

**Algorithmische Anwendungen WS 2006/07**

Referat

# **Time Domain Harmonic Scaling**

**A\_grün\_Ala0607**

**Dennis Ostrop 11035320 ai582@gm.fh-koeln.de**  
**Daniel de Buhr 11035463 ai574@gm.fh-koeln.de**

# Inhaltsverzeichnis

1. Einführung.....	3
2. Time Domain Harmonic Scaling.....	4
2.1 Beschleunigung eines Signals.....	4
2.2 Dreieckmethode zur beschleunigten Wiedergabe.....	6
2.3 Verlangsamung eines Signals.....	8
2.4 Dreieckmethode zur verlangsamten Wiedergabe.....	9
3. Bestimmung der Periodendauer.....	11
4. Java Programm.....	13
5. Fazit.....	14
5. Quellen.....	15

# 1. Einführung

Time Stretching bezeichnet einen Prozess, der ein Signal verlangsamt bzw. beschleunigt, ohne die Frequenz zu beeinflussen. Das Signal klingt also bei langsamerer Wiedergabe nicht tiefer und bei schnellerer Wiedergabe nicht höher.

Anwendung findet dieses Verfahren bei Audiosoftware wie z.B. [Logic Pro](#) von Apple, wo sich sogenannte [Apple Loops](#) in Echtzeit an das Songtempo und der Tonhöhe eines Songs anpassen, oder Adobe Audition.



- 1 Zugriff auf alle Effekte und Instrumente
- 2 Spurweise Zuordnung der Rechenleistung vernetzter Computer
- 3 Speicherbare Kanalzüge einschliesslich Audioeffekt- und Softwareinstrumente Kombinationen pro Spur
- 4 Erweiterte QuickTime Unterstützung mit DV oder DVCPRO Ausgabe über FireWire
- 5 Samplegenaue Automation nahezu aller Parameter
- 6 Globale Spuren für Tonart, Tempo, Taktart, Marker, Video und mehr
- 7 Arrangementmodi für Shuffle und Auto-Crossfade
- 8 Flexible, objektorientierte Bearbeitung von Audio- und MIDI-Material
- 9 Partiturtranskriptionen in Echtzeit

Abbildung: Logic Pro von Apple

Time Stretching kann durch verschiedene Verfahren realisiert werden:

- Resampling
- Phase Vocoder
- Time Domain Harmonic Scaling

## 2. Time Domain Harmonic Scaling

Das Time Domain Harmonic Scaling Verfahren erlaubt es ein Signal im Zeitbereich harmonisch zu strecken bzw. zu stauchen, und wurde 1978 von Lawrence Rabiner und Ronald Schafer entwickelt. Um Signale zu manipulieren, bearbeitet das Verfahren das Signal abschnittsweise (Granularsynthese).

Diese Abschnitte dürfen nur maximal 40 ms lang sein, ansonsten würde unser Gehör die Manipulation des Signales bemerken, da es ein zeitliches Auflösungsvermögen von etwa 40 ms hat. Für den Menschen ist es also nicht möglich, zu unterscheiden, was während einer Zeitspanne von 40 ms vorher und was nachher passiert.

### 2.1 Beschleunigung eines Signals

Hier werden kurze Zeitabschnitte des Signals weggelassen, wodurch sich das Signal verkürzt.

