Klassifikationsbasierte Polyphasen Bildinterpolation

von der Fakultät

Elektrotechnik und Informationstechnik

der Technischen Universität Dortmund

genehmigte

Dissertation

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

von

Sebastian Lenke

Dortmund, 2009

Tag der mündlichen Prüfung: 11.12.2009

Hauptreferent: Prof. Dr.-Ing. Schröder

Korreferent: Prof. Dr.-Ing. Rothermel

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Arbeitsgebiet 'Schaltungen der Informationsverarbeitung' der Technischen Universität Dortmund.
Dem Leiter des Arbeitsgebietes, Herrn Prof. Dr.-Ing. H. Schröder möchte ich für die Anregung, Betreuung und Begutachtung der Arbeit meinen ganz besonderen Dank aussprechen. Durch fruchtbare Diskussionen, seine Hilfsbereitschaft, und durch das von ihm geschaffene hervorragende Arbeitsklima hat er erheblich zum Gelingen der Arbeit beigetragen.
Herrn Prof. Dr.-Ing. A. Rothermel vom Institut für Mikroelektronik der Universität Ulm danke ich für sein Interesse an der Arbeit und für die Übernahme des Koreferats.
Bei der wissenschaftlichen Arbeit habe ich wesentlich von Erfahrungen und Erkenntnissen profitiert, die ich im Vorfeld im Rahmen einer Industriekooperation mit der Firma Sony Deutschland GmbH gesammelt habe. Allen Mitarbeitern der Abteilung EuTeC, mit denen ich zusammenarbeiten durfte, möchte ich dafür ebenfalls sehr herzlich danken.
Weiterhin danke ich allen Kollegen, studentischen Mitarbeitern, Studien- und Diplomarbeitern, die durch ihren Einsatz und anregende Diskussionen zu dieser Arbeit beigetragen haben.
Mein besonderer Dank gilt meiner Frau Sarah, meinen Eltern und Geschwistern für ihre Unterstützung, ihre Geduld und ihr Verständnis vor allem während der Durchführung dieser Arbeit.
Herrn Dipl.-Ing. Lars Ole Keller wie auch meiner Familie danke ich darüber hinaus für die Durchsicht des Manuskripts.

Sebastian Lenke

Notation

Abkürzungen

2D	Zweidimensionale Ausrichtung (z. B. Filter, Signale usw.)
AKF	Autokorrelationsfunktion
CRT	Cathode Ray Tube (Bezeichnung für Fernseher mit einer Kathodenstrahlröhre)
DRC	Digital-Reality-Creation (Signalverarbeitungschip der Firma SONY mit einem adaptiven Interpolationsverfahren nach Kondo [KNF+01])
ELA	Edge-adaptive-Line-Averaging (Adaptives Deinterlacingverfahren [KLL96])
FIR	Finite Impulse Response, (Impulsantwort mit einer begrenzten Ausdehnung)
HDTV	High Definition TeleVision, Bezeichnung für hochaufgelöstes Fernsehen
HD	High Definition, hochaufgelöst
IIR	Infinite Impulse Response (Impulsantwort mit einer unendlichen Ausdehnung)
LCD	Liquid Crystal Display (Flüssigkeitskristalldisplay)
MSE	Mean Square Error (mittlerer quadratischer Fehler, MQF)
MQF	Mittlerer quadratischer Fehler
NEDI	New-Edge-Directed-Interpolation (Adaptives Interpolationsverfahren nach X.Li [LO01])
PDP	Plasma Display Panel
PSNR	Peak Signal to Noise Ratio (Verhältnis des größtmöglichen Signalwertes zum vorliegenden Rauschen)

SDTV	Standard Definition TeleVision, Bezeichnung für normalaufgelöstes Fernsehen
SD	Standard Definition, normalaufgelöst
SW	Schwarz-Weiß

Operatorbezeichnungen

$\mathrm{ggT}\,(\cdot)$	Größter gemeinsamer Teiler
$\mathrm{MAX}\,\{\cdot\}$	Maximum-Operator
$\mathrm{MIN}\,\{\cdot\}$	Minimum-Operator
$\langle\cdot,\cdot\rangle$	Skalar-Produkt zweier Vektoren

Operator- und Funktionsdefinitionen

Signum-Funktion:

$\mathrm{sign}(x)$ $\quad \mathrm{sign}(x) = \begin{cases} -1 & \text{für } x < 0 \\ 0 & \text{für x} = 0 \\ 1 & \text{für } x > 0 \end{cases}$

Abrunden-Funktion:

$\lfloor x \rfloor$ $\quad \lfloor x \rfloor = \mathrm{MAX}\,\{y \in \mathbb{Z}\,|y \leq x\}$

Modulo-Funktion:

$(a \mod m)$ $\quad (a \mod m) = a - \left\lfloor \frac{a}{m} \right\rfloor \cdot m$

Formelzeichen

$_{x,y}$	Parameter in x- oder y-Richtung
a_{ij}	Element der Matrix $\mathbf{A}$
$F(\vec{x}, \tau)$	Luminanz des Bildpunktes an der örtlichen Position $\vec{x}$ zum normierten Zeitpunkt τ

$s(t)$	Kontinuierliche Funktion s am Punkt t
$S(f_t)$	Spektrum S an der Frequenz f_t
$s\,[n]$	diskrete Folge s an der ganzzahligen Position n
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{Z}$	Menge der ganzen Zahlen: $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$

Kurzfassung

Durch die gestiegene Verbreitung von hochauflösenden und großformatigen Displays im Consumerbereich und der zur Zeit noch sehr geringen Verbreitung von hochauflösendem Eingangsmaterial sowie der Notwendigkeit, unterschiedlichste Bildformate und -qualitäten verarbeiten zu müssen, steigt der Anspruch an qualitativ hochwertigen Interpolationsverfahren.
Bisher werden hauptsächlich lineare nichtadaptive Verfahren im Consumerbereich eingesetzt, da zum einen der Aufwand sehr gering ist, und zum anderen die Erzeugung von Artefakten sehr gut vermieden wird. Nichtlineare adaptive Verfahren ermöglichen zwar in vielen Fällen eine gestiegene Interpolationsqualität, tendieren jedoch zur Erzeugung von Interpolationsartefakten und benötigen zumeist einen erhöhten Realisierungsaufwand.
In dieser Arbeit werden zwei inhaltsadaptive örtliche Interpolationsverfahren vorgestellt, welche durch eine einfache lokale Klassifikation jeweils Filterkoeffizienten verwenden, die für die vorliegende Bildsituation optimal sind. Das eine Verfahren benötigt für jede Interpolationsphase einen optimierten Datensatz, wohingegen das zweite Verfahren durch ein einmaliges Training optimierte Polynome für beliebige Interpolationsfaktoren beinhaltet. Zur weiteren Optimierung werden aus Bildqualitätsforderungen Nebenbedingungen abgeleitet und diese in den Trainingsprozess beider Verfahren integriert. Die vorgeschlagenen Verfahren erreichen eine sehr hohe Interpolationsgüte und sind durch die einfache Klassifikation und das Vorhalten bereits optimierter Lösungen sehr aufwandsgünstig.
Abschließend wird dargestellt, wie diese Struktur auch für eine bewegungsfehleradaptive Zwischenbildinterpolation gewinnbringend eingesetzt werden kann. Mit Hilfe einer Klassifikation des Bewegungsschätzungsfehlers und des Bildinhaltes können einfache lineare Filter benutzt werden, die den Fehler kompensieren, und so selbst der Erhalt feinster Details möglich ist.

Abbildungsverzeichnis

Inhaltsverzeichnis

1 Einleitung und Motivation

Großformatige und hochauflösende Displays (HD) haben einen zunehmenden Anteil an neuen Fernsehgeräten. Von wenigen Ausnahmen abgesehen liegen die meisten Signalquellen momentan noch in niedriger Auflösung (SD) vor. Bedingt durch die größeren Bildschirmdiagonalen und auch durch die Charakteristik von Matrixdisplays wird die Sichtbarkeit von Artefakten auf diesen Displays verstärkt [Dol06] [EBU02]. Somit ist eine hochwertige Skalierung von niedrig aufgelösten Bildsignalen zwingend notwendig.
Auf Grund der Vielfalt der Eingangsformate, die heutzutage auf einem HD-fähigen Display angezeigt werden sollen (z. B. Videostreaming aus dem Internet, DVDs mit Kinofilmmaterial etc.), müssen die Verfahren neben einer guten Interpolationsperformance auch in der Lage sein, verschiedenste Interpolationsfaktoren zu berücksichtigen. Dies betrifft zum einen die örtlichen Interpolationsfaktoren, als auch eine zeitliche Interpolation zur Anpassung an die Bildwiederholrate des Displays.
Dieser Trend wird vor allem durch die neue Klasse der 120Hz bzw. 200Hz LCDs vorangetrieben. Im Gegensatz zur Intention der 100Hz-Röhrenfernsehertechnik (CRTs), die Bildinformation flimmerfrei darzustellen, liegt hier die Absicht in der Unterdrückung von Bewegungsunschärfe. Diese wird dadurch hervorgerufen, dass bei Verfolgung eines Bildobjektes die Augen eine kontinuerliche Bewegung machen und das Objekt jedoch während der Dauer eines Bildes stillsteht und dabei dauerhaft angezeigt wird. Somit verwischt das wahrgenommene Bild auf der Netzhaut des Auges.
Sowohl die örtliche als auch die zeitliche Konversion hat also gerade auf Grund der fortgeschrittenen Displaytechnologie stark an Bedeutung zugenommen.
Einfache örtliche Interpolationsverfahren bestehen aus einer Abtastratenkonversion mit anschließender Filterung. Bei dieser Filterung wird versucht, die durch Abtastratenkonversion entstandenen Wiederholspektren zu unterdrücken und dabei das Grundspektrum weitestgehend zu erhalten. Diese Verfahren sind je nach Länge des benutzten Filters sehr aufwandsgünstig, erreichen jedoch auch nur eine eingeschränkte Bildqualität.
Sogenannte Superresolutionsverfahren (SR) hingegen nutzen Informationen über das vorliegende Bildmaterial und versuchen mit unterschiedlichen Methoden den Bildinhalt des Objektes zu erkennen und diese Information gewinnbringend für die Interpolation einzusetzen. Eine große Klasse dieser Verfahren erreichen die verbesserte Interpolationsqualität weitestgehend über die Ausnutzung zeitlicher Redundanzen und mathematischer Modelle für die Wahrscheinlichkeit bestimmter Bildinhalte. Viele dieser Verfahren benötigen einen hohen Rechenaufwand und können stark sichtbare Artefakte erzeugen, da sie in den meis-

ten Fällen den Bildinhalt nichtlinear auswählen bzw. berechnen, und somit sehr harte Unterschiede zwischen benachbarten Bildbereichen entstehen können. Die Kombination von einer adaptiven nichtlinearen Interpretation des Bildinhaltes und einer Ausrichtung oder Auswahl von linearen Filterkoeffizienten, wird von einer weiteren Gruppe adaptiver Interpolationsverfahren vorgenommen. Dieser Ansatz verbindet den Qualitätsgewinn einer nichtlinearen Analyse mit dem moderaten Verhalten einer Filterung des vorliegenden Bildinhaltes. Als Vertreter dieser adaptiven linearen Interpolationsverfahren erreicht neben stark rechenaufwändigen Verfahren wie NEDI [LO01] vor allem Kondo's Methode (auch bekannt als DRC [KNF+01]) eine sehr gute Qualität bei geringem Aufwand. Dieses Verfahren unterteilt den lokalen Bildinhalt in verschiedene Klassen und benutzt optimierte Filterkoeffizienten für die jeweilige detektierte Klasse.
Der Nachteil dieser bekannten Verfahren liegt vor allem in den nur eingeschränkten und ganzzahligen Interpolationsfaktoren. Es existieren in der Literatur zwar einzelne Ansätze zur Kombination von adaptiven Interpolationsverfahren mit starren Faktoren mit linearen einfachen Interpolationsverfahren mit Polyphasentauglichkeit [ML08] [Li00]. Diese Kombinationen führen jedoch systembedingt zu einer eingeschränkten Leistungsfähigkeit.
Da aber hauptsächlich ungerade Interpolationsfaktoren benötigt werden um die unterschiedlichen Bildquellen mit den verschiedenen Displays zu verbinden, liegt das Hauptaugenmerk dieser Arbeit auf der Polyphasentauglichkeit aller hier eingeführten Verfahren.
Aus diesem Grund wird auf der Basis des aufwandsgünstigen Verfahrens nach Kondo eine örtliche Polyphaseninterpolation vorgestellt. Dieses kann für beliebige ungerade Interpolationsfaktoren trainiert werden und daran anschließend auf der Basis der Filterkoeffizienten eine hochqualitative adaptive Interpolation erreichen.
Da jeder Faktor, der bei der Interpolation genutzt werden soll, eine vorher trainierte Filterdatenbank benötigt, wird im weiteren Verlauf dieser Arbeit ein adaptives Interpolationsverfahren auf der Basis optimierter kontinuierlicher Interpolationsformen (Polynome) vorgestellt. Die Grundidee liegt darin, mit einer Form alle (sonst) einzeln anzulernenden Filterkoeffizienten der verschiedenen Interpolationsphasen zu erfassen. Hierzu wird eine Transformation der zu trainierenden Filterkoeffizienten zu Parametern des Interpolationspolynoms vorgenommen. Mit Hilfe dieser Polynome können dann beliebige Interpolationsfaktoren realisiert werden.
In der Literatur bekannte Ansätze wie die Interpolation mit Hilfe eines adaptiven Polynoms nach Shi/Reichenbach [SR06] sind durch Symmetrieforderungen stark eingeschränkt und parametrisieren mit wenigen freien Parametern lediglich vorgegebene Interpolationspolynome auf der Basis einer lokalen Analyse des Bildinhaltes. Die im Rahmen dieser Arbeit erzeugten Polynome unterscheiden sich hiervon vollständig, da sie zum einen völlig frei parametrisierbar sind und zum anderen diese Parameter wie auch die Polyphasenfilterkoeffizienten mit Hilfe von hoch- und niedrig aufgelösten Trainingssequenzen für jeweils einzelne Bildsituationen (Klassen) optimiert werden.
Die Optimierung erfolgt mit Hilfe einer Minimierung des quadratischen Fehlers. Eine Ein-

schränkung dieser Minimierung liegt in einer nur unzureichenden Beschreibung von Bildqualität durch den mathematischen quadratischen Fehler.
Um diese Beschränkungen umgehen zu können werden eine Vielzahl von kleineren nachträglichen Überprüfungen des Trainingsprozesses vorgeschlagen. Des Weiteren werden aus einem mathematischen Modell Nebenbedingungen abgeleitet, die den Minimierungsprozess auf eine Lösungsmenge einschränken, die die Bildung einiger Artefakte systembedingt ausschließt. Um diese Nebenbedingungen auch bei dem Polynomansatz verwenden zu können, wird ein weiteres Verfahren vorgeschlagen, welches auf der Basis stückweise definierter Polynome basiert.
Die Vielfältigkeit dieser Methode einer Klassifikation und anschließenden optimierten Filterung zeigt sich darin, dass sie - wie auch in der Literatur [HdH07] vorgeschlagen - zur Artefaktreduktion und Schärfeanhebung eingesetzt werden kann. Neben dieser Variante, die auch kurz im Rahmen dieser Arbeit beleuchtet wird, wird die Vielfältigkeit vor allem durch einen Transfer des örtlichen Filterverfahrens in die zeitliche Interpolation von Zwischenbildern vorgenommen.
Für die zeitliche Anpassung von Bildsequenzen (z.B. Kino 24Hz) an das darstellende Display (z.B. 120 Hz) ist die Berechnung von Zwischenbildern notwendig. Neben einfachen Verfahren, die hierzu lediglich eine Mittelung der Bildinformationen vornehmen, basieren fortgeschrittene Verfahren auf der Nutzung von Bewegungsinformationen, um die Zwischenbilder bewegungskompensiert berechnen zu können. Bei dieser Bewegungsschätzung entstehen durch Auf- und Verdeckung von Bildinhalten sowie durch Rauschen Fehler, die zu stark sichtbaren Artefakten führen können. Neben Ansätzen diese Artefakte zu detektieren und zu unterdrücken [dH00] gab es Vorschläge von Blume [Blu97] und Franzen [FS02] gewichtete Medianfilter zur Zwischenbildinterpolation einzusetzen, da diese Filter Kanten trotz Bewegungsfehler phasenrichtig interpolieren können. Nachteilig wirkt sich durch die Medianfilter bei einem kompensierten Bewegungsfehler jedoch ein sichtbarer Detailsignalverlust aus.
Das im Rahmen dieser Arbeit vorgeschlagene zeitliche klassifikationsbasierte Verfahren erreicht hingegen neben der ebenfalls phasengenauen Interpolation von Kanten bei einem Fehler der Bewegungsschätzung zusätzlich den Erhalt selbst feinster Details.

Übersicht über die Kapitel

Die Arbeit ist wie folgt aufgebaut. Im Kapitel 2 werden die Grundlagen zur örtlichen Interpolation vorgestellt. Des Weiteren wird dort auf Details zur phasengenauen Interpolation sowie auf die gängigsten Standardverfahren eingegangen.
Die Grundlagen zum Entwurf linearer Interpolationsfilter über Trainingsdaten werden im Kapitel 3 beschrieben. Hier finden sich auch Methoden zur Einbindung von Nebenbedingungen in den Optimierungsprozess der Filter. Des Weiteren wird im selben Kapitel die adaptive Interpolation nach Kondo auf Grund ihrer Bedeutung für die Entwicklung

der hier vorgestellten Verfahren beschrieben. Im darauf folgenden Kapitel 4 werden diese adaptiven Interpolationsverfahren vorgestellt. Zum einen ein polyphasentaugliches örtliches Interpolationsverfahren, welches auf diskreten vortrainierten Filterkoeffizienten beruht. Zum anderen wird ein Interpolationsverfahren vorgestellt, welches mit Hilfe von vortrainierten kontinuierlichen Filterpolynomen quasibeliebige Interpolationsfaktoren ermöglicht. Außerdem wird gezeigt, dass die klassifikationsbasierte Filterung auch für eine zeitliche Zwischenbildinterpolation genutzt werden kann.
Im Kapitel 5 werden die im Rahmen dieser Arbeit vorgeschlagenen Verfahren hinsichtlich ihrer Leistungsfähigkeit geprüft und mit in der Literatur üblichen adaptiven und nichtadaptiven Standardverfahren verglichen.

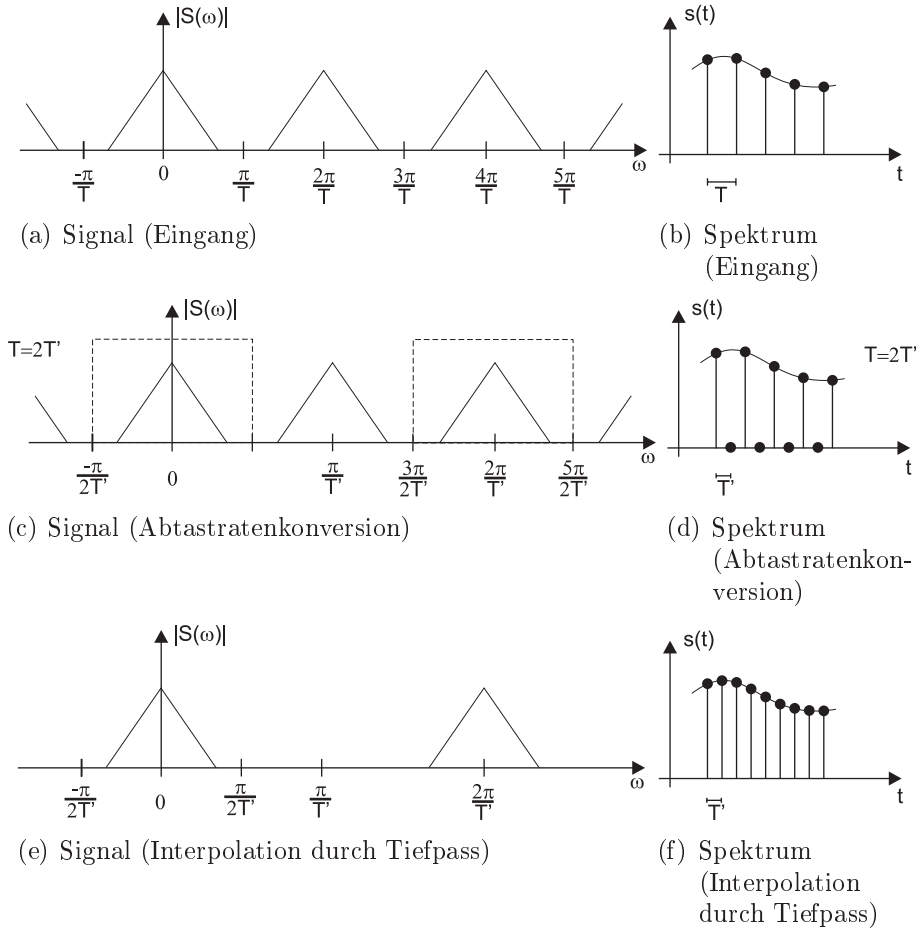
2 Grundlagen und Algorithmen zur Interpolation von Bildern und Bildsequenzen

Unter einer Interpolation versteht man die *Bestimmung von Zwischenwerten einer Funktion aufgrund einer Reihe bekannter Zahlenwerte dieser Funktion* [Bro06]. Im Bereich der Nachrichtentechnik wird die Interpolation als Erhöhung der effektiven Abtastrate $\frac{1}{T}$ angesehen [SB00]. Für einen Verlauf $s(t)$ ergibt sich bei einem Abtastintervall von T die Folge $s[n]$. Will man bei einer gegebenen Folge $s[n]$ die Abtastrate anheben, so benötigt man Verfahren, die die fehlenden Werte $s[m]$ berechnen können. Die Abbildung 2.1 stellt eine Interpolation um den Faktor z. B. $L = 2$ dar. Es wird durch Einfügen von Nullen eine Abtastratenerhöhung und durch Anwendung eines Interpolationstiefpasses eine Unterdrückung der Wiederholspektren erreicht.
Es existieren viele verschiedene Interpolationsverfahren. Diese versuchen möglichst exakt aus den gegebenen Informationen (den vorliegenden Werten) die zusätzlichen Werte zu berechnen. Hierbei unterscheiden sie sich je nach Methode durch Komplexität und Genauigkeit. Grundsätzlich ist bei einer Interpolation keine Erhöhung des Informationsgehaltes möglich.

2.1 Grundlagen zur Polyphasen-Interpolation

2.1.1 Zeitlich

In vielen Fällen muss eine Interpolation mit einem ungeraden Interpolationsfaktor $\frac{L}{M}$ realisiert werden. Theoretisch müsste man zuerst eine Interpolation um den Faktor L und anschließend eine Dezimation um den Faktor M durchführen (siehe auch Abbildung 2.2). Hierbei werden viele Werte berechnet, die entweder ohnehin Null ergeben oder im Anschluss der Berechnung wieder verworfen werden.
Eine Interpretation der Interpolation mit rationalen Faktoren als Digital→Analog→Digital-Wandlung hilft bei der Verdeutlichung des Polyphasenprozesses. Dies ist auch gleichzeitig eine anschauliche Erklärung des Begriffs *Resampling*.



(a) Signal (Eingang)

(b) Spektrum (Eingang)

(c) Signal (Abtastratenkonversion)

(d) Spektrum (Abtastratenkonversion)

(e) Signal (Interpolation durch Tiefpass)

(f) Spektrum (Interpolation durch Tiefpass)

Abbildung 2.1: Interpolation eines Signals um den Faktor 2

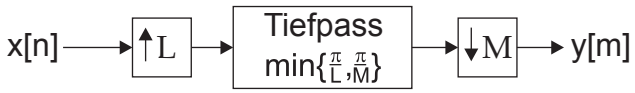


Abbildung 2.2: Interpolation eines Signals um den Faktor $\frac{L}{M}$

In diesem Fall wird die Folge $x[n]$ mit einem idealen D/A Wandler in eine gewichtete δ Impulsfolge $x_{kont}(t) \cdot \text{Ш}_T(t)$ umgewandelt und mit einem Rekonstruktionstiefpass $h_{TP}(t)$ gefiltert. Dies ergibt den kontinuierlichen Signalverlauf $x_{kont}(t)$, welcher an beliebigen neuen Positionen abgetastet werden kann (siehe auch Abbildung 2.3).

