

Grundkurs Codierung

Lösungsvorschläge zu den Fragen in den Unterkapiteln „Was blieb?“

Stand 26.08.2025

Unterkapitel 6.2.7, Seite 404

Zu Frage 1:

Die merkwürdig erscheinende Antwort ist **nein** und **ja**, warum? Betrachten wir zunächst den allgemeinen Fall einer im Intervall $0 \leq t \leq T$ beliebigen stetigen Funktion $s(t)$, wie hier in Bild 1 oben links oder rechts. Teilt man das Intervall in $2N$ gleiche Teile $\Delta t = T/(2N)$ und misst in Abständen Δt die Funktionswerte $s(v \cdot \Delta t)$,

$$s(v) = s(v \cdot \Delta t), \quad v = 0, 1, 2, \dots, 2N-1,$$

so lassen sich daraus gemäß Unterkapitel 6.2.2.1 in Anlehnung an Seite 352 ff. mit den Transformationsformeln der DFT

$$a_\lambda = \frac{1}{2N} \sum_{v=0}^{2N-1} s(v) \cos \lambda \frac{v\pi}{N}, \quad b_\lambda = \frac{1}{2N} \sum_{v=0}^{2N-1} s(v) \sin \lambda \frac{v\pi}{N}, \quad \lambda = 0, \pm 1, \pm 2, \dots, \pm N-1$$

$$a_N = \frac{1}{4N} \sum_{v=0}^{2N-1} s(v) \cos \lambda \frac{v\pi}{N}, \quad b_{-N} = b_0 = b_N = 0$$

die Fourierkoeffizienten a_λ und b_λ berechnen (zu a_N bitte *Hinweis* am Ende beachten).

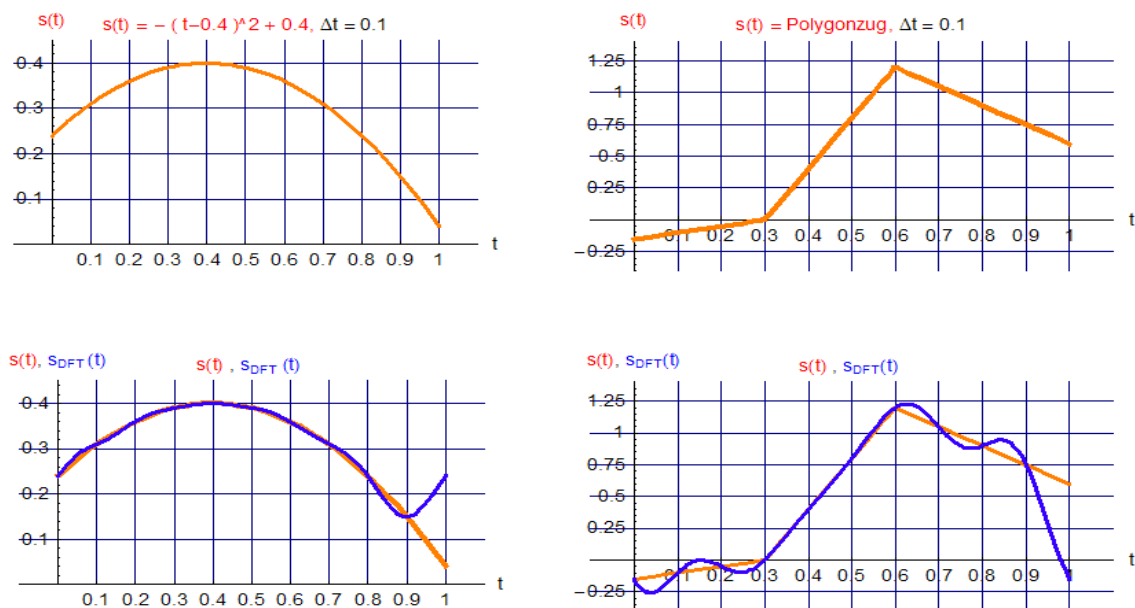


Bild 1: Zwei stetige Funktionen $s(t)$, ihre Abtastwerte in den 10 Punkten $0, \Delta t, 2\Delta t, \dots, 9\Delta t$ und die mit den Koeffizienten der DFT gebildeten Näherungsfunktionen $s_{DFT}(t)$.

Umgekehrt können mit diesen Koeffizienten über $s_{DFT}(v \cdot \Delta t)$

$$s_{DFT}(v) = a_0 + \sum_{\lambda=1}^N 2a_\lambda \cos \lambda \frac{v\pi}{N} + 2b_\lambda \sin \lambda \frac{v\pi}{N} = s(v \Delta t), \quad v = 0, 1, 2, \dots, 2N-1$$

wieder die Abtastwerte $s(\Delta t)$ der ursprünglichen Funktion $s(t)$ bestimmt werden – **und nur diese!** Betrachtet man nämlich die gesamte Funktion $s_{\text{DFT}}(t)$,

$$s_{\text{DFT}}(t) = a_0 + \sum_{\lambda=1}^N 2a_{\lambda} \cos \lambda \frac{2\pi}{T} t + 2b_{\lambda} \sin \lambda \frac{2\pi}{T} t ,$$

so weicht sie zwischen den Abtastpunkten von der Originalfunktion $s(t)$ ab, wie in den Diagrammen der unteren Reihe von Bild 1 zu erkennen ist, links nur wenig, rechts in den beiden Seitenästen schon deutlich. Benutzt man mehr Abtastwerte, etwa mit $2N = 20$ doppelt so viele, dann wird die Näherung in einigen Bereichen genauer, nicht aber notwendigerweise über alles, was Bild 2 verdeutlicht:

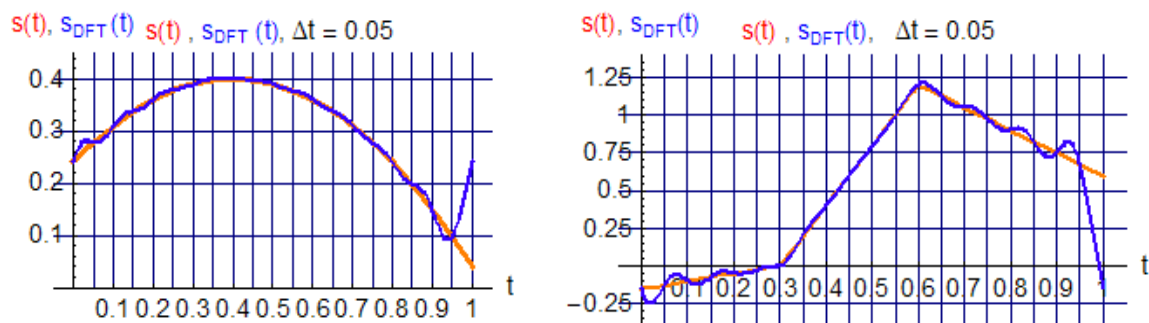


Bild 2: DFT-Näherungen mit doppelter Anzahl von Abtastwerten gegenüber den in Bild 1 verwendeten