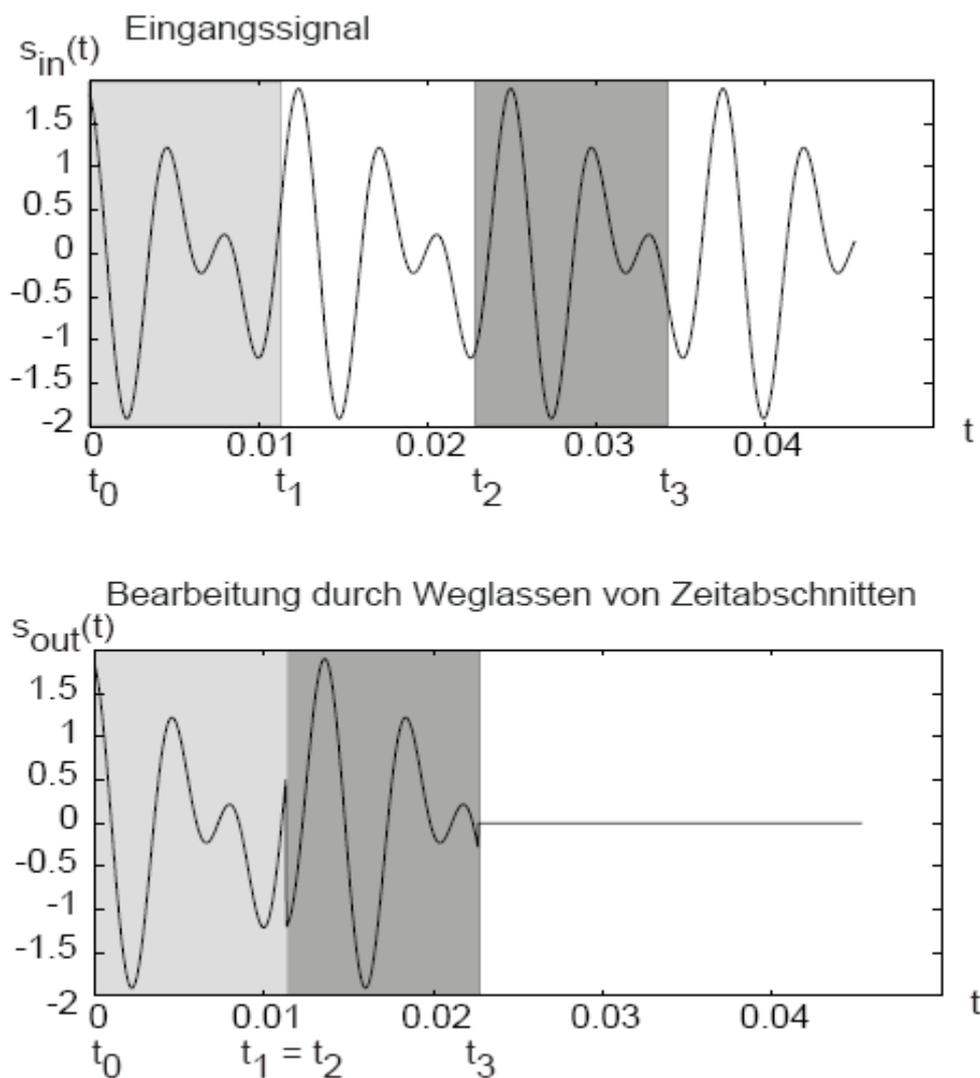
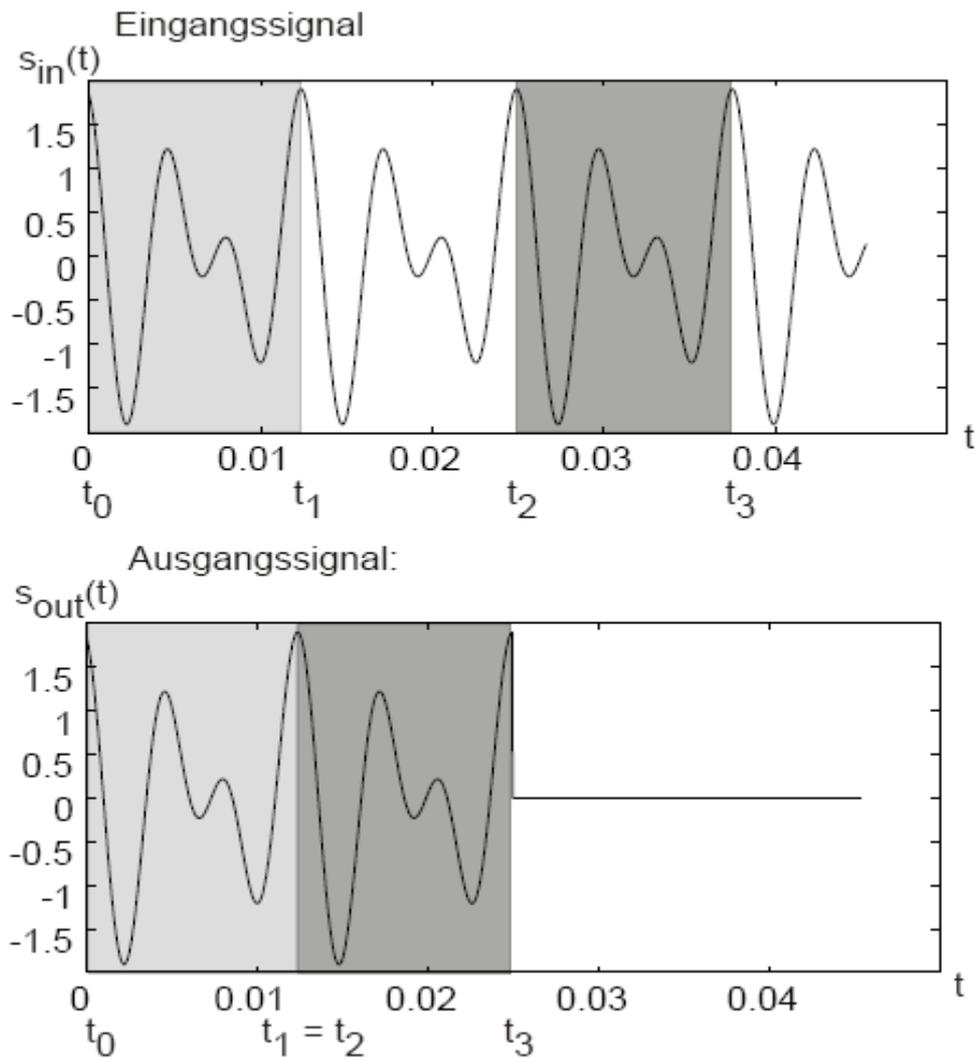