In Abbildung 2.4 wird ein solcher Resampling-Prozess für den Faktor $\frac{5}{4}$ dargestellt. Man erkennt, dass nach M (hier 4) ursprünglichen Abtastwerten L (hier 5) neue Abtastwerte erzeugt worden sind. Bei einer weiteren Unterteilung der alten Abtastrate um den

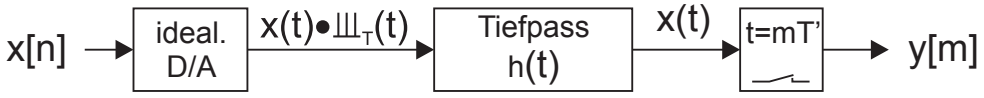


Abbildung 2.3: Interpretation des Resamplings als analoger Prozess

Faktor L, sowie einer weiteren Unterteilung der neuen Abtastwerte um den Faktor M werden auf beiden Rastern $L \cdot M$ Positionen erzeugt. Auf diese Weise erreicht man ein gemeinsames Raster. Dieses entspricht genau demjenigen Raster, welches auch durch eine Überabtastung um den Faktor L erreicht wird.
Der Ausgangswert an einer beliebigen Stelle berechnet sich durch eine Faltung der Eingangssequenz mit dem Interpolationstiefpass [CR83]. Es wird deutlich, dass für einen Ausgangswert an der Stelle m_0 nur wenige Positionen (N_1 bis N_2) der Eingangssequenz von Bedeutung sind.

$$x\left[m\right] = x_{kont}\left(t\right)\big|_{t=m\cdot T'} \tag{2.1}$$

Berücksichtigt man nun, dass die Werte der Eingangssequenz in der diskreten Folge $x\left[n\right]$ vorliegen, so kann man den Ausgangswert $x\left[m\right]$ wie folgt berechnen:

$$x\left[m\right] = \sum_{N_2}^{n=N_1} x\left[n\right] \cdot h_{tp}\left(m \cdot T' - n \cdot T\right)) \tag{2.2}$$

Die Wiederholung des Rasters nach $L \cdot M$ Werten beschränkt auch die Anzahl der relativen Positionen des Tiefpasses bzw. des Ausgangswertes m_0 zu den Werten n. Im Folgenden werden diese Positionen Phasen genannt. Aus diesem Grund wird die Impulsantwort des Interpolationstiefpasses in L zueinander verschobene Teilsignale zerlegt. Jedes dieser Teilsignale ist um einen festen Wert verschoben, so dass die Kombination dieser Polyphasenanteile wiederum die vollständige Impulsantwort des Interpolationstiefpasses (abgetastet mit $\frac{T}{L}$-Werten) ergibt.

$$h_{tp}[k] = h_{tp}\left(k \cdot \frac{T}{L}\right)\Bigg| \forall h_{tp}(t) \neq 0 \tag{2.3}$$

Somit ergibt sich die Berechnung der Werte für die Phasen p mittels folgender Vorschrift.

$$h^p[k] = h\left[kL + p\right] \tag{2.4}$$

Um einen Ausgangswert $y[m]$ zu berechnen, werden sowohl die Phase p als auch die absolute Position in der Eingangsfolge n_m benötigt,

$$\begin{aligned} y[m] \quad &= \quad \sum_k h^p[k] \cdot x\left[n_m - k\right] \\ &mit\ n_m = \lfloor \frac{m \cdot M}{L} \rfloor\ und\ p = (m \cdot M)\ mod\ (L) \end{aligned} \tag{2.5}$$

wobei die Laufvariable k auf den aktiven Bereich des Filters $h[k]$ beschränkt wird.

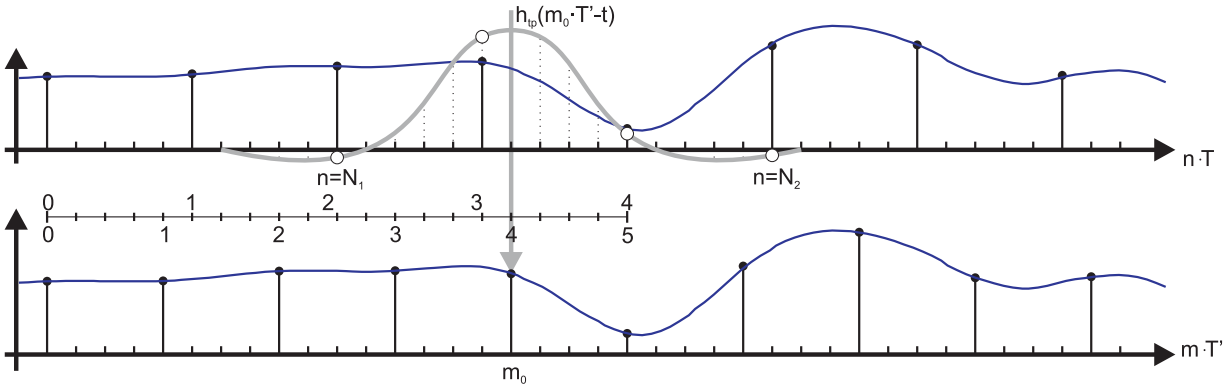


Abbildung 2.4: Resampling Prozess für den Faktor $\frac{5}{4}$

2.1.2 Örtlich

In vielen Bereichen der Signalverarbeitung werden Abtastratenkonversionen vorgenommen. In den meisten Fällen handelt es sich um annähernd unendliche Wertefolgen, deren Auflösung geändert wird, in dem auf der Basis von M Eingangswerten L Werte interpoliert werden. Allgemein wird davon ausgegangen, dass der $0te$ und $(L-1)$te Wert der Eingangswerte mit den Ausgangswerten 0 und $(M-1)$ übereinstimmen.
Bei Betrachtung von zweidimensionalen örtlich-begrenzten Signalen wie Bildern und Bildsequenzen ergeben sich Einschränkungen, da nun die absolute örtliche Position der Start- und End-Werte an Bedeutung zunimmt. Diese Ortsbezogenheit ist bei der Kathodenstrahlröhre, auf Grund der Gaußapertur sowie ihrer linienbasierten und parametrisierbaren Ansteuerung noch gering. Matrixdisplays wie LCD (Liquid Crystal Display) und PDP (Plama Display Panel) hingegen haben eine ortsfeste Position und genaue Breite der Bildpunkte. Dies erfordert eine exakte Definition von Pixelbreiten und Positionen.
Die in vielen Fällen fehlende Berücksichtigung der genauen Position der Pixel führt zu Verschiebungen und Größenabweichungen im Subpixelbereich. Wenn man die Ergebnisse verschiedener Interpolationsverfahren vergleicht, so kann man in einigen Fällen diesen Versatz bemerken. Dies führt vor allem beim objektiven Vergleich von Interpolationsverfahren wie z. B. mittels MQF zu fehlerhaften Werten, wenn die absolute Position und Größe der Referenz von denen des Interpolationsergebnisses abweicht. Man kann diese Abweichungen durch Subpixelverschiebungen des Testmaterials zwar verringern, die Ungenauigkeit der Messung bleibt jedoch vorhanden (siehe auch [LZd03]).

Eine Interpolation von Bildmaterial in SDTV-Auflösung zur HDTV-Auflösung, um z. B. ein SDTV-Fernsehsignal auf einem hochauflösenden Matrixdisplay darzustellen, wird im Folgenden als Beispiel benutzt. Die SDTV-Auflösung liegt bei 720x576 Bildpunkten, die

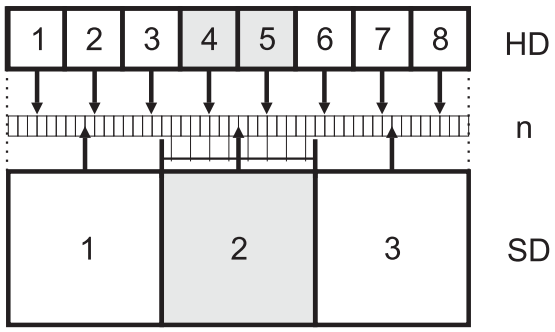


Abbildung 2.5: Phasenzuordnung für einen Interpolationsfaktor $\frac{8}{3}$

HDTV-Auflösung liegt bei 1920 x 1080 Bildpunkten. Der Interpolationsfaktor ergibt sich als gekürzter Bruch der horizontalen und vertikalen Pixelverhältnisse. Im horizontalen Fall liegt also der Faktor $\frac{1920}{720} = \frac{ggT(1920,720)\cdot 8}{ggT(1920,720)\cdot 3} = \frac{240\cdot 8}{240\cdot 3} = \frac{8}{3}$ vor. Wie im Abschnitt 2.1.1 dargestellt, ergeben sich somit acht verschiedene Ausgangsphasen. Da aber, wie zu Beginn dieses Abschnittes erwähnt, bei der Konversion die Pixelbreite und Pixelposition beachtet werden muss, muss die Interpolationsbeziehung (2.5) modifiziert werden.
Hierzu wird jedem Pixel eine Breite zugewiesen. Es wird davon ausgegangen, dass bei einer Auflösung von z. B. drei Pixeln pro Bildschirmbreite, sowie bei einer höheren Auflösung von z. B. acht Pixeln pro Bildschirmbreite sich die Pixel über dieselbe Breite verteilen. Dies bedeutet also, dass die äußeren Ränder des ersten und letzten Pixels beider Auflösungen an derselben örtlichen Position liegen (siehe auch Abbildung 2.5). Zusätzlich wird jedem Pixel eine genaue Position zugewiesen. Im Rahmen dieser Arbeit wird der Schwerpunkt bzw. die Mitte der Pixelfläche für die Position verwendet. In der Abbildung 2.5 sind die Positionen der Pixel mit Pfeilen dargestellt. Man kann hier gut erkennen, dass bei einer Interpolation am Bildrand auch einige Pixel extrapoliert werden müssen (d. h. Interpolationspositionen außerhalb der bekannten Pixelwerte liegen).
Das im vorherigen Abschnitt eingeführte ganzzahlige Raster $L\cdot M$ muss an dieser Stelle um den Faktor 2 erweitert werden, da die Schwerpunkte der Pixel in der Mitte der Pixel und sich somit theoretisch zwischen zwei Rasterpositionen befinden können. Dieses Raster ist ebenfalls in der Abbildung 2.5 als n zwischen den niedrig (SD) und hoch (HD) aufgelösten Pixeln zu erkennen.
Die Phasenberechnung 2.5 verschiebt sich somit um eine halbe Pixelbreite für das SD- und das HD-Pixel. Da sich die Pixel auf dem $L \cdot M \cdot 2$ Raster befinden, ergibt dies einen Versatz von $+M$ und $-L$. Anschließend werden die ganzzahligen Phasenpositionen, da sie mit dem Wert 2 vergrößert wurden, wieder um den Faktor gekürzt. Um Rundungen

zu vermeiden, werden ungerade Phasenpositionen ($(L+M)mod(2)=1$) um 1 erniedrigt. Somit ergibt sich folgende Formel:

$$p = \frac{((m \cdot 2 \cdot M + M - L)\, mod(2 \cdot L) - (L+M)\, mod\,(2))}{2} \tag{2.6}$$

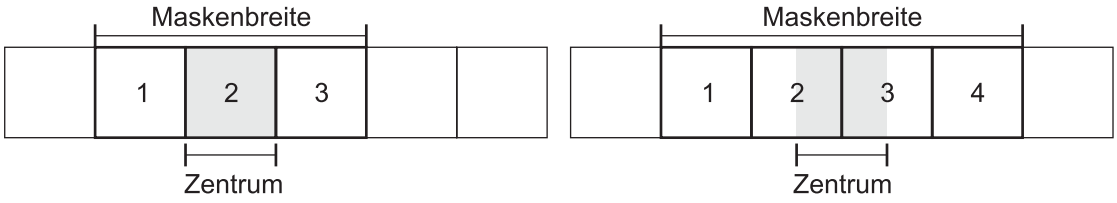
Für das Beispiel einer SDTV $\rightarrow$ HDTV Interpolation lauten die ganzzahligen Phasenpositionen nach Formel 2.6 $[5,0,3,6,1,4,7,2]$. Der absolute Versatz, d. h. die ganzzahlige Phase zwischen den Eingangs- und Ausgangspixeln kann wie folgt bestimmt werden:

$$p_{ganzzahlig} = \frac{((m \cdot 2 \cdot M + M - L)\, mod(2 \cdot L))}{2 \cdot L} \tag{2.7}$$

Dies entspricht den tatsächlichen Phasen $\left[\frac{11}{16}, \frac{1}{16}, \frac{7}{16}, \frac{13}{16}, \frac{3}{16}, \frac{9}{16}, \frac{15}{16}, \frac{5}{16}\right]$. Die Zuordnung einer vorgegebenen HD-Pixelposition m zu einer SD-Pixelposition n_m ist relativ leicht zu bestimmen:

$$n_m = \lfloor \frac{2 \cdot M \cdot m + M}{2 \cdot L} \rfloor \tag{2.8}$$

Innerhalb eines Interpolationsalgorithmus kann auch die umgekehrte Zuordnung, nämlich die Bestimmung zu berechnender HD-Pixel in Abhängigkeit zu den für deren Berechnung genutzten SD-Pixeln entscheidend sein; diese SD-Pixel werden im Folgenden mit SD-Maske bezeichnet. Somit können für eine SD-Maskenposition alle zugehörigen HD-Pixel berechnet, die SD-Maske anschließend weitergeschoben und die für die nächste SD-Maskenposition passenden HD-Pixel berechnet werden. Insgesamt macht es Sinn sich auf diejenigen HD-Pixel zu beschränken, die im Zentrum der SD-Maske liegen. Dieser zentrale Bereich hat die Breite eines SD-Pixels, da somit für die Berechnung der weiter entfernt liegenden HD-Pixel die SD-Maske einfach um ein Pixel verschoben werden kann. Je nach Anzahl der Pixel der SD-Maske (gerade/ungerade) muss ein Versatz des Zentrums von einem halben Pixel berücksichtigt werden. Abbildung 2.6 zeigt dies deutlich für SD-Masken der Größe 3 und 4.



(a) Maskenbreite 3 Pixel (b) Maskenbreite 4 Pixel

Abbildung 2.6: Definition des Zentrums von SD-Masken

Aus diesem Grund muss zwischen geraden und ungeraden Maskenbreiten unterschieden werden:

$$mask_{shift} = \begin{cases} 0.5 & Maskenbreite = gerade \\ 0 & Maskenbreite = ungerade \end{cases} \tag{2.9}$$

Um die Position der SD-Masken beschreiben zu können, wird je nach Maskenbreite das Pixel im tatsächlichen Zentrum oder die nächst kleinere Position benutzt:

$$mask_{mittelpunkt} = \lceil \frac{Maskenbreite}{2} \rceil \tag{2.10}$$

Für den Fall von 3 Pixeln ergibt sich die zentrale Position 2 und für den Fall von 4 Pixeln ebenfalls die Position 2. Die Zuordnung der Position der SD-Masken n zu den zu berechnenden HD-Pixeln kann über die ersten m_n^{first} und letzten m_n^{last} HD-Pixel beschrieben werden:

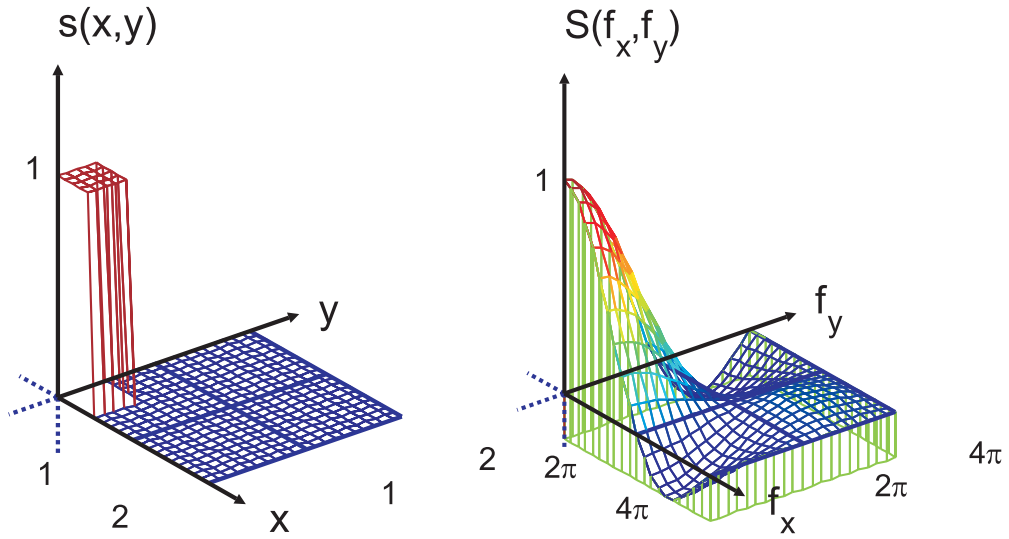
$$m_n^{first} = \lfloor (n + mask_{shift}) \cdot \frac{L}{M} + \frac{1}{2} \rfloor \tag{2.11}$$

$$m_n^{last} = \lfloor (n + 1 + mask_{shift}) \cdot \frac{L}{M} + \frac{1}{2} \rfloor - 1 \tag{2.12}$$

2.2 Lineare örtliche Interpolationsverfahren

Im letzten Jahrzehnt und früher sind sehr viele verschiedene Verfahren für eine örtliche Bildsignalinterpolation vorgeschlagen worden (u. A. [HA78], [TOS92], [RJM96], [KLL96], [Atk98], [LPB+00], [LO01], [HJO+01], [KNF+01], [FJP02], [HCH03], [Hen03], [PPK03], [LZd03]).
Dieser Abschnitt soll die wichtigsten linearen Verfahren kurz darstellen. Hierbei kann man grundsätzlich zwischen statischen und adaptiven Verfahren unterscheiden. Die statischen linearen Verfahren nähern sich mehr oder weniger einem idealen Interpolationstiefpass an (siehe Abschnitt 2.1.1). Ein idealer Tiefpass ist auf Grund seiner unendlichen Impulsantwort nicht realisierbar, so dass die folgenden Interpolationsfunktionen (siehe Abbildungen 2.7 bis 2.11) als aufwandsgünstige Alternativen zu sehen sind.
In vielen Fällen basieren diese Impulsantworten auf eindimensional definierten Funktionen, die separiert in horizontaler und vertikaler Richtung angewendet werden. Auf diese Weise sind horizontal und vertikal unterschiedliche Interpolationsfaktoren sehr leicht realisierbar. Natürlich können diese Impulsantworten auch zu einem zweidimensional definierten Tiefpass durch Multiplikation oder ähnliche Schritte verknüpft werden. Die Anwendung eines zweidimensionalen Tiefpasses erfordert zwar einen erhöhten Rechenaufwand (O^2 anstelle von $2 \cdot O$) andererseits kann auch ein Speicher für den ersten Zwischenschritt eingespart werden und es können mit dieser Struktur auch nicht separierbare 2D Impulsantworten verarbeitet werden.

Zu Beginn werden die gebräuchlichsten eindimensional definierten Interpolationsfunktionen sowie deren multiplikative 2D Impulsantwort vorgestellt.



(a) Impulsantwort Pixelwiederholung

(b) Übertragungsfunktion Pixelwiederholung

Abbildung 2.7: Interpolationsfunktion Pixelwiederholung im Zeit- und Frequenzbereich

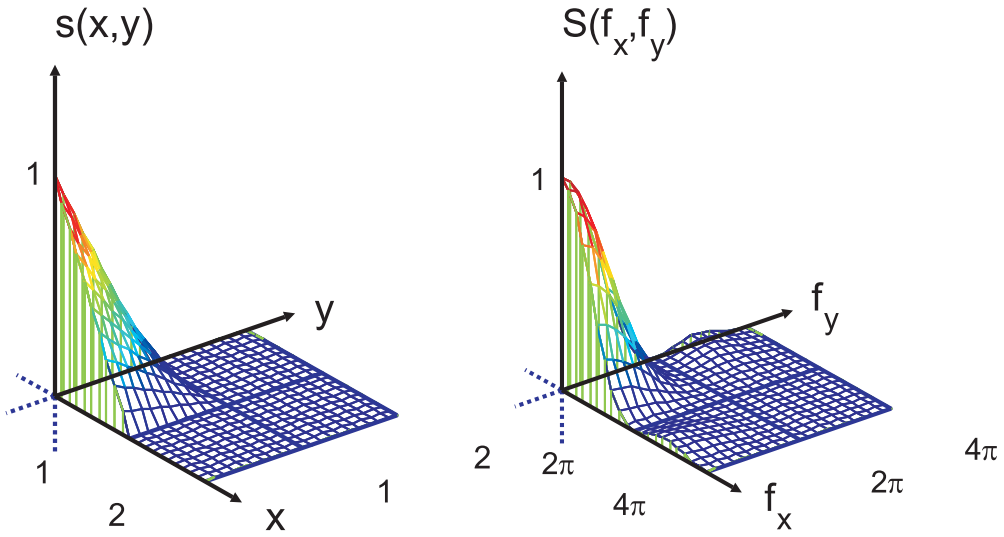
Interpolation durch Pixelwiederholung

Die Interpolation durch Pixelwiederholung (Nearest Neighbour Interpolation) ist die einfachste Interpolationsvariante, bei der die Werte von dem am nächst gelegenen Originalpixel kopiert werden. Dieses Verhalten kann man durch folgende Übertragungsfunktion beschreiben:

$$h(x) = \Pi_{x_0}(x) \tag{2.13}$$

Die Interpolation mit diesem Tiefpass erreicht zwar auf der einen Seite einen guten Erhalt der im Nutzspektrum vorhandenen Frequenzen und somit einen relativ akzeptablen Erhalt der Bildschärfe, andererseits sind Aliasstörungen in Form von Treppeneffekten schon bei geringen Vergrößerungen sichtbar. Die Rechteckfunktion im Zeitbereich korrespondiert mit einer $sinc(x) = \frac{sin(x)}{x}$ Funktion im Frequenzbereich. Die $sinc(x)$ Funktion ist eine sehr schlechte Annäherung eines idealen Tiefpasses, da sowohl im Sperrbereich von π bis 2π zu schwache Dämpfungswerte, als auch im Bereich von 2π bis 4π zu starke Überschwinger existieren.

Des Weiteren kann eine Nearest Neighbour Interpolation auch eine Verschiebung des Gesamtbildes bis zu einem halben Pixel verursachen. Weitere Informationen finden sich u. a. bei [HA78].



(a) Impulsantwort bilinear (b) Übertragungsfunktion bilinear

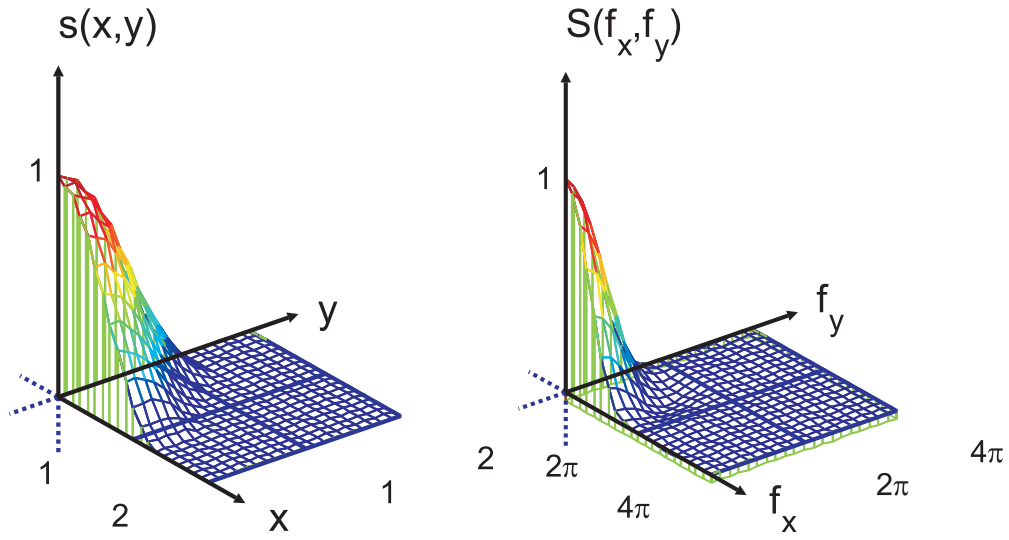
Abbildung 2.8: Interpolationsfunktion bilinear im Zeit- und Frequenzbereich

Bilineare Interpolation

Eine Verbesserung der Interpolationseigenschaften erreicht man durch eine Faltung mit einer Dreiecksfunktion:

$$h(x) = \bigwedge_{x_0}(x) \tag{2.14}$$

Mit Hilfe der Dreiecksfunktion wird eine mit dem Abstand gewichtete Mittelung der angrenzenden Originalwerte erreicht. Man kann die Dreiecksfunktion auch als zweifache Faltung mit der Rechteckfunktion beschreiben. Auf diese Weise erhält man im Spektrum eine Quadrierung des $sinc(x)$ Verlaufes. Auf Grund der stärkeren Dämpfung in der Nähe der Grenzfrequenz entsteht bei der Benutzung dieser Interpolationsfunktion im resultierenden Bild eine leichte Unschärfe. Andererseits erreicht die bilineare Interpolation eine verbesserte Dämpfung oberhalb der Grenzfrequenz.



(a) Impulsantwort Diamondfilter (b) Übertragungsfunktion Diamondfilter

Abbildung 2.9: Interpolationsfunktion Diamondfilter im Zeit- und Frequenzbereich

Diamondfilter

Das Diamondfilter gehört zur Klasse der nichtadaptiven nicht separierbaren Interpolationsfilter. Das in [Hen03] vorgeschlagene Verfahren zur Interpolation basiert wie die bilineare Interpolation auf einer zweifachen Faltung mit einer 2D Rechteckfunktion. Die zweite 2D Rechteckfunktion wird bei diesem Verfahren in seiner Orientierung um 45° gedreht (siehe Abbildung 2.10). Es entsteht somit ein nicht separierbares Gesamtfilter, das wegen seiner Form als Diamondfilter bezeichnet wird. Bei Betrachtung der Übertragungsfunktion des 2D Rechtecks existiert eine unzureichende Unterdrückung von Wiederholspektren in horizontaler und vertikaler Richtung, in diagonaler Richtung jedoch ergeben sich sehr gute Eigenschaften. Da durch die 45° Drehung des zweiten Filters sich das Frequenzverhalten ebenfalls mitdreht, ergibt das multiplikative Ergebnis beider Teilfilter eine gute Unterdrückung der diagonalen, vertikalen und horizontalen Wiederholspektren.
Auf diese Weise erreicht man vor allem ein besseres Filterverhalten an diagonalen Kanten. Durch die starke Dämpfung im Grundspektrum verursacht diese Interpolation jedoch eine starke Unschärfe. Da das Verfahren lediglich auf einer geschickten Addition von einzelnen Pixelwerten basiert und keinerlei Gewichtung der Pixelwerte benötigt, ist der Aufwand sehr gering. Zusätzlich zeichnet sich dieses Verfahren durch eine gute Unterdrückung von Wiederholspektren aus. Der Erfinder dieser Diamondstruktur (Hentschel) schlägt für eine Verbesserung des hochfrequenten Verhaltens vor, die Dämpfung des Nutzspektrums durch

eine Detailsignaladdition auf der Basis einer Quadratur der Detailsignale zu kompensieren (siehe hierzu [HS07]).