Im Allgemeinen ist die Antwort auf unsere Frage also **„nein“**. Gibt es aber vielleicht Sonderfälle, und sei es welche mit nur theoretischer Bedeutung? Betrachten wir dazu als Funktion $s(t)$ eine Summe von je 4 sinus- und cosinus-förmigen Anteilen mit vorgegebenen Frequenzen und einem Gleichanteil:

$$s(t) = a_0 + \sum_{\lambda=1}^N 2a_{\lambda} \cos \lambda \frac{2\pi}{T} t + 2b_{\lambda} \sin \lambda \frac{2\pi}{T} t$$

Die tiefste Frequenz ist $f = 1/T = f_{\min} = 1 \text{ s}^{-1}$, die höchste $f_{\max} = 4 f_{\min} = 4 \text{ s}^{-1}$. Es wird hier also eine Abtastfrequenz von $2f_{\max}$ benötigt, um nach dem **Abtasttheorem** auch die schnellste Komponente $f_{\max}$ identifizieren zu können. Die 8 Messwerte folgen im Abstand $\Delta t = 0.125 \text{ s}$. Wegen der trigonometrischen Anteile ist $s(t)$ vollständig durch die Angabe von Gleichanteil a_0 und Amplituden a_{μ} , b_{μ} , $\mu = 1, 2, 3, 4$ festgelegt. Die Werte könnten, wie in Tabelle 1 angegeben, z. B. sein:

Frequenz	Original	DFT (N=4)	DFT (N=5)	Original	DFT (N=4)	DFT (N=5)
Gleichanteil	$a_0 = -0.3$	$as_0 = -0.3$	$as_0 = -0.3$	$b_0 = 0$	$bs_0 = 0$	$bs_0 = 0$
$1.0 \cdot f = f_{\min}$	$a_1 = 2.0$	$as_1 = 2.0$	$as_1 = 2.0$	$b_1 = 1.0$	$bs_1 = 1.0$	$bs_1 = 1.0$
$2.0 \cdot f$	$a_2 = 0.8$	$as_2 = 0.8$	$as_2 = 0.8$	$b_2 = -0.5$	$bs_2 = -0.5$	$bs_2 = -0.5$
$3.0 \cdot f$	$a_3 = 0.4$	$as_3 = 0.4$	$as_3 = 0.4$	$b_3 = 0.7$	$bs_3 = 0.7$	$bs_3 = 0.7$
$4.0 \cdot f$	$a_4 = 0.5$	$as_4 = 0.5$	$as_4 = 0.5$	$b_4 = -0.8$	$bs_4 = 0.0$	$bs_4 = -0.8$
$5.0 \cdot f$	---		$as_5 = 0.0$	---		$bs_5 = 0.0$

Tabelle 1: Amplituden der Originalfunktion $s(t)$ und der DFT bei $N = 4$ (8 Abtastwerte) und $N=5$ (10 Abtastwerte). Bei $N=5$ wird auch bs_4 richtig wiedergegeben.

Bild 3 zeigt in der oberen Reihe den Verlauf der Originalfunktion, die links mit $2N=8$, rechts mit $2N=10$ abgetastet wurde. Die untere Reihe stellt das Ergebnis der DFT dar. Nach den bisherigen Überlegungen sind in beiden Fällen auch hier die Werte in den Abtastzeitpunkten identisch. Links jedoch weicht das mit den DFT-Koeffizienten ermittelte Diagramm ab. Der Grund liegt darin, dass die Originalfunktion bei der Frequenz $4f$

einen Sinusanteil $b_4 = -0.8$ enthält, der aber bei der DFT als schnellste Komponente prinzipiell Null ist, siehe Transformationsformeln auf Seite 1. Hier wird eben immer im **Nulldurchgang** der maskierenden sin-Funktion abgetastet. Alle anderen DFT-Koeffizienten werden dagegen richtig bestimmt, so dass sich die gezeigte Abweichung ergibt. Dies ist auch aus Tabelle 1 zu erkennen.

Verkürzt man dagegen Δt , indem man das Intervall T nicht in $2N$, sondern in $2(N+1)$ Abschnitte für nun $2N+2$ Abtastwerte teilt, so wird der Koeffizient b_4 ebenfalls richtig berechnet, während a_5 natürlich Null ist (man würde mit dem kürzeren Δt eine weitere Komponente mit $5f$ identifizieren, falls sie vorhanden wäre). Damit verlaufen Original und DFT vollständig gleich, im Diagramm rechts unten wurde lediglich zur Darstellung künstlich ein Gleichanteil von 0.4 addiert, um die Funktionen optisch auseinander zu ziehen.

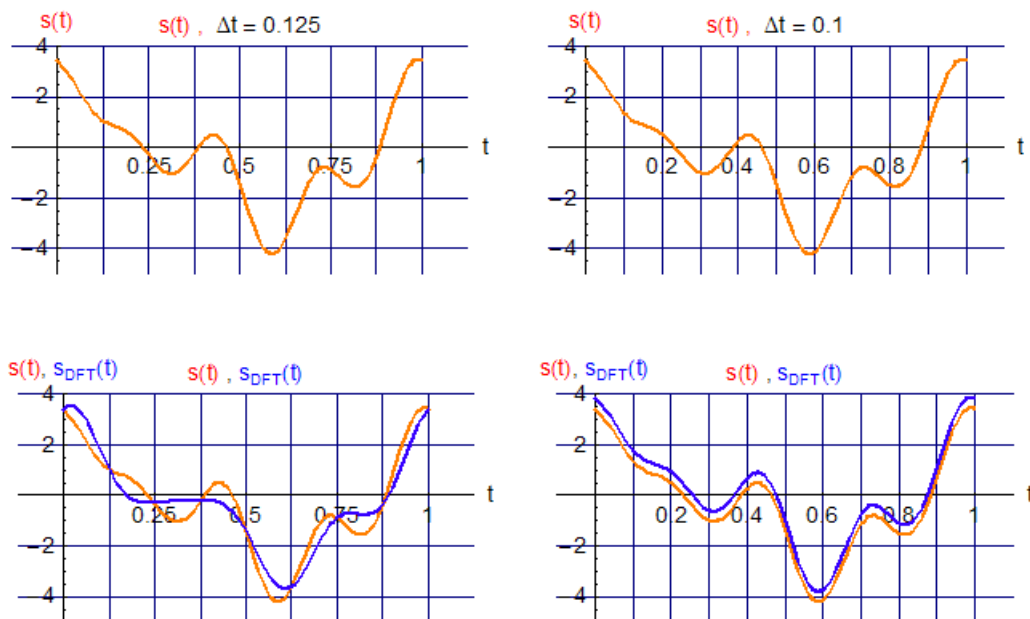


Bild 3: Die Originalfunktion $s(t)$ ist Summe eines Gleichanteils und von je 4 cos- und sin-Funktionen. Links wurde 8 Mal abgetastet, rechts 10 Mal. Im Diagramm rechts unten wurde auf s_{DFT} künstlich ein Gleichanteil von 0.4 addiert, um die Verläufe optisch auseinander zu ziehen.