Abbildung: Beschleunigung eines Signals ohne Analyse der Periodendauer

Da bei einer willkürlichen Wahl der Abschnitte jedoch an den Enden der Abschnitte Sprünge entstehen können, muß das Signal vorher auf seine Periodendauer untersucht werden. Diese wird dann für die Abschnittslänge verwendet.



**Abbildung: Beschleunigung eines Signals mit Analyse der Periodendauer**

Ist das Eingangssignal periodisch, kann auf diese Weise das Signal um beliebige ganze Faktoren verkürzt werden, ohne die Tonhöhe zu verändern. Bei den natürlichen Signalen (Musik, Sprache) ergeben sich zusätzliche Schwierigkeiten, da sich keine zwei völlig identische Abschnitte finden lassen. Somit entstehen beim Zusammenfügen der Abschnitte dennoch wieder Sprünge.

Eine Lösung bietet die nun folgende Dreiecksmethode.

## 2.2 Dreieckmethode zur beschleunigten Wiedergabe

Da zwei aufeinanderfolgende Abschnitte nur ungefähr übereinstimmen, muss der zweite Abschnitt in den ersten eingeblendet werden.

1. Abschnitt A wird mit einer fallenden Dreieckfunktion (ausblenden) multipliziert:

$$s_1(i) = s(i) \cdot \left(1 - \frac{1}{L} \cdot i\right) \quad i = 1, 2, 3 \dots L$$

2. Abschnitt B wird mit einer ansteigenden Dreieckfunktion (einblenden) multipliziert:

$$s_2(i) = s(L + i) \cdot \frac{1}{L} \cdot i \quad i = 1, 2, 3 \dots L$$

3. Das Ausgangssignal ist die Überlagerung der zwei Abschnitte:

$$s_3 = s_1 + s_2$$

Dabei entspricht der Anfangswert des Ausgangssignals dem Anfangswert des ersten Abschnittes und der Endwert des Ausgangssignals dem Endwert des zweiten Abschnittes. Auf diese Weise werden Sprünge im Signalverlauf vermieden.

Da sich die einblendende und ausblendende Funktion ergänzen, bleibt die Lautstärke konstant. Mit dieser Methode wird die Beschleunigung um den Faktor zwei erreicht.

