CWT

PROGRAMMDOKUMENTATION

Inhaltsverzeichnis

1 Überblick

Das Programmpaket besteht aus den beiden Programmen `cwt` und `ind`.

- `cwt` berechnet aus gemessenen Daten oder aus vordefinierten Funktionen die Wavelet-Transformation bezüglich vordefinierter Wavelets. Das Programm kann die Ergebnisse graphisch darstellen und legt außerdem Zusatzinformationen für die Weiterverarbeitung mit `ind` ab.
- `ind` wertet die Ausgabe von `cwt` aus und berechnet Indikatoren wie das Zeit-Energie-Spektrum, das Skalen-Energie-Spektrum oder die Entropie.

Welche Dateien diese Programmen erzeugen und einlesen, ist in Abbildung 1 dargestellt.

2 Installation

Die Programme mit zugehörigen C-Quellen holen Sie bitte aus dem zentralen Lager mit dem Befehl

```
cvs -d /home/zetem/daten/workshops_cvsroot/ checkout wavelet/methoden/grundlagen/cw
```

Das Verzeichnis `wavelet` mit seinen Unterverzeichnissen wird in Ihrem aktuellen Verzeichnis angelegt. Wechseln Sie in das Verzeichnis `cwt`

```
cd wavelet/methoden/grundlagen/cwt
```

und starten Sie dort ein Testbeispiel mit

```
gmake test
```

Ganz ähnlich erzeugen und betrachten Sie diese Dokumentation, indem Sie in das Unterverzeichnis `doc` des Verzeichnisses `cwt` wechseln und dort

```
cd doc
gmake show
```

starten.

3 `cwt`: Berechnung der kontinuierlichen Wavelet-Transformation

3.1 Kommandozeilenparameter

Das Programm `cwt` wird vollständig über die Kommandozeile gesteuert. Zu den meisten Optionen gibt es eine Kurzform, welche mit einem Minuszeichen beginnt und eine Langform, welche mit zwei Minuszeichen beginnt. Folgende Formen sind zulässig und bewirken das gleiche:

- `-p100`
- `-p 100`
- `--nodes 100`
- `--nodes=100`

Optionen, die keine weiteren Argumente erfordern, wie die Optionen `-?` oder `-g` können gesammelt nach einem einfachen Minus aufgeführt werden. Zum Beispiel `-?g`.

Argumente von Optionen dürfen keine Leerzeichen enthalten, diese werden andernfalls als Trennung zwischen zwei Optionen interpretiert.

- `-f 10, 100` - falsch
- `-f 10,100` - richtig

Die Reihenfolge, in der Optionen übergeben werden, beeinflusst deren Interpretation nicht.