Es gibt also wenigstens einen theoretischen Fall, bei dem die DFT den Verlauf der Originalfunktion vollständig wiedergibt. Praktisch hat dies jedoch keine Bedeutung, da man es ja immer mit Originalfunktionen **unbekannter** Zusammensetzung zu tun haben wird. Allerdings könnte man das Signal durch ein **Tiefpassfilter** mit der Grenzfrequenz $4f$ laufen lassen und hätte dann den Idealfall immerhin angenähert. Eine entsprechende Maßnahme gehört unter anderem auch zum MP3-Verfahren, siehe Unterkapitel 6.2.5, Seite 382 ff.

Wie sieht es nun aber für die ursprüngliche Frage aus? Hier kommen zu den trigonometrischen Anteilen in $s(t)$ zusätzlich zu den in Tabelle 1 genannten weitere 4 mit nicht-harmonischen Frequenzen hinzu. Diese sind also keine ganzzahligen Vielfachen der Grundfrequenz f , wie Tabelle 2 zeigt.

Zunächst ist der Verlauf der Originalfunktion gemäß Bild 4, obere Reihe, gegenüber dem in Bild 3 gezeigten leicht verändert. Die Zahl der Abtastwerte blieb mit $2N = 8$, bzw. $2N = 10$ aber gleich. Die DFT-Koeffizienten weichen nun von denen der Originalfunktion ab, siehe Tabelle 3. Die "Aufgabe" der DFT ist es ja, unter Ausnutzung der 8 bzw. 10 Freiheitsgrade Identität in den Abtastzeitpunkten herzustellen. Dies gelingt hier eben nur auf Kosten der Amplituden.

Bild 4 unten rechts zeigt, dass selbst bei $2N = 10$ Abtastwerten keine vollkommene Übereinstimmung zwischen Originalfunktion und der auf DFT-Basis ermittelten hergestellt wurde.

Frequenz	Original	DFT (N=4)	DFT (N=5)	Original	DFT (N=4)	DFT (N=5)
Gleichanteil	$a_0 = -0.3$	$as_0 = -0.15$	$as_0 = -0.19$	$b_0 = 0$	$bs_0 = 0$	$bs_0 = 0$
$1.0 \cdot f = f_{\min}$	$a_1 = 2.0$	$as_1 = 2.93$	$as_1 = 2.87$	$b_1 = 1.0$	$bs_1 = 0.51$	$bs_1 = 0.53$
$1.1 \cdot f$	$a_2 = 1.0$			$b_2 = -1.0$		
$1.2 \cdot f$	$a_3 = 0.8$			$b_3 = 2.0$		
$1.3 \cdot f$	$a_4 = -1.0$			$b_4 = -1.0$		
$2.0 \cdot f$	$a_5 = 0.8$	$as_2 = 0.57$	$as_2 = 0.50$	$b_5 = -0.5$	$bs_2 = -0.99$	$bs_2 = -0.96$
$2.4 \cdot f$	$a_6 = 0.6$			$b_6 = -0.8$		
$3.0 \cdot f$	$a_7 = 0.4$	$as_3 = 0.95$	$as_3 = 0.87$	$b_7 = 0.7$	$bs_3 = 1.08$	$bs_3 = 1.15$
$4.0 \cdot f$	$a_8 = 0.5$	$as_4 = 0.71$	$as_4 = 0.81$	$b_8 = -0.8$	$bs_4 = 0.0$	$bs_4 = -0.68$
$5.0 \cdot f$	---		$as_5 = 0.14$	---		$bs_5 = 0.0$

Tabelle 2: Koeffizienten der um 4 zusätzliche, jedoch nicht-harmonische Anteile erweiterten Originalfunktion aus Tabelle 3 und die über DFT mit $2N = 8$ und $2N = 10$ Abtastwerten ermittelten Amplituden

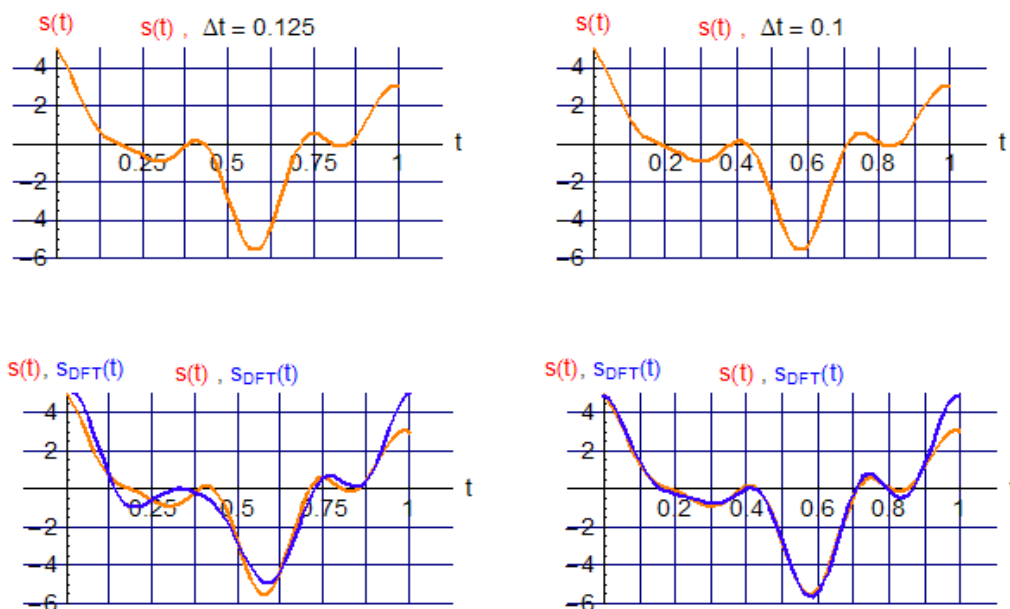


Bild 4: Die Verläufe von Originalfunktion und Näherungen über DFT für die um 4 nicht-harmonische Anteile erweiterten trigonometrischen Summen gemäß Tabelle 2

Dennoch lässt sich auch hier eine vollkommene Übereinstimmung zwischen Original und Näherung erzwingen. Damit sind wir endlich bei der Antwort auf die ursprüngliche Frage. Aus den bisherigen Überlegungen folgt, dass identische Verläufe dann hergestellt werden können, wenn es für alle im Signal $s(t)$ enthaltenen Frequenzanteile eine gemeinsame Grundfrequenz gibt, da diese dann ganzzahlige Vielfache sind. Das ist im vorliegenden Beispiel für $f_{\min} = 0.1 f$ der Fall. Man benötigt hier bei gleichen Intervallen Δt allerdings die 10-fache Anzahl von Abtastwerten, der gesamte Bereich erstreckt sich über das Intervall $10 \cdot T$.