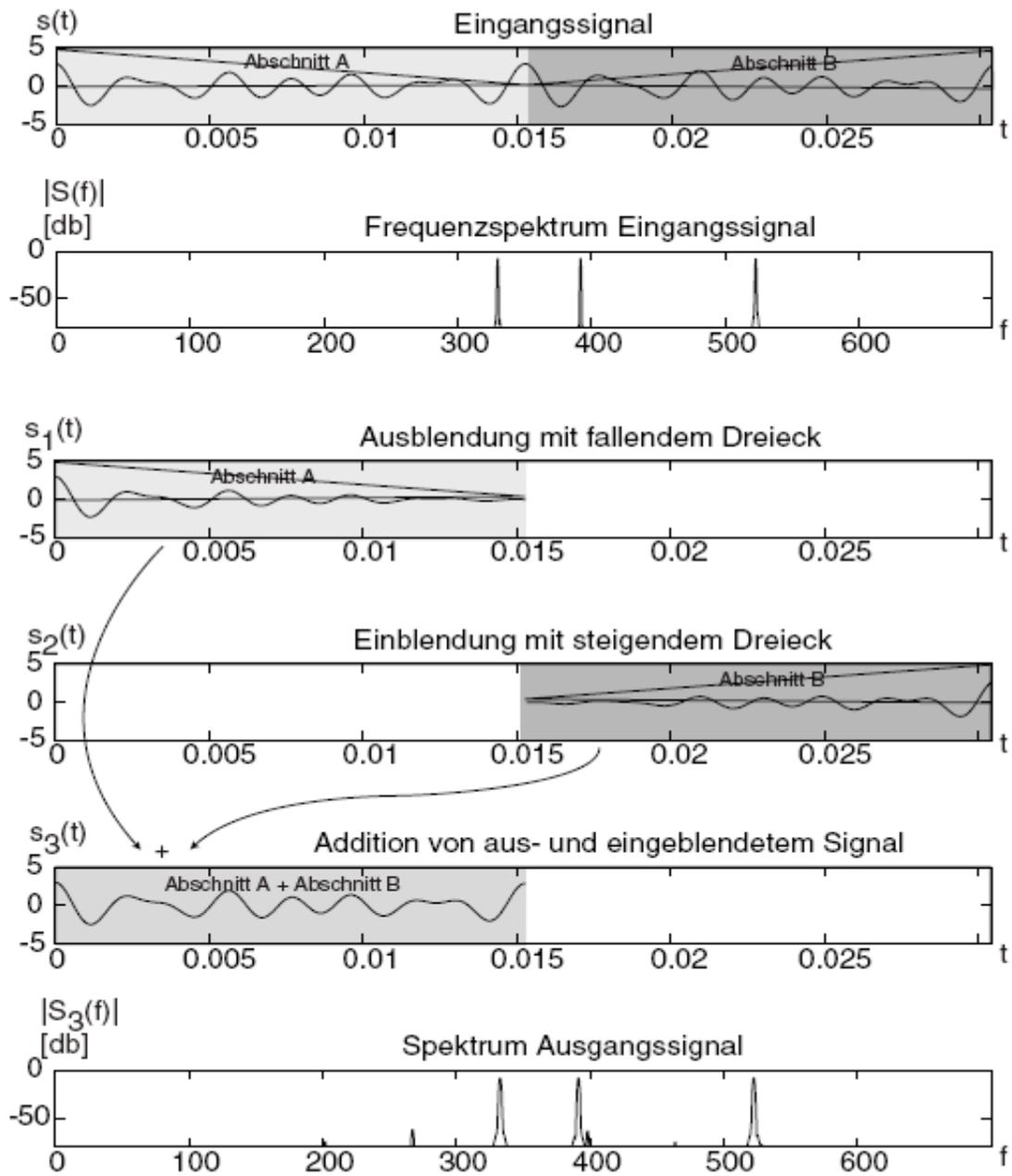


Abbildung: Anwendung der Dreieckfunktion

## 2.3 Verlangsamung eines Signals

Im Gegensatz zur Beschleunigung werden hier kurze Zeitabschnitte des Signals hinzugefügt, wodurch das Signal verlängert wird.