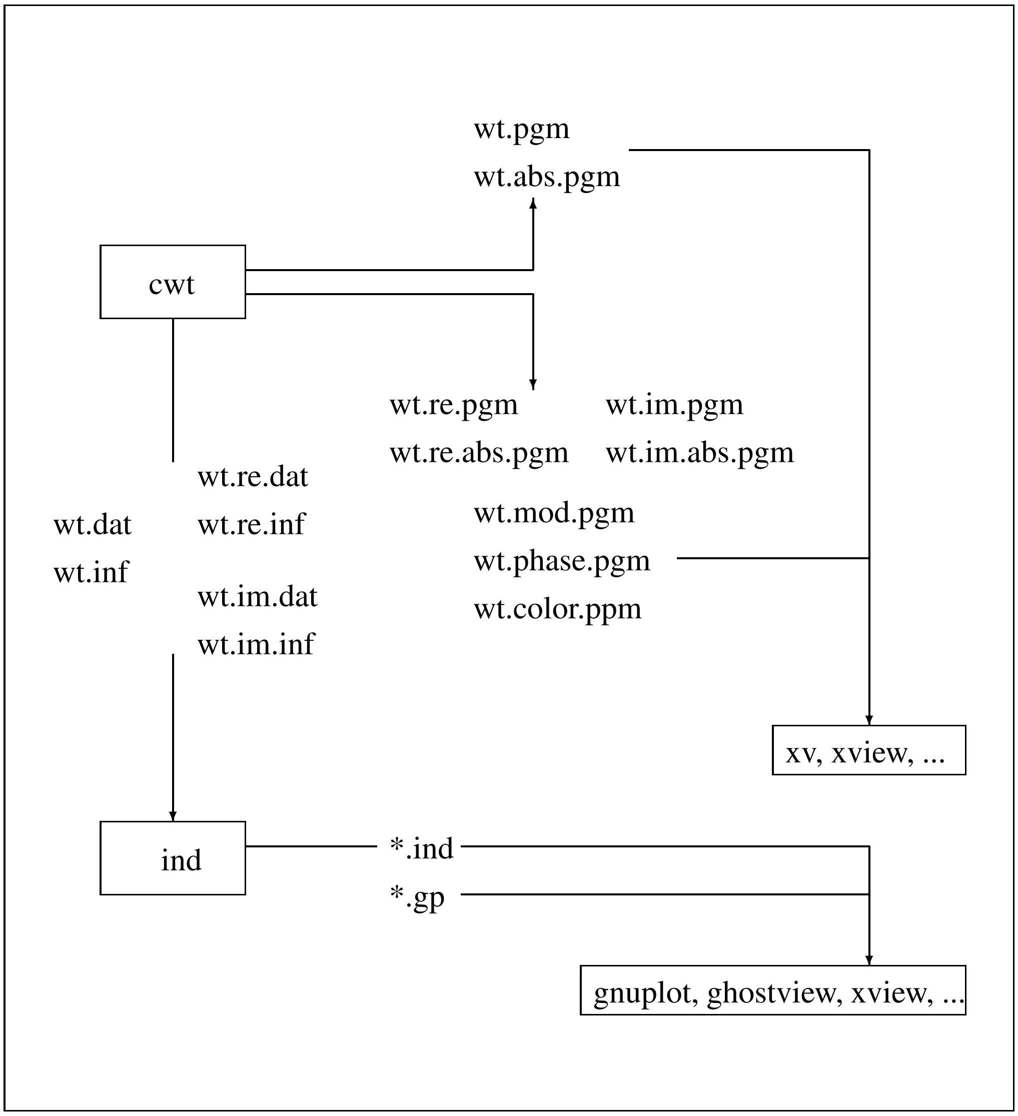

Abbildung 1: Datenfluss zwischen den Programmen `cwt` und `ind`

Allgemeines

`-?`, `--help`
Gibt eine kommentierte Liste aller Programmoptionen aus.

`--usage`
Gibt eine kurze Übersicht aller Programmoptionen aus.

`-V`, `--version`
Gibt die Versionsnummer des Programmes aus.

Eingabe und Ausgabe

`-s`, `--signal` *Name*
Name eines vordefinierten Signals, das transformiert werden soll. Eine Liste aller vordefinierten Signale wird bei Angabe der Option `-s ?` ausgegeben. Sie können sie außerdem in Abschnitt 3.2 finden.

`-i`, `--input` *Datei*
Alternativ kann ein gemessenes Signal transformiert werden. Dessen Abtastrate kann mit der Option `-r` angegeben werden. Das Format wird nach der Endung der Datei bestimmt.

Namensmuster	Format	Beispiel

datei`.dat`	Textdatei mit reeller Zahlenfolge	`0.1`
		`0.2`
		`0.3`
		`0.4`

datei`.cdat`	Textdatei mit komplexer Zahlenfolge	`0.1 + i 0.5`
		`0.2 + i 0.6`
		`0.3 + i 0.7`
		`0.4 + i 0.8`

datei`.raw`	Binärdatei mit vorzeichenbehafteten 16-Bit-Zahlen in Big-Endian-Bytefolge (könnte ja jemanden interessieren :-) welche rationale Zahlen im Bereich $[-1, 1)$ repräsentieren	$x_0, x_1, x_2, \ldots$
datei`.craw`	Analog Binärdatei mit komplexen Zahlen deren Real- und Imaginärteile verzahnt abgelegt sind.	$\Re(x_0), \Im(x_0),$ $\Re(x_1), \Im(x_1)$ $\Re(x_2), \ldots$

`-r`, `--samplerate` *Frequenz*
Die Rate mit der das gemessene Eingangssignal abgetastet wurde. Beispielsweise wird das Audiosignal für Aufnahmen einer CD mit einer Frequenz von 44100 Hz abgetastet. Möchte man ein solches Signal transformieren, übergibt man die abgetasteten Werte in einer Datei mit der Option `-s` und erklärt mit `-r 44100` dem Programm `cwt`, dass 44100 aufeinanderfolgende Abtastwerte einer Zeitdauer von 1 entsprechen (in diesem Falle 1 s). Bei kontinuierlichen Eingangsdaten (Option `-i`) wird das Signal um den angegeben Faktor gestaucht.

`-w`, `--wavelet` *Name*
Name eines vordefinierten Wavelets, mit dem transformiert werden soll. Bei komplexwertigen Wavelets wird eine komplexe Transformation durchgeführt. Eine Liste aller vordefinierten Wavelets wird bei Angabe der Option `-w ?` ausgegeben. Sie können sie außerdem in Abschnitt 3.3 finden. Die Verwendung von Wavelets in Form von Abtastwerten ist derzeit nicht vorgesehen.

`-o`, `--output` *Datei*
Name der Ausgabedatei oder `wt.dat`. Je nach Transformationsart (reell oder komplex) werden verschiedene andere Dateien mit gleichem Namensstamm erzeugt.

Reelle Transformation:

datei.dat	Waveletkoeffizienten
datei.pgm	Bild der Wavelettransformierten
datei.abs.pgm	Bild der Absolutbeträge der Waveletkoeffizienten

Komplexe Transformation:
Für den Imaginär- und den Realteil werden jeweils die drei Dateien wie bei der reellen Transformation erzeugt, jeweils bezüglich der erweiterten Dateinamen *datei*.re und *datei*.im - Das versteht doch kein Mensch oder ?
Hinzu kommen folgende Dateien:

datei.{im,re}.phase.pgm	Bild der Phaseninformation in den Waveletkoeffizienten
datei.{im,re}.mod.pgm	Bild der komplexen Absolutbeträge der Waveletkoeffizienten
datei.{im,re}.color.ppm	Bild in dem die Phasenlage jedes Wavelets durch eine Farbe auf dem Regenbogenspektrum und die Amplitude durch die Helligkeit des Bildpunktes repräsentiert wird

Vergleiche auch mit Abbildung 1

-g, --graphic
Wird diese Option angegeben, wird nach Abschluss jeder Berechnung das zugehörige Bild angezeigt. Die .pgm-Bilder werden aber in jedem Fall erzeugt.