Die Grundfrequenz $f_{\min}$ und einige weitere Vielfache hiervon treten in diesem Fall allerdings gar nicht auf, weshalb die DFT für diese Anteile eben Null-Koeffizienten ermittelt. Folgende Angaben werden für die Lösung gebraucht:

- Da sich die höchste Frequenzkomponente $f_{\max}$ nicht geändert hat, dürfen die Abtastintervalle weiterhin höchstens $\Delta t = 0.125$ s lang sein.

- Die Grundfrequenz $f_{\min} = 0.1 f$ ist gegenüber vorher zehn mal kleiner, daher benötigt man statt $2N=8$ nun $2N = 80$ Abtastwerte.
- Das Messintervall hat dann eine Länge von $T = 10$ s. In der oben stehenden Formel für $s_{\text{DFT}}(t)$ ist dies bereits berücksichtigt.
- Um auch die Amplitude der schnellsten Sinus-Komponente richtig zu identifizieren wird die Zahl der Abtastwerte um ein weiteres Paar erhöht, **$2N = 82$** .
- Da das Messintervall wegen des Zusammenhangs mit $f_{\min}$ die Länge $T = 10$ s beibehalten muss, ist das Abtastintervall **$\Delta t = 10/82 = 0.122$ s** groß.

Die Koeffizienten sind im Prinzip gemäß Tabelle 3 aufgebaut. Bild 5 zeigt schließlich die Verläufe der Originalfunktion $s(t)$ und der Näherung $s_{\text{DFT}}(t)$, die wie erwartet identisch sind. Zur Übersichtlichkeit wurde aus dem gesamten Intervall $T = 10$ s nur der Anfang von 0 bis 1 s und das Ende von 9 bis 10 s dargestellt. Außerdem ist wie in Bild 4 zur optischen Erkennung der Gleichheit beider Diagramme die Näherung künstlich um den Betrag 1.0 angehoben.

Frequenz	Original	DFT (N=41)	Original	DFT (N=5)
Gleichanteil	$a_0 = -0.3$	$as_0 = -0.3$	$b_0 = 0$	$bs_0 = 0.0$
$1 f_{\min} = 0.1 f$	$a_1 = 0.0$	$as_1 = 0.0$	$b_1 = 1.0$	$bs_1 = 0.0$
$2 f_{\min} = 0.2 f$	$a_2 = 0.0$	$as_2 = 0.0$	$b_2 = -1.0$	$bs_2 = 0.0$
$3 f_{\min} = 0.3 f$	$a_3 = 0.0$	$as_3 = 0.0$	$b_3 = 2.0$	$bs_3 = 0.0$
...	...	...	...	...
$9 f_{\min} = 0.9 f$	$a_9 = 0.0$	$as_9 = 0.0$	$b_9 = 0.0$	$bs_9 = 0.0$
$10 f_{\min} = 1.0 f$	$a_{10} = 2.0$	$as_{10} = 0.0$	$b_{10} = 1.0$	$bs_{10} = 0.0$
$11 f_{\min} = 1.1 f$	$a_{11} = 0.0$	$as_{11} = 0.0$	$b_{11} = 0.0$	$bs_{11} = 0.0$
$12 f_{\min} = 1.2 f$	$a_{12} = 0.0$	$as_{12} = 0.0$	$b_{12} = 0.0$	$bs_{12} = 0.0$
...	...	...	...	...
$39 f_{\min} = 3.9 f$	$a_{39} = 0.0$	$as_{39} = 0.0$	$b_{39} = 0.0$	$bs_{39} = 0.0$
$40 f_{\min} = 4.0 f$	$a_{40} = 0.5$	$as_{40} = 0.5$	$b_{40} = -0.8$	$bs_{40} = -0.8$
$41 f_{\min} = 4.1 f$	---	$as_{41} = 0.0$	---	$bs_{41} = 0.0$

Tabelle 3: Die Koeffizienten zur Aufgabenstellung in Frage 1

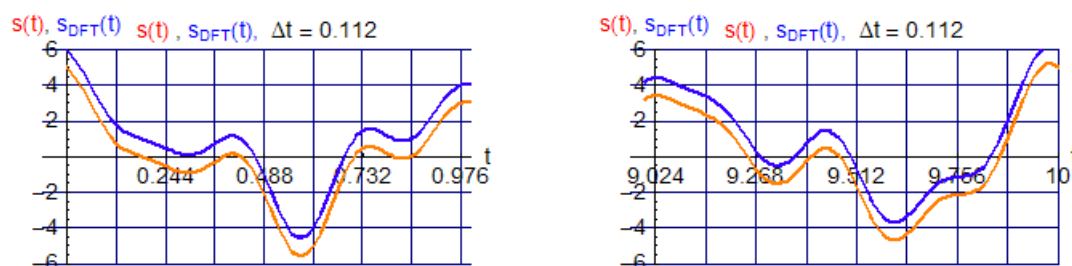


Bild 5: Anfangs- und Endbereich von $s(t)$ und Näherung $s_{\text{DFT}}(t)$. Zur besseren Erkennung der Identität wurde die Näherungskurve in dieser grafischen Darstellung um den Betrag 1.0 angehoben.

Die Überlegungen zu diesem letztgenannten Fall erscheinen sehr theoretisch, sind aber im Zusammenhang mit dem Aufbau von Filterbänken wie z. B. für die MP3-Kompression von Bedeutung, siehe auch Frage 3.

Hinweis zum Koeffizienten a_N in der Formel auf der hiesigen Seite 1: Gegenüber den Angaben zur Berechnung der Fourierkoeffizienten a_k auf Seite 352 des Unterkapitels 6.2.2.1 gilt im allgemeinen Fall für den Koeff-

fizienten mit dem höchsten Index $\lambda = N$, dass er nur mit seinem halben Wert eingeht. Für Signale $s(t)$, die als Summe von cos- und sin-Anteilen einen höchsten Frequenzanteil mit $\lambda = N-1$ enthalten, ist $a_N = 0$, wie wir bei Tabelle 1 oder 3 sehen. In diesen Fällen stimmen dann auch die Formeln auf Seite 352 ff.