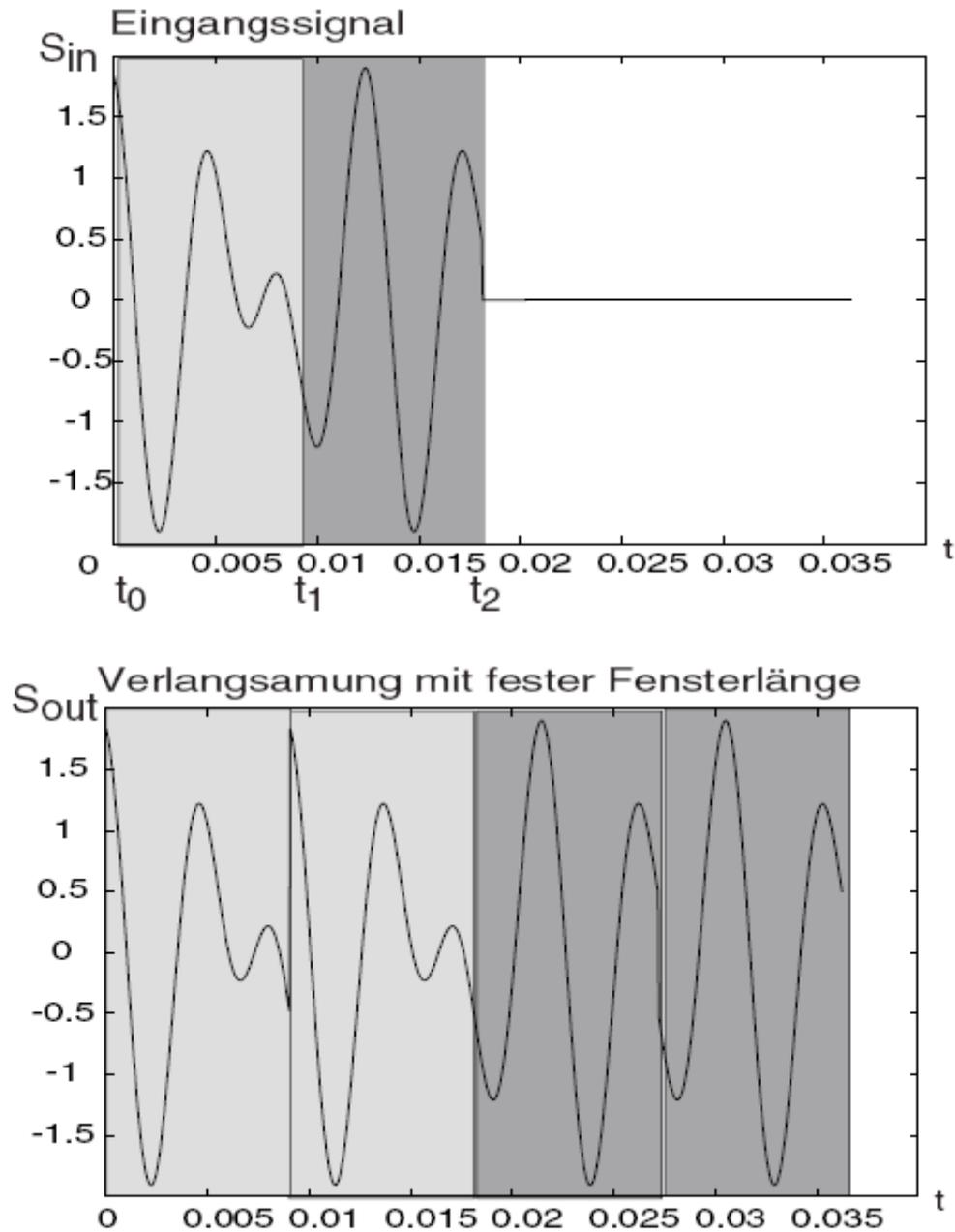


Abbildung: Verlangsamung eines Signals ohne Analyse der Periodendauer

Das Verdoppeln kurzer Zeitabschnitte des Signals führt auch hier zu Sprüngen und damit zur Verformung des Signals.

## 2.4 Dreiecksmethode zur verlangsamten Wiedergabe

Zuerst wird das Signal in Blöcke unterteilt. Um einen Block zu verdoppeln, wird sowohl der vorhergehende Block, der aktuelle Block, wie auch der folgende Block benutzt. Auf diese Weise entstehen die neu generierten Werte als eine Art "Signalinterpolation". Die maximale Blocklänge beträgt 40ms und wird, falls eine kürzere Periodendauer detektiert wird, kleiner gewählt.

1. Signal in der Periodendauer angepasste Blöcke unterteilen.
2. Abschnitt A beginnt eine Abschnittslänge vor des zu verdoppelnden Blockes und geht bis zum Ende des zu verdoppelnden Blockes. Somit hat Abschnitt A die zweifache Länge der aktuellen Periodendauer. Der ganze Abschnitt wird mit einem ansteigenden Dreieck multipliziert:

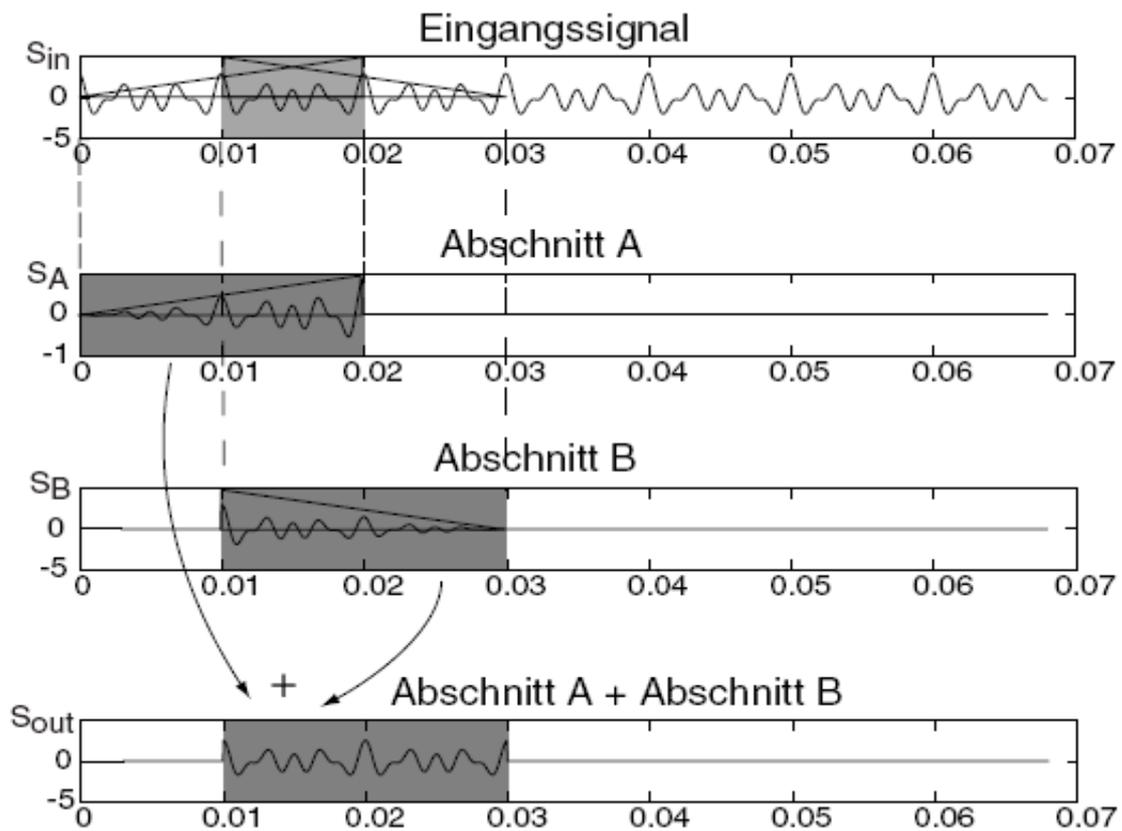
$$s_A(i) = s_{in}(i) \cdot \frac{i}{2L} \quad i = 1, 2, 3 \dots 2 \cdot L$$