-x, --viewer *Programm*
Grafikanzeigeprogramm mit welchem die Ergebnisbilder angezeigt werden sollen. Voreingestellt ist xv.

Skalen

-f, --frequencies *von,bis*
Angabe der niedrigsten und der höchsten Frequenz für die die Wavelettransformation durchgeführt werden soll. Hiermit wird gewissermaßen die senkrechte Begrenzung für den Ausschnitt aus dem Zeit-Frequenz-Diagramm festgelegt. Beispielsweise können Sie mit --frequencies 50,15000 den gesamten Frequenzbereich hörbarer Schallwellen analysieren – sofern Sie mit der Option -r die richtige Abtastrate Ihrer Daten eingestellt haben. Voreingestellt ist eine äquidistante Einteilung der Frequenzachse, also zum Beispiel 50, 60, 70, 80 . . . 14990, 15000.

-a, --scales *von,bis*
Alternative Angabe des Frequenzausschnittes über die Skalen. In cwt sind alle Wavelets auf die Bandmittenfrequenz 1 normiert, das heißt dass sich Skale und Frequenz reziprok zueinander verhalten. So bedeutet die Skale 10, dass die dominierende Schwingung eines Wavelets auf dieser Skale die Periode (Schwingungsdauer) 10 besitzt. Entsprechend untersucht man mit --scales 10,100 alle Skalen mit Schwingungen der Längen 10 bis 100. Voreingestellt ist auch hier eine äquidistante Einteilung, nun allerdings bezüglich der Skalen, im Beispiel 10, 20, 30, . . . , 90, 100.

-m, --na *Anzahl*
Wie viele Skalen sollen zwischen der kleinsten und der größten analysiert werden.
Beispiel:

```
cwt -a 10,20 -m 10 ...
```

bearbeitet die Skalen 10, 11, 12, 13, . . . , 19, 20.

-d, --scalediv *Typ*
Wie soll die vorgegebene Anzahl Skalen zwischen der kleinsten und der größten aufgeteilt werden.

-d linear -a 10,20 -m 10	$10, 11, 12, \ldots, 19, 20$
-d reciprocal -a 0.1,0.2 -m 5	$\frac{1}{10}, \frac{1}{9}, \frac{1}{8}, \ldots, \frac{1}{5}$
-d exponential -a 1,256 -m 8	$1, 2, 4, 8, \ldots, 256$

Translationen

`-b`, `--translations` *von,bis*
: Bereich der Waveletpositionen auf der Zeitachse. Hiermit kann man die waagerechten Begrenzungen für den Ausschnitt aus dem Zeit-Frequenz-Diagramm vorgeben. Der erste Abtastwert eines gemessenen Signals entspricht immer dem Zeitpunkt 0.

`-n`, `--nb` *Anzahl*
: Feinheit der Unterteilung des Zeitbereiches. Es wird immer äquidistant unterteilt.

Berechnung der Skalarprodukte Im Wesentlichen besteht die Wavelet-Transformation aus der Berechnung von Skalarprodukten des Signals mit verschobenen und skalierten Wavelets. In `cwt` sind zwei verschiedene Varianten für die numerische Integration implementiert.

1. Das Wavelet wird einmal mit einer festen Anzahl Stützstellen diskretisiert und die Skalarprodukte werden mit skalierten Version des Signals berechnet. Diese Vorgehensweise empfiehlt sich bei *kontinuierlichen* Eingangssignalen (Option `-s`), weil nur kontinuierliche Funktionen beliebig genau abgetastet werden können. Dies ist das voreingestellte Verfahren.

2. Das Wavelet wird für jede Skale neu diskretisiert so dass es zu der Abtastung des Eingangssignals passt. Diese Methode ist für *diskrete* Eingangssignale (Option `-i`) angeraten. Bei zu kleinen Skalen arbeitet das Verfahren nicht zufriedenstellend, was einfach daran liegt, dass das Signal nicht fein genug abgetastet wurde. Dieses Verfahren muss mit der Option `-u` angewählt werden.

Die Berechnung lässt sich mit diesen Optionen steuern:

`-u`, `--usesignalrate`
: Schaltet das zweite der oben beschriebenen Verfahren ein, also Orientierung an der Abtastung des Signals.

`-p`, `--nodes` *Anzahl*
: Falls Sie das erste Verfahren anwenden wollen, geben Sie hier an, an wievielen Stützstellen das Wavelet auf seinem (numerisch relevanten) Träger ausgewertet werden soll.

3.2 vordefinierte Signale

`silence`	Absolute Stille
`primitives`	ein linearer Anstieg und ein Plateau
`twosines`	je eine Sinusschwingungen mit einfacher und mit doppelter Frequenz
`cosine`	Cosinusschwingung
`cosineperiod1`	Cosinusschwingung mit Periode 1
`twocosines`	je eine Kosinusschwingung mit einfacher und zehnfacher Frequenz
`pulse`	ein kurzer Impuls
`plateau`	Die charakteristische Funktion des Intervalls [-5,5]
`heavyside`	Die charakteristische Funktion des Intervalls $[0, \infty)$
`cosinerolloff`	$\frac{\sin x}{x}$
`chirp`	Sinusschwingung ansteigender Frequenz
`doublechirp`	zwei überlagerte Sinusschwingung, eine fallend und eine steigend