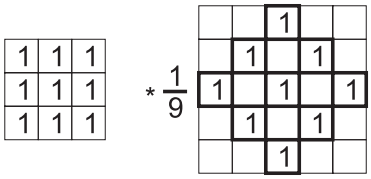
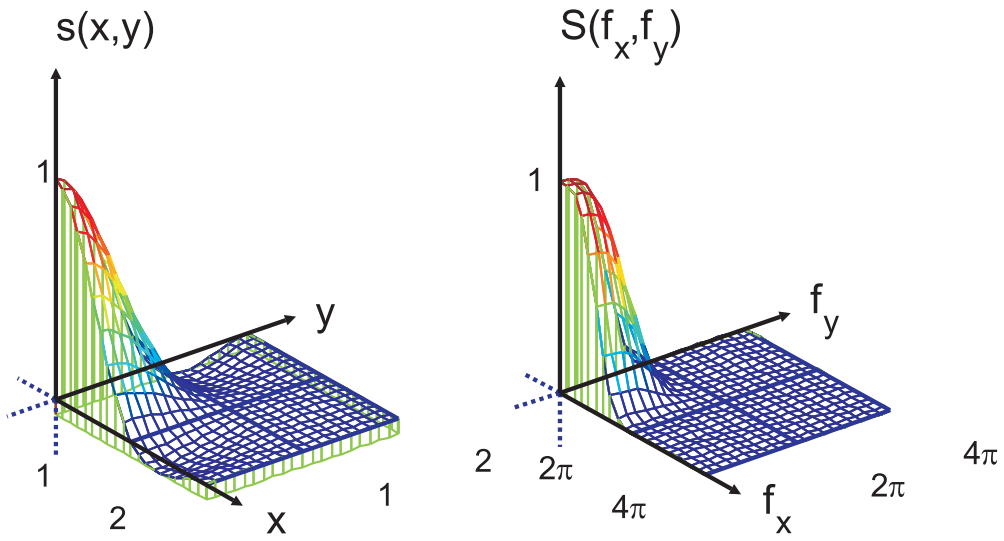


Abbildung 2.10: Diamondfilter (Skalierung um den Faktor 3)

Kubische Interpolation

Über insgesamt vier Pixel erstreckt sich die High-Resolution-Spline Funktion, eine stückweise definierte kubische Funktion:

$$f(x) = \begin{cases} (a+2)x^3 - (a+3)x^2 + 1 & | \, 0 < |x| < 1 \\ ax^3 - 5ax^2 + 8ax - 4a & | \, 1 < |x| < 2 \end{cases} \tag{2.15}$$



(a) Impulsantwort High-Resolution Spline $\alpha = -0.50$

(b) Übertragungsfunktion High-Resolution-Spline $\alpha = -0.50$

Abbildung 2.11: Interpolationsfunktion High-Resolution-Spline im Zeit- und Frequenzbereich

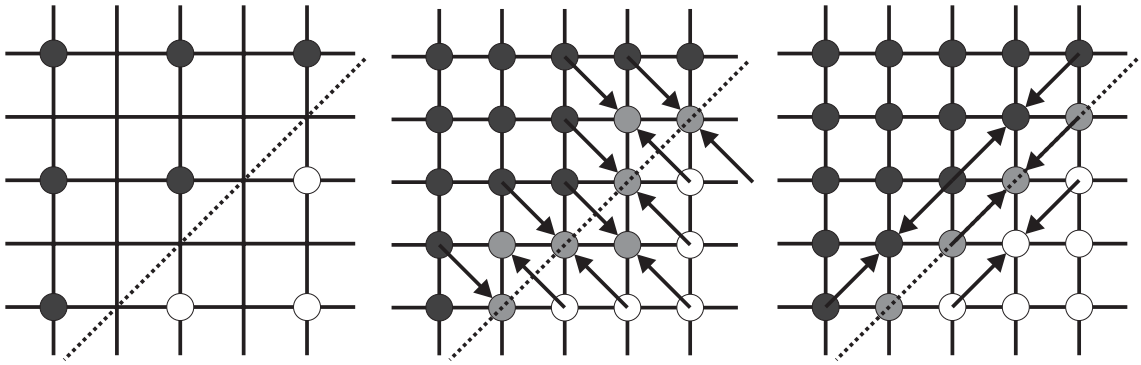
Bei der Wahl eines negativen Wertes für a ist die Funktion im Bereich von 0,0 bis 1,0 positiv und im Bereich 1,0 bis 2,0 negativ. Es sind in der Literatur für a mehrere Werte analysiert worden (-1; -0,75; -0,65 ; 0,5). Hierbei hat sich gezeigt, dass die Übertragungsfunktion bei der Wahl von $a = -0,5$ in den niedrigen Frequenzen nahezu konstant und in nächster Nähe zur Grenzfrequenz eine hohe Flankensteilheit besitzt. In Abbildung 2.11 ist der Frequenzverlauf für diesen Fall ($a = -0,5$) skizziert. Dieses Frequenzverhalten ist vorteilhaft, wenn nachfolgend weitere Bildsignalverarbeitungsalgorithmen (wie z. B. Enhancement oder Rauschreduktion) angewendet werden sollen. Kubische B-Spline Funktionen wurden eingehend u. a. von Hou und Andrews [HA78] untersucht.

2.3 Superresolution

Einfache lineare Verfahren - wie die im vorherigen Abschnitt beschriebenen - versuchen weitestgehend, das Originalspektrum zu erhalten und die Wiederholspektren zu unterdrücken. Die durch die Interpolation erhöhte Bandbreite des Bildes, nämlich genau der Bereich in dem sich vorher die Wiederholspektren befanden, werden nicht genutzt. Bei einer informationstheoretischen Betrachtung wird klar, dass einmal verlorene Informationen nicht mehr wiederhergestellt werden können [Pro00]. Für ein völlig zufälliges Bild (z. B. ein Bild mit weißem Rauschen) ist es deswegen nicht mehr möglich die hochfrequenten Anteile des Bildes wiederherzustellen. In natürlichem Bildmaterial existiert bekanntermaßen eine hohe Redundanz der für das menschliche visuelle System relevanten Informationen. Unter Ausnutzung dieser relevanten Informationen sind Superresolutionsverfahren in der Lage, aus den noch vorhandenen Bildobjekten in bestimmtem Umfang bei Vergrößerung der Redundanz hochfrequente Anteile zu erzeugen. An dieser Stelle kann man die Analogie zur Bildcodierung sehen. Bei der Bildcodierung wird versucht Redundanz im Bild und der Bildsequenz zu detektieren und durch geschickte Transformationen und Quantisierungen zu reduzieren, so dass idealerweise nur noch Bildinformationen übertragen werden. Die Superresolutionsverfahren stellen eine genaue Umkehrung dieses Prinzips dar. Es wird versucht, alle im Bild vorhandenen Informationen zu detektieren und daraufhin durch Redundanzerhöhung eine subjektiv verbesserte Qualität zu erreichen. Um dies zu erreichen, bedienen sich Superresolutionsverfahren der Hilfe von Bildmodellen. Die Bildmodelle dienen zum einen dazu, Informationen im Bild zu detektieren, und zum anderen, die Information unter voller Ausnutzung der zur Verfügung stehenden Auflösung bestmöglich darzustellen. Hierbei ist es möglich, dass Informationen im Bild genutzt werden, die nur in bestimmten Eigenschaftsräumen sichtbar sind (so z. B. Kanten oder komplexere Strukturen im Ortsbereich). Im Folgenden werden einige solcher Interpolationsverfahren vorgestellt. Die jeweiligen Bildmodelle basieren meistens auf der Detektion einer Kante und deren Richtung sowie einer darauf optimierten Interpolation.

2.3.1 Adaptive lineare Interpolation

Die meisten adaptiven linearen Interpolationsverfahren richten ihre Filterkoeffizienten so aus, dass die Interpolationsfilterung möglichst parallel zu Kanten und nicht orthogonal zu ihnen erfolgt. In Abbildung 2.12 ist diese Art der Interpolation schematisch im Ortsbereich dargestellt. Man sieht deutlich den Einfluss der Filterung auf die Breite der entstehenden Kante. In der linken Abbildung ist das Filter orthogonal zur tatsächlichen Kante ausgerichtet und verursacht im Gegensatz zur parallelen Filterung eine stärker sichtbare Unschärfe. Im Folgenden werden einige Verfahren vorgestellt, die auf dieser Grundidee aufbauen.



(a) SW-Kante vor einer Interpolationsfilterung

(b) SW-Kante nach einer ungünstig ausgerichteten Interpolationsfilterung

(c) SW-Kante nach einer gut ausgerichteten Interpolationsfilterung

Abbildung 2.12: Einfluss gerichteter Interpolationsfilter bei der Interpolation einer SW-Kante

Adaptive Filterpositionierung

Einfache Ansätze basieren auf einer Kantenrichtungsdetektion und einer anschließenden Positionierung des Filters in Kantenrichtung. Ivanov hat schon 1994 ein solches Verfahren als Teil seines kombinierten örtlich-zeitlichen Deinterlacers genutzt ([Iva94]). Aus dem Bereich des rein örtlichen Deinterlacings sind außerdem das Edge-Adaptive-Line-Averaging (ELA, [KLL96]) und das modifizierte ELA [LPB+00] zu nennen.
Auf Grund ihres eindimensionalen Aufbaus sind diese Verfahren nur beschränkt für eine zweidimensionale Anwendung geeignet. Es gibt zwar auch zweidimensionale Verfahren wie das Fast-Edge-Oriented-Image-Interpolation (FEI, [HCH03]). Dieses ist durch eine Beschränkung auf vier Kantenrichtungen in der Leistungsfähigkeit stark eingeschränkt.

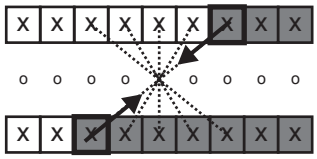


Abbildung 2.13: Grundprinzip ELA und modified ELA

Die Abbildung 2.13 zeigt das Prinzip dieser Verfahren. Auf Grund einer einfachen Kantenrichtungsdetektion, z. B. bei ELA die geringste Differenz von Pixeln in unterschiedlichen Richtungen (hier als gestrichelte Linie dargestellt), wird in Richtung der geringsten Differenz interpoliert, indem die Interpolationskoeffizienten an die entsprechenden Pixel positioniert werden. Es sind auch andere Kantenrichtungsdetektionen oder auch breitere Filter denkbar, wobei alle diese Verfahren auf festen Filterkoeffizienten basieren, die lediglich verschoben werden.

Adaptive Filterkoeffizienten

Eine Verbesserung der Interpolationsqualität erreichen viele Verfahren über eine adaptive Wahl von Filterkoeffizienten. Auf diese Weise ist eine Adaption auch an ungerade Kantenrichtungen möglich. Es existiert zusätzlich durch die zumeist größeren Filter eine Reduktion des Fehlers bzw. der Stärke der nichtlinearen Richtungsentscheidungen.

New Edge Directed Interpolation
X. Li und M. Orchard haben mit NEDI [LO01] ein adaptives Verfahren entwickelt, welches auf der Idee basiert, dass sich die Richtung einer in einem begrenzten Bildausschnitt vorherrschenden Kante bei einer Interpolation nicht ändert. Auf diese Weise können aus den vorliegenden (niedrig aufgelösten) Originalpixeln F_{SD} optimale Filterkoeffizienten abgeleitet werden, die für eine Interpolation von hochaufgelösten Pixeln genutzt werden. Die Interpolation vergrößert um den Faktor 2 in horizontaler und vertikaler Richtung. Hierbei wird der zu interpolierende Bildpunkt F_{HD} (in Abbildung 2.14 durch den zentralen schwarzen Punkt dargestellt) durch eine lineare Filterung der vier in den Diagonalen angrenzenden Originalpixel F_{SD} berechnet. Die Filtergewichte sind als schwarze Pfeile dargestellt.
Diese Gewichte werden folgendermaßen berechnet:
Zu Beginn wird eine Pixelumgebung B definiert. Für jeden einzelnen dieser Pixel werden Originalpixel bestimmt, die in derselben diagonalen Struktur aufgebaut sind wie bei der Interpolation des HD-Pixels. Nun wird angenommen, dass man jedes dieser Pixel der Umgebung B durch die vier benachbarten Diagonalpixel C interpolieren kann. Unter der Bedingung, dass sich die Ausrichtung einer Kante in einem bestimmten Bereich nicht stark

ändert, wird angenommen, dass die für die Interpolation jeweils notwendigen Gewichte ähnlich sind und dementsprechend auch als Filtergewichte für die HD-Interpolation genutzt werden können.
Durch eine Minimierung des mittleren quadratischen Fehlers (MQF, siehe Kapitel 3.1.2) der Originalpixelwerte aus der Matrix B und der Werte, die bei einer Interpolation mit Hilfe der zugehörigen Diagonalpixel entstehen, werden die für diesen Ausschnitt optimalen Gewichte bestimmt.

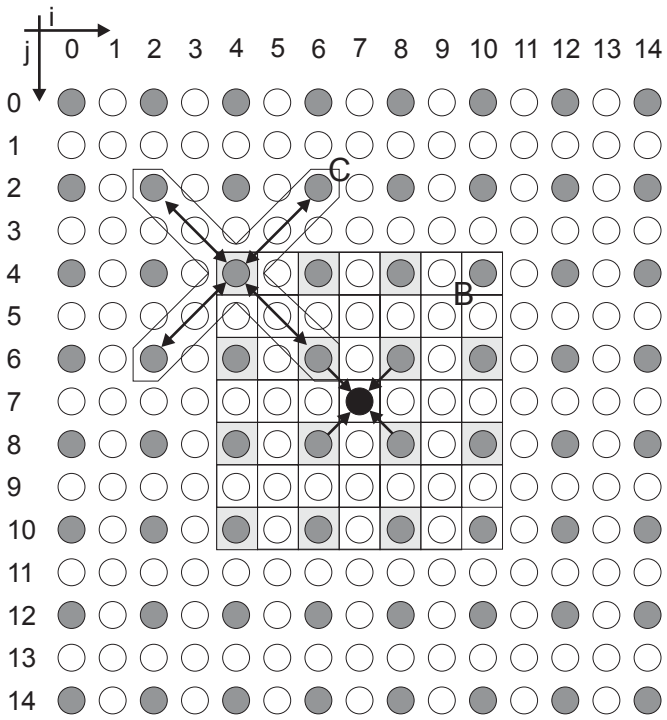


Abbildung 2.14: Apertur für NEDI. Das Originalpixel wird aus vier Original-Pixeln interpoliert, Matrix B dient zur Bestimmung der vier Gewichte, C stellt die Trainingsdaten dar. Die Originalpixel sind dunkelgrau und die zu interpolierenden Pixel weiß bzw. das aktuell zu interpolierende schwarz dargestellt [LO01]

.

Resolution Synthesis
Das Verfahren Resolution Synthesis [ABA01] basiert ebenfalls auf einer bildinhaltsabhängigen Filtergewichtung. Im Gegensatz zu NEDI wird an dieser Stelle zu Beginn eine Klassifikation des vorliegenden Bildmaterials und anschließend eine lineare Interpolation

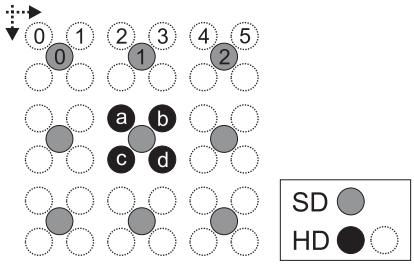


Abbildung 2.15: Pixelzuordnung Kondo's Methode

mit optimierten Filterkoeffizienten vorgenommen.

Die Optimierung der Filterkoeffizienten erfolgt hierbei nicht zur Laufzeit sondern vorher. Resolution Synthesis basiert hierbei auf einer Berechnung der Wahrscheinlichkeit der Zugehörigkeit des aktuellen Bildinhaltes zu einer (oder mehreren) vortrainierten Klassen mit Hilfe des Gaussian Mixture Models (siehe auch [Atk98] sowie [ABA01]).

Hierbei wird der vorliegende Bildinhalt auf die lokale mittlere Helligkeit bezogen und anschließend diese Differenz auf verschiedene in einer Datenbank abgespeicherten Verläufe abgebildet. Mit der gewonnenen Abbildung werden dann die den Klassen entsprechenden Filter auf den Bildinhalt angewendet und abschließend aufaddiert. Somit ergibt sich ein weiches Umschalten zwischen den einzelnen Klassen. Zusätzlich wird in der Dissertation von Atkins ([Atk98]) eine alternative Ansteuerung in Form einer harten Entscheidung der Bildklassen vorgestellt.

Kondo's Methode
Ein zur Resolution Synthesis sehr ähnliches Verfahren ist Kondo's Methode, vielfach auch bekannt als Teil des Digital Reality Creation Chips (DRC, Sony, [KNF+01]). Auch hierbei ist die Grundidee eine Klassifizierung des Bildinhaltes. Im Gegensatz zu dem RS-Verfahren wird die Klassifizierung hart durch eine Binarisierung der Differenzen des Bildinhaltes zur mittleren lokalen Helligkeit vorgenommen.

Nach der Klassifizierung der benachbarten SD-Pixel (z.B. 3x3) erfolgt die Interpolation, wobei jeweils vier HD-Pixel berechnet werden müssen (siehe Abbildung 2.15). Hierzu sind in der Datenbank für die erkannte Klasse Filterkoeffizienten für alle vier HD-Pixel (a, b, c, d) abgespeichert, die der Reihe nach interpoliert und ins Ausgangsbild geschrieben werden. Die jeweilige Interpolation ist eine einfache FIR Filterung der benachbarten SD-Pixel.

Die Werte für die einzelnen Klassen werden anhand mehrerer Trainingssequenzen mit Hilfe einer Minimierung des mittleren quadratischen Fehlers (MQF, siehe auch 3.1.2) im Labor optimiert und können während der Interpolation des aktuellen Bildbereichs einfach aus einer Tabelle ausgelesen werden.
Durch diese strikte Unterteilung in ein aufwendiges Training der Filterkoeffizienten und eine einfache Anwendung dieser sowie durch die Tatsache, dass die Qualität der Interpolation trotz der einfachen Struktur sehr hoch ist, ist dieses Verfahren für die aktuelle Fernsehtechnik sehr attraktiv. Dies zeigen auch bisherige Realisierungen als dedizierte Hardware (wie der DRC-Chip) und selbst Software-Echtzeitrealisierungen (wie auf dem Cell-Chip der Playstation 3). Auf Grund der hohen Bedeutung und Leistungsfähigkeit dieses Verfahrens und weil die Verfahren dieser Arbeit auf der selben Grundidee einer einfachen lokalen Klassifikation und der Nutzung von optimierten Filterkoeffizienten für die jeweilige Bildklasse beruhen, wird es im Kapitel 3.2 genauer beschrieben.

2.3.2 Detailsignaladdition

Im Gegensatz zu einer adaptiven linearen Filterung basieren viele Superresolutionsansätze auf der Idee, die fehlenden hochfrequenten Detailinformationen additiv hinzuzufügen. Diese additiven Detailsignalverfahren werden in der Literatur oft als Interpolationsverfahren aufgeführt. Auf Grund der Addition auf das schon interpolierte Bild sollte man diese jedoch zur Klasse der Enhancementverfahren zählen, wobei der Übergang jedoch fließend ist.
Um passende Details auf ein interpoliertes Bild addieren zu können, werden in vielen Fällen große Datensätze mit tieffrequenten und passenden hochfrequenten Signalanteilen benötigt. Informationen aus den vergrößerten tieffrequenten oder auch direkt aus den niedrigaufgelösten Bildausschnitten werden dann in einer Datenbank gesucht und die passenden Informationen hinzuaddiert. Beispielhaft können hier W. T. Freeman [FJP02] sowie A. Hertzman [HJO+01] genannt werden.
Für die Wahl der passenden Bildausschnitte gibt es ebenfalls mehrere Verfahren, da hierbei sowohl der Grad der Übereinstimmung als auch die Umgebung des aktuellen Ausschnittes berücksichtigt werden müssen.
Ein weiterer Ansatz existiert für Bildbereiche mit unregelmäßigen Texturen. Da die Wahrscheinlichkeit sehr gering ist, für unregelmäßige Texturen passende Inhalte in einer Datenbank vorliegen zu haben, wird bei [LE07] vorgeschlagen, Detailsignale durch eine örtliche Resynthese der hochfrequenten Anteile der niedrig aufgelösten Eingangsbilder zu erzeugen. Trotz der guten Ergebnisse, kann dieses Verfahren noch nicht für eine SD→HD-Konversion benutzt werden, da bisher keine ausreichende Qualität in der automatischen Erkennung und Trennung von unregelmäßigen Bildbereichen erreicht wird.

2.3.3 Interpolation durch mathematische Modelle

Eine weitere große Klasse der Superresolutionsverfahren baut auf mathematischen Modellen zur Erzeugung von Bildinhalten auf. Es wird auch hier versucht unter Zuhilfenahme örtlicher und vor allem zeitlicher Redundanzen Informationen aus den bandbegrenzten Bildern zu gewinnen.

Nonuniform-Interpolation

Ausgehend von einfachen Modellen, wie Nonuniform-Interpolation [PPK03], die in drei getrennten Schritten versucht, zuerst Informationen in niedrigaufgelösten Bildsequenzen zu erkennen, zu interpolieren und anschließend zu rekonstruieren, sind heutzutage komplexere Modelle von höherer Bedeutung, die nicht mehr in derartige Einzelschritte aufgeteilt werden können.

Iterative-Backward-Projection

Bei dem Iterative-Backward-Projection Verfahren wird von einer hochaufgelösten Szene ausgegangen. Diese wird durch ein Abbildungsmodell auf eine niedrig aufgelöste Testsequenz abgebildet und diese mit der bekannten niedrigaufgelösten Bildsequenz verglichen. An Hand der festgestellten Unterschiede beider Sequenzen wird nun wiederum auf die notwendige Änderung der hochaufgelösten Szene „zurück projiziert“.

POCS

Eine weitere Methode zur Superresolution ist das Projection-onto-convex-sets (POCS) Verfahren. Diese Methode wird vor allem in der Bildrekonstruktion häufig verwendet. Der Vektorraum $\mathbb{R}^{N_1 x N_2}$, der alle hochaufgelösten Bilder der Auflösung N_1xN_2 enthält, wird mit Hilfe von Nebenbedingungen in der Form von konvexen Mengen eingeschränkt. Hierzu zählen positiver Signalverlauf, begrenzte Energie und eine gewisse Glattheit des zu erzeugenden Bildes.
Die POCS-Methode stellt den Lösungspunkt, der alle Nebenbedingungen erfüllt, iterativ fest. Hierbei wird solange zwischen den Projektionen alterniert bis ein Punkt in der Schnittmenge aller Nebenbedingungen konvergiert. Es kann bei diesem Ansatz keine eindeutige Lösung garantiert werden, zumal die Lösung stark vom Startpunkt abhängt (siehe auch [TOS92] und [RJM96]).

Stochastische Modelle

Eine weitere große Gruppe der mathematischen Superresolutionsverfahren basiert auf stochastischen Methoden. Es werden die beobachteten niedrigaufgelösten Bilder y einer Szene, die hochaufgelöste Szene z und additives Rauschen als Realisierungen von Zufallsgrößen aufgefasst und mit Hilfe von Verteilungsdichten beschrieben. Die Schätzung erfolgt durch eine Maximierung der a-posteriori Wahrscheinlichkeit $P(z|y)$.
Aus diesem Grund werden diese Verfahren auch MAP-Verfahren genannt. In vielen Fällen wird die Wahrscheinlichkeit durch das Bayes'sche Theorem umformuliert, so dass zwei Wahrscheinlichkeiten maximiert werden müssen: Die Wahrscheinlichkeit des Auftretens der bekannten niedrigaufgelösten Szene y bei Schätzung einer hochaufgelösten Szene z ($P(y|z)$) sowie die Wahrscheinlichkeit der hochaufgelösten Szene selbst ($P(z)$). Diese beiden Elemente können jeweils als Abbildungsmodell und als Bildmodell interpretiert werden [SS96]. Für weitere Details wird auf die genannte Literatur verwiesen.

3 Optimierung linearer Interpolationsfilter

Dieses Kapitel gibt zu Beginn eine kurze Übersicht zu Entwurfsverfahren linearer Interpolationsfilter. Anschließend wird die für diese Arbeit wichtige Optimierung auf der Basis von Trainingsdaten beschrieben. Hierzu wird vor allem auf die mathematischen Grundlagen der Optimierung von linearen Filterstrukturen durch die Minimierung des quadratischen Fehlers eingegangen. Abschließend wird ein auf dieser Optimierung basierendes Verfahren, die Interpolationsmethode nach Kondo [KNF+01], vorgestellt. Dies dient zum einen als Anwendungsbeispiel für eine Optimierung von Interpolationsfiltern mittels Trainingsdaten. Zum anderen basieren die im Rahmen dieser Arbeit eingeführten Verfahren ebenfalls auf dieser Optimierung und benutzen das selbe Grundprinzip einer einfachen lokalen Klassifikation des vorliegenden Bildinhalts mittels einer Binarisierung, so dass diese Punkte detailliert beschrieben werden.

Neben dem Entwurf von Filtern mittels Trainingsdaten existieren sehr viele Entwurfsverfahren für FIR-Filter, in denen z. B. ein gewünschter Frequenzgang oder andere Wunschkriterien vorgegeben werden.

Entwurf auf Basis eines Sollfrequenzgangs

Der einfachste Ansatz ist der Entwurf der Filterkoeffizienten durch Minimierung der quadratischen Abweichung des Sollfrequenzganges vom Approximationsfrequenzgang. Dies führt dazu, dass sich die Koeffizienten des FIR Filters einfach durch eine Ausschnittsbildung aller Filterkoeffizienten des in den Zeitbereich transformierten Sollfrequenzganges ergeben. Auf Grund der Ausschnittbildung ergeben sich jedoch nicht gewünschte Überschwinger, die durch Wahl „weicherer“ Fenster (Kaiser, Hamming, Hanning, Cos-Roll off) bei Verlust von Flankensteilheit weitestgehend gemindert werden können.
Ein weiteres Entwurfsverfahren für FIR-Filter ist das Tschebyscheff-Verfahren. Im Gegensatz zur Minimierung der quadratischen Abweichung wird hier die maximale Differenz zwischen dem Soll- und dem Approximationsfrequenzgang minimiert. Um dies zu erreichen, gibt es Verfahren wie den Parks-McClellan Algorithmus, der sich in iterativer Form der Lösung annähert.