3. Abschnitt B beginnt am Anfang des zu verdoppelnden Blockes und hat die zweifache Länge der aktuellen Periodendauer. Der ganze Abschnitt wird mit einem fallenden Dreieck multipliziert:

$$s_B(i) = s_{in}(i) \cdot \left(1 - \frac{i-L}{2L}\right) \quad i = L + 1, L + 2, L + 3 \dots 3 \cdot L$$

4. Die Überlagerung von Abschnitt A mit Abschnitt B ergibt das Ausgangssignal:

$$s_{out}(L + 1 + i) = s_A(i) + s_B(L + 1 + i) \quad i = 1, 2, 3 \dots 2 \cdot L$$



**Abbildung: Verdoppeln mit der Dreiecksmethode und angepasster Abschnittslänge**

Durch die Multiplikation mit den aufsteigenden bzw. fallenden Dreiecksfunktionen entspricht der Anfangswert des verdoppelten Blockes, sowie dessen Endwert genau den Werten des ursprünglichen Signales, und es entstehen keine Signalsprünge.

### 3. Bestimmung der Periodendauer

Wie bereits erwähnt, muss die beste Fensterlänge für einen Signalausschnitt bestimmt werden. Um die Periode eines Signals zu messen, wird die Autokorrelationsfunktion (AKF) angewandt. Ein periodischer Anteil lässt sich anhand der größten Amplitude im Signal feststellen. Das größte Maximum stellt eine dominante Periodizität dar. Damit ist die Position des Maximums die geeignete Fensterlänge.

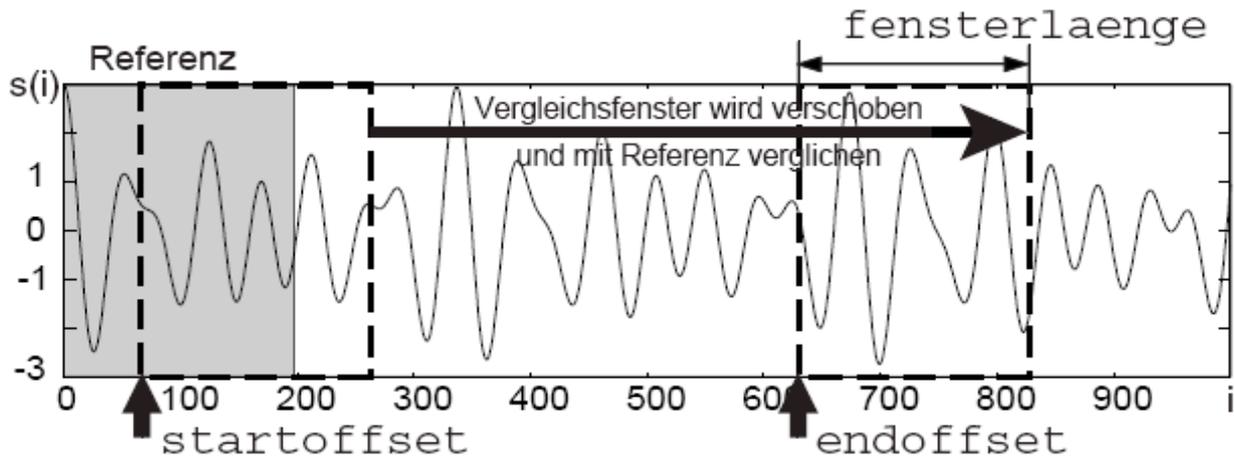


Abbildung: Periodensuche mittels Autokorrelationsfunktion

Die AKF entspricht einem Skalarprodukt der Samples in einem Referenzfenster und einem Vergleichsfenster. Begonnen wird mit der oben erwähnten Startverschiebung ( $n_{start}$ ) und rechnet bis zu einer Endverschiebung ( $n_{end}$ ). Für jede Verschiebung wird das berechnete Skalarprodukt in den Resultatvektor  $y$  geschrieben. Damit enthält der Resultatvektor  $y$  also die AKF mit Startverschiebung und hat eine Länge von  $n_{end} - n_{start}$ . Das Referenzfenster  $r$  enthält einen Ausschnitt der Länge  $n$  aus dem Eingangssignal  $s$ .

