

Abhängig von den Toleranzgrenzen aller Merkmale, bleibt nun eine kleine Auswahl an Datensätzen übrig, die in der Regel deutlich unter den genannten 1.000 Coils liegt. Wir wollen jedoch den (laufzeitmäßig) schlimmsten Fall betrachten.

### 6.5 Ergebnisse

In diesem Abschnitt werden wir die Machbarkeit einer Zuordnung evaluieren. Da wir damit nicht nur die theoretische Machbarkeit meinen, werden wir auch

die Laufzeiten der Suchvorgänge untersuchen, um eine praktische Machbarkeit zu zeigen. Wir wollen zunächst die Bedingungen der Versuchsreihen, also vor allem die Datenlage und die Wahl des Alignmentverfahrens, klären.

### 6.5.1 Datenbasis

Eine quantitative Aussage über die Treffgenauigkeit unserer Suchalgorithmen ist aufgrund mangelnder Daten sehr schwer. Die 23 Datensätze, die für die Untersuchungen in Kapitel 5 verwendet wurden, sind zwar qualitativ sehr gut geeignet, jedoch hätte ein Ergebnis, das auf so wenigen Datensätzen beruht, keine Aussagekraft.

Es liegt zwar eine größere Menge an Messdaten von Warm- und Kaltbändern vor, allerdings gibt es hier teilweise weder eine vorgegebene Zuordnung, so dass das Ergebnis der Algorithmen nicht verifiziert werden kann, noch ist überhaupt bekannt, ob jede Kaltbandmessreihe ein Pendant als Warmband in dieser Datenauswahl hat.

Bei dem Teil der Daten, der einander aufgrund von Identifikationsnummern klar zuzuordnen ist, gibt es häufiger Messausfälle<sup>1</sup>, die eine automatisierte Zuordnung mit Hilfe der vorgestellten Algorithmen erschweren bzw. das Ergebnis verzerren. Die Untersuchungen in diesem Kapitel stützen sich auf einen Datenpool von 191 Kaltbändern und 1000 Warmbändern. Zu diesen Kaltbändern ist die zugehörige Warmbandmessung jeweils bekannt.

Bei den Kaltbanddatensätzen handelt es sich um Messungen ohne die angesprochenen Messausfälle. Diese wurden herausgefiltert, da diese Art von Störung eher untypisch ist. Da die Auswahl der Kaltbänder ansonsten zufällig ist, sind andere typische Störeinflüsse sehr wohl in den Daten enthalten.

Ebenfalls enthalten sind umgewickelte Coils (siehe Abschnitt 1.1). Die meisten Kaltbanddatensätze müssen vertikal gespiegelt werden, um sie in den Warmbanddaten wiederzufinden. Bei umgewickelten Coils ist dies jedoch nicht nötig. Um auch die sogenannten „Umwickler“ zu finden, suchen wir jedes Kaltband zweimal und zwar einmal gespiegelt und einmal ungespiegelt. Der bessere Treffer bezüglich des Gütekriteriums wird dann gewertet.

---

<sup>1</sup>Der Dickenmesswert fällt über eine gewisse Zeit auf 0.

### 6.5.2 Parameter

Um einen Kaltbanddickenschrieb einem passenden Warmbanddickenschrieb zuzuordnen, müssen wir uns für die Verwendung eines passenden Alignmentverfahrens aus Kapitel 4 entscheiden. Nach den Laufzeitergebnissen in Abschnitt 4.2.6 ist naheliegend, dass wir Algorithmus 4.6 mit Algorithmus 4.5, also die Alignmentbestimmung mittels FFT und Gitterheuristik, wählen.

Das gewählte Alignmentverfahren fordert einige Parameter, die wir nun für die Versuche in den nächsten Abschnitten einheitlich und sinnvoll festsetzen:

**Fixedseries** Die Warmbandmessung wird stets als feste Messreihe übergeben, da sie länger als die Kaltbandmessung ist.

**Alignseries** Als auszurichtende Messreihe übergeben wir dementsprechend die Kaltbandmessung.

**Offsetintervall** Aufgrund eines fehlerhaften Geschwindigkeitssignals bei der Extraktion des meterbezogenen Dickenschriebs ist es denkbar, dass eine Kaltbandmessung mit einem optimalen Alignment wenige Punkte über die Warmbandmessung hinaus ragt, also einen geringen Overlap hat. Deswegen erlauben wir einen Overlap von 5% der Länge der Warmbandmessung.

**Skalierungsintervall** Auf den vorliegenden Daten wählen wir das Skalierungsintervall  $[0, 9; 1, 0]$ .

**Startschrittweite** Der Empfehlung aus Abschnitt 4.2.6 folgend, wählen wir die Startschrittweite der Gitterheuristik als 1% der Sequenzlänge des Warmbandes.

Um die Trefferwahrscheinlichkeit zu erhöhen, haben wir die Daten geeignet vorverarbeitet. So wurden alle Messungen um ihren Mittelwert bereinigt, um die Berechnung des mittleren quadratischen Fehler sowie des Korrelationskoeffizienten zu verbessern. Diese Vorverarbeitung spielt für den Gleichläufigkeitskoeffizienten keine Rolle. Des Weiteren wurden alle Dickenschriebe an den Rändern um 10 Messpunkte beschnitten. Das entspricht etwa 2‰ der Messpunktanzahl und beeinträchtigt das Alignmentergebnis somit nicht negativ. Im Gegenteil: Durch das Beschneiden der Randbereiche, in denen typischerweise die größten Messungenauigkeiten auftreten, kann die Güte der Alignments deutlich verbessert werden, was zu einer höheren Trefferquote bei der Zuordnung führt.

### 6.5.3 Vergleich der Gütekriterien

Wir wollen nun zuerst die verschiedenen Gütekriterien auf ihre Treffgenauigkeit testen. Wir haben in Abschnitt 6.1 bereits überschlagen, dass es mit der naiven Herangehensweise ohne Gitterheuristik und den dort gewählten Parametern mindestens 16 Minuten dauert ein Stahlband wiederzufinden. Nun wollen wir 191 Stahlbänder jeweils zweimal mit drei verschiedenen Gütekriterien suchen. Daraus ergibt sich eine theoretische Rechenzeit von etwa 300 Stunden, also mehr als 12 Tagen.

Da das schnellere Siebverfahren ausschließlich mit dem mittleren quadratischen Fehler als Gütekriterium einsetzbar ist, wurden die Daten für die nächste Versuchreihe mit der PAA komprimiert, um die Rechenzeit zu verkürzen. Dadurch erhalten wir zwar nicht die bestmögliche Trefferquote für jedes Gütekriterium, aber zumindest eine vergleichende Aussage ist möglich.

Gütekriterium	PAA 30	PAA 100	PAA 200
mittlerer quadratischer Fehler	98,4%	98,4%	89,0%
Korrelationskoeffizient	98,4%	96,9%	86,9%
Gleichläufigkeitskoeffizient	100,0%	98,4%	86,9%

*Tabelle 6.1: Trefferquote der verschiedenen Gütekriterien*

In Tabelle 6.1 sehen wir die Trefferquoten der Gütekriterien auf unserer Datenbasis. Um mehr Falschtreffer zu provozieren und somit den Unterschied zwischen den Kriterien deutlicher hervorzuheben, wurde der Versuch mit den PAA-Kompressionsstufen 30, 100 und 200 durchgeführt. Bemerkenswert ist die Tatsache, dass sich die Trefferquoten der verschiedenen Gütekriterien kaum unterscheiden. Außerdem sind die Quoten trotz der typischen Messstörungen relativ hoch.