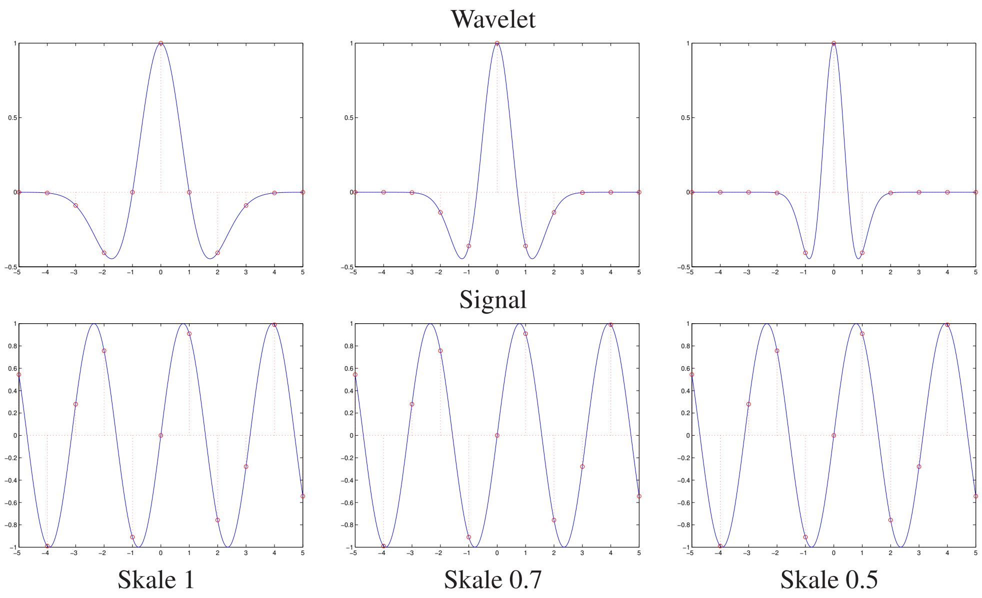

Abbildung 2: Die Wavelettransformation wenn sie mit der Abtastung des Signals arbeitet, `--usesignalrate`

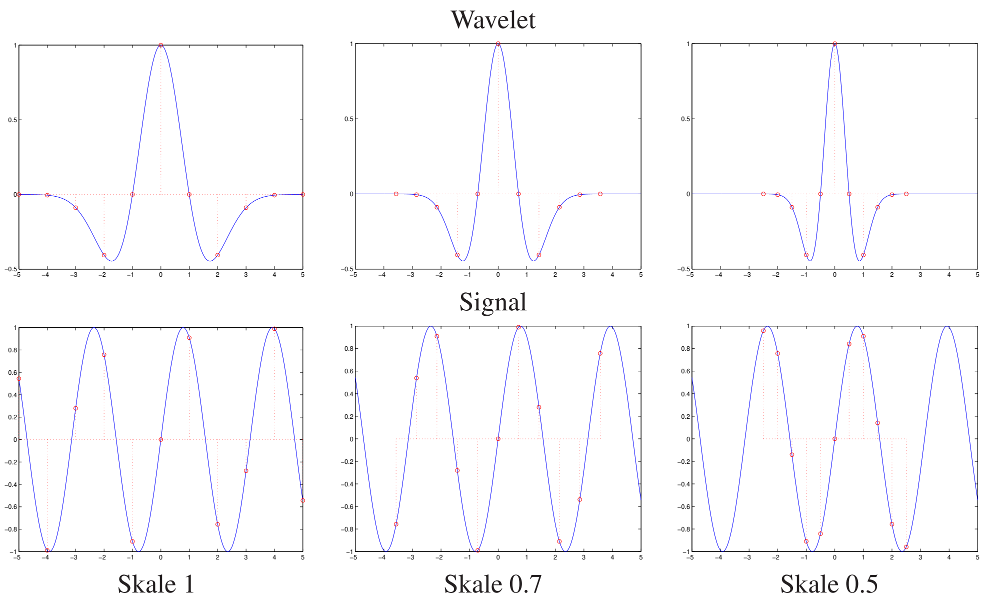

Abbildung 3: Die Wavelettransformation wenn sie mit fest diskretisiertem Wavelet arbeitet, es wurden 10 Wavelet-Stützstellen verwendet, `--nodes 10`

3.3 unterstützte Wavelets

`haar`	Haar-Wavelet
`mexicanhat`	Mexikanischer Hut
`sinus`	Sinuswelle
`poisson`	Poisson-Wavelet
`cauchy`	Cauchy-Wavelet beliebiger Ordnung, zusätzliche Parameter siehe unten
`spinning`	Verallgemeinertes Cauchy-Wavelet für die Analyse von Spinnrotordaten
`morlet`	Morlet-Wavelet mit unabhängig wählbarer Skale, zusätzliche Parameter siehe unten
`bspline`	B-Spline-Wavelet, zusätzliche Parameter siehe unten

Folgende Wavelets unterstützen zusätzliche Modifikationen:

- `-w cauchy,`*Ordnung*
  Das Cauchy-Wavelet n. Ordnung ψ_n ist als n. Ableitung einer rationalen Funktion, dem Cauchy-Kern C, definiert:

$$\begin{aligned} C(t) &= \frac{1}{2\pi(1-it)} \\ \psi_n(t) &= \frac{\mathrm{d}^n}{i^n\,\mathrm{d}\,t^n}C(t) \\ &= \frac{n!}{2\pi(1-it)^{n+1}} \end{aligned}$$

- `-w spinning,`μ_1, μ_2
  Entspricht einem Cauchy-Wavelet mit anderer Parametrisierung.

$$\begin{aligned} \psi(t) &= (1-ikt)^\alpha \\ \alpha &= -\frac{1}{2}+\frac{\mu_2}{k}+i\mu_1 \end{aligned}$$

  k wird intern so gewählt, dass das Wavelet die Bandmittenfrequenz 1 besitzt.