$$y(i) = \sum_{n=0}^{n_{end}-n_{start}} s(i+n) \cdot r(i+n)$$

Die Position  $x$  mit  $0 < x < (n_{end} - n_{start})$  des Maximums im Resultatvektor  $y$  ergibt zusammen mit der Startverschiebung die Fensterlänge  $L$  für die Verwendung im TDHS Algorithmus:

$$L = n_{start} + x$$

Die Lautstärke eines Signals kann sich von Abschnitt zu Abschnitt ändern. Solche Schwankungen haben keinen Einfluss auf die Verarbeitung mit dem TDHS Algorithmus. Bei der AKF wird das laute Signal aber stärker gewichtet als das leise. Da dies in diesem Fall unerwünscht ist, wird ein normiertes Skalarprodukt benutzt. Der Normierungsfaktor berechnet sich folgendermaßen:

$$\rho = \frac{1}{\sqrt{\sum_{i=0}^N v(i)^2 \cdot \sum_{i=0}^N r(i)^2}}$$

Hierbei ist  $v(i)$  der zu vergleichende Ausschnitt aus  $s(i)$ , und  $r(i)$  das Referenzfenster. Es wird für jede Verschiebung ein neuer Korrekturfaktor berechnet.

Um den interessanten Bereich der AKF deutlicher hervorzuheben, wird die AKF geglättet, indem die Eingangsdaten tiefpassgefiltert werden.

## 4. Java Programm

Das Java Programm ermöglicht es, Wav Samples zu laden, und mit dem Time Domain Harmonic Scaling Algorithmus zu halbieren bzw. zu verdoppeln. Die Suche der Periode mittels der Autokorrelationsfunktion kann mit drei Parametern beeinflusst werden. Der Parameter Start gibt die Startverschiebung, der Parameter Ende die Endverschiebung und der Parameter Länge die Fensterlänge des Vergleichsfenster an. Das bearbeitete Sample kann bei Bedarf als Wav Sample abgespeichert werden.