Mit dem Gleichläufigkeitskoeffizienten ließen sich sogar alle 191 Kaltbänder wiederfinden. Ein Vorteil bei der Verwendung des Gleichläufigkeitskoeffizienten auf den (ungefilterten) vorliegenden Daten ist, dass Messausfälle im Kaltband das Ergebnis bei der Zuordnung kaum beeinflussen, wenn trotzdem noch genügend restliches Band gemessen wurde. Ein Messausfall im Kaltband ist nämlich nichts anderes als ein konstanter Abschnitt in der Messung. Nach unserer Berechnung des Gleichläufigkeitskoeffizienten in Abschnitt 4.2.2 werden diese Abschnitte pauschal als gleichläufig gewertet. Wäre hingegen ein längerer Messausfall in einer beliebigen Warmbandmessung im Datenpool, so würde dieses Warmband bei einer Suche nach einem beliebigen Kaltband stets unverhältnismäßig gut

bewertet werden.

Der mittlere quadratische Fehler schneidet bei geringer PAA-Komprimierung etwas schlechter als der Gleichläufigkeitskoeffizient ab. Das lässt sich damit erklären, dass Messstörungen sich in einem Versatz oder einem Peak in  $y$ -Richtung auswirken. Aus der Definition der Kriterien ist klar, dass solch eine Störung den mittleren quadratischen Fehler weitaus mehr verfälscht als den Gleichläufigkeitskoeffizienten. Hieraus wird auch sofort ersichtlich, weshalb der MQF für höhere PAA-Stufen eine, relativ zu den anderen Kriterien, höhere Trefferquote hat. Wie bereits am Ende von Abschnitt 2.5 angemerkt wurde, werden mit einer hohen PAA-Komprimierung nämlich genau die beschriebenen Messstörungen geglättet bzw. gefiltert.

Eine weitere Versuchsreihe hat für den Gleichläufigkeitskoeffizienten auf Daten, die mit einem Kompressionsfaktor von 50 komprimiert wurden, eine Trefferquote von 98,2% ergeben. Dieser Versuch wurde mit einer Datenbasis von 4008 Kaltbändern und 4308 Warmbändern durchgeführt. In dieser Datenbasis waren alle typischen und untypischen Messstörungen enthalten, darunter insbesondere Messausfälle, Bandteilungen, fehlende Warmbandmessungen und Fehler beim Exportieren der Daten aufgrund fehlender Gültigkeitssignale (siehe Kapitel 3). Die Suche eines Datensatzes ohne Umwickler hat in diesem Versuch durchschnittlich 9,84 Sekunden gedauert. Aufgrund der äußerst schlechten Datenlage ist das Ergebnis zwar nicht repräsentativ, zeigt aber durchaus wie tolerant der Gleichläufigkeitskoeffizient gegenüber Messstörungen ist.

### **Einschub: Der Korrelationskoeffizient zum Auffinden horizontal gespiegelter Zeitreihen**

Der Korrelationskoeffizient hat nach den Ergebnissen in Tabelle 6.1 am schlechtesten abgeschnitten. Er hat jedoch unter Bedingungen, die in dieser Arbeit nicht betrachtet werden, einen Vorteil in der Laufzeit, denn diese kann mit dem Korrelationskoeffizienten um den Faktor 2 schneller sein als eine Zuordnung mit dem mittleren quadratischen Fehler, falls auch horizontal gespiegelte Zeitreihen gesucht werden sollen.

Am Beispiel der Stahlband-Fingerprints kann dieser Fall nicht eintreten, da es sich hier um Dickenmessungen handelt. Zwei Dickenmessungen von ein und demselben Coil können unter keinen Umständen horizontal gegeneinander gespiegelt sein.

Wenn in einer anderen Anwendung jedoch eine Zeitreihe gesucht werden soll, die potentiell (aber nicht sicher) horizontal gespiegelt wurde, so reicht es diese nur



ein einziges Mal mit dem Betrag der Korrelation als Gütekriterium zu suchen, statt mit dem mittleren quadratischen Fehler zweimal, nämlich gespiegelt und ungespiegelt.

Nach Lemma 2.11 wissen wir, dass

$$\text{Kor}(M, -S) = -\text{Kor}(M, S)$$

ist. Da es sich bei  $-S$  um die horizontale Spiegelung der Zeitreihe  $S$  handelt, entspräche eine Suche nach der gespiegelten Zeitreihe einer Suche nach der Originalreihe unter *Minimierung* der Korrelation. Wir könnten also beide Suchschritte vereinigen, indem wir den Betrag der Korrelation maximieren.

In ähnlicher Weise kann man diese Idee auch mit dem Gleichläufigkeitskoeffizienten umsetzen, da die horizontal gespiegelte Zeitreihe vollkommen gegenläufig zur Originalreihe ist, also einen GLK von 0 hat. Wir könnten in diesem Fall  $|\frac{1}{2} - \text{GLK}|$  maximieren und so wieder das beste Alignment beider Spiegelungen finden.

#### 6.5.4 Vergleich der Verfahren

Wir haben durch den Vergleich der Gütekriterien bereits experimentell belegt, dass sich eine Warmbanddickenmessung eines gegebenen Kaltbandes in einer großen Menge anderer Dickenmessungen wiederfinden lässt. Selbst leichte Störungen und umgewickelte Coils verhindern eine Zuordnung nicht. Wenn das Coil, welches gesucht werden soll, potentiell umgewickelt wurde, so verdoppelt sich lediglich die Laufzeit der Zuordnung.

In der Praxis ist neben der Treffgenauigkeit auch die Laufzeit der Verfahren von größter Bedeutung. So ist ein Verfahren, das ein Coil mit hundertprozentiger Wahrscheinlichkeit wiederfindet, aber dafür mehrere Minuten, Stunden oder gar Tage benötigt, praktisch wertlos. Deswegen wollen wir abschließend vor allem die absoluten Laufzeiten der Verfahren evaluieren.

Mit dem Siebverfahren und den PAA-Siebstufen 100, 30, 15, 8, 5, 3, 1 benötigt die Suche einer Messreihe in der Datenbasis aus Abschnitt 6.5.1 durchschnittlich 49 Sekunden, das heißt etwa  $1\frac{1}{2}$  Minuten mit Umwicklern. Diese Laufzeit variiert jedoch äußerst stark mit den gewählten Siebstufen und der Beschaffenheit der Daten.

In Anhang B.2 findet sich eine detaillierte Übersicht der Ergebnisse des Siebverfahrens. Hier sehen wir auf der einen Seite wie effektiv oder auch ineffektiv die einzelnen Siebstufen Datensätze aussieben und auf der anderen Seite

die Laufzeiten für die zweimalige Suche jedes einzelnen Kaltbandes. Für Kaltbandmessreihen, die so beschaffen sind, dass schon sehr früh viele Messungen ausgeschlossen werden können, dauert der Suchvorgang lediglich 5 Sekunden (vgl. Ident-Nr. 986085), wohingegen es Messreihen gibt, bei denen bis zur letzten Siebstufe kaum Datensätze ausgesiebt wurden und der Suchvorgang bis zu 19 Minuten und somit länger als mit dem naiven Zuordnungsverfahren dauert (vgl. Ident-Nr. 996986).