Zu Frage 2:

Die DCT ergibt Koeffizienten, die sich im allgemeinen nur durch **nicht-rationale** (reelle) Zahlen ausdrücken lassen und daher für praktische Verwendung **gerundet** werden müssen. Insbesondere im Rahmen schneller Verfahren werden kurze Zahlendarstellungen mit wenigen Bits Länge erzwungen. Die Rundungsfehler sind bei der Rücktransformation nicht rückgängig zu machen, ziehen also bereits Verluste nach sich. Dies ist aber kein im Sinne der Kompression beabsichtigter Effekt.

Zu Frage 3:

Ein **Polyphasen-Filter-Algorithmus** (Mehrphasen-Filter) gestattet es, auf Basis eines Tiefpassfilters mit gewünschter Bandbreite B in rechentechnisch einfacher Weise durch Verschiebung der Mittenfrequenz beliebige weitere Filter mit **Bandpasswirkung** zu erzeugen (= "Phasenverschiebung der Mittenfrequenz"). Die Verwendung dieser Filtervariante ist zwar keineswegs zwingend und könnte auch durch andere Filterlösungen realisiert werden, sie ist aber gerade für **Filterbänke**, wie sie für die Verarbeitung eines Audiosignals zu einem MP3-Komprimat erforderlich sind, besonders einfach auszuführen.

Ehe wir uns die prinzipielle Technik genauer anschauen, ist es nützlich, sich nochmals die Aufgabenstellung der Filterbank im MP3-Verfahren in Erinnerung zu rufen:

- Da man aufgrund von experimentellen Untersuchungen das physiologische Verhalten des Gehörsinns in etwa kennt, weiss man auch einiges über die zeitliche und dynamische Verarbeitung der verschiedenen wahrgenommenen Frequenzanteile. Diese gruppieren sich in die so genannten "kritischen" Frequenzbänder.

Um das berücksichtigen zu können, ist also die Kenntnis der Frequenzanteile im Originalsignal erforderlich. Hierzu wird das erwartete Spektrum des Audiosignals $s(t)$ durch eine Polyphase-Filterbank zunächst in 32 gleichgroße Bänder zwischen 0 Hz oder 20 Hz und 20 KHz unterteilt (diese entsprechen aus Aufwandsgründen nicht exakt den physiologischen kritischen Bänder). Aus den PCM-Abtastwerten des Originalsignals entstehen am Ausgang der Filterbank also die PCM-Abtastwerte der 32 bandbegrenzten Spektralanteile.

- Diese PCM-Werte werden nun durch eine Fourier-Transformation (z. B. DFT, hier aber MDCT = Modified Discrete Cosine Transform)) für die weitere Verarbeitung in ihre Frequenzanteile abgebildet, siehe Kapitel 6.2.2.2, Seite 362 ff.

Wie würden wir mit unseren Kenntnissen diese Aufgabe prinzipiell lösen? Um den Aufwand in Grenzen zu halten, unterteilen wir den Frequenzbereich des Audiosignals in 10 Bereiche von je 2 KHz Breite, also von 0 bis 20 KHz und fordern für die Näherung durch eine DFT, dass eine Frequenzauflösung in Schritten von 100 Hz erfolgt. Das heisst unter anderem:

- Da die höchste Frequenzkomponente 20 KHz ist (oder genauer: sein kann), muss $s(t)$ mit 40 KHz abgetastet werden. Das Abtastintervall beträgt also 25 μ s.
- Das unterste Band enthält die 20 Anteile 100 Hz, 200 Hz, ... , 2000 Hz
- Das nächste Band enthält die 20 Anteile 2100 Hz, 2200 Hz, ... 4000 Hz
- usw.

Man muss also wegen der gewünschten harmonischen Anteile im Abstand von 100 Hz in der DFT-Näherung immer von der Intervallbreite $T = 1/100 \text{ Hz} = 10 \text{ ms}$ ausgehen, siehe Lösungsvorschlag zu Frage 1, Seite 5 ff. Im Einzelnen:

- Die Intervallbreite des untersten Bandes ist wegen 100 Hz als langsamster Frequenz 10 ms, das benötigte Abtastintervall beträgt wegen der obersten zu identifizierenden Frequenz von 2000 Hz genau $1/4000 = 0.25$ ms. Man kann also wegen des sowieso notwendigen Abtastintervalls von 25 μ s jeden zehnten Abtastwert hierfür verwenden.
- Für das nächste Band von 2000 Hz bis 4000 Hz wird wegen der Aufteilung in 100 Hz- Schritte ebenfalls eine Grundfrequenz von 100 Hz benötigt. Das Intervall hat also ebenfalls eine Breite von 10 ms. die Abtastfrequenz wäre hier $1/8000$ Hz = 0.125 ms.
- Das dritte Band von 4000 Hz bis 6000 Hz erfordert wieder eine Intervallbreite von 10 ms, für die höchste Komponente mit 6000 Hz sind $1/12.000 = 83.33$ μ s erforderlich. Da dieser Wert kein genaues Vielfaches von 25 μ s darstellt, wird der nächst kleinere Wert gewählt, also 75 μ s. Die Intervallbreite wäre hier 9.975 ms, die Frequenzen Vielfache von 100.251 Hz..
- Das vierte Band von 8000 Hz bis 10000 Hz benötigt für die höchste Komponente $1/20000 = 50$ μ s, also jeden zweiten Wert der Abtastfolge.
- Das fünfte Band von 10.000 Hz bis 12.000 Hz muss $1/24.000 = 41.66$ μ s haben, man könnte 25 μ s oder 50 μ s nehmen.
- und so fort.
- Das oberste Band von 18.000 Hz bis 20.000 Hz schließlich braucht 25 μ s Abtastzeit. Bei einer Intervallbreite von 10 ms enthält die DFT-Näherung dann alle Komponenten mit 18.000 Hz, 18.100 Hz, 18.200 Hz usw.

Generell lassen sich demnach bei Verlängerung des Zeitintervalls T auch nicht ganzzahlige Vielfache der jeweils tiefsten Frequenz $f_{\min}$ des Bandes identifizieren, wenn deren Existenz an sich bekannt ist. In praktisch vorkommenden Fällen, in denen die spektrale Zusammensetzung eines Signalabschnittes von $s(t)$ aber gerade nicht vorliegt, wird dadurch immerhin die Näherung an die tatsächlich vorhandenen Anteile verbessert.

Dies gilt allerdings nur, wenn diese Anteile **stationär** sind, sich also im betrachteten Intervall weder in der Amplitude noch in der Phasenlage ändern. Auch dies trifft in praktischen Fällen nicht wirklich zu. Amplituden schwanken im Sinne von Modulationen (z. B. lauter/leiser werdende Instrumente), Frequenzanteile verschwinden oder kommen neu hinzu (z. B. durch einsetzende Schlagzeuge). Bei Verlängerung des Zeitintervalls T wird dann zwar die **mittlere Frequenzauflösung** verbessert, zugleich verschlechtert sich aber die **Zeitauflösung** bei **instationären** Änderungen im Intervall.