  – μ_1 entspricht einer Verzerrung, durch die das Wavelet weniger symmetrisch wird und dadurch so ähnlich wirkt, als ob das Wavelet verschoben worden wäre.
  – μ_2 entspricht der Güte des Filters. Je höher der Wert desto schärfer werden die Frequenzen aufgelöst.

- `-w morlet,`*Skale*
  Das Morlet-Wavelet besteht aus einer komplexwertigen Schwingung welche von einer Gaußschen Glockenkurve eingehüllt ist. Die Frequenz der komplexwertigen Schwingung wird über die Frequenzoptionen `-a` oder `-f` eingestellt, während die Breite der einhüllenden Glockenkurve im Verhältnis zu dieser Schwingungsperiode angegeben wird. Bei `-w morlet,2` ist die Einhüllende doppelt so breit wie bei `-w morlet,1`. Zur Verinfachung kann `-w morlet` an Stelle von `-w morlet,1` verwendet werden.

$$\psi(t) = e^{2\pi it-t^2}$$

- `-w bspline,`*Breite 1, Breite 2, …*
  Dieses Wavelet besteht ähnlich dem Morlet-Wavelet aus einer komplexwertigen Schwingung welche von einem B-Spline eingehüllt ist. Diese besondere Form ermöglicht eine besonders effiziente Berechnung der Wavelet-Transformation. Das B-Spline wird durch Faltung mehrerer charakteristischer Funktionen erzeugt, deren Breiten in einer Liste nach dem Wort `bspline` angegeben werden müssen. `-w bspline,1` entspricht zum Beispiel einem Wavelet, dessen Einhüllende (eine charakteristische Funktion) genau so breit wie eine Schwingung ist. Weitere Beispiele sind in Abbildung 4 zu sehen.

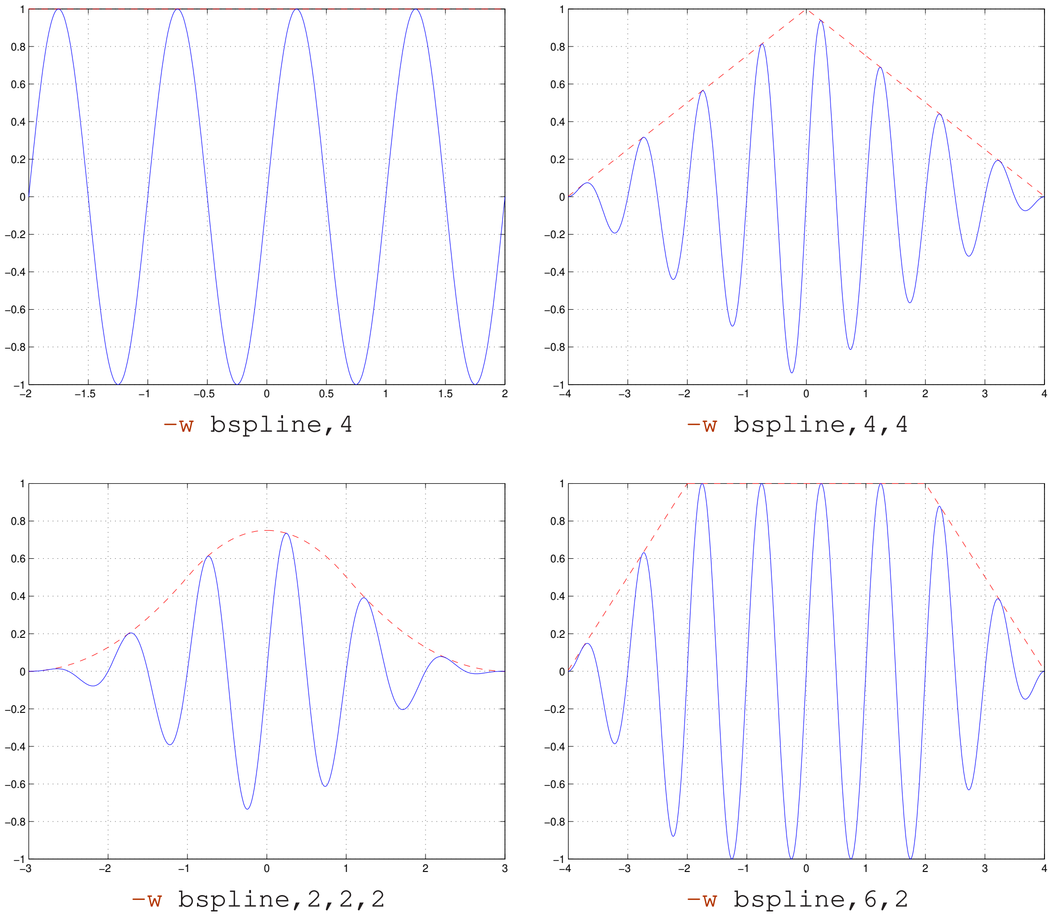

Abbildung 4: Beispiele für B-Spline basierte Wavelets, abgebildet sind die Imaginärteile der Wavelets und ihre einhüllenden B-Splines