Grundsätzlich können wir feststellen, dass die Suche eines Datensatzes dann sehr lange dauert, wenn er viele Peaks enthält bzw. sehr „unglatt“ ist. Denn dann ist der Fehler zu seiner PAA-Darstellung sehr hoch, wodurch das Intervall aus Gleichung (6.5) groß wird. Somit können nur wenige Datensätze in jedem Siebschritt gesiebt werden. Das Beschneiden der Datensätze um 10 Messpunkte an den Rändern konnte die Laufzeit des Siebverfahrens beinahe halbieren, da damit die typischen Peaks in den Randbereichen eliminiert wurden.

In Tabelle 6.2 sehen wir zwar, dass eine Laufzeit unter einer Minute ein durchaus gutes Ergebnis im Vergleich zur unkomprimierten Suche ist, jedoch kommen Zweifel auf, ob es überhaupt vorteilhaft ist, den besten Treffer bezüglich des Gütekriteriums auf den Originaldaten zu finden.

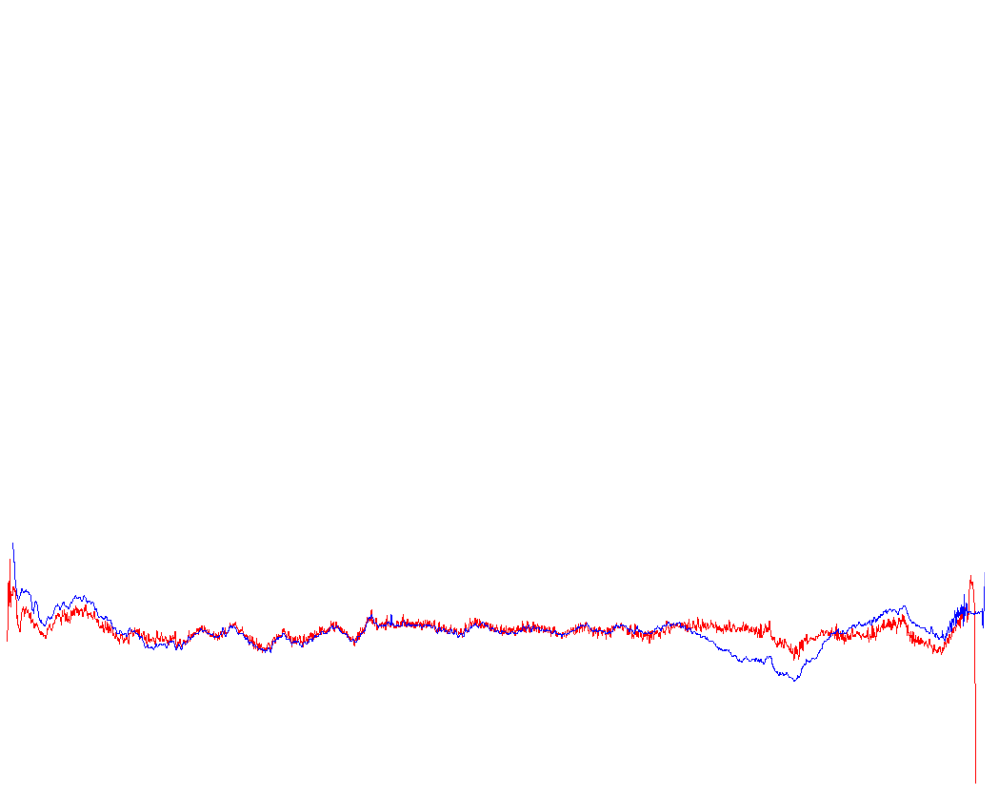
PAA	Falschtreffer (MQF)	Trefferquote	∅ Dauer für ein Band
keine	–	–	ca. 6-7 min
Sieb	3	98%	48,86 s
20	2	99%	7,85 s
30	3	98%	4,58 s
50	3	98%	2,20 s
75	3	98%	1,29 s
100	3	98%	0,78 s
125	6	97%	0,54 s
150	8	96%	0,37 s
200	21	89%	0,23 s
300	34	82%	0,12 s

**Tabelle 6.2:** Trefferquote mit verschiedenen PAA-Stufen und dem MQF als Gütekriterium. Die Dauer bezieht sich auf die Suche ohne Umwickler.

Die absoluten Laufzeiten<sup>2</sup> der Suche auf komprimierten Daten ist nicht nur, wie zu erwarten, drastisch besser, sondern auch die Trefferquote leidet nicht

---

<sup>2</sup> Referenzrechner: Lenovo ThinkPad T60; Intel Core 2 1,83GHz; 3,0GB RAM  
Implementierung: C#



*Abbildung 6.1: Der komprimierte Datensatz wurde gefunden, der unkomprimierte hingegen nicht (Ident-Nr.: 980675)*

unter der Komprimierung der Daten. In einem Fall ließ sich sogar ein Kaltband mit Messstörung mit einer PAA-Komprimierung von 20 finden, das auf den Originaldaten nicht gefunden wurde. Diesen Datensatz sehen wir in Abbildung 6.1.

Die extremen Peaks, die in Warmband- (rot) und Kaltbandmessung (blau) vorhanden sind, wurden durch das Beschneiden der Randbereiche bereits abgetrennt, jedoch zeigt sich im Kopfbereich ein deutlicher  $y$ -Versatz, der den mittleren quadratischen Fehler stark beeinflusst. Des Weiteren sehen wir, dass das Kaltband im Kopfbereich länger als das Warmband ist. Dies deutet auf einen Fehler in der Geschwindigkeitsaufzeichnung hin.

Abschließend sollte noch erwähnt werden, dass die hier angegebenen Laufzeiten sich auf die reine Suchzeit beziehen. Die Vorverarbeitung (Einlesen, PAA-Komprimierung, Mittelwertbereinigung, etc.) ist nicht darin enthalten, da die Daten in der Praxis bereits vorverarbeitet in einer Datenbank liegen werden. Doch selbst, wenn man die gesamte Vorverarbeitung zu den Laufzeiten hinzuzählt, ergeben sich für die Praxis vertretbare Wartezeiten. So dauert die

komplette Vorverarbeitung der hier betrachteten Datenbasis inklusive Datei-  
zugriffen auf dem Referenzrechner lediglich etwa 10 Sekunden.

# Kapitel 7

## Fazit und Ausblick

Wir haben in dieser Arbeit Verfahren zur Bestimmung eines optimalen horizontalen Alignments von Zeitreihen bezüglich einem von drei Gütekriterien kennengelernt. Es ist uns zwar gelungen, diese Verfahren deutlich zu beschleunigen, jedoch mussten wir letztendlich die Garantie auf ein optimales Alignment zu Gunsten einer Laufzeit im Subsekundenbereich aufgeben. In der praktischen Anwendung hat sich herausgestellt, dass dies bei korrekter Wahl der Startschrittweite der Gitterheuristik keinerlei Auswirkungen auf die Ergebnisse hat und somit durchgehend optimale Alignments von Stahlband-Fingerprints bestimmt werden.

Um die Alignmentverfahren zu evaluieren und Aufgabenstellungen aufzuzeigen, die mit Hilfe unserer Alignmentverfahren gelöst werden können, haben wir die Berechnung der Länge von Kopf- und Fußschrott und die Zuordnung von Stahlband-Fingerprints zwischen Kalt- und Warmwalzwerk aufgrund realer Dickenmessdaten der voestalpine Stahl GmbH betrachtet.