Die Folgen zeigen sich bei der Rückkehr vom Frequenz- in den Zeitbereich: Die rücktransformierten Abtastwerte $s_{\text{DFT}}(v\Delta T)$ enthalten keine im Intervall T aufgetretenen instationären Vorgänge mehr. Die aus den berechneten Frequenzkomponenten hergestellten komprimierten Daten gestatten dann keine vollständige Rekonstruktion solcher Vorgänge. Für eine ausreichende **Zeitauflösung** ist es daher notwendig, das Intervall T kurz genug zu wählen, was letztlich einen Kompromiss mit der Frequenzauflösung erfordert. Die passende Länge für T hängt davon ab, wie schnell sich die Spektren in $s(t)$ ändern und wird im allgemeinen durch Messungen im Rahmen des betrachteten Einsatzfalles ermittelt werden müssen.

Die Zeitauflösung lässt sich im Übrigen auch durch andere Transformations-Formen beeinflussen, z. B. durch die **Wavelet-Technik**.

Nun noch einmal zurück zur Technik der Polyphase-Filterbank: Eines der Kernelemente des MPEG-Verfahrens für Audiosignale ist ja die Aufteilung aller in einem Zeitintervall T eines bandbegrenzten Zeitsignals $s(t)$ enthaltenen Frequenzanteile in 32 (gleich-) breite Bereiche. Dazu durchläuft eine Abtastfolge $s(v\Delta T)$ eine "Bank" aus 32 Bandfiltern, deren Ausgänge 32 diskrete Zeitfolgen liefern, die dann nur Frequenzanteile ihres Bandes enthalten. Jedes Bandfilter ist gekennzeichnet durch seine **Bandbreite** B_v und seine **Mitten- oder Resonanzfrequenz** $f_{\text{res } v}$, mit $v = 0, 1, 2, \dots$ Im einfachsten Fall hat jedes Filter die gleiche Bandbreite $B_v = B$, im vorliegenden Fall des Hörbereichs 0 bis 20 KHz also $B = 20.000 : 32 = 625$ Hz. Die Mittenfrequenzen besitzen z. B. die Werte $f_{\text{res } v} = (v+1) B$, mit $v = 0, 1, 2, \dots, 31$, für das tiefste Band kann ein Tiefpassfilter mit der Bandbreite B einsetzen.

Man kann diese Filter gemäß dem in Unterkapitel 6.2.5, Seite 388 ff., beschriebenen Verfahren konstruieren, dort ist auch das Ergebnis der Filterung eines Eingangssignals und des Ausgangssignals dargestellt, siehe Bilder 6.8 und 6.9. Der Aufbau einer ganzen Bank mit 32 Filtern wird besonders einfach, wenn man die

Technik der Polyphase-Filter verwendet, siehe, Seite 391 ff.. Zum Vergleich nochmals die dargestellten **Frequenzgänge *** der "klassischen" Filterausführung und der Polyphase-Variante, beides in der einfachsten, weiter verbesserungsfähigen theoretischen Ausführung zweiter Ordnung:

$$F(j2\pi f)_{v, \text{ klassisch}} = \frac{1}{\left(j \frac{f}{f_{\text{RESoD } v}}\right)^2 + j2D_v \frac{f}{f_{\text{RESoD } v}} + 1}, \quad v=0, 1, 2, \dots, 31$$

$$F(j2\pi f)_{\text{PolyphaseDesign-Tiefpass}} = \frac{1}{j \frac{f}{f_B} + 1}$$

$$F(j2\pi f)_{v, \text{ PolyphaseBandfilter}} = \frac{1}{j \cdot 2 \frac{f - v f_{\text{res}}}{f_B} + 1}, \quad v=0, 1, 2, \dots, 31$$

Die Bandbreite B (der Abstand der beiden Frequenzen, bei denen der Betrag des Frequenzganges den Wert $\frac{1}{\sqrt{2}}=0.71$ annimmt) der klassischen Filter muss mit Hilfe des Dämpfungsfaktors D und der Resonanzfrequenz ohne Dämpfung, f_{RESoD} , für jedes Frequenzband gesondert berechnet werden. Demgegenüber benötigt die Ausführung in Polyphase-Technik nur einmal die Festlegung von B für den sogenannten Design-Tiefpass, die Frequenzgänge der Bandfilter gehen in einfacher Weise durch eine "**Phasen-Verschiebung**" der Mitten- oder Resonanzfrequenz f_{res} hervor, daher "**Polyphase-Filter**" (diese Bezeichnung kollidiert allerdings etwas mit dem sonst üblichen Gebrauch des Begriffs "Phase", bei dem die gegenseitige Phasenlage zweier sin- oder cos-Funktionen beschrieben wird). In den Diagrammen Bild 6 ist als Beispiel der Betrag des Design-Tiefpasses erster Ordnung mit $B = 625$ Hz dargestellt:

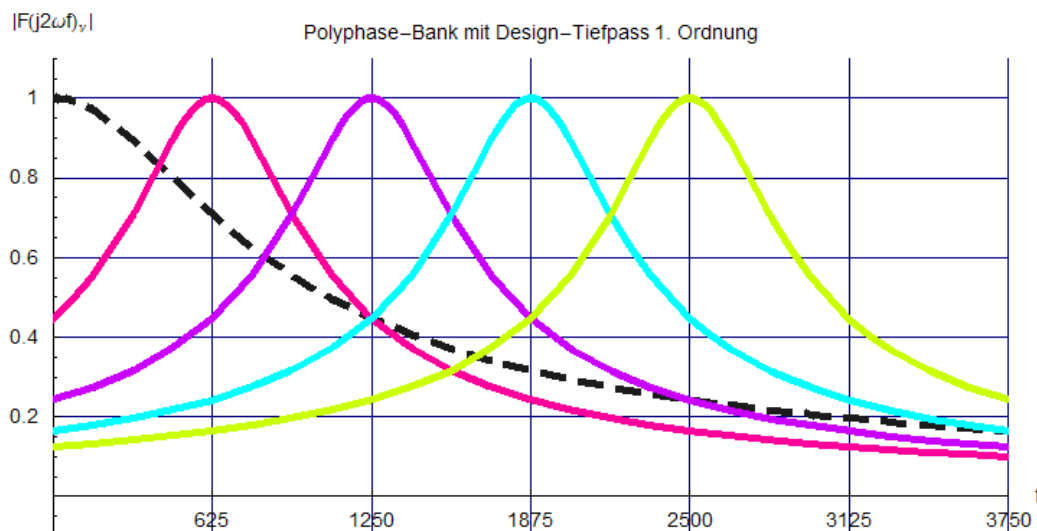


Bild 6: Betragskurven eines Design-Tiefpasses 1. Ordnung (gestrichelt) mit der 3 db (= 0.71) - Bandbreite $B = 625$ Hz und der ersten 4 Bandfilter einer Polyphase-Filterbank mit den Mittenfrequenzen 625, 1250, 1875 und 2500 Hz.