Weitere Sollgrößen

Die Abweichung vom Sollfrequenzgang ist natürlich nur ein mögliches Qualitätskriterium beim Entwurf von FIR-Filtern in der Bildsignalverarbeitung. Kantensteilheit, Position und Größe der ersten und zweiten Überschwinger im Ortsbereich haben ebenfalls einen großen Einfluss auf das Verhalten eines Filters.
Da der Entwurf eines derartigen Filters von vielen verschiedenen Parametern abhängt, muss eine Gesamtgütefunktion definiert werden. In dem darauffolgenden Schritt muss dann die Optimierung vorgenommen werden, was in vielen Fällen zu einem nichtlinearen Optimierungsproblem führt. Diese sind mit gängigen Gradienten-basierten Verfahren nicht zu lösen. Vielmehr sind evolutionäre Optimierungsstrategien denkbar (siehe auch [Rec94], [SB00]). Diese sind jedoch sehr rechenaufwendig und garantieren zum anderen nicht, das globale Maximum finden zu können.
Bei [FBS98] wird gezeigt, dass die Optimierung solcher Filter sehr gut mittels evolutionärer Strategien erreicht werden kann, wobei auch hier der Aufwand sehr hoch ist und für eine beschleunigte Konvergenz zusätzlich ein hohes Wissen über das zu optimierende System vorhanden sein muss.

3.1 Entwurf linearer Interpolationsfilter durch Trainingsdaten

Eine weitere Möglichkeit lineare Filterkoeffizienten zu bestimmen, ist die Nutzung von Trainingsdaten. Für diese Optimierung muss natürlich auch ein geeignetes Optimierungskriterium herangezogen werden. Bewertet und minimiert werden in den meisten Fällen die Abweichungen der Filterergebnisse (bzw. der approximierten Daten) von den Referenzausgangsdaten bei unterschiedlichen Filterkoeffizienten und gegebenen Referenzeingangsdaten. Im Rahmen dieser Arbeit wird die Optimierung mittels Minimierung des quadratischen Fehlers (MQF bzw. Mean Square Error, MSE) vorgenommen und in den folgenden Abschnitten vorgestellt.

3.1.1 Definition des mittleren quadratischen Fehlers

Ein sehr stark verbreitetes objektives Fehlermaß ist der mittlere quadratische Fehler (MQF bzw. Mean Square Error, MSE). Der MQF entspricht im statistischen Sinne der Varianz der Differenz der approximierten und der Referenzausgangsdaten. Bei Bildern ergibt sich der MQF für die N_xxN_y Pixel großen Bilder F_{ref} und F_{approx} wie folgt:

$$MQF(F_{ref}, F_{approx}) = \frac{1}{N_x \cdot N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \left[F_{ref}\left(x, y\right) - F_{approx}\left(x, y\right)\right]^2 \tag{3.1}$$

Da der MQF zur Optimierung linearer Filter benutzt werden soll, kann man die approximierten Ausgangspixel (F_{approx}) aus einer Linearkombination von Eingangspixeln (F_{input} z. B. niedrig aufgelöste Bildpixel) errechnen:

$$F_{approx}(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j} \cdot F_{input}(x+i, y+j) \tag{3.2}$$

$w_{i,j}$ steht hierbei für die Filterkoeffizienten, die für die Linearkombination der Eingangsdaten F_{input} benutzt werden. Die Definition des Mittelpunktes bzw. der Position und der Größe des Filters ist an dieser Stelle willkürlich gewählt und dient nur einer vereinfachten anschaulichen Darstellung.
Anstelle der Interpretation als Varianz kann man den MQF auch als Maßeinheit der Länge eines Fehlervektors interpretieren. Die Eingangspixel (F_{input}) werden im Folgenden zur Vereinfachung als bekannte Eingangsdaten $\vec{sd}$ und die relevanten Filterkoeffizienten ($w_{i,j}$) als Vektor $\vec{w}$ dargestellt. Dies gilt analog für die approximierten Ausgangspixel (F_{approx}).
Ein Ausgangswert hd_{approx} wird durch die Multiplikation von $\vec{sd}^T$ mit den passenden Filterkoeffizienten $\vec{w}$ ausgedrückt. Da mehrere Werte als Trainingsdaten zur Verfügung stehen, kann man alle Ausgangswerte als einen Vektor $\vec{hd}_{approx}$ schreiben. Der transponierte Vektor $\vec{sd}^T$ wird in diesem Fall zu der Matrix SD, in deren Zeilen sich die zusammenhängenden Eingangswerte befinden.

$$\vec{hd}_{approx} = SD \cdot \vec{w} \tag{3.3}$$

Der Wert hd_{ref} repräsentiert den gewünschten Referenzausgangswert. Der Unterschied zwischen dem Referenzausgangswert und dem approximierten Ausgangswert ergibt sich wie folgt:

$$e = hd_{ref} - \vec{sd}^T \cdot \vec{w} \tag{3.4}$$

$$\vec{e} = \vec{hd}_{ref} - SD \cdot \vec{w} \tag{3.5}$$

Der Vektor $\vec{e}$ ist also der Vebindungsvektor der Vektoren die durch die Referenzwerte und die approximierten Werte aufgespannt werden (siehe auch Abbildung 3.1).
Die Länge des Vektors ist also eine Größe für den Unterschied von Referenz und Approximation im n-dimensionalen Raum, wobei n der Anzahl der Referenzwerte entspricht.
Die euklidische Länge des Vektors $\vec{e}$ kann nun also als Fehlermaß genutzt werden, wobei zur Vereinfachung der nachfolgenden Optimierungen natürlich auch das Quadrat dieser Summe genutzt werden kann.

$$||\vec{e}||_2 = \sqrt{(e_1^2 + e_2^2 + ... + e_n^2)} \tag{3.6}$$

$$alternativ: ||\vec{e}||_2^2 = \left(e_1^2 + e_2^2 + ... + e_n^2\right) \tag{3.7}$$

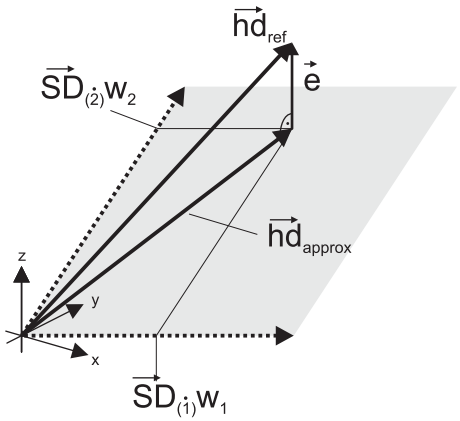


Abbildung 3.1: Graphische Interpretation des mittleren quadratischen Fehlers

3.1.2 Minimierung des mittleren quadratischen Fehlers

Für eine Verbesserung von Filterkoeffizienten kann nun das Maß des mittleren quadratischen Fehlers genutzt werden. Es existieren in der Literatur sehr viele verschiedene Herleitungen und Motivationen zur Minimierung des mittleren quadratischen Fehlers. Selbst im Bereich der Bildsignalverarbeitung, speziell der örtlichen Interpolation, wird der MQF auf unterschiedliche Weise hergeleitet [KNF+01, LO01].
Auf der Basis der vektoriellen Schreibweise von $\vec{e}$ und der Formel 3.5 ergibt sich die mathematische Herleitung wie folgt. Gesucht sind die optimalen Filtergewichte $\vec{w}$, die das Fehlermaß $\vec{e}$ am besten minimieren:

$$min \left\|\vec{e}\right\|_2^2 = min \left\langle \vec{e}, \vec{e} \right\rangle = min \left\langle \vec{hd} - SD\vec{w}, \vec{hd} - SD\vec{w} \right\rangle \tag{3.8}$$

Eine notwendige Bedingung für dieses Minimum ist, dass die erste Ableitung nach $\vec{w}$ Null ergibt. Die Ableitung des $MQF(Skalar)$ nach einem Vektor ergibt wiederum einen Vektor. Unter Berücksichtigung der Summenschreibweise nach 3.8 ergibt sich:

$$\frac{\partial}{\partial \vec{w}} MQF\left(\vec{w}\right) = \frac{\partial}{\partial \vec{w}} \left\langle \vec{hd}, \vec{hd} \right\rangle - 2\frac{\partial}{\partial \vec{w}} \left\langle \vec{hd}, SD\vec{w} \right\rangle + \frac{\partial}{\partial \vec{w}} \left\langle SD\vec{w}, SD\vec{w} \right\rangle \stackrel{!}{=} \vec{0} \tag{3.9}$$

Da $\vec{hd}$ nicht von $\vec{w}$ abhängig ist, ergibt die Ableitung des ersten Terms Null. Die Ableitung wird in die Skalarmultiplikation des zweiten und dritten Terms hineingeschoben und dann mit Hilfe der Produktregel aufgespalten. Als Hilfsgröße wird hier ein „·“ eingefügt, um eine anschließende Multiplikation mit einem Vektor anzudeuten. Dies ist notwendig, da die Ableitung eines Vektors eine Matrix ergibt und das Skalarprodukt eigentlich nur für Vektoren definiert ist.

$$\begin{aligned} \vec{0} = \quad & \vec{0} - 2\left\{ \left\langle \tfrac{\partial}{\partial \vec{w}} \vec{hd} \cdot, SD\vec{w} \right\rangle + \left\langle \vec{hd}, \tfrac{\partial}{\partial \vec{w}} \left(SD\vec{w}\right) \cdot \right\rangle \right\} \\ & + \left\{ \left\langle \tfrac{\partial}{\partial \vec{w}} \left(SD\vec{w}\right) \cdot, SD\vec{w} \right\rangle + \left\langle SD\vec{w}, \tfrac{\partial}{\partial \vec{w}} \left(SD\vec{w}\right) \cdot \right\rangle \right\} \end{aligned} \tag{3.10}$$

Die Ableitung des ersten Terms ergibt wiederum $\vec{0}$ und unter Ausnutzung des Kommutativgesetzes können die letzten beiden Terme zusammengefasst werden:

$$\vec{0} = -2\left\langle \frac{\partial}{\partial \vec{w}} (SD\vec{w}) \cdot, \vec{hd} \right\rangle + 2\left\langle \frac{\partial}{\partial \vec{w}} (SD\vec{w}) \cdot, SD\vec{w} \right\rangle \tag{3.11}$$

Da die Ableitung von $(SD\vec{w})$ nach $\vec{w}$ die Matrix SD ergibt und $\left\langle \vec{a}, \vec{b} \right\rangle$ auch durch $\vec{a}^T \cdot \vec{b}$ beschrieben werden kann, folgt:

$$\begin{aligned} \vec{0} &= -2SD^T \cdot \vec{hd} + 2SD^T \cdot SD\vec{w} && (3.12)\\ \Leftrightarrow SD^T \cdot SD \cdot \vec{w} &= SD^T \cdot \vec{hd} && (3.13)\\ \Leftrightarrow \vec{w} &= \left(SD^T \cdot SD\right)^{-1} \cdot SD^T \cdot \vec{hd} && (3.14) \end{aligned}$$

Eine alternative Herleitung der Formel 3.14 nach [Göt06] kann über eine geometrische Betrachtung erfolgen. In Abbildung 3.1 wird für den vereinfachten Fall von 3 Trainingssamples und 2 gewünschten optimalen Filterkoeffizienten (Dimension von $\vec{w} = 2$) der Referenz-Vektor $\vec{hd}_{ref}$, der approximierte Vektor $\vec{hd}_{approx}$ sowie der Differenzvektor $\vec{e}$ dargestellt. Die Spalten der Matrix SD können als Vektoren $\vec{SD}_{(1)}$ und $\vec{SD}_{(2)}$ interpretiert werden, die gemeinsam mit den skalaren Filterkoeffizienten w_1 und w_2 eine zweidimensionale Lösungsebene im dreidimensionalen Lösungsraum aufspannen.
Bei der Optimierung mittels MQF-Minimierung wird, wie ausgeführt, der kleinste Differenzvektor $\vec{e}$ gesucht. Da der Referenzvektor sehr wahrscheinlich nicht in der aufgespannten Ebene liegt, ist offensichtlich der Vektor $\vec{e}$ am kleinsten, wenn er senkrecht auf der aufgespannten Ebene steht.
Dies wiederum bedeutet, dass die Skalarprodukte des Fehlervektors mit den aufspannenden Vektoren $\vec{SD}_{(1)}$ und $\vec{SD}_{(2)}$ Null ergeben müssen:

$$\begin{aligned} 0 &\stackrel{!}{=} \left\langle \vec{SD}_{(0)}, \vec{e} \right\rangle = \vec{SD}_{(0)}^T \cdot \vec{e} \;\cap\; 0 \stackrel{!}{=} \left\langle \vec{SD}_{(1)}, \vec{e} \right\rangle = \vec{SD}_{(1)}^T \cdot \vec{e} \\ \Rightarrow \vec{0} &\stackrel{!}{=} SD^T \cdot \vec{e} \end{aligned} \tag{3.15}$$

Multipliziert man nun die Formel 3.5 mit SD^T, erhält man:

$$SD^T \cdot \vec{e} = SD^T \cdot \vec{hd} - SD^T \cdot SD \cdot \vec{w} \stackrel{!}{=} \vec{0} \tag{3.16}$$

Somit gelangt man über die geometrische Betrachtung ebenfalls zu der Formel 3.13.
Wenn nun die Inverse der Matrix $SD^T \cdot SD$ definiert ist, kann eine Lösung für das Gleichungssystem erreicht werden.

3.1.3 Berücksichtigung von Nebenbedingungen

Die Minimierung des mittleren quadratischen Fehlers ist eine zwar einfache und leistungsfähige Optimierung, sie kann in bestimmten Fällen jedoch zu nicht idealen Lösungen führen. Diese beiden Eigenschaften (die einfache Optimierung, sowie die Gefahr einer nichtidealen Lösung) liegen in dem einfachen Fehlermaß begründet, welches weiteres Wissen über das zu optimierende Filter nicht berücksichtigt.
Man kann diese Probleme reduzieren, indem man das Bekannte der Lösungsmenge in Form von Nebenbedingungen beschreibt und somit die Lösungsmenge einschränkt.

Mit Hilfe der Nebenbedingungen stellt man die bekannten Zusammenhänge zwischen den Parametern dar und lässt dieses Wissen in die Optimierung einfließen. Auf diese Weise wird die Dimension der Lösungsmenge verringert und gleichzeitig aus dem Optimierungsproblem mit Nebenbedingungen ein Optimierungsproblem ohne Nebenbedingungen mit niedrigerer Ordnung erzeugt. Zur Veranschaulichung wird ein einfaches Beispiel gegeben. Es liegt eine Optimierung der Parameter w_1 bis w_4 vor. Wenn man nun weiß, dass der Parameter w_1 mit dem Parameter w_4 übereinstimmen soll, dann kann dies dazu genutzt werden, um w_4 durch w_1 zu beschreiben und die Optimierungsmenge auf w_1 bis w_3 zu beschränken. Der genaue Formalismus soll im Folgenden vorgestellt werden.

Man geht von einem Optimierungsproblem wie in Formel 3.8 aus. Der Vektor $\vec{w}$ besteht aus m Filterkoeffizienten. Zusätzlich nutzt man das Wissen über die Filterkoeffizienten in Form des Gleichungsystems $A \cdot \vec{w} \stackrel{!}{=} \vec{b}$. Die Matrix A (Dimension m x n) beinhaltet n Filterbedingungen:

$$\min_{\vec{w}} \left|\left|\vec{hd} - SD\vec{w}\right|\right|_2^2 \quad \text{mit NB} \quad A \cdot \vec{w} \stackrel{!}{=} \vec{b} \tag{3.17}$$

Im ersten Schritt werden die Nebenbedingungen umformuliert, so dass sie später in das Optimierungsproblem eingefügt werden können. Hierzu bietet sich eine QR-Zerlegung der transponierten Matrix A^T an.
Die QR-Zerlegung erzeugt zwei Matrizen, deren Produkt $Q \cdot R$ wiederum die Ursprungsmatrix A^T erzeugt. Die Matrix Q ist eine orthogonale Matrix der Dimension m x m, so dass $Q^T = Q^{-1}$ gilt. Die Matrix R hat die Dimension n x m, wobei der obere Teil aus einer oberen Dreiecksmatrix der Dimension n x n besteht und der untere Teil mit der Dimension n x $m-n$ Null ist (die QR-Zerlegung kann mit Hilfe von Givens-Rotationen oder Householder-Transformationen erfolgen). Hierzu bzw. auch zu weiteren Eigenschaften der

Matrizen Q und R sei auf entsprechende Literatur verwiesen, z. B. [LH74].
Mit Hilfe der QR-Zerlegung wird die transponierte Matrix A^T in Q und R aufgespalten.

$$A \cdot \vec{w} = \vec{b} \tag{3.18}$$
$$\left(A^T\right)^T \cdot \vec{w} = \vec{b} \tag{3.19}$$
$$(Q \cdot R)^T \cdot \vec{w} = \vec{b} \tag{3.20}$$
$$R^T \cdot Q^T \cdot \vec{w} = \vec{b} \tag{3.21}$$

Die Matrix Q wird nun benutzt um die Koeffizienten $\vec{w}$ zu $\vec{w}_{mod}$ zu transformieren.

$$Q^T \cdot \vec{w} = \vec{w}_{mod} \tag{3.22}$$

Somit ergibt sich:

$$R^T \cdot \vec{w}_{mod} = \vec{b} \tag{3.23}$$

Die Matrix R besteht aus einer oberen Dreiecksmatrix und einem Bereich, der Null ist. Die Dreiecksmatrix wird im Folgenden mit $R_{Dreieck}$ beschrieben. Sofern die Nebenbedingungen linear unabhängig voneinander sind, existiert die Dreiecksmatrix $R_{Dreieck}$, sowie deren Inverse.

$$R = \begin{pmatrix} R_{Dreieck} \\ 0 \end{pmatrix} \tag{3.24}$$

Da der zweite Teil der Matrix R aus Nullen besteht, werden durch die Gleichung 3.23 die unteren Elemente des Vektors $\vec{w}_{mod}$ nicht beschrieben.

$$\begin{aligned} R^T \cdot \vec{w}_{mod} &= \vec{b} \\ \begin{pmatrix} R_{Dreieck}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_{mod,1} \\ \vec{w}_{mod,2} \end{pmatrix} &= \vec{b} \end{aligned} \tag{3.25}$$

$$R_{Dreieck}^T \cdot \vec{w}_{mod,1} = \vec{b} \tag{3.26}$$

Somit lässt sich der obere Teil des Lösungsvektors berechnen:

$$\vec{w}_{mod,1} = R_{Dreieck}^{-T} \cdot \vec{b} \tag{3.27}$$

Im nächsten Schritt wird in der Formel 3.17 der Ausdruck $\vec{w}$ nach Formel 3.22 durch $Q \cdot \vec{w}_{mod}$ ersetzt.

$$\min_{\vec{w}_{mod}} \left|\left| SD \cdot Q \cdot \vec{w}_{mod} - \vec{hd} \right|\right|_2^2 \tag{3.28}$$

Es sei an dieser Stelle nochmals darauf hingewiesen, dass der Vektor $\vec{w}_{mod}$ aus zwei Teilen besteht (aus dem Teil, der durch die Lösung der Nebenbedingungen bestimmt ist, und dem Teil, der durch die Minimierung gelöst werden muss).

$$\vec{w}_{mod} = \begin{pmatrix} \vec{w}_{mod,1} \\ \vec{w}_{mod,2} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{w}_{mod,2} \end{pmatrix} + \begin{pmatrix} \vec{w}_{mod,1} \\ 0 \end{pmatrix} \tag{3.29}$$

Teilt man nun den Vektor $\vec{w}_{mod}$ auf, so lässt sich der Minimierungsterm ebenfalls aufteilen:

$$\min_{\vec{w}_{mod}} \left|\left| SD \cdot Q \cdot \begin{pmatrix} 0 \\ \vec{w}_{mod,2} \end{pmatrix} - \left(\vec{hd} - SD \cdot Q \cdot \begin{pmatrix} \vec{w}_{mod,1} \\ 0 \end{pmatrix} \right) \right|\right|_2^2 \tag{3.30}$$

$SD \cdot Q$ wird jetzt durch die Matrix SDQ dargestellt. Auch an dieser Stelle kann durch die Multiplikation mit 0 eine Dimensionsreduktion erfolgen. Im ersten Teil ist nur der rechte Bereich der Matrix SDQ von Relevanz (also die Spalten $n+1$ bis m) und im zweiten Teil der linke Term (also die Spalten 1 bis n). Da $w_{mod,1}$ durch die Nebenbedingungen bestimmt ist, erfolgt die Optimierung lediglich auf $w_{mod,2}$, so dass sich folgender Ausdruck ergibt

$$\min_{\vec{w}_{mod,2}} \left|\left| SDQ_2 \cdot \vec{w}_{mod,2} - \left(\vec{hd} - SDQ_1 \cdot \vec{w}_{mod,1} \right) \right|\right|_2^2 \tag{3.31}$$

Dies entspricht wieder der MQF-Minimierung für Optimierungen ohne Nebenbedingungen, so dass $\vec{w}_{mod,2}$ sich errechnet als:

$$\vec{w}_{mod,2} = \left(SDQ_2^T \cdot SDQ_2 \right)^{-1} \cdot SDQ_2^T \cdot \left(\vec{hd} - SDQ_1 \cdot \vec{w}_{mod,1} \right) \tag{3.32}$$

Die endgültigen Filterkoeffizienten ergeben sich über die Rekombination beider Filterkoeffiziententerme und Multiplikation mit Q nach den Formeln 3.22 und 3.29:

$$\vec{w} = Q \cdot \begin{pmatrix} \vec{w}_{mod,1} \\ \vec{w}_{mod,2} \end{pmatrix} \tag{3.33}$$

Die QR Zerlegung stellt nur eine Variante zur Lösung von MQF Minimierungen mit Nebenbedingungen dar, weitere Möglichkeiten finden sich z. B. bei [LH74].

3.2 Anwendungsbeispiel: Kondo's Interpolationsmethode

In diesem Abschnitt wird das Verfahren nach Kondo im Detail vorgestellt. Ein wesentlicher Punkt und Vorteil dieses Verfahrens ist die strikte Unterteilung in eine einfach zu realisierende adaptive Interpolation und eine aufwendigere Optimierung von Filterkoeffizienten, die im vorhinein z. B. im Labor durchgeführt werden kann. Die einfache Anwendung macht eine solche Struktur z.B. ideal für die Abbildung auf eine Hardware (siehe auch [ML08]). Bemerkenswert ist vor allem, dass trotz der einfachen Struktur die Qualität der Interpolation sehr hoch ist.
Auf Grund der Möglichkeit mit geringem Aufwand eine hohe Interpolationsqualität erreichen zu können wird diese Grundidee aufgegriffen und für die im Kapitel 4 eingeführten

örtlichen und zeitlichen Interpolationsverfahren verwendet.

Die Abbildungen 3.2 und 3.3 zeigen deutlich die wesentliche Trennung in einen komplizierten Trainingsprozess und eine einfache Anwendung. In Abbildung 3.2 ist die Interpolation um den Faktor 2, und in der Abbildung 3.3 die Methode zum Erstellen der Filterkoeffizienten dargestellt. Die Verbindung beider Teile ist ein Speicher in dem die optimierten Filterkoeffizienten abgelegt werden.

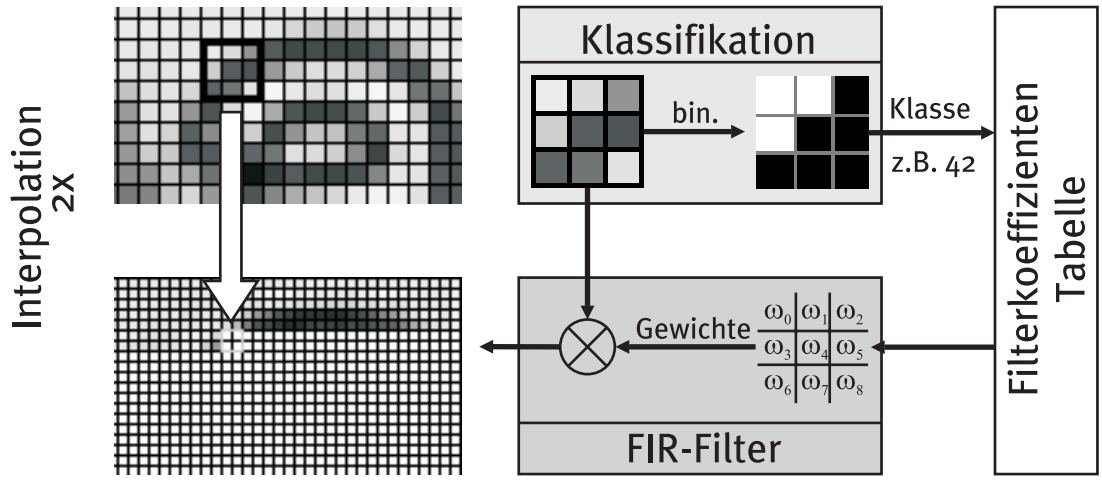


Abbildung 3.2: Interpolation nach Kondo's Methode [KNF+01]

Die Interpolation besteht aus wenigen und einfachen Schritten (siehe Abbildung 3.2). Ein Bereich eines niedrig aufgelösten Bildes (SDTV) wird zuerst klassifiziert, dann werden die Filterkoeffizienten für diese Klasse aus einer Tabelle gelesen, und anschließend wird eine lineare Filterung mit für diese Klasse optimalen Filterkoeffizienten durchgeführt.