Während die Ergebnisse der Zuordnung sehr gut waren, hatten wir Probleme mit der Genauigkeit der Kopf- und Fußschrottberechnung. Diese Probleme lassen sich mit den messtechnischen Schwierigkeiten erklären, die es bei der Warmbandmessung im Bereich von Kopf und Fuß gibt. Hierdurch werden die Ergebnisse der Schrottberechnung verfälscht, während die Kurven im sogenannten Filetbereich des Bandes, der für das Endprodukt relevant ist, fast perfekt zur Deckung gebracht werden.

Die genauere Betrachtung der wenigen Falschtreffer bei der Zuordnung aus Abschnitt 6.5 lässt darauf schließen, dass auch hier eine Optimierung der Messtechnik die Ergebnisse weiter perfektionieren kann. Unter optimalen Voraussetzungen sollte es gar möglich sein, jedes Band bei einer Suche wiederzufinden.

Diese Voraussetzungen ließen sich eventuell bereits durch eine Plausibilisierung der Messwerte erreichen, da es in unserem Datenpool viele unvollständige und somit sehr kurze Messprofile gab, die das Ergebnis verfälschten.

Das Alignmentverfahren, welches auch für die Versuche in dieser Arbeit verwendet wurde – der FFT-Alignmentalgorithmus mit Gitterheuristik –, ist seit der Version 5.13.0 in den ibaAnalyzer, einem Tool zur Offline-Analyse von Messdaten, das von der iba AG entwickelt wird, als neue Funktionalität integriert (siehe auch Anhang C).

Aufgrund der starken Verbreitung von iba-Messtechnik in Walzwerken konnten seitdem auch viele weitere Beispiele getestet werden. Aus diesen Erfahrungen haben sich einige neue Anwendungsgebiete der sogenannten Fingerprint-Methode erschlossen:

- Das Verfahren wurde erfolgreich an Daten aus anderen Prozessschritten getestet. Obwohl die Stahlbänder nach dem Kaltwalzwerk äußerst präzise gewalzt sind und somit das Dickenprofil eines Stahlbandes nur noch wenig charakteristisch scheint, konnten selbst Dickenschriebe von Stahlbändern am Ausgang des Kaltwalzwerks und am Eingang der Verzinkungslinie sehr gut aligniert werden.
- In der Verzinkungslinie wird das Stahlband oberflächenveredelt, indem eine extrem dünne und homogene Zinkschicht aufgetragen wird. Zusätzlich wird das Stahlband in dieser Anlage dressiert, also zur Planheitsverbesserung leicht gewalzt, wodurch eine Bandlängung von etwa 1% erreicht wird. Trotz der Bearbeitung des Bandes zwischen den beiden Messungen, konnten Alignments von Dickenprofilen am Eingang der Verzinkungslinie und am Ausgang der Verzinkungslinie bereits erfolgreich durchgeführt werden.
- Bei dem ibaQDR (Quality Data Recorder) handelt es sich um ein Messsystem, das in der Lage ist, im laufenden Betrieb die zeitbezogenen Messsignale verschiedener Messorte innerhalb einer kontinuierlichen Bandanlage längenbezogen zu transformieren und auf die endgültige Länge des Bandes zu normieren. Transformations- und Normierungsfehler, beispielsweise aufgrund fehlerhafter Bandverfolgungssignale oder Konfiguration des ibaQDR, sind in den erzeugten Messdateien nicht unmittelbar erkennbar. Mit einem Alignment zwischen Dickenprofilen aus verschiedenen Messorten des ibaQDR lässt sich aber sehr einfach und ohne großen Aufwand verifizieren, ob das System konsistente Daten liefert. Auch an diesen Daten wurde die Fingerprint-Methode bereits erfolgreich getestet.
- In dieser Arbeit wurden lediglich *Dicken*profile von Stahlbändern betrach-

---

tet. Da die Implementierung des Alignmentverfahrens im ibaAnalyzer keineswegs auf diese beschränkt ist, lassen sich hiermit auch Breitenprofile aneinander ausrichten. Es hat sich herausgestellt, dass auch diese genügend Charakteristik aufweisen, um deckungsgleiche Alignments zwischen zwei getrennten Messungen zu ermöglichen.

Ein Problem, das in allen bisherigen und zukünftigen Anwendungen der Fingerprint-Methode ein Rolle spielt, ist die Datenqualität. Wenn diese zu schlecht ist, also große Messstörungen vorhanden sind, können zwei Kurven nicht deckungsgleich übereinander gelegt werden. Wenn Alignments in Zukunft automatisiert stattfinden sollen, wird also die Erkennung oder sogar Korrektur verfälschter Daten eine große Rolle spielen. Hierzu wurden bereits erste Versuche der Universität Mainz unternommen, die auf einer statistischen Analyse der Daten beruhen und in denen es gelungen ist, einige Messstörungen zu detektieren und diese anschließend durch Manipulation der Daten zur verringern.

Während in dieser Arbeit lediglich horizontale Alignments mit einem konstanten Skalierungsfaktor betrachtet wurden, sind auch horizontale Alignments mit nichtlinearen Skalierungen interessant. Der Shotgun-Aligner aus Abschnitt 4.3 wäre hierfür prinzipiell geeignet, da dieses Verfahren in der Lage ist, abschnittsweise Skalierungsfaktoren zu bestimmen. Laut [GGK<sup>+</sup>08] ist das Dynamic Time Warping bei qualitativ vergleichbaren Ergebnissen jedoch schneller als das Shotgun-Verfahren. Des Weiteren werden beim Dynamic Time Warping keine abschnittswise Skalierungsfaktoren gesucht, sondern es wird eine echte Punkt-zu-Punkt-Zuordnung zwischen den beiden Messreihen bestimmt. Somit ist die gefundene Skalierung vollkommen nichtlinear.

Das Dynamic Time Warping spielt auch eine Rolle bei der Online-Erkennung von Stahlbändern. Hier geht es darum, ein Stahlband mit einem Dickenschrieb, der in einem vorherigen Prozessschritt angefertigt wurde, zu alignieren, während es durch das Walzwerk läuft. Es soll also ein Alignment berechnet werden, das unter Hinzunahme neuer Messwerte kontinuierlich verbessert werden soll. Hierzu scheint eine neue Variation des Dynamic Time Warping, die in [TSI08] beschrieben ist, besonders geeignet zu sein. Mit dieser Variante soll es möglich sein, das Alignment nach Hinzunahme eines neuen Messwertes in Zeit  $\mathcal{O}(n)$  zu aktualisieren. Erste Tests der Universität Mainz an Stahlband-Fingerprints stehen zur Zeit jedoch noch aus.

In Zukunft soll mit Hilfe der Fingerprint-Methode ein zertifizierter Stammbaum eines jeden Stahlbandes aufgebaut werden. Mit diesem Stammbaum soll es möglich sein, Messwerte eines Stahlbandes aus verschiedenen Prozessschritten präzise übereinander zu legen. Die Schnittstelle zwischen den Prozessschritt-

ten bildet hier der Fingerprint, also der Dickenschrieb, eines Stahlbandes. Anhand des berechneten Skalierungsfaktors und der Verschiebung des Fingerprints lassen sich auch sämtliche andere Messwerte übereinander legen, so dass ein Fehler im Band, der an einer gewissen Position des Endproduktes festgestellt wird, auch Messsignal- und Prozess-übergreifend analysiert werden kann.

Auf diese Weise lassen sich Informationen über den Produktionsprozess gewinnen, die *unmittelbar* zur Verbesserung der Produktqualität beitragen.