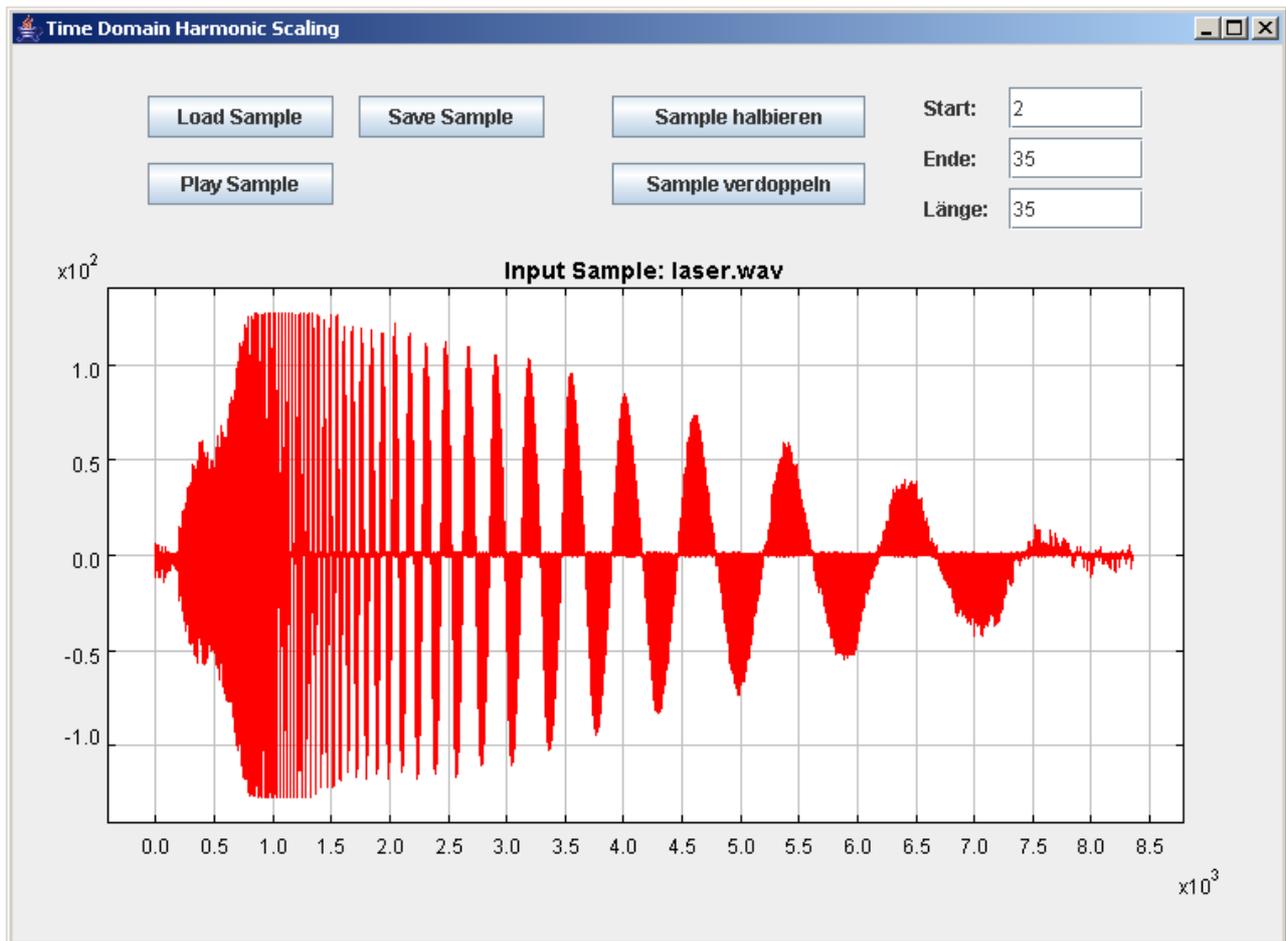


Abbildung: Das Time Domain Harmonic Scaling Programm in Java

## 5. Fazit

Die Qualität des Ausgangssignals hängt stark von der Ermittlung der Abschnittslänge ab. Signale, die einen periodischen Verlauf haben, können mit dieser Methode sehr gut manipuliert werden.

Klangelemente von kurzer zeitlicher Dauer und einer ausgeprägten Anschlagscharakteristik, wie beispielsweise Klicks, Drums und Percussion, sind nur schwer zu bearbeiten, da sie einen impulshaften Charakter besitzen und nicht periodisch sind. Durch die Bildung von max. 40ms langen Blöcken, ertönt die Anschlagscharakteristik zweimal hintereinander. Dies kann umgangen werden, wenn die maximale Länge eines Abschnittes verkürzt wird. Als Folge davon verliert das bearbeitete Signal jedoch stark an Bässen, was gegen kurze Abschnitte spricht. Man muss also das Optimum der Abschnittslänge bestimmen. Die Bestimmung der max. Abschnittslänge ist abhängig vom Signal.

Bei polyphonen Klangquellen bleibt das Einfügen bzw. Auslassen von Abschnitten ebenfalls nicht ohne hörbare Konsequenz. Bedingt durch die Mehrstimmigkeit enthält das Signal nicht nur eine einzelne Grundtonhöhe. Dadurch ist es schwer, einen periodischen Abschnitt zu finden.

Eine bessere Qualität kann durch Auftrennung des Signals in mehrere Frequenzbänder erreicht werden. Bei diesem verbesserten Verfahren ergeben sich allerdings Probleme an den Übergängen der Frequenzbänder und Schwierigkeiten bei der zeitlichen Synchronisation.

Der derzeitige Standard ist der DIRAC Algorithmus, der eine Zeit-Frequenz-Transformation verwendet. Er kombiniert die Vorteile des Time Domain Harmonic Scaling Verfahrens und des Phase Vocoder Verfahrens, wodurch er sich je nach Einstellung für periodisches oder auch für perkussives Material eignet. Er wird z.B. in der Audiosoftware WaveLab 6 der Firma Steinberg verwendet.

Genrell kann man sagen, dass eine genaue Kenntnis der Technologie benötigt wird, um ein optimales Ergebnis zu erzielen.

## 5. Quellen

- [1] *Introduction to Signal Processing*  
Sophocles J. Orfanidis  
Prentice Hall (1995)  
ISBN 0132091720
  
- [2] *C Algorithms for Real-Time DSP*  
Paul Embree  
Prentice Hall (1995)  
ISBN 0133373533
  
- [3] *Time Stretching And Pitch Shifting of Audio Signals*  
Stephan M. Bernsee  
<http://www.dspdimension.com/html/timepitch.html>
  
- [4] *Mit Helium gegen Godzilla - Zeitkorrektur mit Steinbergs WaveLab 6*  
Stephan M. Bernsee  
<http://www.musicianslife.de/2006/09/23/mit-helium-gegen-godzilla-zeitkorrektur-mit-steinbergs-wavelab-6-2/>
  
- [5] *Veränderung der Wiedergabegeschwindigkeit von Musiksignalen*  
Christian Bühler, Christian Liechti  
[http://www.medialab.ch/archiv/pdf\\_studien\\_diplomarbeiten/2sa99/2sa1999-106\\_WiedergabegeschwindigkeitVonMusik%20\(Buehler&Liechti\).pdf](http://www.medialab.ch/archiv/pdf_studien_diplomarbeiten/2sa99/2sa1999-106_WiedergabegeschwindigkeitVonMusik%20(Buehler&Liechti).pdf)
  
- [6] *Logic Pro*  
Apple  
<http://www.apple.com/de/logic/>