Klassifikation

Die Wahl der Koeffizienten für die Gewichtung wird mit Hilfe einer Klassifizierung des Bildmaterials durchgeführt. Die Klassifizierung eines Blocks (*Block*) mit 3x3 Pixeln führt bei einer 8 Bit Quantisierung der Luminanz zu $256^9 = 4.722.366.482.869.645.213.696$ verschiedenen Klassen bzw. Bildsituationen. Um die Anzahl der möglichen Klassen reduzieren zu können, werden die Luminanzwerte (F_{SD}) nach dem ADRC Verfahren (Adaptive Dynamic Range Coding, [KK95], [KNF+01]) in ihrer Bittiefe reduziert. Hierbei wird zuerst

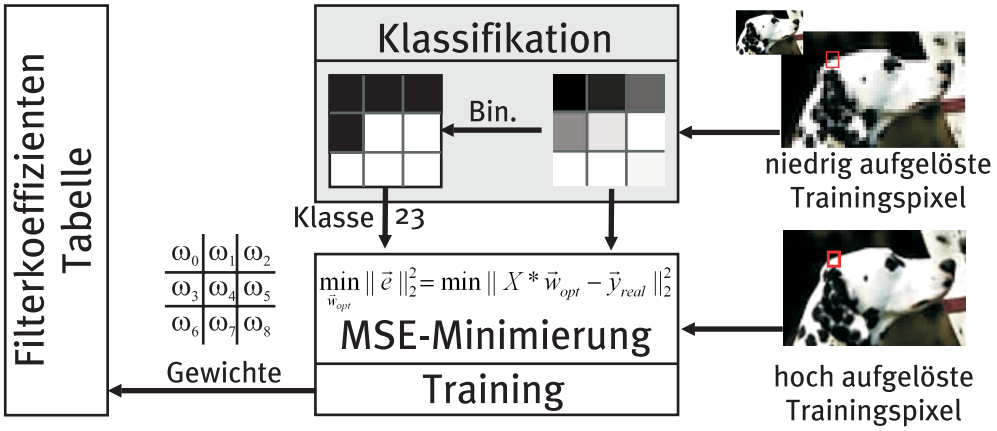


Abbildung 3.3: Training der Filterkoeffizienten nach Kondo's Methode [KNF+01]

die maximale (MAX) und minimale Helligkeit (MIN) in dem aktuellen Block (3x3 Pixel) bestimmt.

$$\begin{aligned} MAX(x,y) &= \max_{i,j \in Block} F_{SD}(x+i, y+j) \\ MIN(x,y) &= \min_{i,j \in Block} F_{SD}(x+i, y+j) \end{aligned} \tag{3.34}$$

Die Differenz dieser Werte ergibt den im aktuellen Block vorhandenen Luminanzbereich. Dieser wird mit DR (Dynamic Range) angegeben:

$$DR(x,y) = MAX(x,y) - MIN(x,y) \tag{3.35}$$

Auf der Basis dieses Bereiches wird nun die Quantisierungsschwelle in Abhängigkeit der gewünschten Bittiefe K festgelegt.

$$DR_K(x,y) = \frac{(MAX(x,y) - MIN(x,y))}{2^K} \tag{3.36}$$

Die im Block vorhandene Luminanz F_{SD} wird nun durch Subtraktion von MIN auf den Wertebereich $0 \rightarrow (MAX - MIN)$ verschoben, durch die Quantisierungsschwelle DR_K dividiert und abgerundet.

$$Q_{x,y}(i,j) = \left\lfloor \frac{(F_{SD}(x+i, y+j) - MIN(x,y))}{DR(x,y)/2^K} \right\rfloor \quad \forall i,j \in Block \tag{3.37}$$

Somit erhält man für jeden Pixel des aktuellen Blockes eine auf K-Bit quantisierte Information der vorliegenden Helligkeit. Bei $K = 1$ Bit und einem Block der Größe (3x3) ergeben sich somit $3 \cdot 3 = 9$ quantisierte Informationen mit jeweils 1 Bit und somit $2^9 = 512$ unterschiedliche Klassen.

Man trifft in der Literatur (u. a. [LZd03]) auch ähnliche Klassifikatoren an, wie z. B. eine 1-Bit Quantisierung (Binarisierung) mit Hilfe des lokalen mittleren Grauwerts $F_{Mittelwert}(x, y)$ des zu klassifizierenden Blockes.

$$F_{Mittelwert}(x,y) = \frac{1}{Anzahl\ der\ Pixel\ im\ Block} \cdot \sum_{i,j \in Block} F_{SD}(x+i, y+j) \tag{3.38}$$

Mit Hilfe dieses Wertes kann die Klassifikation wie bei der vorherigen Methode durch eine Division und Abrunden erfolgen:

$$Q_{x,y}(i,j) = \left\lfloor \frac{(F_{SD}(x+i, y+j))}{F_{Mittelwert}(x,y)} \right\rfloor \ \forall i,j \in Block \tag{3.39}$$

Alternativ lässt sich diese Klassifikation auch als Vergleich des aktuellen Pixels F_{SD} mit dem $F_{Mittelwert}$ beschreiben.

$$Q_{x,y}(i,j) \quad = \begin{cases} 0 & |F_{SD}(x+i, y+j) < F_{Mittelwert}(x,y) \\ 1 & |F_{SD}(x+i, y+j) \geq F_{Mittelwert}(x,y) \end{cases} \quad \forall i,j \in Block \tag{3.40}$$

Die hier beschriebenen Varianten der Klassifikation unterscheiden sich zwar bezüglich ihrer Klassifizierungsqualität bei einzelnen Störpeaks, aber auf Grund der laplace-förmigen Verteilung von lokalen Bildinhalten um die lokale mittlere Helligkeit ähneln sich die Ergebnisse beider Formeln sehr stark. Es sind Unterschiede in dem Aufwand bei einer Hardwarerealisierung denkbar, welche hier aber nicht weiter betrachtet werden.

Unabhängig von der Art der Binarisierung erhält man eine starke Reduktion der Anzahl an unterschiedlichen Klassen. Bei einer Größe von 3x3 wird eine Reduktion der Klassen um den Faktor 9.223.372.036.854.775.808 (4.722.366.482.869.645.213.696→512) erreicht.

In [KFC01] wird vorgeschlagen, einen Helligkeitsübergang von hell nach dunkel genauso zu behandeln wie von dunkel nach hell. Somit halbiert sich die Gesamtzahl der Klassen und reduziert sich auf $2^{9-1} = 256$.
Die Abbildung 3.2 zeigt ein Beispiel für eine solche Klassifikation. Der Ausschnitt (bzw. aktuelle Block) stellt eine diagonal verlaufene Kante dar. Die Information einer diagonal verlaufenden Kante ist auch nach der Binarisierung weiterhin erkennbar. Neben der reinen Richtungsbestimmung von Kanten liefert die Klasse auch Zusatzinformationen wie die Position der Kante (auf ganze Pixelpositionen beschränkt) und die Breite. Zusätzlich ermöglicht die Klasse auch die Beschreibung komplizierter Bildsituationen (Ecken, Abzweigungen, etc.).
Auf Grund der sehr einfachen und symmetrischen Pixelanordnung ist es sehr leicht möglich eine Reduktion der notwendigen Klassen vorzunehmen. Hierzu wird bei [ZLd02] die

Ausnutzung einer horizontalen und vertikalen Symmetrie angedeutet. Hierbei werden die Koeffizienten für die HD-Pixel b,c,d in „gespiegelten“ Klassen gesucht. Der Grundgedanke einer Symmetrieausnutzung wird auch bei den im folgenden Kapitel 4 eingeführten Verfahren aufgegriffen. Diese Reduktion erfolgt jedoch nicht in Form einer Reduktion der Phasen, sondern in Form einer Reduktion der Bildklassen.

Interpolation

Nach der Klassifikation eines Bildbereiches, erfolgt die zweifache Interpolation. Hierzu werden die vier HD-Pixel, die im Mittelpunkt der Maske liegen (siehe auch Abbildung 2.15, HD-Pixel a, b, c, d) nacheinander interpoliert. Die Interpolation eines einzelnen HD-Pixels F_{HD} (z. B. Position a und detektierte Klasse 42) ergibt sich aus einer einfachen FIR-Filterung der niedrigaufgelösten SD-Pixel F_{SD} mit den passenden Filterkoeffizienten $w_{Pixel\,a,Klasse\,42}(i,j)$:

$$F_{HD,a}(x,y) = \sum_{i,j \in Block} F_{SD}(x+i, y+j) \cdot w_{a,42}(i,j)) \tag{3.41}$$

Nach der Interpolation der vier HD-Pixel wird der Block im SD-Bild um einen Pixel verschoben und die nächste Klassifikation beginnt.
Auf diese Weise werden für jedes einzelne SD Pixel vier HD Pixel erzeugt. Dies entspricht der zweifachen horizontalen und zweifachen vertikalen Auflösungserhöhung.

Training der Filterkoeffizienten

Der Trainingsprozess zur Erzeugung der Filterkoeffizienten ähnelt dem Interpolationsprozess sehr stark. Im Gegensatz zu einer Interpolation, die mit vorhandenen Koeffizienten aus niedrig aufgelösten Bildern hochaufgelöste Bilder erzeugt, werden zum Training mehrere Bildsequenzen benötigt, die schon sowohl in niedriger als auch hoher Auflösung existieren. In der Abbildung 3.3 ist ein kleiner Bildausschnitt in beiden Auflösungen dargestellt. Für einen tatsächlichen Trainingsprozess sind natürlich längere und größere Trainingssequenzen notwendig.
Das gesamte niedrig aufgelöste Bild wird abgearbeitet und für jede Position wird ein Block aus benachbarten Eingangspixeln definiert und dessen Klasse bestimmt.

$$SD_{Klasse\,...} = \{F_{SD}(x+i, y+j)\} \,|i,j \in Block \tag{3.42}$$

Diese Schritte erfolgen in derselben Art und Weise wie bei der Interpolation.
Anschließend werden in einer Datenbank für die detektierte Klasse der niedrig aufgelöste Block des Eingangsbildes und die vier Pixel des hochaufgelösten Bildes abgespeichert.
Somit erhält man für jede Klasse eine Menge an Trainingsdaten, die jeweils die niedrigaufgelösten Soll-SD-Pixel und das HD-Soll-Pixel enthält.

Für mehrere Trainingsdaten einer Klasse ergeben die Soll-Ausgangswerte einen Vektor($\vec{hd}$ HD-Werte) und die Soll-Eingangswerte eine Matrix (SD, SD-Werte).
Es werden diejenigen Filterkoeffizienten gesucht, die aus den SD-Werten die HD-Werte möglichst genau erzeugen. Die Interpolation ist ein linearer Filterprozess und somit kann die Optimierung in die Form der Formel 3.8 gebracht werden, in der diejenigen Filterkoeffizienten gesucht werden, die - mathematisch gesehen - den geringsten quadratischen Fehler (MQF) erzeugen.
Nach Formel 3.14 ergeben sich die optimalen Filterkoeffizienten für eine bestimmte Klasse und für z. B. die HD Pixelposition a wie folgt:

$$\vec{w}_{Pixel\,a,Klasse\,42} = \left(SD_{42}^T \cdot SD_{42}\right)^{-1} \cdot SD_{42}^T \cdot \vec{hd}_{a,42} \tag{3.43}$$

Für die Erstellung der Koeffizientendatenbank muss nun für alle vier Bildpositionen (a, b, c, d) und für alle Klassen die Lösung nach obiger Formel erzeugt werden. Dies ist zwar mit erhöhtem Rechenaufwand verbunden, muss aber lediglich einmal z.B. im Labor erfolgen.

4 Klassifikationsbasierte lineare Interpolation

In diesem Kapitel werden verschiedene Interpolationsverfahren vorgestellt, die für ihre Anwendung lediglich einen geringen Aufwand benötigen, aber trotz ihres geringen Aufwands dennoch hochqualitative Interpolationsergebnisse liefern.

Der geringe Aufwand bei einer Interpolation wird ermöglicht durch eine Unterscheidung von verschiedenen Bildsituationen in Form eines einfachen Klassifikators und einer anschließenden einfachen linearen Filterung mit im Vorhinein optimierten Filterkoeffizienten.

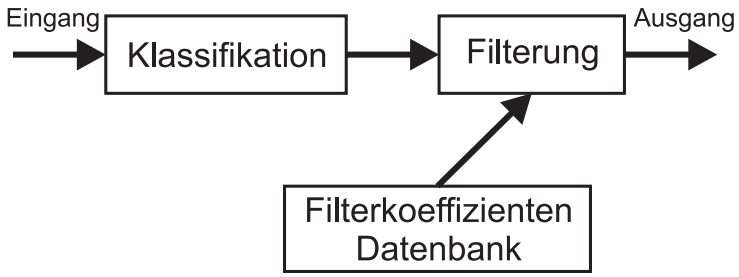


Abbildung 4.1: Schema der klassifikationsbasierten Signalverarbeitung

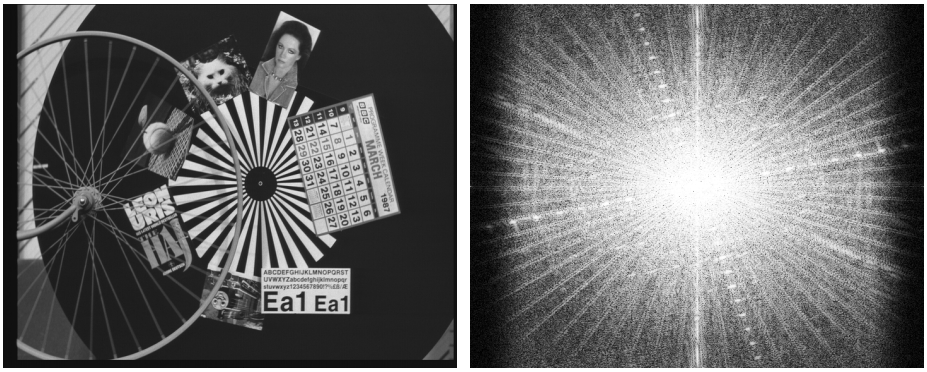
Bild 4.1 stellt diesen Prozess dar. Man kann deutlich die einfache Struktur erkennen. Da für die Anwendung derartiger Algorithmen lediglich klassifiziert und linear gefiltert werden muss, können verschiedenste Algorithmen mit sehr geringem Rechenaufwand realisiert werden. Die im vorherigen Kapitel 3.2 angegebene adaptive Interpolation nach Kondo [KNF+01]) basiert auch auf diesem Prinzip.
Ziel der Klassifikation ist es, eine Linearität der Eingangspixel zu den zu berechnenden Ausgangspixeln zu erreichen. Somit können optimale Filter zur Verfügung gestellt werden, so dass insgesamt eine weit über eine statische Filterung hinausgehende Qualität erreicht werden kann.
Die hohe Qualität der Verfahren basiert zum großen Teil auf der adaptiven örtlichen Ausrichtung von Filtern und der damit verbundenen vollständigen Ausnutzung von Frequenzen bis zur Grenze des Grundspektrums. Der folgende Abschnitt analysiert dies unter dem

Aspekt der Superresolution und zeigt, dass mit diesem Ansatz eine noch höhere Interpolationsqualität erreicht werden kann. Selbst die Erzeugung höherer spektraler Signalanteile, als sie im Grundspektrum vorhanden sind, ist durch die Ausnutzung von Aliasanteilen möglich.
In den darauf folgenden Abschnitten werden zwei solcher Verfahren im Detail vorgestellt, sowohl eine örtliche Interpolation zur SD→HD-Konversion als auch eine zeitliche Zwischenbildinterpolation (siehe Abschnitt 4.4).

4.1 Anwendbarkeit und allgemeiner Gewinn

4.1.1 Spektrale Analyse

In diesem Abschnitt wird die Ursache dieser hohen Qualität sowie die grundsätzliche Anwendbarkeit der Methode auf unterschiedliche Bereiche der Bildsignalverarbeitung näher untersucht.
Das Spektrum einer natürlichen Sequenz konzentriert sich bekanntermaßen hauptsächlich auf den niederfrequenten Bereich. Dennoch haben vor allem die höheren Frequenzen einen großen Einfluss auf die subjektive Schärfe eines Bildes. Für natürliche Bilder kann man die vorliegenden Frequenzen sehr gut mit einer laplace-förmigen Verteilung beschreiben (siehe auch [RG83]). Deswegen besteht das Spektrum aus sehr vielen verschiedenen Frequenzen. Abbildung 4.2 zeigt dies beispielhaft an dem Spektrum der Sequenz Wheel. Man erkennt hier einerseits die starke Konzentration auf den niederfrequenten Bereich, aber auch die Vielzahl an unterschiedlichen höherfrequenten Spektralanteilen.

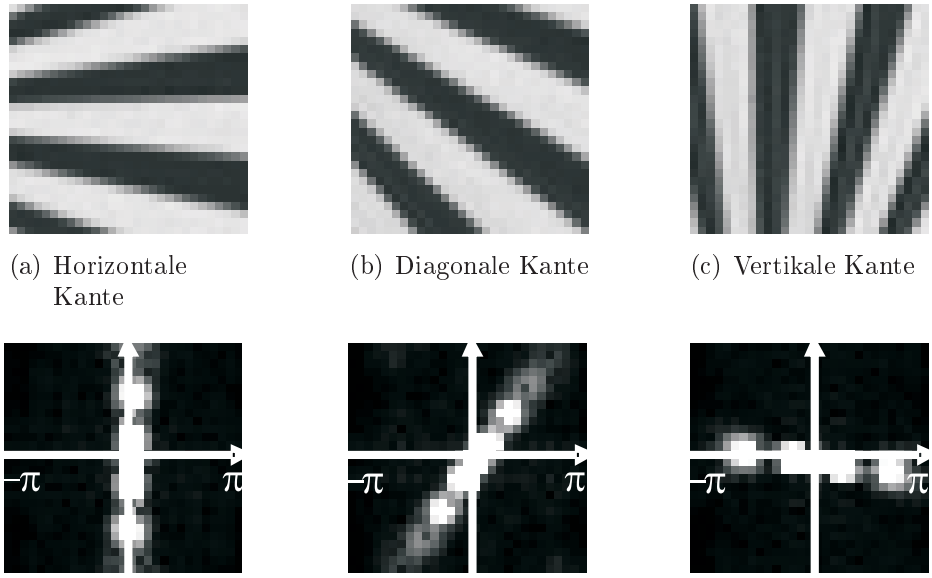


(a) Ortsbereich

(b) Spektrum (kontrastverstärkt und logarithmische Darstellung)

Abbildung 4.2: Spektrum und Originalbild der Sequenz Wheel

Im Gegensatz zu der Verteilung der Frequenzen eines Gesamtbildes wird bei einer lokalen Betrachtung (kleine Bildausschnitte) eine deutliche Reduktion der vorhandenen Frequenzen sichtbar. In Abbildung 4.3 sind mehrere lokale Ausschnitte der Sequenz WHEEL und



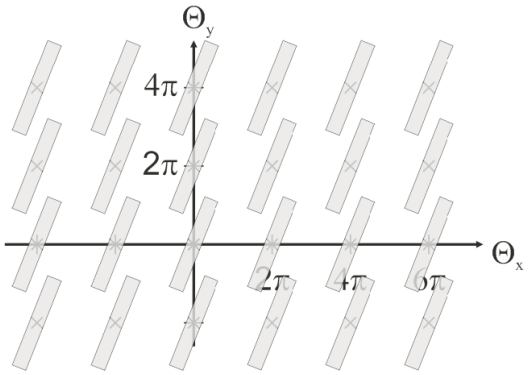
(a) Horizontale Kante (b) Diagonale Kante (c) Vertikale Kante

(d) Spektrum einer horizontalen Kante (e) Spektrum einer diagonalen Kante (f) Spektrum einer vertikalen Kante

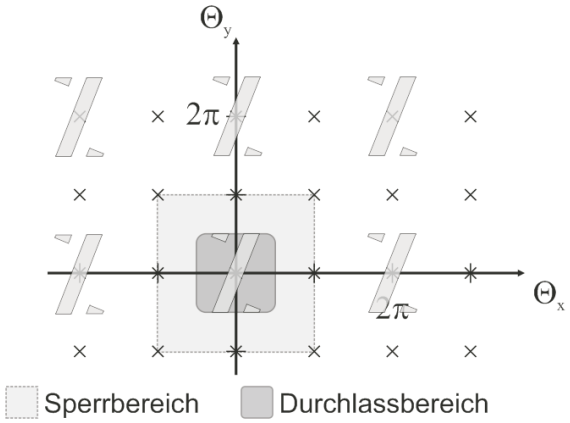
Abbildung 4.3: Bildausschnitte aus der Sequenz WHEEL und deren Spektren

deren zugehörigen Spektren dargestellt. Abhängig von der Richtung der Kanten im Bild besitzen auch die Frequenzen eine deutliche Ausrichtung. Im Bild vorhandene Kanten bewirken eine zum Kantenverlauf orthogonal ausgerichtete Fourierreihe. Da das menschliche visuelle System stark auf Kanten ausgerichtet ist, befinden sich somit in diesen Frequenzbereichen die wichtigsten Informationen.
Für eine qualitativ hochwertige Bildsignalverarbeitung müssen also diese Bereiche möglichst erhalten werden. Bei einer Interpolation müssen bekannterweise nach einer Abtastratenerhöhung die ausgerichteten Basis- und Wiederholspektren weitestgehend erhalten und die störenden Zwischenspektren unterdrückt werden. Hierbei macht es Sinn die Ausrichtung der Filter an die vorliegenden Frequenzen anzupassen. Abbildung 4.4 zeigt den Unterschied zwischen statischen und adaptiven Filtern für die Anwendung einer Interpolation um den Faktor 2. Bei einem statischen Filter (Abbildung 4.4(b)) werden hohe Frequenzen abgeschnitten bzw. stark gedämpft. Des Weiteren werden Frequenzen von Zwischenspektren (Alias) nicht ausreichend unterdrückt und bleiben somit als störende Aliasanteile im Grundspektrum erhalten (Abbildung 4.4(b)).

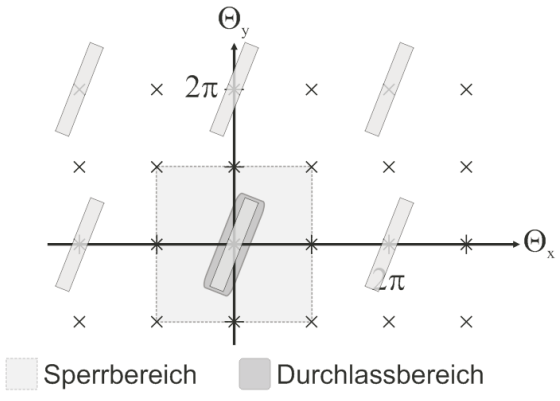
In der Abbildung 4.4 überlappen sich die Frequenzen des Grund- und der Wiederholspektren (als Rechteck dargestellt). Somit sind im Grundspektrum und den Wiederholspektren Aliasanteile vorhanden. Mit einem adaptiv ausgerichteten Filter ist man nun in der Lage das vorliegende Grundspektrum zu bewahren und aus den angrenzenden Aliasanteilen der Nachbarspektren, Frequenzen, die über das Grundspektrum hinausgehen, zu rekonstruieren. Zusätzlich werden Aliasanteile, die im Grundspektrum vorliegen und nicht gewünscht sind, unterdrückt (siehe hierzu Abbildung 4.4(c)).



(a) Spektrum eines Bildausschnittes



(b) Ergebnis eines statischen Interpolationsfilters

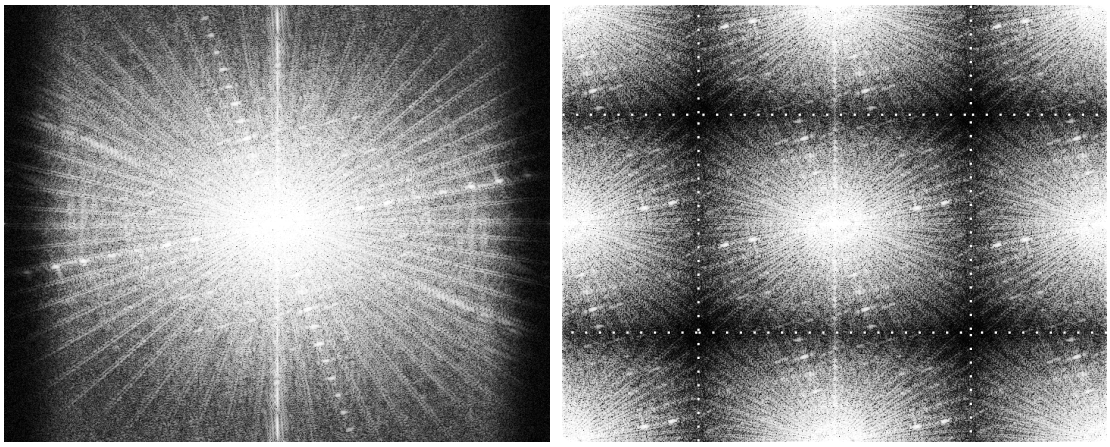


(c) Ergebnis eines adaptiven Interpolationsfilters

Abbildung 4.4: Schema einer adaptiven und einer statischen Interpolation (Faktor 2)

Die Abbildung 4.5 zeigt beispielhaft die Leistungsfähigkeit dieses Verfahrens. Es sind die Spektren einer adaptiven Interpolation im Vergleich zu einer statischen Interpolation dargestellt. Bei einer statischen Interpolation werden die Wiederholspektren weitestgehend unterdrückt. Bei der adaptiven Verarbeitung hingegen erkennt man sehr deutlich, dass

das Spektrum des Eingangsbildes (als weißer Kasten sichtbar) durch im Bild vorhandene Aliasanteile erweitert werden kann.



(a) Originalspektrum eines Bildes der Sequenz Wheel

(b) Zweifache Abtastratenerhöhung einer vorher um den Faktor 2 dezimierten Version von Wheel

(c) Statische Filterung von 4.5(b)

(d) Adaptive Filterung von 4.5(b)

Abbildung 4.5: Originalspektrum und Wiederholspektren einer zweifachen Interpolation der Sequenz Wheel

Wenn mit Hilfe der Klassifikation Bildausschnitte unterschieden werden können, so dass es spektrale Vorzugsrichtungen für diese Klassen gibt, dann können auch lineare Filter gefunden werden, die die Spektren erhalten. Die im Kapitel 2.3.1 dargestellte Binarisierung auf der Basis des mittleren Grauwertes erreicht für die Anwendung einer Interpolation genau diese notwendige Unterscheidung der unterschiedlichen spektralen Ausrichtungen.