Es wurden die oben angegebenen Formeln verwendet. Die Filter sind bei der Polyphase-Technik automatisch von 2. Ordnung, da im Nenner des Frequenzganges für Frequenzen unterhalb der Mittenfrequenzen auch negative Differenz-Frequenzen auftreten.

Der Vorteil der einfachen Konstruktion von Filterbänken mit der Polyphase-Technik liegt vor allem in den darüber ableitbaren, verhältnismäßig einfachen Algorithmen für die benötigten **Differenzgleichungen**,

siehe weiter unten.

Da die Flanken der einzelnen Bandfilter nicht ideal steil sind, treten in benachbarten Filtern jeweils noch schwache Frequenzanteile auf, die das Ausgangssignal verfälschen. Die **Flankensteilheit** der Betragsverläufe lässt sich aber durch Hintereinanderschaltung mehrerer gleichartiger Bandfilter verstärken. Damit die Bandbreiten B unverändert bleiben muss eine Anpassung in den Frequenzgängen vorgenommen werden, die sicherstellen, dass die Betragsverläufe im halben Abstand $B/2$ links und rechts der Mittenfrequenzen $f = \nu f_{\text{res}}$ gerade auf $3 \text{ dB} = 0.71$ abgefallen sind. Bezeichnet man mit m die Anzahl hintereinander geschalteter Einzelfilter 2. Ordnung, dann erfüllen die folgenden Frequenzgänge diese Anforderungen:

$$F(j2\pi f)_{\text{Polyphase Design - Tiefpass}} = \frac{1}{j\sqrt{2^{1/m}-1}\frac{f}{f_B} + 1}$$

$$F(j2\pi f)_{\nu \text{ Polyphase Bandfilter}} = \frac{1}{j2\sqrt{2^{1/m}-1}\frac{f - \nu f_{\text{res}}}{f_B} + 1}, \quad \nu = 0, 1, 2, \dots, 31$$

Die Bild 6 entsprechende Diagramme sind in Bild 7 für $m=10$ dargestellt:

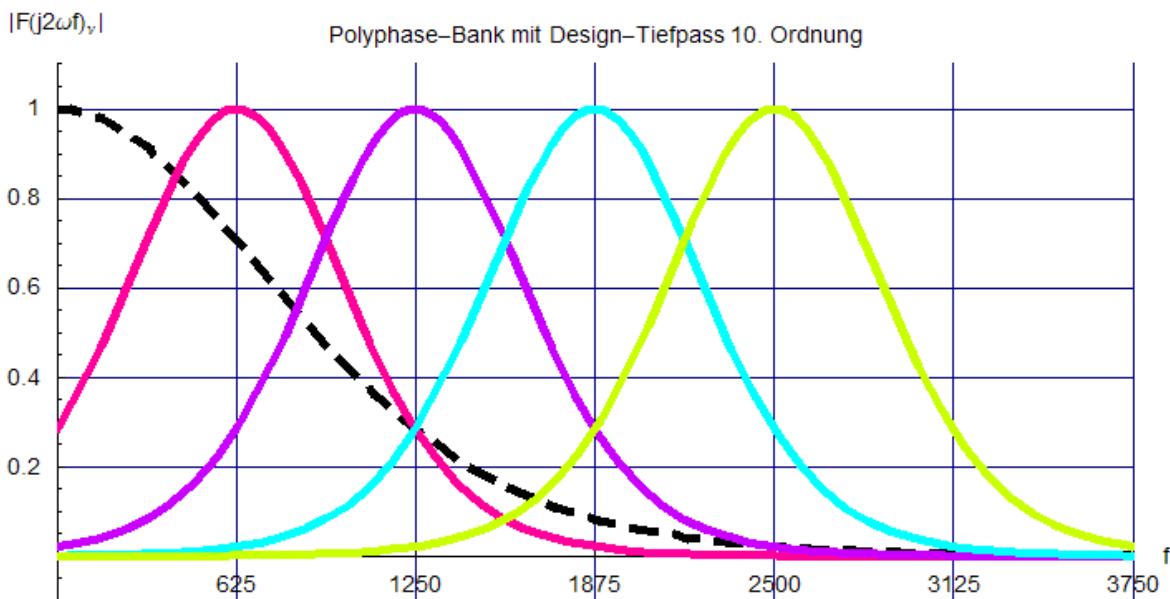


Bild 7: Wie Bild 6, aber Betragskurven eines Design-Tiefpasses 10. Ordnung (gestrichelt) mit der 3 dB ($= 0.71$) - Bandbreite $B = 625 \text{ Hz}$ und der ersten 4 Bandfilter einer Polyphase-Filterbank mit den Mittenfrequenzen 625, 1250, 1875 und 2500 Hz.

Man erkennt deutlich die bei gleicher Bandbreite $B = 625 \text{ Hz}$ steileren Flanken, die den "Übersprechfehler" benachbarter Filter verringern.

Im Übrigen ist die Festlegung der **Filterparameter** für die Frequenzgänge nur ein Schritt im gesamten Verfahren. Da alle Berechnungen in Echtzeit direkt aus den Abtastwerten des Zeitsignals $s(t)$ gemacht werden, sind im nächsten Schritt die zugehörigen **Differenzgleichungen** zu bestimmen, wie in Unterkapitel 6.5 auf Seite 391 ff., dargestellt, allerdings ist dies nur einmal erforderlich.

***) Hinweis:** Der komplexe Frequenzgang einer elektronischen Schaltung wird sowohl durch den **Betrags-** als auch durch den **Phasenverlauf** beschrieben. Die vorstehenden Betrachtung beschränkt sich aber nur auf den Betragsverlauf. Bei Filteruntersuchungen hat der Betrag oft einen höheren Stellenwert, da es sich bei Filtern in vielen Fällen, so wie hier, um "Vorwärts-" Elemente handelt. Sind Filter dagegen in

rückgekoppelte Schaltungsstrukturen integriert, spielt auch der Phasenverlauf eine entscheidende Rolle.

Die Phasen-Nacheilung darf selbst bei reinen Vorwärts-Anordnungen, insbesondere bei Filtern höherer Ordnung, nicht ganz ausser Acht gelassen werden, da sie mit einer Laufzeit verbunden ist. So erzeugt ein Filter 20. Ordnung ($m = 10$) bereits eine Nacheilung der hohen Anteile im Bandbereich von 5 Perioden. Das kann im Extremfall zu merkbar störenden Effekten führen, so dass der beliebigen Erhöhung von m Grenzen gesetzt sind.