## Anhang A

# Illustration des Produktionsprozesses

Auf den folgenden Seiten finden sich Skizzen von Warm- und Kaltbandstraße sowie eine Einordnung derselben in den Gesamtproduktionsprozess. Die Abbildungen wurden mit freundlicher Genehmigung der angegebenen Rechteinhaber in diese Arbeit übernommen. Für nähere Informationen und detaillierte Beschreibungen empfiehlt es sich, die referenzierten Quellen zu lesen.

ANHANG A. ILLUSTRATION DES PRODUKTIONSPROZESSES

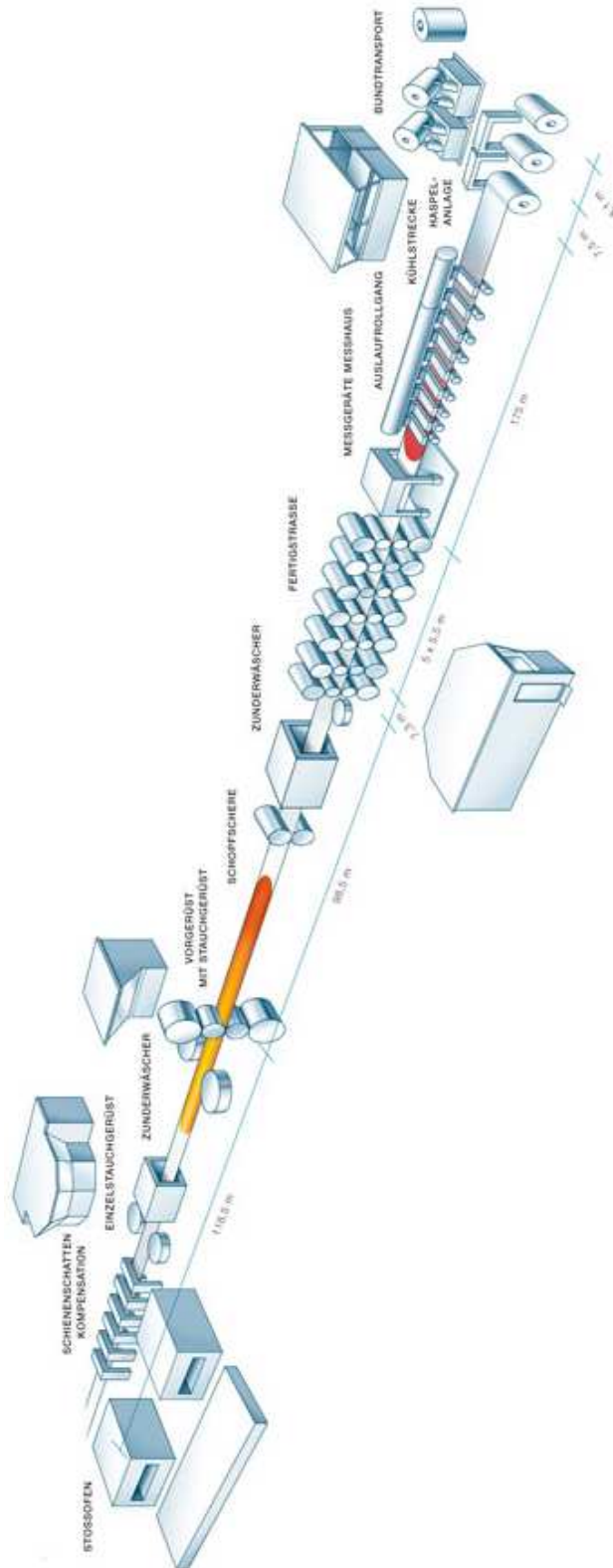
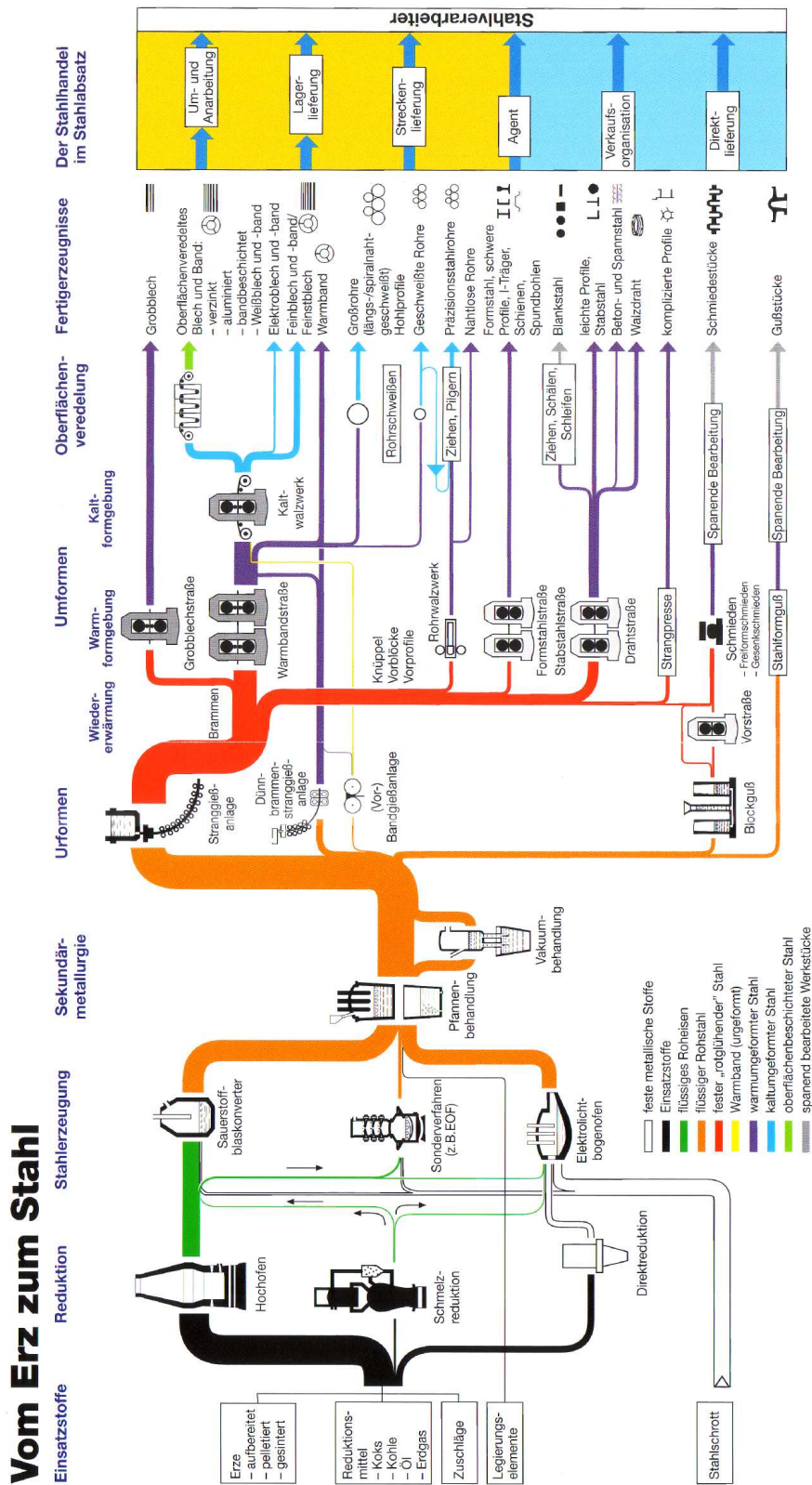


Abbildung A.1: Skizze der Warmbreitbandstraße [vSGb]





Herausgeber: Stahlinstitut VDEh, Düsseldorf  
© 2007 Verlag Stahleisen GmbH, Düsseldorf

Abbildung A.3: Einordnung in den Gesamtproduktionsprozess [DFE<sup>+</sup> 07]

# Anhang B

## Daten

### B.1 Alignmentergebnisse für den Kopfschrott

Die Ergebnisse basieren auf einer Datensammlung von Mai 2008, die aus Dicken-  
schriften aus Warmwalzwerk und Beize von 23 Stahlbändern besteht. Bei die-  
sen Bändern hat die voestalpine Stahl GmbH Angaben zum Kopfschrott ge-  
macht. Die Bänder wurden mit durchschnittlich etwa 9.000 Messpunkte bei  
einer Auflösung von 10 cm pro Messpunkt auf Meterbezug extrahiert.