Die Ausrichtung von Filtern ist unter anderem auch in der zeitlich-räumlichen Formatkonversion sehr verbreitet [SB00]. Selbst einfache adaptive De-Interlacingverfahren besitzen eine Unterscheidung (und somit eine Klassifikation) zweier Filterrichtungen. Zum einen gibt es eine örtliche Filterung (F_y-Achse) und zum anderen ein temporales Filter (F_t-Achse). Je nach Bewegungssituation werden die wichtigsten spektralen Anteile somit erhalten und die störenden Anteile unterdrückt.
Somit wird eine Interpolation bzw. Filterung durch eine Erweiterung auf mehrere Dimensionen nicht schwieriger, sondern eröffnet vielmehr Möglichkeiten, Frequenzen unterschiedlich zu behandeln, was wiederum bei einer niedrigeren Dimensionsstufe nicht möglich ist.
Eine Interpretation im Ortsbereich führt zu der Erkenntnis, dass man durch die Klassifikation einen lokalen linearen Zusammenhang zwischen den Eingangs- und Ausgangspixeln erreicht, so dass eine hochwertige Interpolation durch eine einfache lokale FIR-Filterung möglich ist.
Gerade dieser Punkt ist vor allem für die Erzeugung der Filter sehr wichtig, da nur bei einem sehr stark ausgeprägten linearem Zusammenhang Filter hoher Qualität angelernt bzw. definiert werden können.
Bei fehlender Klassifikation könnte eine Filteroptimierung nur ein statisches Filterergebnis erzielen. Diese wäre lediglich eine inverse Abbildung des Globalfilters, welches den Unterschied zwischen den niedrig aufgelösten und hochaufgelösten Trainingssequenzen beschreibt. Erst durch die Klassifikation können Filter entstehen, die über die Qualität eines inversen Filters hinausgehen.
Im Rahmen dieser Arbeit werden die Vorzüge der adaptiven Filterung hauptsächlich am Beispiel der Interpolation dargestellt. Dieses Verfahren ist aber natürlich auch für andere Bereiche der Bildsignalverarbeitung anwendbar. So kann eine spektrale Vorzugsrichtung auch im Zeitbereich (z. B. Zwischenbildinterpolation, siehe Abschnitt 4.4) liegen, oder das Filter kann auch ausschließlich zur Unterdrückung störender Spektralanteile dienen (z. B. zur Rausch- und Blockingreduktion).

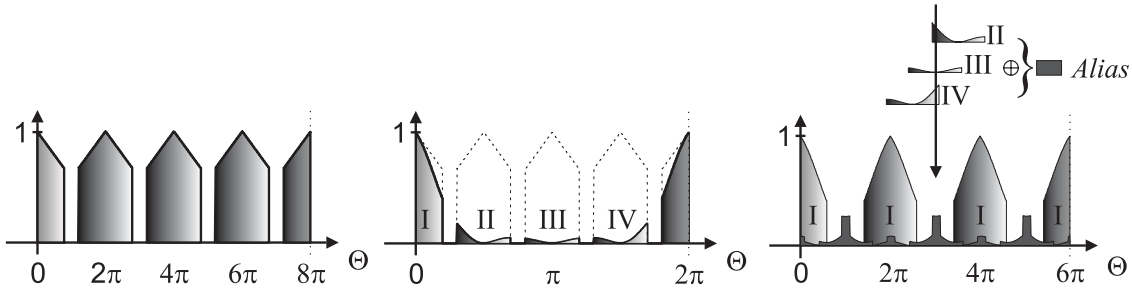
4.1.2 Motivation einer Polyphasenstruktur

Eine Interpolation von niedrig aufgelöstem Bildmaterial (wie z. B. SDTV, PAL Signale) für hochauflösende Displays (HDTV) erfordert auf Grund der Fülle an unterschiedlichen Eingangs- und Ausgangsauflösungen viele verschiedene nicht ganzzahlige Interpolationsfaktoren. Die im Kapitel 2.3.1 vorgestellten gebräuchlichen fortgeschrittenen Verfahren besitzen jedoch nur die Möglichkeit ganzzahlige Vergrößerungen zu realisieren.
Es gibt Ansätze diese bildinhaltsadaptiven Verfahren mit statischen einfachen Verfahren (z. B. bilinearer Interpolation wie bei [Li00]) zu kombinieren, um nicht ganzzahlige Faktoren zu ermöglichen. Diese Kombination verschlechtert jedoch die Gesamtqualität, wie im

Folgenden kurz dargestellt wird.
Für den angenommenen Fall einer SDTV->HDTV Interpolation ergibt sich z. B. der Faktor $\frac{8}{3}$. Um diesen zu realisieren wird in vielen Fällen die Interpolation in zwei Schritte unterteilt: Eine adaptive Interpolation mit dem Faktor 2 und eine bilineare Interpolation mit dem Faktor $\frac{4}{3} \approx 1,3333$.
Für diese Interpolation muss man von einer Überabtastung um den Faktor 4 und einer anschließenden Dezimation um den Faktor 3 ausgehen. Die Überabtastung um den Faktor L (hier 4) bedeutet, dass das Eingangsspektrum $S(\Theta)$ (wobei Θ die normierte Kreisfrequenz ist) um den Wert L skaliert wird ($S(L \cdot \Theta)$). Nun wird zur Unterdrückung der Wiederholspektren die bilineare Interpolation angewendet. Abschließend wird eine Reduktion der Abtastpunkte um den Faktor M (hier 3) durchgeführt. Das Filter der bilinearen Interpolation hat in den Wiederholspektren jeweils seinen Nulldurchgang, so dass es grundsätzlich die DC-Anteile der Wiederholspektren komplett unterdrückt. Die restlichen Frequenzen der Wiederholspektren werden jedoch nur unzureichend unterdrückt. Außerdem wird das Hauptspektrum ebenfalls gedämpft.
Abbildung 4.6 zeigt die Auswirkung einer bilinearen Interpolation um den Faktor $\frac{4}{3}$ auf das Spektrum. Durch die Abtastratenkonversion erfolgt eine Stauchung der Frequenzachse, so dass bei einer vierfachen Interpolation drei Zwischenspektren (I, II, III) entstehen. Diese werden durch die lineare Interpolation nur unzureichend unterdrückt. In Abbildung 4.6(b) kann man auch die Dämpfung der hochfrequenten Anteile des Originalspektrums (I) sehen. Durch die Dezimation um den Faktor 3 überlagern sich die Anteile aller vier Spektren, so dass starker Alias entsteht. Um den Versatz der Frequenzen deutlich zu machen, sind den einzelnen Spektralanteilen in Abbildung 4.6 verschiedene Grauwerte zugeordnet. Durch den ungeraden Interpolationsfaktor überlagern sich unterschiedliche Frequenzen an einem Punkt.
Da das Eingangsspektrum durch eine vorherige Interpolation bedingt den Bereich $[0 : \frac{\pi}{2}]$ nicht vollständig ausfüllt, wird der hier sehr stark sichtbare negative Effekt der nachgeschalteten bilinearen Interpolation etwas abgeschwächt.

4.2 Adaptive Polyphaseninterpolation

Auf Grund der negativen Eigenschaften eines nichtadaptiven Teils innerhalb einer Interpolation wird im Folgenden ein adaptives Verfahren zur Interpolation mit rationalen Faktoren vorgestellt.
Grundlage der Überlegungen ist das adaptive Interpolationsverfahren nach Kondo [KNF+01], welches ausführlich in Kapitel 3.2 vorgestellt wurde.
Das Verfahren beruht auf einer einfachen Klassifikation des Bildinhaltes und einer davon abhängigen linearen Filterung, welche klassenabhängige Filterkoeffizienten aus einer vortrainierten Datenbank verwendet. Unter Berücksichtigung der örtlichen Lage zwischen



(a) Original Spektrum

(b) Spektrum nach der Abtastratenerhöhung (Faktor 4) und der bilinearen Filterung

(c) Spektrum und Alias nach der Dezimation (Faktor 3)

Abbildung 4.6: Auswirkung einer ganzrationalen Interpolation um den Faktor $\frac{4}{3}$

den niedrig aufgelösten und den hochaufgelösten Bildpunkten (siehe Kapitel 2.1.2), wird im Folgenden das inhaltsadaptive Interpolationsverfahren entsprechend erweitert.

4.2.1 Anwendung

Die Abbildung 4.7 zeigt die Struktur einer adaptiven Polyphaseninterpolation. In Abhängigkeit der vorliegenden Eingangs- und Ausgangsauflösungen wird der Interpolationsfaktor bestimmt. Darauf basierend wird das gesamte SD-Bild abgearbeitet, und es werden je nach aktueller Position im SD-Bild die passenden HD-Pixel berechnet.
Hierzu wird in einem ersten Schritt für die aktuelle SD-Position der lokale Bildinhalt im Eingangsbild klassifiziert. Mit Hilfe der Klassifikation werden aus einer Tabelle die Filterwerte für diese Klasse ausgelesen. Da immer nur die HD-Pixel, die sich im Zentrum der

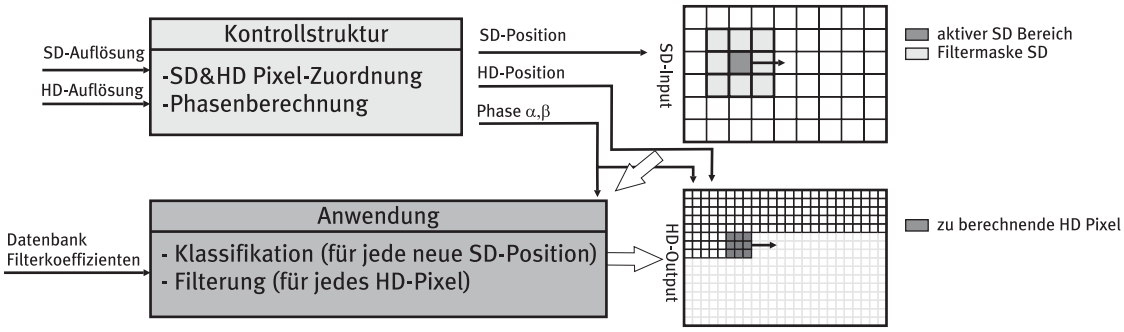


Abbildung 4.7: Schema der adaptiven Polyphaseninterpolation (Anwendung)

SD-Maske befinden, berechnet werden, werden für diese die jeweilige Position und damit ihre Phase bestimmt. Hierzu wird die Start- und die Endposition der zu berechnenden HD-Pixel bestimmt (siehe Formeln 2.11 und 2.12) und in Form einer linearen Filterung der Eingangspixel der SD-Maske berechnet.
Eine solche organisatorische Struktur (siehe Abbildung 4.8), die für jede Eingangspo-

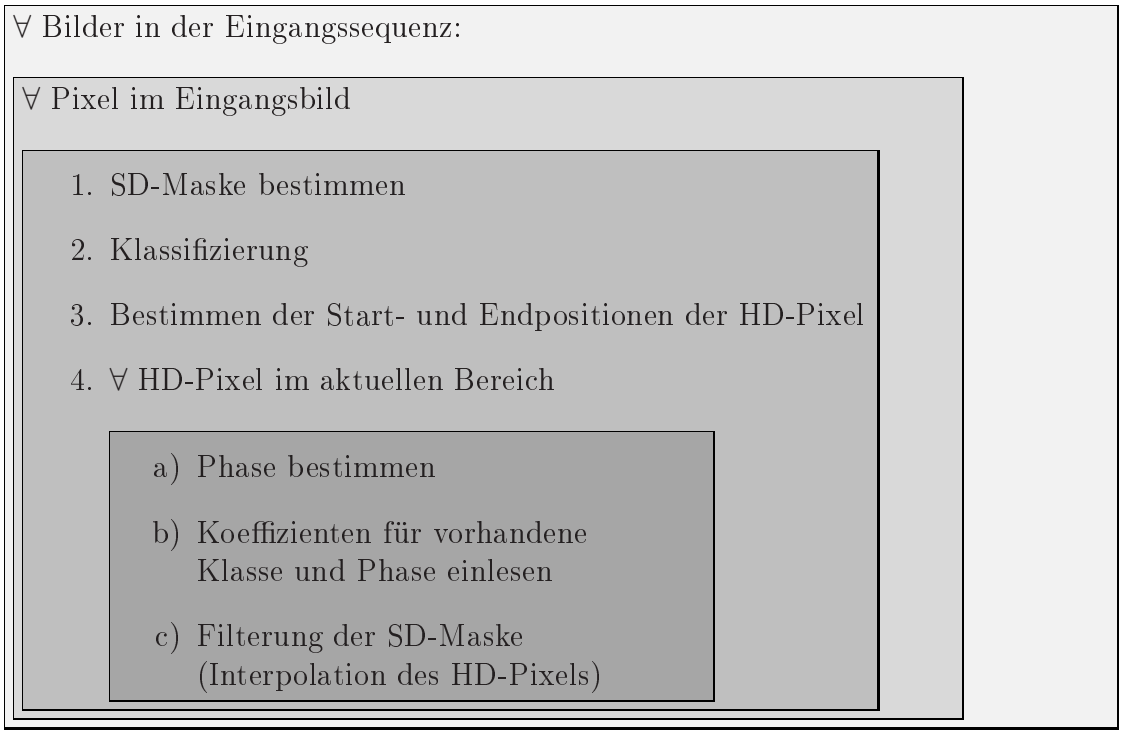


Abbildung 4.8: Programmablauf der Polyphaseninterpolation

sition die Ausgangspositionen bestimmt, hat den Vorteil, dass lediglich einmal auf die Eingangspixel zugegriffen werden muss. Zusätzlich muss diese Situation auch nur einmal klassifiziert und anschließend mehrere Ausgangspixel auf dieser Basis berechnet werden. Für die Berechnung einer Interpolation mit dem Faktor $\frac{8}{3}$ werden je nach Maskenposition zwischen vier und neun Ausgangspixel berechnet (zwei bis drei Ausgangsphasen je x- und y-Richtung).

4.2.2 Training

Im vorherigen Abschnitt wurde davon ausgegangen, dass die optimierten Filterkoeffizienten bereits vorliegen. Um Filterkoeffizienten für eine derartige Polyphaseninterpolation

1. ∀ Klassen, ∀ Phasen: Erstellen der leeren Matrix $SDTSD_{Klasse,Phase}$
2. ∀ Klassen, ∀ Phasen: Erstellen des leeren Vektors $SD\vec{T}hd_{Klasse,Phase}$
3. ∀ Eingangsbilder
   a) ∀ Pixel im Eingangsbild
      i. SD-Maske bestimmen
      ii. Klassifizierung
      iii. Bestimmen der Start- und Endpositionen der HD-Pixel
      iv. ∀ HD-Pixel im aktuellen Bereich
         A. Phase bestimmen
         B. SD-Pixel auf Matrix $SDTSD_{Klasse,Phase}$ akkumulieren
         C. HD-Pixel auf Vektor $SD\vec{T}hd_{Klasse,Phase}$ akkumulieren
4. ∀ Klassen, ∀ Phasen: Erstellen der Koeffiziententabelle $W_{Klasse,Phase}$
5. ∀ Klassen, ∀ Phasen: Lösen der aktuellen Gleichungsmatrix $W_{Klasse,Phase} = (SD^T SD_{Klasse,Phase})^{-1} \cdot SD^T \vec{hd}_{Klasse,Phase}$

Abbildung 4.9: Programmablauf des Trainings der Filterkoeffizienten

erstellen zu können, sind auch am Trainingsprozess (siehe Kapitel 3.2) mehrere Modifikationen notwendig. Dies betrifft die unterschiedliche Behandlung der Pixelpositionen und -phasen im SD- und HD-Bild, welche jedoch auf dieselbe Art und Weise wie bei der Interpolation berücksichtigt werden können. Die Ähnlichkeit zwischen der Anwendung und dem Training wird vor allem bei Betrachtung der Abbildung 4.9 deutlich. Hier wird der Programmablauf des Trainingsprozesses dargestellt. Im Unterschied zur Anwendung fällt auf, dass zu Beginn eine Initialisierung einiger Matrizen und Vektoren erfolgt. Diese werden während des Trainings gefüllt und nach der Abarbeitung der gesamten Sequenz zur Lösung der Filterkoeffizienten verwendet. Im Gegensatz zur eigentlichen Interpolation werden beim Training passende Paare von hochaufgelösten und niedrigaufgelösten Pixeln gebildet und auf diese Matrizen akkumuliert. Der Sinn dieser Akkumulation wird nun in diesem Abschnitt genauer vorgestellt.
Bei einer Interpolation, z. B. um den Faktor $\frac{8}{3}$, ergeben sich nach Formel 2.6 acht verschiedene Phasen, die jedoch auf Grund der Modulo-Operation sehr unterschiedlich ausfallen. Zusätzlich werden je nach Position im niedrigaufgelösten Bild unterschiedlich viele Pixel im HD-Bild interpoliert. Aus diesem Grund ist eine genaue Beachtung der zu realisierenden Phasen und vor allem eine hieran angepasste Speicherverwaltung der Trainingsdaten notwendig, die sich zudem dynamisch an die Anzahl der gerade abzuspeichernden Daten anpassen muss.

Auf Grund der hohen Anzahl an zu realisierenden Phasen ergibt sich auch ein hoher Trainingsaufwand.
Zum Vergleich: Sind bei einer 2D Interpolation um den Faktor 2 lediglich vier, bei dem Faktor 3 lediglich neun Phasen notwendig, so benötigt man für den dazwischen liegenden Faktor $\frac{8}{3} \approx 2.6666$ schon 64 verschiedene Phasen.
Bezeichnet man die einzelnen Trainingspixel des Eingangs mit SD und die des Ausgangs mit $\vec{hd}$, so ergibt sich nach Formel 3.8 die Minimierung des mittleren quadratischen Fehlers für eine Phase wie folgt:

$$\min \left\| \begin{bmatrix} SD_{11} & SD_{12} & \dots & SD_{19} \\ \dots & & & \dots \\ SD_{n1} & SD_{n2} & \dots & SD_{n9} \end{bmatrix} \cdot \begin{pmatrix} w_{1,\alpha} \\ w_{2,\alpha} \\ \dots \\ w_{9,\alpha} \end{pmatrix} - \begin{pmatrix} hd_{1,\alpha} \\ \dots \\ hd_{n,\alpha} \end{pmatrix} \right\|_2^2 \tag{4.1}$$

Die Matrix SD muss alle niedrigaufgelösten Trainingspixel einer Klasse enthalten und hat somit die Größe $NuTs \cdot SzMsk$, wobei $NuTs$ die Anzahl der Trainingsdaten (**Nu**mber of **T**raining **s**amples) und $SzMsk$ die Anzahl der Eingangspixel (Größe der Maske, **Siz**e of **Msk**) beschreibt. Die Größe des Trainingsvektors $\vec{hd}$ für die hochaufgelösten Trainingspixel ergibt sich ähnlich zu $NuTs \cdot 1$, wobei jedoch für jede Phase ein eigener Vektor $\vec{hd}$ erstellt werden muss. Da sowohl die Matrix SD als auch der Vektor $\vec{hd}$ für die Berechnung

der Filterkoeffizienten (siehe Formel 3.14) durch die Multiplikation mit der Transponierten von SD ($SD^T \cdot SD$ bzw. $SD^T \cdot \vec{hd}$) lediglich nur noch $SzMsk \cdot SzMsk$ bzw. $SzMsk \cdot 1$ groß sind, werden die Trainingsdaten direkt in der transponierten Weise abgespeichert. Die Informationen in SD sind zeilenweise gespeichert und somit ist eine Akkumulation der Trainingsdaten möglich. Wenn z. B. die Matrix SD aus den beiden Zeilen SD_1 und SD_2 besteht, kann die Operation $SD^T \cdot SD$ sequentiell abgearbeitet werden:

$$SD^T \cdot SD = SD_1^T \cdot SD_1 + SD_2^T \cdot SD_2 \tag{4.2}$$

Definiert man die Speichergröße für einen Trainingswert der Matrix SD als $SzTv$, die Anzahl der vorhandenen Klassen als $NuCl$(Number of Classes) und die Anzahl der Phasen mit $NuPh$(Number of Phases), so kann man den Speicheraufwand mit folgender Formel abschätzen.

$$Speicher_{Training} = NuCl \cdot SzMsk \cdot (SzMsk + 1) \cdot SzTv \cdot NuPh \tag{4.3}$$

Für die Akkumulation wird ein großer Datentyp (double, 64 Bit) benutzt, da ansonsten durch Rundungs- und Begrenzungsfehler die Konditionierung der Matrix stark abnehmen kann [Göt06]. Somit ergeben sich für den Fall der SD→HD-Interpolation bei 64 Phasen, 64 Bit (8 Byte) für den Trainingswert und einer Maskengröße von 3x3 und 256 Klassen $256 \cdot 8\,Byte \cdot 64 \cdot (9 \cdot (9 + 1)) = 11,25\,MB$. Natürlich ist dies nur eine grobe Abschätzung, da durch Verwaltungsmehraufwand der tatsächliche Speicherbedarf etwas größer ausfällt. Der Speicherbedarf der Filterkoeffiziententabelle ergibt sich (wenn die Speichergröße für einen Filterkoeffizient mit $SzFv$ festgelegt wird) wie folgt:

$$Speicher_{Koeffizienten} = NuCl \cdot SzMsk \cdot SzFv \cdot NuPh \tag{4.4}$$

Geht man von 32 Bit für die Speicherkoeffizienten aus, so ergibt sich (bei 256 Klassen, 64 Phasen, 32 Bit Filterkoeffizienten, 9 Pixel) ein Speicheraufwand von knapp $576KB$.

In Abbildung 4.10 ist beispielhaft das Ergebnis der Filterkoeffizienten einer Klasse dargestellt (Interpolationsfaktor $\frac{8}{3}$). Da für jeden der 3x3 Eingangspixel in Abhängigkeit der Phase unterschiedliche Filterkoeffizienten vorhanden sind, werden die Filterkoeffizienten für jedes Pixel phasenrichtig nebeneinander gesetzt. Aus diesem Grund ergeben sich $(3 \cdot 8) \cdot (3 \cdot 8)$ Filterkoeffizienten.

Man erkennt, dass die einzelnen Koeffizienten grob einer kontinuierlichen Form folgen. Somit entsprechen die Filterwerte Stützstellen eines unbekannten kontinuierlichen Tiefpasses, der für die vorliegende Bildsituation eine optimale Interpolation ermöglicht.

Man erkennt zusätzlich leichte Abweichungen der Filterwerte von dieser Form. Diese deuten auf ein grundsätzliches Problem des vorgestellen Algorithmus hin:

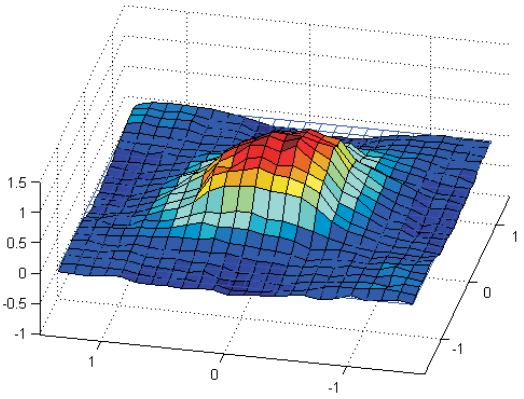


Abbildung 4.10: Impulsantwort für den Faktor $\frac{8}{3}$

Die Filterkoeffizienten werden für jede Phase einzeln angelernt, ohne dass die Ergebnisse der Nachbarkoeffizienten berücksichtigt werden. Beide Punkte (die kontinuierliche Form und die Probleme beim Training) werden im Abschnitt 4.3 wieder aufgegriffen und dienen als Grundlage für einen Trainingsprozess für kontinuierliche Interpolationsformen.

Trainingssequenzen

Um die Filterkoeffizienten richtig anlernen zu können, muss sichergestellt sein, dass die niedrig und hochaufgelösten Trainingssequenzen im passenden phasenrichtigen Größenverhältnis zueinander stehen, da ansonsten bei einem Trainingsprozess eine fehlerhafte Zuordnung der SD- und HD-Pixel sowie der Phasen entsteht, die dazu führt, dass im Datensatz eine Verschiebung integriert wird. Dies bedeutet weiterhin, dass sich die Gesamtqualität des Datensatzes verschlechtert, da der Schwerpunkt der Interpolation in diesem Fall nicht im Zentrum der SD-Maske liegt.
Aus diesem Grund werden für den Trainingsprozess die Sequenzen mit einem bikubischen Polyphaseninterpolationsalgorithmus verkleinert, der dieselbe Phasen- und SD/HD-Pixelzuordnung besitzt, wie auch der anzulernende adaptive Interpolationsalgorithmus.

4.2.3 Optimierung des Trainingsprozesses

Auf Grund der einfachen Struktur in der Anwendung liegt der Gewinn einer adaptiven Filterung vor allem in qualitativ hochwertigen Filterkoeffizienten.
Im Rahmen dieses Abschnittes werden verschiedene Möglichkeiten aufgezeigt die Qualität der Filterdatenbank zu verbessern. Dies umfasst die Untersuchung einer sinnvollen Anzahl

von Trainingssequenzen und deren Qualität, wie auch den Einfluss verschiedener Maskengrößen, Ausnutzung von Symmetrien sowie die Nutzung von Metadaten zum vorliegenden Bildmaterial.