Zu Frage 4:

Als **Abtastrate** = $1/\text{Abtastfrequenz}$ wird für CD-ähnliche Stereo-Qualität meistens $1/44.100 \text{ Hz} = 22,68 \mu\text{s}$ pro Kanal gewählt. Die höchste im Audiosignal damit identifizierbare Frequenzkomponente beträgt nach dem Abtasttheorem also $44.100/2 = 22.050 \text{ Hz}$ und liegt damit über der **Hörgrenze**, die mit (sehr groben) 20.000 Hz angenommen wird. Die **Abtastfrequenz** lässt sich bei den Encodern auf Soundkarten usw. meistens aus mehreren Möglichkeiten auswählen, z. B. 48.000 Hz oder 32.000 Hz und weitere.

Genau genommen sind das allerdings die für .wav-Dateien direkt wählbaren Abtastraten. Für .mp3-Files kann demgegenüber oft nur die **Bitrate** festgelegt werden. Sie liegt zwischen 64 und 320 kBit/s und bezeichnet die für Echtzeitübertragung (= **streaming**) verwendbare höchste Bitanzahl pro Sekunde. Je höher diese Bitrate ist, desto bessere Qualität kann das dekomprimierte Audiosignal haben. Die Auswahl ist natürlich von der Kapazität des aktuellen Übertragungskanals abhängig. Bei den früher eingesetzten ISDN-Kanälen standen im allgemeinen nur 64 kBit/s zur Verfügung, während die Encodierung und Dekodierung mit Soundkarten und PCs keine Grenzen setzen. Bei 320 kBit/s entstehen längere Files als bei kleineren Abtastraten. Als Abtastrate für CD-ähnliche Qualität wird davon unabhängig allerdings wie für .wav-Dateien meistens **$22,68 \mu\text{s}$** gewählt.

Zu Frage 5:

Wie wir bei der Beantwortung von Frage 1 gesehen haben, gehen bei ungefiltertem Signal alle Frequenzanteile in die Berechnung der DFT-Koeffizienten ein. Die über der höchsten zu berücksichtigenden Frequenz liegenden Beiträge verfälschen daher das gewünschte Ergebnis eines bandbegrenzten Signals.

Zu Frage 6:

In Unterkapitel 6.2.4.1; Seite 381 und 382, ist beschrieben, wie Video-Bildfolgen bei MPEG-1 prinzipiell behandelt werden. Die Grundidee besteht darin, dass sich bei vielen Bildfolgen die Einzelbilder "stetig" entwickeln, sich also von Einzelbild zu Einzelbild nur wenig unterscheiden. Dann ist es vorteilhaft, nicht die Vollbilder, sondern nur deren Differenzwerte (= Pixelunterschiede) zu übertragen. Die sich dabei ergebenden Zahlen haben entweder den Wert 0 oder sind sehr klein, weshalb sich ihre Gesamtmenge stark komprimieren lässt (mit RLE oder Huffman für wenige unterschiedliche Werte nahe "0"). Allerdings ist dieses Verfahren sehr empfindlich gegen Verfälschungen des Datenstroms durch Störungen. Deshalb müssen zur Synchronisation in Abständen immer wieder Vollbilder eingeschoben werden.

Zu Frage 7:

JPEG 2000 bietet alternativ zur Bilddatenkompression mit Hilfe der bewährten Fouriertransformation auch den Einsatz der **Wavelet-Transformation**, siehe Unterkapitel 6.2.6. Der Vorteil liegt in höheren Kompressionsraten bei vergleichbarer Qualität, da die Wavelet-Technik Vorteile bei der Behandlung kontrastreicher Bilder mit scharfen Übergängen bietet. Da die Standardbrowser aber in der Grundversion bislang nicht immer Dekompressions-Algorithmen bieten, ist der Einsatz dieser Option beschränkt.

Zu Frage 8:

Man benötigt wenigstens

- a) ein geeignetes Maß, welches über die Ähnlichkeit zweier Pixelblöcke – des Referenz-Blockes PB_R und des Kandidaten-Blockes PB_K - Auskunft gibt. Hierbei ist nicht nur die Ähnlichkeit ohne Veränderung des Kandidaten-Blocks gemeint, sondern auch diejenige nach Verschiebung, Verzerrung (Stauchung oder Streckung) und Rotation.
- b) ein Vorgehen, um aus einem aktuellen Ähnlichkeits-Maßwert eine Aktion abzuleiten, die diesen Wert durch Verschiebung, Verzerrung und/oder Rotation des Kandidatenblocks verbessert.

Bezüglich a) könnte ein solches Maß in einem sehr einfachen Fall die Summe der Differenzquadrate zwischen den Pixelwerten $PB_R(i,j)$, $PB_K(i,j)$ mit gleichen Koordinaten sein. Sind die Pixelblöcke 8 x 8 Pixel groß, so ist diese Summe q :

$$q = \sum_{i=1}^8 \sum_{j=1}^8 (PB_R(i,j) - PB_K(i,j))^2$$

Je geringer diese Summe ausfällt, umso stärker wird vermutlich die Ähnlichkeit sein. Allerdings muss auch festgelegt werden, was als "ausreichend" anzusehen wäre. Dies kann man durch Tests ähnlich wie beim psychoakustischen Modell für Audiodaten ermitteln.

Nachteilig bei diesem Differenzverfahren bleibt, dass die Pixelwert-Paare unabhängig von ihrer Position innerhalb des Pixelfeldes eingehen. Da sich Bildpunkte aber oft stetig, d. h. ohne abrupte Sprünge ändern, bleibt dabei die "nachbarschaftliche" Information ungenutzt. Hier können Korrelationsverfahren zusätzliche Informationen liefern, sie sind allerdings auch aufwändiger, siehe Unterkapitel 4.1, Seite 247 ff.

Geht z. B. das Kandidatenfeld PB_K durch horizontale Verschiebung aus PB_R hervor, so kann dies durch Kreuzkorrelation leicht erkannt werden, wie es für zeitliche Signale in Bild 4.7 auf Seite 252 zu sehen ist. Mit zweidimensionalen Kreuzkorrelationen lassen sich sowohl horizontale wie auch vertikale Verschiebungen identifizieren. Auch Streckungen oder Stauchungen und Rotationen sind identifizierbar.

Zu Punkt b) sind numerische Extremwert-Suchverfahren geeignet. Ein einfacher Weg besteht darin, zwei Maßwerte vor und nach Veränderung zu betrachten und je nach Vorzeichen die Veränderung in gleicher Richtung fortzusetzen oder umzukehren. Dies ist sowohl für eindimensional für Veränderungen in nur einem Parameter (z. B. nur Translation) als auch mehrdimensional in mehreren Parametern (z. B. Translation und Verzerrung) möglich.