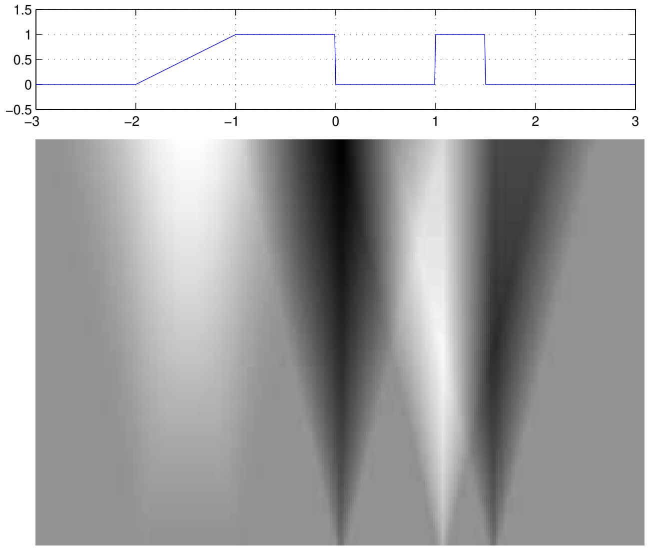

Abbildung 5: Vordefinierte Funktion `primitives` und ihre Wavelettransformierte

4 Beispiele

4.1 Kantenerkennung mit dem Haar-Wavelet

Mit der Kommandozeile

```
gmake examplehaar
```

wird `cwt` mit folgenden Parametern gestartet:

```
cwt -s primitives -w haar -a 0.1,2 --na 200 -b -3,3 --nb 300 -g
```

`-s primitives`	Untersuche eine der vordefinierten Funktionen, welche aus Sprüngen und einem linearen Anstieg besteht, wie in Abbildung 5 dargestellt.
`-w haar`	Benutze zur Analyse das Haar-Wavelet, welches auf feinen Skalen das Signal näherungsweise differenziert und daher Sprünge gut sichtbar macht.
`-a 0.1,2`	Analysiere mit Haar-Wavelets mit Breiten 0.1 bis 2.0
`--na 200`	Wähle 200 Zwischenschritte zwischen 0.1 und 2.0
`-b -3,3`	Untersuche das Signal auf dem Intervall $[-3, 3]$ …
`--nb 300`	… und zwar an 300 Stellen
`-g`	gib das Ergebnis (siehe Abbildung 5) graphisch mit `xv` aus

4.2 Frequenzerkennung mit dem Morlet-Wavelet

Mit der Kommandozeile

```
gmake exampledoublechirp
```

wird `cwt` mit folgenden Parametern gestartet:

```
cwt -s doublechirp -w morlet --nodes 200 -f 0.02,0.2 --na 200 -b0,200
--nb 500 -g
```

`-s doublechirp`	Untersuche eine der vordefinierten Funktionen, welche aus zwei überlagerten Sinusschwingungen besteht. Beide Schwingungen ändern ihre Frequenz im Laufe der Zeit, die eine Schwingung oszilliert zunehmend schneller und die andere zunehmend langsamer.
`-w morlet`	Die komplexe Schwingung im Morlet-Wavelet stellt sicher, dass Wavelet und Schwingung im Signal unabhängig von der Phasendifferenz korrelieren, wenn die Frequenzen übereinstimmen. Bei reellen Wavelets erreicht man das nicht, da zwei Schwingungen nicht miteinander korrelieren, wenn sie um $90°$ in der Phasenlage abweichen. Das führt zu Schwingungen in der Wavelettransformierten. Stattdessen erzeugt die komplexe Wavelettransformation mit dem Morlet-Wavelet durchgehend Waveletkoeffizienten großen Betrages, wenn die Frequenzen von Wavelet und Schwingung im Signal übereinstimmen.
`--nodes 500`	Das Morlet-Wavelet wird konstant mit 500 Werten abgetastet. Diese relativ hohe Auflösung ist nötig, da auf großen Skalen die Abtastung des Wavelets immer noch feiner sein muss, als die höchste auftretende Frequenz im Signal, sonst tritt der Moire-Effekt auf, d.h. hohe Frequenzen im Signal werden nahezu willkürlich in tiefe Frequenzen verwandelt.
`-f 0.02,0.2`	Es werden die Frequenzen von 0.02 bis 0.2 untersucht, wobei die tiefen Frequenzen im Bild unten zu sehen sind.
`--na 200`	Es werden in diesem Frequenzbereich 200 verschiedene Frequenzen analysiert und zwar linear abgestuft, da keine anderen Angaben gemacht wurden.
`-b 0,200`	Zeitintervall $[0,200]$
`--nb 200`	unterteilt in 200 Schritte
`-g`	graphische Ausgabe siehe Abbildung 6

4.3 Analyse von Audiosignalen

Mit der Kommandozeile

```
gmake examplegiana
```

wird `cwt` mit folgenden Parametern gestartet:

```
cwt -i examples/giana.raw -r 44100 -u -w bspline,10,10 -f 110,1760
--scalediv exponential --na 240 -b 0,13 --nb 1000 -g
```