Anzahl der Trainingsdaten

Die Auswahl der Trainingssequenzen hat einen sehr großen Einfluss auf die resultierende Qualität der Filterkoeffizienten. Da die SD-Sequenzen meistens einen geringeren Informationsgehalt als die HD-Sequenzen besitzen, ist selbst bei einer perfekten Interpolation von einem Unterschied zwischen dem interpolierten SD-und dem HD-Bild auszugehen.
Dieser Fehler ist für jeden Trainingswert unterschiedlich. Wenn zu wenig Trainingsdaten berücksichtigt werden, besteht die Gefahr, dass nicht die optimale Lösung für diese Klasse angelernt wird, sondern lediglich der Fehler der Trainingsdaten minimiert wird. Im Extremfall sind so wenig Trainingsdaten wie freie Filterkoeffizienten vorhanden, so dass zwar eine vollständige Unterdrückung des Fehlers möglich ist, aber diese Lösung nur für die vorhandenen Trainingsdaten funktioniert. Unter Umständen führt eine leicht veränderte Eingangssituation zu völlig falschen Werten. Dies ergibt bei einer Interpolation störende Artefakte z. B. helle Peaks in dunklen Bildbereichen.
Aus diesem Grund ist es wichtig, den Umfang des Trainingsmaterials so zu wählen, dass alle Filterkoeffizienten fehlerfrei angelernt werden können.
Durch die Auflösung ($SzHDx$ x $SzHDy$, Size of HD picture for x/y) und Bildanzahl ($NuFr$, Number of Frames) der hochaufgelösten Trainingssequenz wird die Anzahl aller Trainingswerte bestimmt.

$$\sum_{i=Phasen,j=Klassen} NuTs_{i,j} = (SzHDx \cdot SzHDy \cdot NuFr) \tag{4.5}$$

Die Gesamtzahl der Trainingswerte teilt sich in unterschiedliche Phasen und Klassen auf. Bezogen auf die Gesamtzahl aller Klassen und aller Phasen ergibt sich als durchschnittlicher Wert pro Phase und Klasse:

$$\varnothing NuTs = \frac{\sum_{i=Phasen,j=Klassen} NuTs_{i,j}}{NuCl \cdot NuPh} \tag{4.6}$$

Bis auf vernachlässigbare Abweichungen durch Randbereiche ist die Verteilung der Trainingswerte über die einzelnen Phasen konstant. Da aber die Häufigkeit einer bestimmten Klasse sehr stark vom Inhalt des Trainingsmaterials abhängt, ist die Verteilung über die verschiedenen Klassen sehr unterschiedlich. Aus diesem Grund ist die durchschnittliche Häufigkeit eine sehr stark vereinfachte Größe.
Um grob abschätzen zu können, wieviele Bilder für das Training notwendig sind, kann

man von einer gewünschten durchschnittlichen Anzahl der Trainingswerte ausgehen und die beiden vorherigen Formeln umstellen:

$$\varnothing NuFr = \frac{NuCl \cdot NuPh \cdot \varnothing NuTs}{(SzHDx \cdot SzHDy)} \tag{4.7}$$

Mit Hilfe dieser Formel ergibt sich für 395 Bilder der Sequenz BIGSEQ bei 64 Phasen und 256 Klassen eine durchschnittliche Verteilung von 10000 Trainingswerten. Diese Gleichverteilung ist nur eine sehr grobe Schätzung, da man tatsächlich eher eine laplace-förmige Verteilung der Klassen beobachtet (siehe auch Abbildung 4.11). Ein starkes Ungleichgewicht zwischen der kleinsten und größten Anzahl der Klassen ist deutlich erkennbar. Um daraus resultierende Fehler einzudämmen, werden im Folgenden eine obere und eine untere Schranke definiert.
Es muss eine obere Grenze für die Trainingsdaten existieren, um eine ungenaue Berechnung der Filterkoeffizienten zu vermeiden. Da für die Berechnung im Rahmen dieser Arbeit *double*-Werte benutzt und in die Matrix $SDTSD$ die Werte quadriert aufaddiert werden, ergibt sich bei einer maximalen Luminanz von 255 (8 Bit) ein Wert von maximal 65025. Bei 300.000 Trainingswerten können sich die Werte auf $2 \cdot 10^{10}$ akkumulieren. Da der Datentyp double ungefähr 15 Dezimalstellen fehlerfrei darstellen kann, wird die Grenze der Trainingswerte im Rahmen dieser Arbeit auf 300.000 festgelegt, um für die abschließende Berechnung der Filterkoeffizienten noch eine sichere Reserve zur Verfügung zu stellen.
Klassen, die nur sehr selten im Trainingsmaterial vorkommen, sind schwieriger anzulernen. Es besteht die Gefahr, dass die Matrix $SDTSD$ singulär und damit nicht invertierbar ist, und dass die Filterkoeffizienten nicht allgemeingültig angelernt sind und bei der Interpolation zu Artefakten neigen. Deswegen wird zusätzlich eine untere Schranke eingeführt. Für diejenigen Klassen, die weniger als das zehnfache der freien Parametern (also hier Filterkoeffizienten) besitzen, wird keine Lösung mittels MQF Optimierung durchgeführt, sondern stattdessen werden Filterkoeffizienten einer HRS-Impulsantwort benutzt (siehe auch Abschnitt 2.2). Diese entsprechen zwar keiner optimalen Lösung, vermeiden aber die Gefahr von Artefakten (Fallback-Lösung).
Für eine 3x3 Maske ergibt sich eine untere Schranke von $10 \cdot 3 \cdot 3 = 90$ Trainingswerten. Beide Schranken sind in der Abbildung 4.11 zu erkennen.

Durch die Interpolation mit vordefinierten Koeffizienten wird die Bildqualität stark verbessert, da stark sichtbare Artefakte (durch unzureichendes Trainingsmaterial) weitestgehend vermieden werden können. Abbildung 4.12 zeigt exemplarisch die Positionen der 25 seltensten Klassen aus dem in Abbildung 4.11 dargestellten Datensatz. Die Stellen (als weiße Punkte dargestellt) liegen hauptsächlich in unstrukturierten und homogenen Bildbereichen. Ein Ersetzen dieser Klassen verursacht keinen nennenswerten Schärfeverlust.

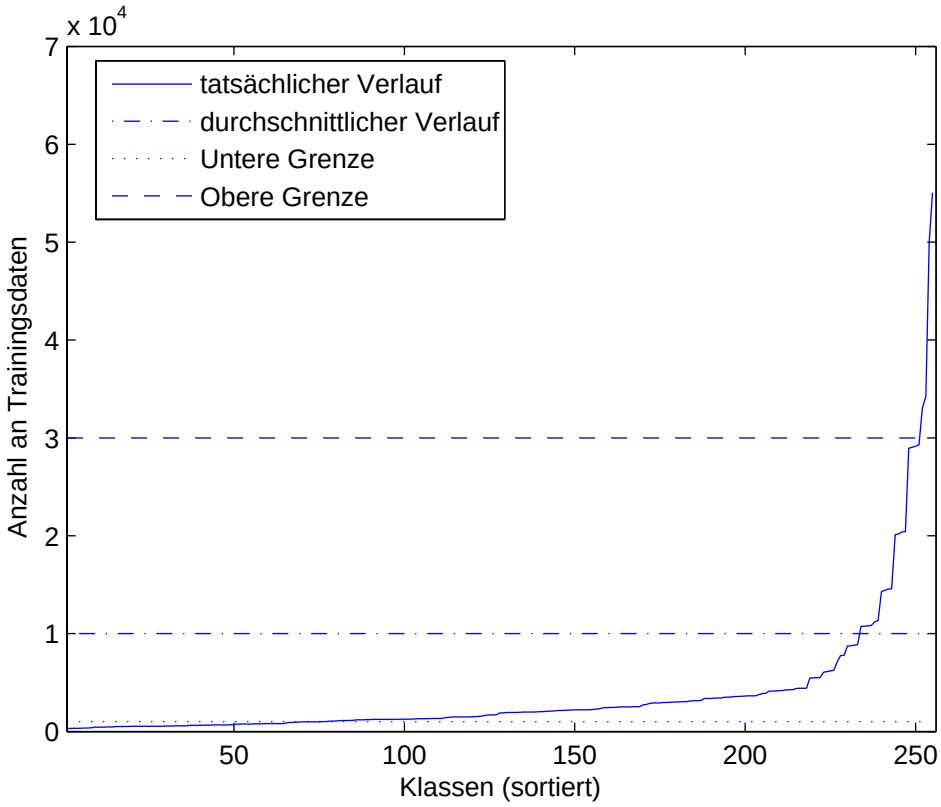


Abbildung 4.11: Klassenverteilung der Sequenz BIGSEQ mit oberer und unterer Schranke

Abbildung 4.12: Darstellung von den Positionen der 25 am seltensten vorhandenen Klassen

Verifizierung der angelernten Daten

Eine weitere Optimierung des Trainingsprozesses findet in Form einer Überprüfung der Filterkoeffizienten statt. Da beim Training eines Interpolationsfilters eine tiefpassähnliche Form wahrscheinlich ist, müssen die Filterkoeffizienten ungefähr den Verlauf eines $sinc(x)$ vorweisen. Da große Abweichungen von der erwarteten Lösung auf einen nicht ausreichend trainierten Datensatz hindeuten, muss man in diesem Fall davon ausgehen, dass die Lösung nicht ideal ist und zu Artefakten führen wird.
Eine typische Verteilung von trainierten Filterkoeffizienten ist in Abbildung 4.13 dargestellt. Man erkennt grob eine Einhüllende in *sinc*-Form. In dieser Abbildung sind auch die im Rahmen dieser Arbeit eingesetzten Maximalwerte (-1,0 und 2,5) dargestellt, die nicht überschritten werden dürfen. Man sieht an diesem Beispiel den großen Abstand zwischen den trainierten Filterkoeffizienten und den Begrenzungen. Die Begrenzung beeinflusst also nicht den Trainingsprozess, sondern filtert lediglich sehr schlecht trainierte Koeffizienten heraus.
Eine weitere Überprüfung der Filterkoeffizienten findet in Form einer Kontrolle des Gesamtfiltergewichtes statt. Es wird überprüft, ob die Gesamtsumme der Filterkoeffizienten in einem Bereich um 1,0 liegt. Ein Gesamtfiltergewicht von eins bewahrt den Gleichanteil des Bildbereiches (keine Abschwächung oder Verstärkung der Helligkeit). Da dies ein charakteristisches Merkmal eines Interpolationstiefpasses ist, deutet eine Abweichung (hier über 20 % Abweichung) auf eine fehlerhafte Lösung der betroffenen Klasse hin. Verworfene Filterwerte werden auch hier durch Filterkoeffizienten einer HRS Impulsantwort ersetzt (siehe auch Abschnitt 2.2).

Einfluss der Maskengröße

Die von Kondo [KNF+01] vorgeschlagene Größe der SD-Maske von 3x3 ergibt bei einer Klassifikation auf der Basis einer Binarisierung insgesamt 256 verschiedene Klassen.
Auf Grund der Basis von lediglich drei Pixeln pro Richtung im Eingangsbild und der damit verbundenen starken Ausschnittsbildung sowie der eingeschränkten Möglichkeit verschiedene Bildsituationen unterscheiden zu können, stellt sich die Frage nach einer Verbesserung der Interpolationsqualität durch vergrößerte Masken.

Erhöht man nun die Größe der SD-Maske, so erhöht sich die Gesamtzahl der Klassen drastisch. Bei einer 4x3 Maske ergeben sich schon 2048, bei 5x3 16384 und bei einer 4x4 Maske 32768 Klassen. Bei der Benutzung einer 5x5 Maske ergeben sich schon 16 Millionen verschiedene Klassen. Um eine derartig hohe Zahl von Klassen anlernen zu können, würde man bei einer SD-Trainingssequenz von 720x405 Pixeln, der Annahme, dass ∅ 10.000 Werte pro Klasse und Phase ausreichen, und bei einem Interpolationsfaktor von 2 nach Formel 4.7 $(16777216 \cdot 10000 \cdot 2 \cdot 2)/(720 \cdot 405 \cdot 4) = 0,5$ Millionen Bilder benötigen. Mit

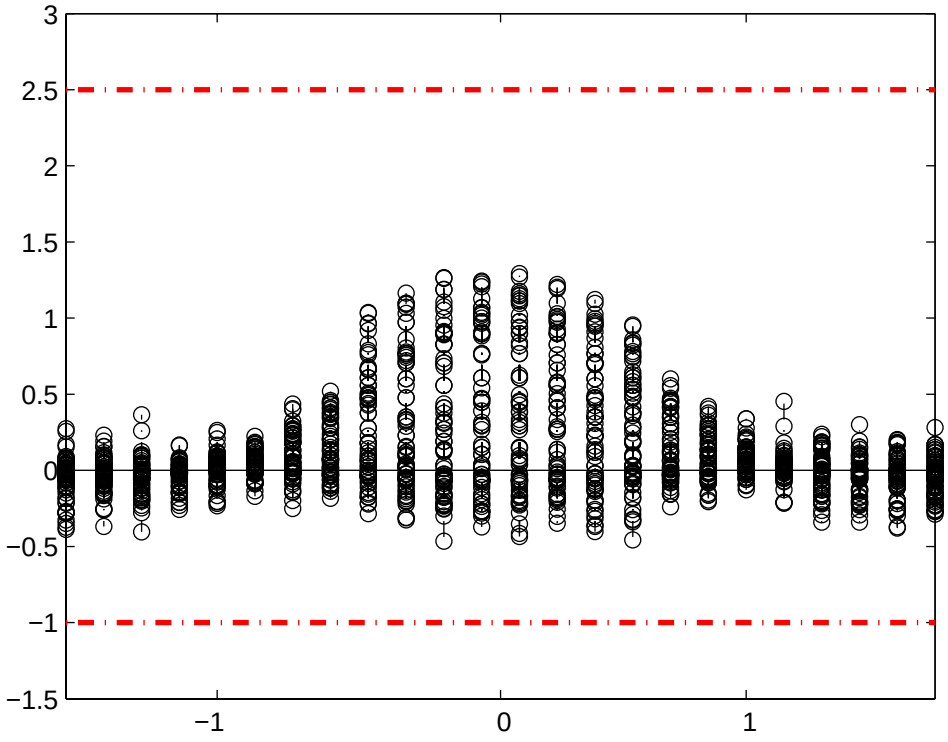


Abbildung 4.13: Typische Verteilung von Filterkoeffizienten (2D Ansicht) und Darstellung der Schranken für die Filterkoeffizienten

Maske	Klassen	Anzahl Trainingsbilder	Speicheraufwand Training	Speicheraufwand Koeffizienten
3x3	256	9	730 KB	36 KB
4x3	2.048	70	9.75 MB	384 KB
5x3	16.384	562	120 MB	3,75 MB
4x4	32.768	1.124	272 MB	8 MB
5x5	16.777.216	575.350	325 GB	6,25 GB

Tabelle 4.1: Einfluss der Größe der SD-Maske auf Training und erforderlichen Speicher am Beispiel einer Interpolation um den Faktor 2

den eben genannten Annahmen wird für verschiedene Maskengrößen in der Tabelle 4.1 eine Übersicht über den benötigten Speicher (nach den Formeln 4.4 und 4.3) sowie über die Anzahl der notwendigen Trainingsdaten gegeben.

Man sieht, dass eine Klassifikation eines Bildbereiches von 5x5 auf der Basis einer Binarisierung eine zu hohe Anzahl an Klassen erzeugen würde. Dies würde eine zu hohe Anforderung an den Trainingsprozess, aber auch an die tatsächliche Anwendung stellen, da eine Filterdatenbank von 6,25 GB(siehe Tabelle 4.1) eindeutig zu groß ist.

Andererseits führt eine Vergrößerung der Maskengröße zu einer Reduktion der Effekte, die durch die Ausschnittsbildung der lokalen Verarbeitung entstehen. Des Weiteren können die Filter genauer an unterschiedliche Bildsituationen angelernt werden, z. B. feinere unterscheidbare Winkel von Kanten.
Somit muss die gewonnene Qualität mit dem Trainingsaufwand, der Größe des Koeffizientenspeichers und dem vergrößerten Aufwand der eigentlichen Interpolation ins Verhältnis gesetzt werden und je nach Anwendungsfall ein optimaler Punkt bestimmt werden.
Abschließend sei an dieser Stelle bemerkt, dass der hier dargestellte Aufwand für lediglich vier Phasen dargestellt ist. Bei einer Interpolation um z. B. den Faktor $\frac{8}{3}$ ergibt sich ein um den Faktor $\frac{8 \cdot 8}{2 \cdot 2} = 16$ höherer Aufwand.

Reduktion der Klassenzahl durch Symmetrien

Wie im vorherigen Abschnitt dargestellt wird, führt eine Vergrößerung der Eingangsmaske zu einer exponentiellen Vergrößerung der Klassenanzahl. Damit wird es zum einen schwieriger die Filterkoeffizienten anzulernen. Zum anderen steigt auch die Größe des Koeffizientenspeichers. Somit ist es sinnvoll die Anzahl der Klassen zu reduzieren. Dies ist vor allem durch Ausnutzung von Symmetrien möglich.
Bei [ZLd02] findet sich ein Vorschlag, für das Verfahren nach Kondo, eine horizontale und vertikale Symmetrie auszunutzen, um lediglich die Koeffizienten von einem von vier

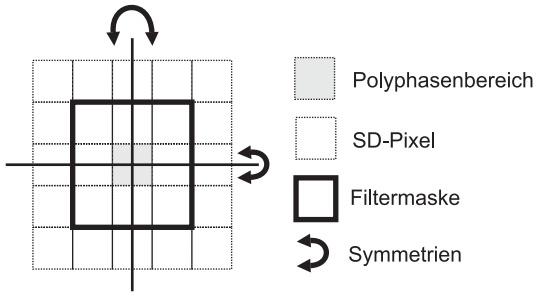


Abbildung 4.14: Horizontale und vertikale Symmetrie

HD-Pixeln abspeichern zu müssen (z. B. a, siehe auch Abbildung 2.5). Hierzu werden die Bildausschnitte gespiegelt, um die Filterkoeffizienten der somit gespiegelten Positionen (b,c,d aus Abbildung 2.5) durch eine Berechnung aus dem gespiegelten Bildausschnitt vornehmen zu können. Dies bedingt jedoch einen symmetrischen Aufbau der Phasen. Ein eventueller Einsatz dieser Reduktionsmethodik bei einer anderen Bildsignalanwendung (z. B. als Rausch- oder Deblockingfilter ohne örtliche Interpolation) würde keine Reduktion ermöglichen, da in diesen Fällen lediglich eine Phase vorhanden ist und diese nicht reduziert werden kann). Aus diesem Grund wird im Folgenden ein Verfahren zur Reduktion der Klassen vorgenommen. Dieser Ansatz hat den Vorteil, dass die Reduktion unabhängig von der Anzahl der vorhanden Phasen vorgenommen werden kann und somit dieser Schritt unabhängig von der tatsächlichen Bildsignalanwendung ist.

Abbildung 4.14 zeigt die im Rahmen dieser Arbeit verwendeten Symmetrien. Es wird nur eine horizontale und eine vertikale Symmetrie ausgenutzt und auf andere Formen der Symmetrie verzichtet (z. B. diagonal), um nicht auf eine quadratische Maskenform angewiesen zu sein. Die Klassifizierung ordnet dem Bildbereich diejenige Klassenzahl zu, die unter Ausnutzung der Symmetrien am niedrigsten ist, und merkt sich die verwendete Symmetrie.
Da sich bei einer Symmetrie (z. B. die horizontale Spiegelung) auch das Verhältnis zwischen den Eingangs- und den Ausgangspixeln ändert, müssen die Interpolationsphasen ebenfalls gespiegelt werden. Dies ist jedoch von der Größe der Eingangsmaske und dem verwendeten Interpolationsfaktor abhängig. In der Abbildung 4.15 wird für den eindimensionalen Fall die unterschiedliche Ausprägung der Phase bei ungerader (links) und gerader (rechts) Maskengröße dargestellt.
Zusätzlich unterscheiden sich die Fälle durch die Anzahl der Interpolationsphasen. In Abbildung 4.15 (a)+(b) sind die Phasen bei einer ungeraden Anzahl an Interpolationsphasen (hier sieben) dargestellt. In 4.15 (c)+(d) sind acht verschiedene Phasen zu sehen. Des Weiteren muss unterschieden werden, ob auf Grund des Interpolationsfaktors eine Ver-

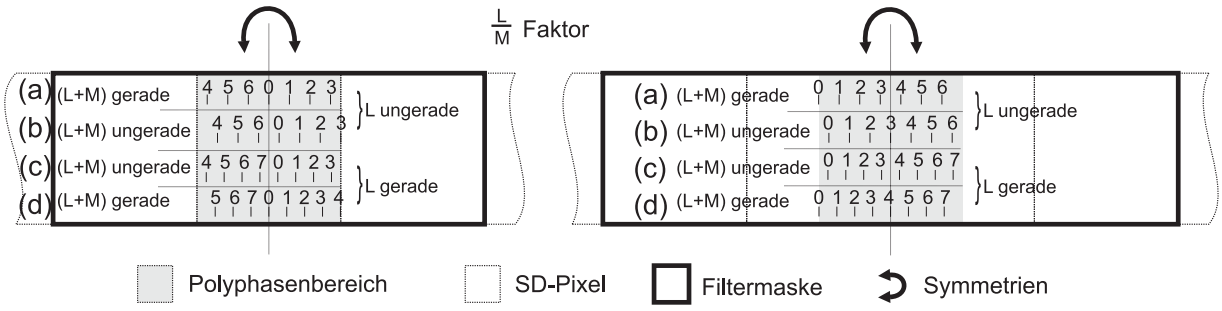


Abbildung 4.15: Einfluss einer Spiegelung auf die Interpolationsphasen

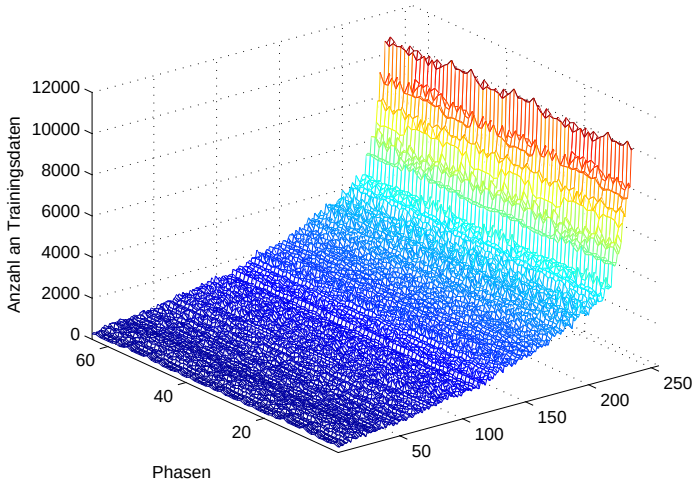
schiebung der Nullphase (4.15 (b)+(d)) oder keine Verschiebung vorliegt (4.15 (a)+(c)). Formal lässt sich dies für einen Interpolationsfaktor $\frac{L}{M}$ folgendermaßen feststellen:

$$Phaseshift = \left\{ \begin{array}{ll} 1\,(ja) & |\,(L+M)mod(2) = 1 \\ 0\,(nein) & |\,(L+M)mod(2) = 0 \end{array} \right\} = (L+M)mod(2) \tag{4.8}$$

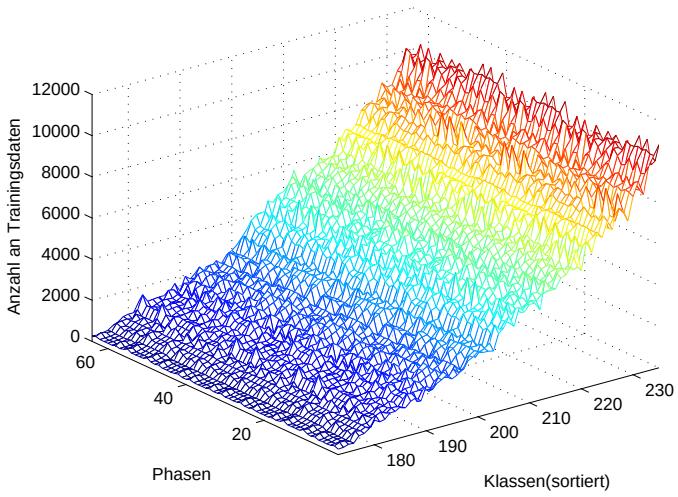
Die Modulo-Summe von $L + M$ kann nicht 0 ergeben, wenn L gerade ist, da somit M ebenfalls gerade wäre und weiter gekürzt werden müsste bis L und/oder M wieder ungerade sind. Somit braucht der Fall (d) aus der Abbildung 4.15 nicht beachtet werden.
Für den Fall (b) bei einer ungeraden Maskenbreite bzw. Fall (a) bei einer geraden Maskenbreite fehlt bei einer Spiegelung für eine Phase das Pendant. Somit ist für derartige Interpolationsfaktoren die Ausnutzung von Symmetrien nicht möglich.
Berücksichtigt man mit dem Modulo-Operator noch "Bereichsüberschreitungen der Phase", so ergibt sich für die gespiegelte Phase:

$$Phase_{spiegel} = (L - Phaseshift - Phase)\,mod(L) \tag{4.9}$$

Mit Hilfe der Symmetrieausnutzung wird die Anzahl der Klassen stark reduziert. Theoretisch müsste nur noch ein Viertel übrigbleiben. Da einige Klassen auf Grund einer symmetrischen Struktur sich bei einer Spiegelung nicht verändern und somit auch nicht in ihrer Anzahl reduziert werden können, ergibt sich bei z. B. 256 Klassen lediglich eine Reduktion auf 84 Klassen (also auf ungefähr ein Drittel von 256).
In Abbildung 4.16 wird die Anzahl von Trainingswerten pro Phase und Klasse für das Training mit und ohne Symmetrieausnutzung dargestellt. Man erkennt in der rechten Abbildung (b) eine starke Reduktion der besetzten Klassen und gleichzeitig einen stärkeren Anstieg der Trainingswerte pro Klassenzahl (die Klassen 0 bis 171 sind unbesetzt).
Um die Klassenreduktion besser beurteilen zu können, werden die Trainingsdaten mit und ohne einer Symmetrieausnutzung zueinander ins Verhältnis gesetzt. Auf diese Weise sind die Gewinne durch Ausnutzung von horizontaler und/oder vertikaler bzw. keiner Symmetrie möglich. Dies ist in Abbildung 4.17 zu erkennen. Lediglich acht Klassen können nicht



(a) Normales Training



(b) Training unter Ausnutzung von Symmetrien

Abbildung 4.16: Verteilung der Trainingsdaten mit und ohne Symmetrien der Sequenz BigSeq bei 395 Trainingsbildern (10.000 ∅ Trainingswerte pro Klasse und Phase)

reduziert werden und besitzen trotz Optimierung keine höhere Anzahl an Trainingsdaten. 28 weitere Klassen können um den Faktor 2 vergrößert werden und 48 Klassen wiederum werden mit viermal so vielen Trainingsdaten angelernt ($48 \cdot 4 + 28 \cdot 2 + 8 \cdot 1 = 256$). Tabelle 4.2 zeigt im Vergleich zur Tabelle 4.1 einen deutlichen Gewinn durch die Reduktion der Klassen.