Ident-Nr.	Schätzung voest [cm]	mittl. quadr. Fehler [cm]	Korrelations- koeffizient [cm]	Gleichläufigkeits- koeffizient [cm]
442285	177	120	120	130
442286	193	520	520	520
446844	197	180	180	200
446845	177	290	300	290
446847	180	210	210	210
446848	200	340	340	350
446849	177	280	280	250
447504	195	180	170	180
447505	174	160	160	160
447507	199	200	200	200
447508	180	140	140	150
447509	200	160	160	170
447510	177	190	190	190
451285	205	90	90	110
451289	177	150	150	170
463202	174	150	140	140
463203	200	470	470	460
463204	180	120	110	100
463205	195	150	150	140
465645	180	310	300	340
466593	200	200	200	200
466594	177	180	180	180
468703	195	160	160	150

## B.2 Ergebnisse des Siebverfahrens

Die Ergebnisse basieren auf der in Abschnitt 6.5.1 beschriebenen Datensammlung. Diese Datensammlung ist eine zufällige Stichprobe aus einer großen Menge von Datensätzen, die die voestalpine Stahl GmbH im Februar 2008 für die Untersuchungen zur Verfügung gestellt hat. Die Stichprobe wurde lediglich um Messungen mit Messausfällen bereinigt.

Die Zeitangaben in der Tabelle beziehen sich auf die Suchzeit für ein Kaltband in 1000 Warmbandmessungen. Hierbei wurden auch Umwickler gesucht, so dass die einfache Suchzeit lediglich die Hälfte der angegebenen Dauer ist.

Im mittleren Teil der Tabelle ist aufgetragen, wie viele Datensätze nach der jeweiligen PAA-Siebstufe noch für den insgesamt besten Treffer in Frage kommen. Der Suchvorgang wurde jeweils abgebrochen, wenn nur ein Datensatz übrig war. Bei einem Großteil der Versuche war es auf diese Weise nicht einmal nötig einen unkomprimierten Vergleich durchzuführen.

Während der Versuchsreihe traten drei Falschtreffer auf, die am rechten Rand mit einem „F“ markiert sind.

Ident-Nr.	PAA 100	30	15	8	5	3	keine	Dauer in Sekunden
400877	287	281	204	12	3	1		32,445
400904	61	60	44	22	5	1		14,835
400936	172	121	77	37	14	2	1	26,562
400969	962	944	856	504	89	3	1	164,781
401122	961	102	14	1				13,267
401530	778	648	172	11	1			39,208
401583	725	400	47	4	1			23,618
401584	816	722	103	5	1			36,553
402325	966	951	583	18	2	1		67,515
402336	844	825	820	818	818	814	1	1040,757
402341	855	823	264	9	1			49,881
402353	933	879	585	77	5	1		76,982
402361	778	713	451	46	9	1		65,929
971715	843	798	722	428	57	2	1	147,193
971808	962	953	873	84	7	1		90,386
971866	412	238	58	4	3	1		20,765
972727	971	838	195	12	3	1		42,9
977834	726	673	528	192	14	3	1	94,688
977987	867	850	844	817	733	155	1	478,126
979183	450	316	84	5	3	1		24,756
979307	811	526	240	21	5	2	1	45,589
979323	940	750	383	95	8	4	1	70,667
980146	387	92	6	3	3	1		11,78
980602	828	564	231	40	5	3	1	49,795

## B.2. ERGEBNISSE DES SIEBVERFAHRENS

980603	839	710	559	387	174	45	1	197,457	
980610	829	678	500	334	225	127	1	276,366	
980642	888	515	171	14	6	1		35,816	
980668	720	478	149	8	4	1		34,243	
980675	703	657	556	366	166	44	1	194,821	F
980682	742	479	261	31	6	3	1	49,643	
980779	971	914	510	30	6	1		64,591	
980787	837	751	621	384	319	5	1	202,185	
980954	951	861	287	7	3	1		48,807	
981097	413	384	346	260	156	5	1	136,517	
981113	956	783	455	130	11	3	1	75,705	
981143	799	411	48	6	1			24,559	
981144	721	340	38	5	3	1		22,713	
981150	842	652	187	8	3	1		40,832	
981294	862	477	62	5	3	1		27,684	
981778	892	707	355	35	8	1		55,886	
981779	812	579	282	43	8	1		50,196	
981812	837	337	36	3	3	1		22,448	
981852	348	108	9	3	3	1		12,288	
981869	908	419	13	3	3	1		22,369	
981904	857	203	6	3	1			16,187	
981910	855	418	30	4	3	1		24,114	
981964	975	473	29	4	1			21,413	
982158	742	96	7	2	1			12,28	
982180	938	689	67	5	1			31,995	
982194	806	424	19	3	3	1		23,683	
982571	970	680	40	6	1			28,482	
982581	864	823	740	584	432	190	1	422,14	
982797	777	448	92	9	3	1		29,362	
982828	937	310	19	3	1			19,405	
982829	931	567	115	9	3	1		33,143	
982884	952	152	13	2	1			14,145	
983072	987	924	82	4	2	1		33,408	
983084	923	8	2	1				9,795	
983089	897	270	11	1				17,359	
983109	937	408	30	4	1			22,181	
983225	941	725	245	5	3	1		44,487	
983377	560	329	141	10	5	2	1	33,598	
983571	840	682	489	252	90	11	1	124,567	
983616	382	201	32	3	3	1		17,142	
983725	962	897	501	35	8	2	1	68,478	
984051	925	846	307	16	3	1		52,291	
984176	971	894	416	22	3	1		56,585	
984283	592	415	122	7	3	1		30,056	
984290	575	528	286	23	4	1		46,018	
984720	967	960	942	896	786	529	1	761,646	
984785	838	634	366	74	11	1		62,17	
984926	915	32	4	2	1			10,674	
985078	976	760	282	42	5	2	1	48,411	
985114	830	167	5	2	1			15,022	