`-i examples/giana.raw`	Lies den Datensatz `examples/giana.raw` ein.
`-r 44100`	Der Datensatz wurde von einer CD eingelesen und hat der daher die Abtastrate 44100 Hz.
`-u`	Diskretisiere Wavelet für jede Skale neu und zwar auf dem Raster des diskreten Eingangssignals.
`-w bspline,10,10`	Benutze ein Wavelet bestehend aus einem linearen Spline. Das Wavelet umfasst 20 Perioden der eingehüllten Schwingung.
`-f 110,1760`	Analysiere die Frequenzen vom Ton A bis zum Ton a'''

`--scalediv exponential`	Exponentielle Frequenzeinteilung bedeutet, dass Tonintervalle unabhängig von der Frequenz den gleichen Zeilenabstand in der Wavelettransformierten besitzen.
`--na 240`	Das untersuchte Frequenzband umfasst 4 Oktaven, also $4 \cdot 12 = 48$ Töne. Damit sind zwei benachbarte Töne in der Wavelettransformierten $\frac{240}{48} = 5$ Zeilen voneinander entfernt.
`-b 0,13`	Von dem Signal sollen die ersten 13 Sekunden untersucht werden …
`--nb 1000`	… und zwar zu 1000 verschiedenen Zeitpunkten.
`-g`	Im Anzeigeprogramm sollte man im Farbeditor die dunklen Farben verstärken (γ-Korrektur), da sonst der Kontrast zu schwach ist. Vergleiche mit Abbildung 7.

5 ind: Berechnung verschiedener Indikatoren

Das Programm `ind` liest eine von `cwt` erzeugte Wavelettransformierte `*.dat` ein und berechnet daraus einen Indikator wie das Zeit-Energie-Spektrum, Skalen-Energie-Spektrum oder die Entropie. Die Zahlenlisten (Ausgaben von `ind`) haben die Endung `.ind`.

5.1 Kommandozeilenparameter

Für die Kommandozeilenoptionen von `ind` gilt das gleiche wie für das Programm `cwt` in Abschnitt 3.1.

Allgemeines

`-?, --help`
Gibt eine kommentierte Liste aller Programmoptionen aus.

`--usage`
Gibt eine kurze Übersicht aller Programmoptionen aus.

`-V, --version`
Gibt die Versionsnummer des Programmes aus.

Eingabe und Ausgabe

`-i, --input` *Datei*
Eine Datei mit der Endung mit der Endung `.dat`, welche die Wavelettransformierte eines Signals enthält so wie sie vom Programm `cwt` ausgegeben wird.

`-g, --gnuplot` *Programm*
Programm welches mit Hilfe der GNUplot-Skripte `*.gp` die Diagramme im Postscript-Format erzeugt. Voreingestellt ist `gnuplot`.

`-p, --psviewer` *Programm*
Anzeigeprogramm für die erzeugten Diagramme im Postscript-Format, wie z.B. Ghostscript `gs` oder Ghostview `gv`. Voreingestellt ist `gv`.

`-x, --imgviewer` *Programm*
Grafikanzeigeprogramm mit welchem die Ergebnisbilder angezeigt werden sollen. Voreingestellt ist `xv`.

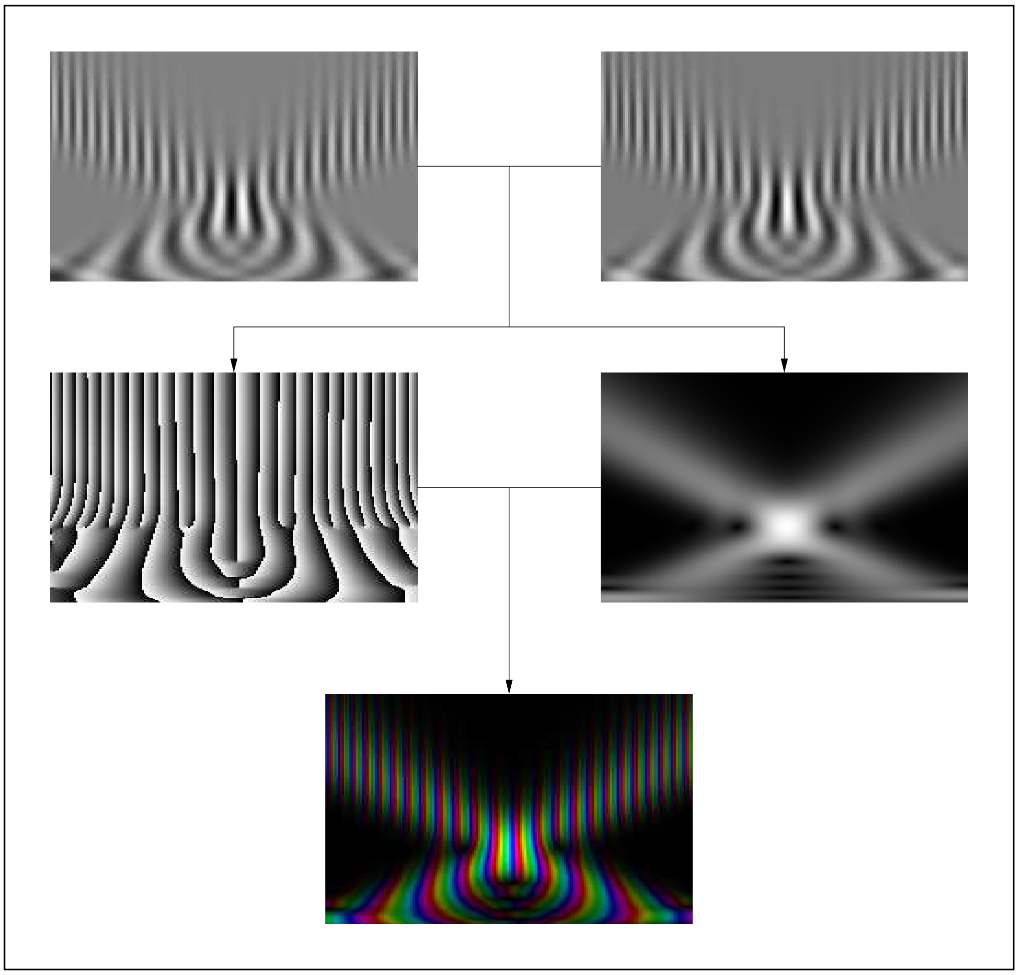

Abbildung 6: Analyse der `doublechirp`-Funktion

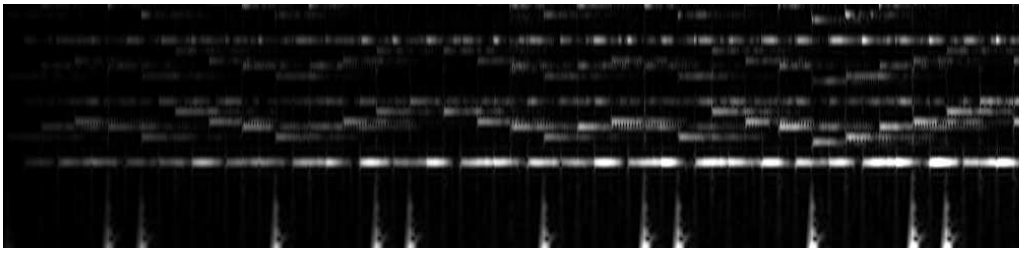

Abbildung 7: Analyse der Audiodaten `examples/giana.raw`

Berechnung

`-d`, `--indicator` *Name*
Indikator, der aus der Wavelet-Transformierten berechnet werden soll.

Folgende Indikatoren sind implementiert:

- `nf` - Frequenzen-Energie-Spektrum

$$n_{\mathrm{f}}(x) = \int_{\mathbb{R}} \left| L_\psi f\left(\tfrac{1}{x}, b\right) \right|^2 \mathrm{d}\, b$$

- `na` - Skalen-Energie-Spektrum

$$n_{\mathrm{a}}(x) = \int_{\mathbb{R}} \left| L_\psi f(x, b) \right|^2 \mathrm{d}\, b$$

- `nb` - Zeit-Energie-Spektrum

$$n_{\mathrm{b}}(x) = \int_{\mathbb{R}} \left| L_\psi f(a, x) \right|^2 \mathrm{d}\, a$$

- `ef` - Frequenz-Entropie-Spektrum

$$E_{\mathrm{f}}(x) = -\int_{\mathbb{R}} \left| L_\psi f\left(\tfrac{1}{x}, b\right) \right| \cdot \log \left| L_\psi f\left(\tfrac{1}{x}, b\right) \right| \mathrm{d}\, b$$

- `ea` - Skalen-Entropie-Spektrum

$$E_{\mathrm{a}}(x) = -\int_{\mathbb{R}} \left| L_\psi f(x, b) \right| \cdot \log \left| L_\psi f(x, b) \right| \mathrm{d}\, b$$

- `eb` - Zeit-Entropie-Spektrum

$$E_{\mathrm{b}}(x) = -\int_{\mathbb{R}} \left| L_\psi f(a, x) \right| \cdot \log \left| L_\psi f(a, x) \right| \mathrm{d}\, a$$

- `pf` - Skalen-Energie+Zeit-Energie

$$n_{\mathrm{a,b}}(x) = \sqrt{n_{\mathrm{a}}(x)} + \sqrt{n_{\mathrm{b}}(x)}$$

5.2 Gnuplot Skripte

Das Programm `ind` benutzt `gnuplot` um die berechneten Indikatorfunktionen graphisch auszuwerten. Dazu erzeugt `ind` die Datei `ind.gp` welche `gnuplot` steuert:

```
set terminal postscript eps     # latex
set output "wt.mod.nf.eps"
set grid
set title "Frequency energy spectrum"
set xlabel "frequency f"
set ylabel "nf(f)"
set logscale x
plot "wt.mod.nf.ind" using 1:2 with lines      # points
```

Soll etwas an der Ausgabe der graphischen Darstellung geändert werden, muss man daher das C++-Programm `ind.cc` ändern.

Das Programm `gnuplot` zeigt Funktionsgraphen allerdings nicht selbst an, sondern erzeugt nur Grafiken im Postscript-`*.ps`-Format. Um diese Dateien wiederum anzuzeigen, startet `ind` noch ein entsprechendes Anzeigeprogramm wie `gv` (siehe Option `-p`).

6 Fragen und Antworten

- *Ich habe zwei verschiedene Transformationen durchgeführt und finde, dass die Helligkeitsunterschiede zwischen den Bildern beider Transformationen nicht die Intensität der Signale widerspiegeln.*

  Die Intensitäten werden bei der Ausgabe als Bild auf die darstellbaren Helligkeiten normiert, daher kann man die Bilder nicht direkt vergleichen.