ANHANG B. DATEN

---

985197	983	978	977	970	947	155	1	476,808
985671	983	467	60	6	2	1		23,008
985837	936	129	12	4	1			13,943
986085	79	39	6	2	1			5,411
986428	962	955	948	927	872	588	1	843,483
986477	262	25	3	2	1			5,841
986508	563	320	83	7	4	1		24,668
986510	715	538	347	129	27	4	1	73,498
986634	943	638	32	4	1			28,218
986644	973	973	947	65	4	1		88,915
986673	924	863	271	15	4	1		49,982
986850	960	599	30	4	1			25,837
986872	941	567	14	3	1			25,823
986921	719	12	4	1				9,328
986947	905	678	389	163	24	5	1	81,747
986974	388	294	153	51	8	3	1	40,938
987323	959	841	60	5	3	1		35,504
987324	961	914	650	218	22	3	1	103,897
987460	985	980	978	967	946	772	1	926,739
987480	945	98	5	1				12,67
987519	412	300	20	3	3	1		19,249
987972	530	26	3	1				8,665
987989	294	111	4	2	1			9,982
988001	359	324	307	250	4	1		74,325
988092	554	295	71	8	3	1		23,603
988107	318	206	33	3	3	1		17,322
988116	318	171	19	3	1			13,737
988139	976	826	83	7	3	1		34,485
988140	976	885	211	9	4	1		42,587
988170	967	640	5	3	1			26,084
988179	703	290	6	3	1			18,368
988520	854	9	1					9,763
988578	853	432	18	3	1			23,15
988579	844	707	128	17	3	1		39,958
988619	934	20	4	1				10,543
988956	973	83	5	1				11,401
988963	853	409	22	3	3	1		24,429
989076	781	177	6	2	1			15,348
989931	969	847	147	11	3	1		41,328
989960	316	117	22	3	2	1		12,5
989983	205	77	13	3	2	1		9,711
990307	978	974	973	745	4	1		152,091
990476	976	858	220	11	3	1		42,025
990478	928	682	185	11	3	1		39,578
990513	904	748	251	40	6	1		48,847
990527	65	54	40	20	6	3	1	20,561
990938	954	775	194	7	3	1		40,526
991021	933	557	109	8	4	1		33,51
991108	962	943	733	281	45	8	1	118,313
991171	971	840	209	16	2	1		42,117



B.2. ERGEBNISSE DES SIEBVERFAHRENS

991184	814	475	27	3	1			24,824	
991202	839	605	130	7	3	1		35,209	
991303	980	976	976	946	883	5	1	331,817	
991366	974	887	666	338	81	13	1	129,063	
991444	700	425	84	5	1			26,345	
991459	681	401	100	16	6	1		31,389	
991463	962	946	916	822	487	9	1	279,749	
991478	943	87	13	3	1			13,254	
991485	837	402	41	6	1			24,196	
991514	272	159	43	8	4	1		15,841	
991608	945	639	30	3	1			27,817	
991613	952	917	757	230	64	11	1	119,897	
991639	412	322	224	117	52	19	1	85,861	
991643	448	271	135	19	8	1		30,171	
991648	279	205	102	28	9	1		26,078	
991671	490	404	355	219	84	2	1	102,132	
991684	675	391	64	10	3	1		27,16	
991898	965	753	86	6	1			32,599	
991946	457	161	24	4	3	1		15,55	
991947	488	68	9	2	1			10,446	
992006	867	680	404	123	16	5	1	77,272	
992013	879	664	384	108	14	4	1	72,115	
992026	743	701	584	436	304	166	1	355,102	F
992031	727	696	682	645	578	483	1	721,242	F
992064	716	519	313	117	32	7	1	75,667	
992108	918	724	115	9	2	1		36,425	
992170	842	814	724	524	338	154	1	372,905	
992937	944	915	803	247	43	3	1	115,966	
992942	843	596	19	3	1			27,279	
992944	873	858	781	361	17	1		126,921	
992971	886	698	73	3	2	1		33,112	
993032	853	689	124	7	3	1		37,21	
993034	896	626	122	6	3	1		34,691	
993079	723	510	156	7	4	1		34,975	
993104	942	924	887	814	703	533	1	776,118	
993292	716	529	311	110	15	4	1	66,622	
993742	969	967	965	962	99	99	1	273,113	
993807	921	801	119	15	1			37,97	
994084	670	516	300	67	5	1		53,317	
994107	389	352	310	173	28	1		74,172	
994111	318	188	77	6	3	1		19,553	
994403	957	551	70	14	1			28,143	
994622	973	829	401	26	3	1		51,056	
995520	950	892	340	26	2	1		53,021	
995559	990	980	943	716	411	17	1	246,307	
995562	967	952	875	538	106	6	1	167,102	
995584	509	452	294	5	3	1		42,863	
995671	544	284	42	3	3	1		20,542	
995676	388	334	307	185	51	3	1	84,422	
995777	283	271	208	100	29	3	1	59,47	

ANHANG B. DATEN

---

995791	64	51	32	16	2	1		11,042
995969	967	256	13	2	1			16,518
995978	989	737	36	3	1			25,64
996436	927	643	137	11	4	1		37,212
996926	983	972	954	757	182	3	1	193,308
996986	952	944	944	941	941	940	1	1133,945
997220	959	931	712	196	10	2	1	97,463
997247	302	276	205	45	6	1		38,655
997254	219	142	72	26	5	1		21,208
997271	330	297	187	12	3	1		31,533
997304	681	638	588	467	287	31	1	238,862
998236	841	751	619	439	266	45	1	234,479
998318	854	815	539	77	10	1		74,401
998444	912	872	804	510	209	4	1	195,041
998534	53	44	35	14	4	1		11,932
998876	923	874	643	120	9	1		85,451
998950	887	869	817	639	377	37	1	277,501

---

## Anhang C

# Integration in den ibaAnalyzer

Mit dem ibaAnalyzer, einem Tool der iba AG, lassen sich Messdaten jeglicher Art offline analysieren. Zum Laden und Speichern der Messdaten wird das hauseigene Dateiformat `.dat` verwendet. Die Messdaten lassen sich mit Anweisungsfolgen, die sowohl aus einfachen Berechnungen, wie auch komplexen Funktionsaufrufen bestehen, analysieren. Solche Anweisungsfolgen werden als Analysen bezeichnet und lassen sich in dem Analysedateiformat `.pdo` speichern.

Der FFT-Alignmentalgorithmus mit Gitterheuristik ist seit der Version 5.13.0 in den ibaAnalyzer integriert. Der Funktionskopf des FFT-Alignmentalgorithmus ist dort als

```
XALIGNFFT([Fixed],[Align],[Offset],[Scale],[Step],[QFactor])
```

definiert. Dabei sind die Parameter analog zu den Bezeichnungen in dieser Arbeit zu verstehen:

**Fixed** Feste Messreihe bzw. Fixedseries

**Align** Auszurichtende Messreihe bzw. Alignseries

**Offset** Suchintervall für die Verschiebung in x-Achseneinheiten der Fixedseries

**Scale** Suchintervall für den Skalierungsfaktor der Alignseries

**Step** Startschrittweite für die Gitterheuristik in Metern

**QFactor** Gütekriterium (0 = MQF, 1 = GLK, 2 = Korrelation)

Ein beispielhafte Analyse zum Alignment von längenbezogener Warmbandmessung [WB] und Kaltbandmessung [KB] könnte wie folgt aussehen. Hierbei gibt [i] das Ergebnis des Aufrufs in Zeile *i* an.

```

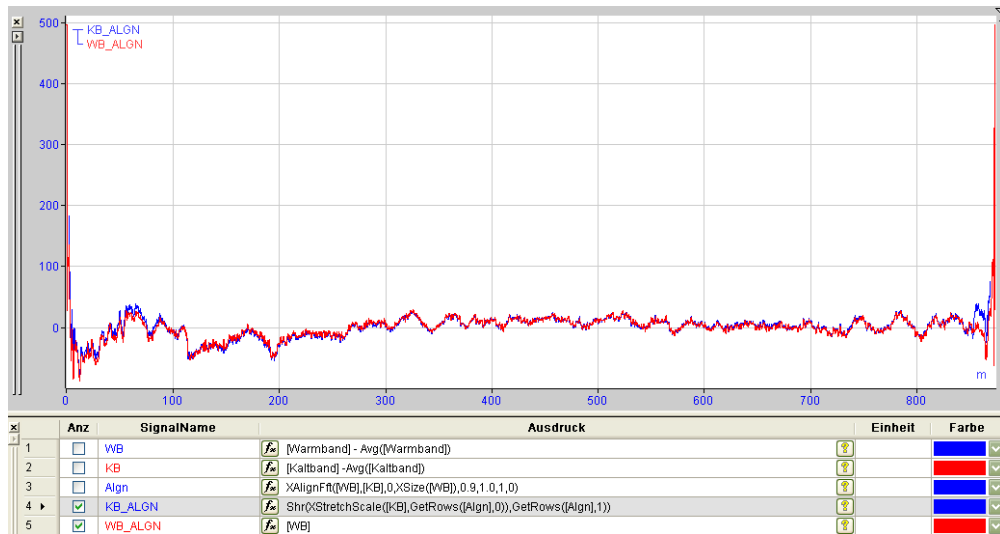
1: [WB] - Avg( [WB] ) // Mittelwertbereinigung
2: [KB] - Avg( [KB] )

3: XSIZE( [1] ) // Länge der Warmbandmessung in Metern
4: XALIGNFFT( [1] , [2] , 0 , [3] , 0.8 , 1.1 , 1 , 0 )

5: GETROWS( [4] , 0 ) // Auslesen der Skalierung (Arrayzugriff)
6: GETROWS( [4] , 1 ) // Auslesen des Offset (Arrayzugriff)
7: GETROWS( [4] , 2 ) // Auslesen des Gütekriteriums (Arrayzugriff)

8: XSTRETCHSCALE( [2] , [5] ) // Kaltband skalieren
9: SHR( [8] , [6] ) // Kaltband verschieben (Shift right)
    
```

Das Kernstück der Analyse ist der Funktionsaufruf des Alignmentalgorithmus. Dieser liefert als Ergebnis eine Art Array, das ich mit der Funktion `GetRows` auslesen lässt. An den entsprechenden Arraypositionen sind Skalierung, Offset und der Wert des Gütekriteriums vermerkt.



**Abbildung C.1:** Eine Analysevorschrift zum Alignment von Warm- und Kaltbandmessung im ibaAnalyzer (unten) und die visuelle Darstellung des Ergebnisses (oben)

# Abbildungsverzeichnis

1.1	Stahlcoil beim Aufhaspeln . . . . .	11
1.2	Bramme und warmgewalzter Stahl . . . . .	12
1.3	Coils im Freilager . . . . .	13
1.4	Skalierungsproblematik . . . . .	14
1.5	Coil mit deformierten Bandenden . . . . .	15
2.1	Transformation des Ortsgitters nach Skalierung . . . . .	20
2.2	Interpolationsfunktionen . . . . .	22
2.3	Optimales Alignment . . . . .	23
2.4	Chronologie durch Crossdating . . . . .	29
2.5	PAA-Komprimierungsstufen . . . . .	31
3.1	Aufbereitung des Dickenprofils . . . . .	37
3.2	Dicken- und Geschwindigkeitsmessung . . . . .	38
4.1	Überlappungsbereich . . . . .	42
4.2	Dickenschriebe mit verschiedener Messgeräteaflösung . . . . .	55
4.3	Kleinster Fehler für jeden Skalierungsfaktor . . . . .	58
4.4	Fehlerauswertungen bei Gitterheuristik . . . . .	59
4.5	Laufzeit für verschiedene Startschrittweiten . . . . .	62
4.6	Shotgunverfahren mit Schnipseln . . . . .	63
4.7	Messstörung durch Ablagerungen . . . . .	64

5.1	Deformierte Außenwindungen . . . . .	67
5.2	Visualisierung des Fußschrott . . . . .	69
6.1	Falschtreffer bei unkomprimierter Zuordnung . . . . .	83
A.1	Übersicht Warmbreitbandstraße . . . . .	90
A.2	Übersicht Beiz-Tandem-Straße . . . . .	91
A.3	Übersicht Produktionsprozess . . . . .	92
C.1	Analysevorschrift im ibaAnalyzer . . . . .	100

# Literaturverzeichnis

- [Anh07] H. Anhaus. Verfahren und Vorrichtung zur Identifizierung eines Teilstücks eines Halbzeugs. Schutzrecht DE102006006733B3, iba AG, D-90762 Fürth, August 2007.
- [Bac08] H. Bachl. Fingerprintmethode bei der voestalpine Stahl. Präsentation auf dem iba-Tag 2008, voestalpine Stahl GmbH / Anlagentechnik Warmwalzwerk, 2008.
- [Bär96] G. Bär. *Geometrie*. Teubner, 1996.
- [BC94] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. *AAAI-94 Workshop on Knowledge Discovery in Databases*, pages 229–248, 1994.
- [But05] T. Butz. *Fouriertransformation für Fussgänger*. Teubner, 2005.
- [CT65] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19(90):297–301, 1965.
- [Dey00] Deyrmon. *La Salamandre*, 141, 2000. <http://www.fe.ethz.ch/lab/Crossdating>.
- [DFE<sup>+</sup>07] M. Degner, R. Fandrich, G. Endemann, T. Ghenda, K. Letz, B. Lungen, I. Steller, H.-J. Wieland, A. Winkhold, R. Bartos, and R. Winkelgrund. *Stahlfibel*. Verlag Stahleisen GmbH, Düsseldorf, 2007.
- [FJ05] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on Program Generation, Optimization, and Platform Adaptation.
- [For77] O. Forster. *Analysis 2: Differentialrechnung im  $\mathbb{R}^n$ , gewöhnliche Differentialgleichungen*. Vieweg, 1977.

- [GGK<sup>+</sup>08] H. Göttler, T. Gottron, M. Kemeter, C. Lipowsky, and E. Schömer. Alignment and comparison of two data series with different horizontal scaling. 2008. [noch in Arbeit; Stand Oktober 2008].
- [Gno06] H. Gnoyke. Genealogie von Stahlband-Fingerprints. Diplomarbeit, Universität Lübeck, 2006.
- [HB06] M. Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Teubner, 2006.
- [Hub43] B. Huber. Über die Sicherheit jahrringchronologischer Datierung. *Holz als Roh-und Werkstoff*, 6(10):263–268, 1943.
- [Pfa63] J. Pfanzagl. Über die Parallelität von Zeitreihen. *Metrika*, 6(1):100–113, 1963.
- [PTVF02] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 2002.
- [Rem92] R. Remmert. Kapitel 4. Fundamentalsatz der Algebra. *Zahlen*, 1992.
- [RK04] C.A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. *3rd Workshop on Mining Temporal and Sequential Data, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.
- [TSI08] M. Toyoda, Y. Sakurai, and T. Ichikawa. Identifying Similar Subsequences in Data Streams. *Lecture Notes in Computer Science*, 5181:210–224, 2008.
- [VAM<sup>+</sup>01] J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, et al. The Sequence of the Human Genome. *Science*, 291(5507):1304–1351, 2001.
- [vSGa] voestalpine Stahl GmbH. <http://www.voestalpine.com>. [Online; Stand Oktober 2008].
- [vSGb] voestalpine Stahl GmbH. expedition voestalpine. <http://www.expeditionvoestalpine.com>. [Online; Stand Oktober 2008].
- [Wen97] C. Wenk. Algorithmen für das Crossdating in der Dendrochronologie. Diplomarbeit, Freie Universität Berlin, 1997.