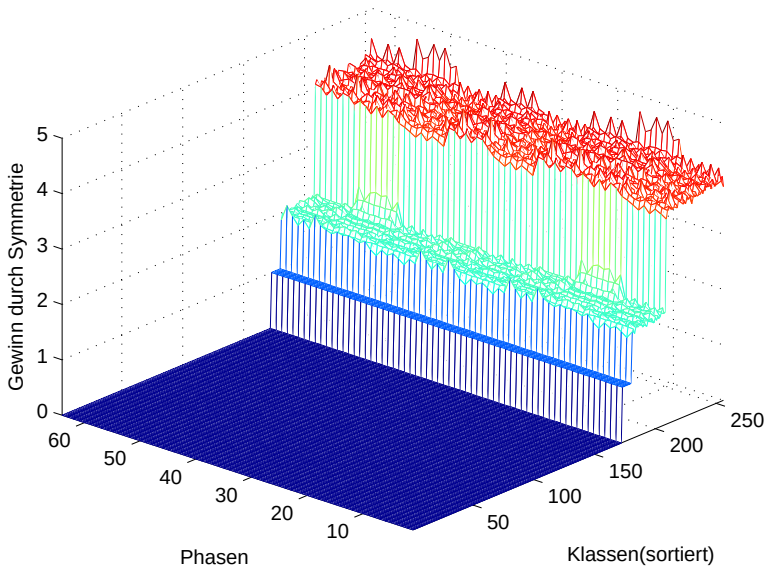


Abbildung 4.17: Gewinn durch Ausnutzung von Symmetrien

Globale und lokale Klassifikation

Auf Grund der einfachen Struktur in der Anwendung ist der Mehrgewinn der adaptiven Filterung vor allem durch die qualitativ hochwertigen Filterkoeffizienten bedingt.
Für den Trainingsprozess werden die gleichen Referenzbilder sowohl in SD- als auch in HD-Auflösung benötigt. Da nur in wenigen Ausnahmefällen solche Sequenzen im Original vorliegen, werden in der Regel HD-Bilder genommen und auf SD-Auflösung dezimiert.
Der verwendete Algorithmus zur Dezimation hat einen großen Einfluss auf das Trainingsergebnis. Die Optimierung minimiert den quadratischen Fehler zwischen den auf der Basis der niedrigaufgelösten Pixel interpolierten Werten und den tatsächlichen hochaufgelösten Werten. Somit ist es möglich durch die geeignete Veränderung der Referenzdaten bestimmte Eigenschaften der Filterkoeffizienten anzutrainieren.
Es finden sich in der Literatur einige Vorschläge, Bildartefakte wie z. B. Rauschen oder Blocking zu reduzieren, indem durch eine adaptive lineare Filterung mit Hilfe einer

Maske	Klassen	Anzahl Trainingsbilder	Speicheraufwand Training	Speicheraufwand Koeffizienten
3x3	84	3	231 KB	12 KB
4x3	576	20	2,74 MB	106 KB
5x3	4.320	148	31,64 MB	0,99 MB
4x4	8.383	288	69,6 MB	2,04 MB
5x5	XXXX	XXX	XXX GB	XXX GB

Tabelle 4.2: Einfluss der Größe der SD-Maske auf Training und erforderlichen Speicher bei Ausnutzung von Symmetrien am Beispiel einer Interpolation um den Faktor 2

ADRC Klassifikation in Kombination mit zusätzlichen Klassifikatoren (wie z. B. der lokalen Bildaktivität) durchgeführt wird. Bei [HdH07] wird z. B. ein klassifikationsbasiertes Filter zum Deblocking und zur gleichzeitigen Bildschärfung vorgestellt.
An dieser Stelle soll jedoch nicht ein einzelnes optimiertes Verfahren vorgestellt werden, sondern die Aufmerksamkeit auf die globale Ausrichtung der gesamten Filterdatensätze gelenkt werden [ML08].
Je nach Trainingsmaterial unterdrücken die adaptiven Filter mehr oder weniger bestimmte zusätzliche Frequenzanteile und besitzen auch im Ortsbereich unterschiedliche Sprungantworten.
Aus diesem Grund ist es sinnvoll, weitere Informationen über das zu interpolierende Bildmaterial zu nutzen (z. B. durch Auswertung von Metadaten, Messungen von Störgrößen, etc.), um somit die am besten passenden Filterkoeffizientensätze auszuwählen.
Bei einem nahezu perfekten und artefaktfreien Eingangsmaterial sollten fast alle vorhandenen Frequenzanteile (bis auf die Wiederholspektren) erhalten und unter Umständen sogar verstärkt werden. In die Filterkoeffizienten kann eine implizite Bildschärfeanhebung eintrainiert werden, indem beispielsweise die SD-Referenzdaten künstlich unscharf gemacht, während die HD-Referenzen in ihrer Schärfe unverändert bleiben.
Ebenso ist es möglich, Rauschen oder andere Artefakte zu reduzieren, indem man artefaktbehaftete SD-Bilder gemeinsam mit fehlerfreien HD-Bildern als Trainingsmaterial benutzt.
Dies ist möglich, da derartig trainierte Filter eine sehr schmale Ausrichtung in Richtung der zu erhaltenen Kanteninformationen besitzen und alle anderen Frequenzanteile stark unterdrücken. Sofern z. B. der Eingang stark artefaktbehaftet ist, sollte eine Datenbank benutzt werden, die Konturen erhält, alle anderen Inhalte jedoch unterdrückt [ML08].

4.2.4 Nebenbedingungen zur Verbesserung der Bildqualität

Im Abschnitt 4.2.3 werden Verfahren zur Überprüfung der entstandenen Filterkoeffizienten vorgestellt. Eine weitere Verbesserung der Qualität der Filter kann erreicht werden, wenn anstatt eines nachträglichen Verwerfens von nicht idealen Filterkoeffizienten, direkt eine Optimierung in einem eingeschränkten Lösungsraum erfolgt. Auf diese Weise wird zum einen die Lösung vereinfacht, zum anderen erfüllen die daraus resultierenden Filterkoeffizienten direkt bestimmte Anforderungen.

Da die MQF Optimierung lediglich eine einfache mathematische Minimierung der quadratischen Abweichung zwischen Soll und Ist realisiert, stimmt die gefundene Lösung oft nicht mit dem Optimum einer subjektiven Bildqualität überein. Aus diesem Grund ist es sehr wichtig, mit Hilfe von Nebenbedingungen den MQF Prozess zu verbessern.
Eine einfache Bedingung ist zum Beispiel der im Kapitel 4.2.3 vorgestellte DC-Erhalt:

$$\sum_i W_i \stackrel{!}{=} 1.0 \tag{4.10}$$

Diese Nebenbedingung kann dann in der Form $A\vec{w} = \vec{b}$ dargestellt werden und somit eine Optimierung mit Nebenbedingungen (siehe Formel 3.17) vorgenommen werden. Diese lässt sich durch eine Transformation der Eingangs- und Ausgangswerte sowie einer anschließenden MQF Optimierung ohne Nebenbedingungen beschreiben. Hierbei wird der Freiheitsgrad um die Anzahl der linear unabhängigen Nebenbedingungen verringert.

Da die Trainingswerte für diesen reduzierten Lösungsraum nicht direkt verwendet werden können, müssen sie dementsprechend während des Trainingsprozesses umtransformiert werden.
Auch in diesem Fall ist eine akkumulative Abspeicherung der Trainingsdaten (niedrigaufgelöst SD und hochaufgelöst $\vec{hd}$), wie sie im Abschnitt 4.2 vorgestellt wird (Formel 3.32) sinnvoll. Somit ergibt sich aus der Formel 4.2:

$$\begin{aligned}(SD \cdot Q_2)^T \cdot (SD \cdot Q_2) &= (SD_1 \cdot Q_2)^T \cdot (SD_1 \cdot Q_2) + (SD_2 \cdot Q_2)^T \cdot (SD_2 \cdot Q_2) \\ &= \cdot Q_2^T \left(SD_1^T \cdot SD_1 + SD_2^T \cdot SD_2\right) \cdot Q_2\end{aligned} \tag{4.11}$$

Nach Formel 4.2 müssen auch die hochaufgelösten Trainingsdaten angepasst werden:

$$\begin{aligned}(SD \cdot Q_2)^T \cdot \left(\vec{hd} - SD \cdot Q_1 \cdot \vec{w}_{mod,1}\right) &= Q_2^T \cdot \left(SD_1^T \cdot (hd_1 - SD_1 \cdot Q_1 \cdot \vec{w}_{mod,1})\right. \\ &\quad \left. + SD_2^T \cdot (hd_2 - SD_2 \cdot Q_1 \cdot \vec{w}_{mod,1})\right)\end{aligned} \tag{4.12}$$

Im Folgenden werden weitere Nebenbedingungen direkt aus Bildqualitätsforderungen hergeleitet. Der Optimierungsprozess (wie oben dargestellt) bleibt jedoch unverändert.

Auf Grund der begrenzten Ausdehnung des Interpolationsfilters und der einfachen Beschreibung von Bildqualität über den mittleren quadratischen Fehler ist dieser Trainingsprozess begrenzt. So will man zwar einen möglichst optimalen Schärfeerhalt, hat aber nur eine begrenzte Filterausdehnung. Durch diese Ausschnittsbildung können unerwünschte Artefakte entstehen, die vor allem in homogenen Bildbereichen störend sichtbar sind. In Abbildung 4.18 sind diese Artefakte abgebildet, die vor allem beim Training durch sehr hochfrequentes SD-Material entstehen (hohe Bildschärfe und Detailreichtum). Man kann die originale Pixelstruktur des Eingangsbildes im Ausgangsbild wiedererkennen. Dies macht sich vor allem bei niederfrequenten Bildbereichen wie Gesichtern oder anderen Luminanzverläufen bemerkbar.
Die Ursache dieser Störungen liegt in einer unzureichenden Unterdrückung der Wieder-

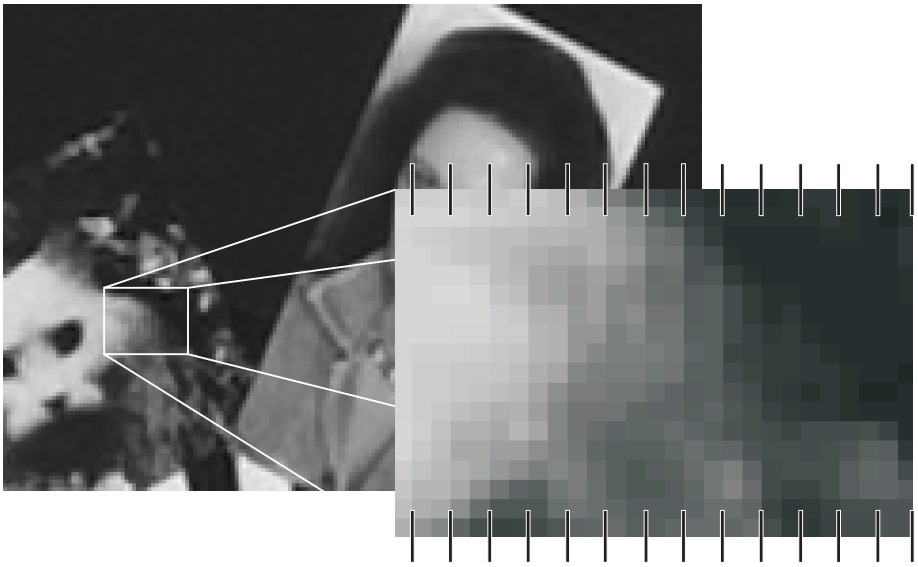
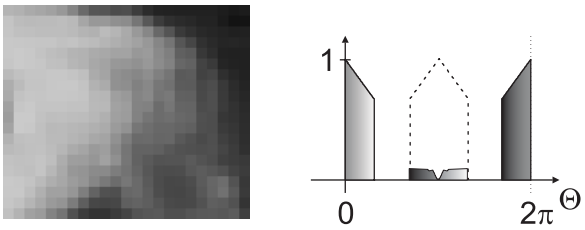


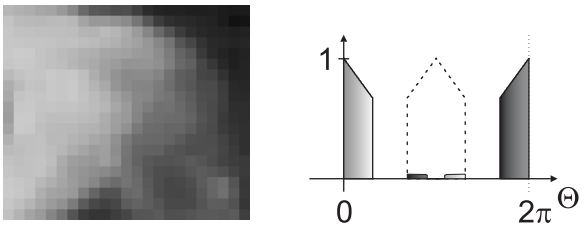
Abbildung 4.18: Artefakte im interpolierten Bild durch sichtbares Raster der Originalpixel (schwarze Markierung)

holspektren, wie in Abbildung 4.19 dargestellt wird. Auf der linken Seite ist das Spektrum eines Signals nach einer unzureichenden Filterung bei einer zweifachen Interpolation zu sehen. Durch einen Nulldurchgang im Spektrum wird die DC Frequenz des Wiederholspektrums zu Null gesetzt. Die übriggebliebenen niederfrequenten Spektralanteile bewirken jedoch eine störende Sichtbarkeit des Originalrasters (siehe (a) und (b)). Entfernt man diese Anteile von dem Spektrum, so verschwinden die störenden Artefakte weitestgehend.

Um die Bedingungen zu definieren, die diese Artefakte unterdrücken, müssen also Nebenbedingungen erzeugt werden, die diese Frequenzanteile weitestgehend unterdrücken.
Betrachtet man das Spektrum von einfachen Polynomen (bis zweiter Ordnung) so fällt



(a) Bild (links) mit Artefakten, Spektrum mit niederfrequenten Anteilen im Bereich der Störspektren (rechts)



(b) Bild (links) nahezu artefaktfrei, stärkere Unterdrückung der Störspektren (rechts)

Abbildung 4.19: Beispielhafter Ausschnitt von Artefakten mit schematischem Spektrum zur Erklärung der Artefakte

auf, dass ihre spektrale Verteilung hauptsächlich in dem niederfrequenten Bereich liegt und zu höheren Frequenzen stark abfällt. Im Bereich der Wiederholspektren ist diese Verteilung genau dort groß, wo Frequenzen zu sichtbaren Störungen im Ausgangsbild führen.

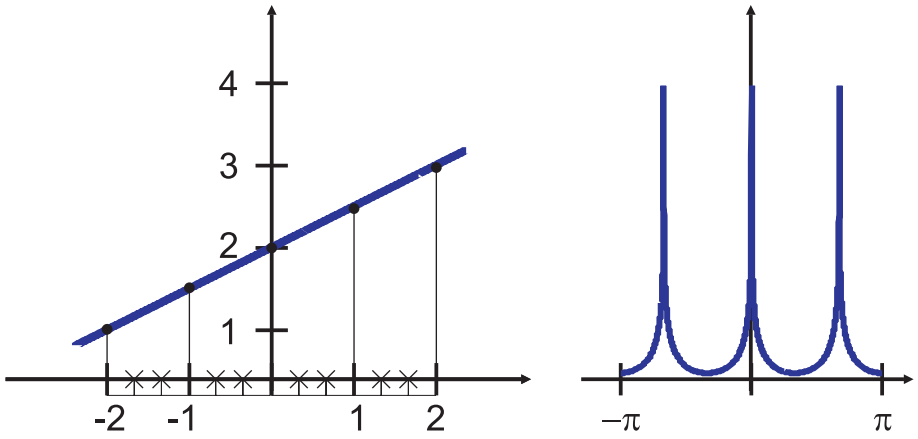
Aus diesem Grund wird im Folgenden der Erhalt solcher einfachen Strukturen bei einer Interpolation gefordert und es werden hieraus die entsprechenden Nebenbedingungen abgeleitet.
Wenn einfache Signalverläufe (Polynome bis zum zweiten Grad) perfekt interpoliert werden, kann dies im Spektrum als perfekte Unterdrückung ihrer Wiederholspektren interpretiert werden. Abbildung 4.20 zeigt dies an Hand einer dreifachen Interpolation. Man erkennt in Abbildung 4.20 (a) das Signal nach einer dreifachen Abtastrateninterpolation durch Einfügen von zwei Nullen zwischen den ursprünglichen Abtastwerten. Das zugehörige Spektrum besteht deswegen aus dem Originalspektrum und zwei Störspektren jeweils bei $\frac{2\pi}{3}$ und $-\frac{2\pi}{3}$. Es wird nun angenommen, dass diese Störspektren perfekt unterdrückt werden können (siehe 4.20 (d)), so dass im Ortsbereich das Signal fehlerfrei rekonstruiert wird (siehe 4.20 (c)).

Natürlich entspricht einerseits die Annahme eines Polynoms begrenzter Ordnung nicht dem tatsächlichen Eingangssignal, andererseits sind in natürlichen Sequenzen die Spektren ebenfalls nahezu laplace-förmig verteilt. Somit ähneln sich die Spektren. Des Weiteren entsprechen die Bildbereiche, in denen die Artefakte sichtbar sind, den angenommenen einfachen Signalen sehr gut.

Die Forderungen werden zuerst im Detail für den eindimensionalen Fall hergeleitet, und im weiteren Schritt auf die zweite Dimension erweitert. Eine perfekte Interpolation bedeutet, dass ein Signal $F_{Orig}(x)$, welches als abgetastete Werte $F_{Orig}(x_n)$ an diskreten ganzzahligen Positionen $[n]$ bekannt ist, auch an jeder Zwischenposition α (zwischen 0 und 1) berechnet werden kann.
Formal ergibt sich somit:

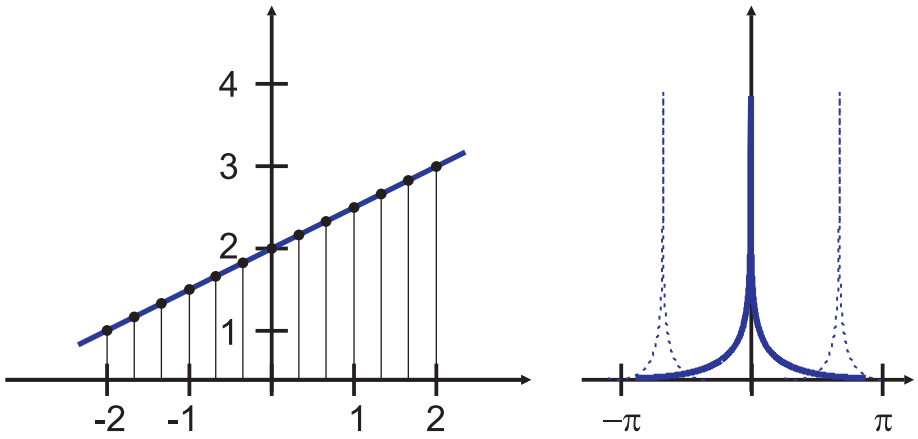
$$F_{Ipol}(x_n + \alpha) - F_{Orig}(x_n + \alpha) \stackrel{!}{=} 0 \;\; \forall \alpha \in [0:1[\tag{4.13}$$

Die Interpolation erfolgt auf der Basis von vier bekannten Eingangswerten $F_{Orig}[n-1, n, n+1, n+2]$, bzw. $F_{Orig}(x_n - 1, x_n, x_n + 1, x_n + 2)$, die mit vier passenden Filterkoeffizienten $A(\alpha), B(\alpha), C(\alpha), D(\alpha)$ multipliziert werden, um den Wert an der Stelle $x_n + \alpha$ interpolieren zu können (siehe auch Abbildung 4.21).



(a) Abgetastete Gerade mit eingefügten Nullen

(b) zu (a) zugehöriges Spektrum



(c) Fehlerfrei interpoliertes Signal

(d) zu (c) zugehöriges Spektrum

Abbildung 4.20: Spektrale Auswirkung der Forderung einer fehlerfreien Interpolation

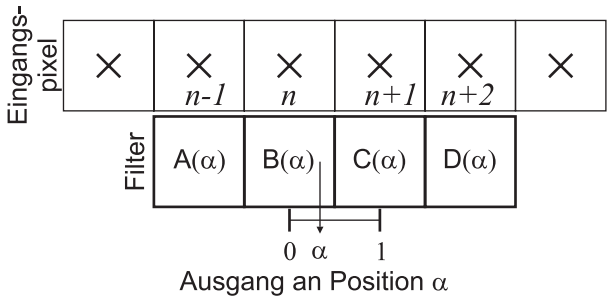


Abbildung 4.21: Interpolation der Phase α auf der Basis von vier Eingangspixeln

Die Anzahl der Filterkoeffizienten ist willkürlich, jedoch kann im Folgenden gezeigt werden, dass bei einem Erhalt von Polynomen zweiten Grades, drei Nebenbedingungen erzeugt werden, so dass die Anzahl der Filterkoeffizienten größer oder gleich drei sein muss:

$$\begin{aligned} F_{Ipol}(x_n+\alpha) = \;& F_{Orig}(x_n-1)\cdot A(\alpha) + F_{Orig}(x_n)\cdot B(\alpha) \\ + \;& F_{Orig}(x_n+1)\cdot C(\alpha) + F_{Orig}(x_n+2)\cdot D(\alpha) \end{aligned} \tag{4.14}$$

Nun wird das Eingangssignal allgemein durch eine Taylorentwicklung an der Stelle x_n beschrieben:

$$\begin{aligned} F_{Taylor}(x_n+\alpha) &= F_{Orig}(x_n+\alpha) \\ &= F_{Orig}(x_n) + (\alpha)\cdot F'_{Orig}(x_n) + \tfrac{1}{2}\cdot(\alpha)^2\cdot F''_{Orig}(x_n) + \cdots \end{aligned} \tag{4.15}$$

Durch die Annahme, dass das Eingangssignal ein einfaches Polynom zweiten Grades ist, fallen höhere Anteile der Taylorentwicklung weg, da die weiteren Ableitungen der Funktion Null ergeben. Zusätzlich werden alle Eingangspixel des Interpolationsprozesses (siehe Gleichung 4.14) als Taylorentwicklungen angegeben.

$$\begin{aligned} F_{Taylor}(x_n-1) &= F_{Orig}(x_n) - 1\cdot F'_{Orig}(x_n) + \tfrac{1}{2}\cdot 1^2\cdot F''_{Orig}(x_n) \\ F_{Taylor}(x_n) &= F_{Orig}(x_n) \\ F_{Taylor}(x_n+1) &= F_{Orig}(x_n) + 1\cdot F'_{Orig}(x_n) + \tfrac{1}{2}\cdot 1^2\cdot F''_{Orig}(x_n) \\ F_{Taylor}(x_n+2) &= F_{Orig}(x_n) + 2\cdot F'_{Orig}(x_n) + \tfrac{1}{2}\cdot 2^2\cdot F''_{Orig}(x_n) \end{aligned} \tag{4.16}$$

Kombiniert man die Ausdrücke 4.13, 4.14 und 4.16, so erhält man:

$$\begin{aligned} F_{Ipol}(x_n+\alpha) - F_{Orig}(x_n+\alpha) &= 0 \\ = F_{Orig}(x_n)\cdot &\;[1 - A(\alpha) - B(\alpha) - C(\alpha) - D(\alpha)] \\ +1\cdot F'_{Orig}(x_n)\cdot &\;[\alpha + A(\alpha) - C(\alpha) - 2D(\alpha)] \\ +\tfrac{1}{2}\cdot 2^2\cdot F''_{Orig}(x_n)\cdot &\;[\alpha^2 - A(\alpha) - C(\alpha) - 4D(\alpha)] \end{aligned} \tag{4.17}$$

Unter Berücksichtigung, dass dieser Ausdruck für alle Werte von $\alpha \in [0:1[$ Null ergibt, erhält man drei verschiedene Nebenbedingungen:

$$\begin{aligned} A(\alpha) + B(\alpha) + C(\alpha) + D(\alpha) &= 1 && (4.18)\\ -A(\alpha) + C(\alpha) + 2D(\alpha) &= \alpha && (4.19)\\ A(\alpha) + C(\alpha) + 4D(\alpha) &= \alpha^2 && (4.20) \end{aligned}$$

Gleichung 4.18 ist der bekannte DC-Erhalt (Formel 4.10) aus dem vorherigen Abschnitt, den man als Forderung für den Erhalt von Polynomen nullter Ordnung verstehen kann. Die beiden anderen Formeln 4.19 und 4.20 stehen für den Erhalt von Geraden (erster Grad) und Polynomen zweiten Grades.
Die Nebenbedingungen für zweidimensionale Filterstrukturen können analog erzeugt werden. Geht man von dem Signal $F_{Orig}(x,y)$ aus und definiert man mit α die zu interpolierende x- und mit β die zu interpolierende y-Richtung, so erhält man als Interpolationsvorschrift bei einer 4x4 Maske:

$$\begin{aligned} F_{Ipol}\ (x_i+\alpha, y_j+\beta) = \\ & F_{Orig}(x_i-1, y_j-1)\cdot A_{\alpha,\beta} && +F_{Orig}(x_i, y_j-1)\cdot B_{\alpha,\beta}\\ & +F_{Orig}(x_i+1, y_j-1)\cdot C_{\alpha,\beta} && +F_{Orig}(x_i+2, y_j-1)\cdot D_{\alpha,\beta}\\ & +F_{Orig}(x_i-1, y_j)\cdot E_{\alpha,\beta} && +F_{Orig}(x_i, y_j)\cdot F_{\alpha,\beta}\\ & +F_{Orig}(x_i+1, y_j)\cdot G_{\alpha,\beta} && +F_{Orig}(x_i+2, y_j)\cdot H_{\alpha,\beta}\\ & +F_{Orig}(x_i-1, y_j+1)\cdot I_{\alpha,\beta} && +F_{Orig}(x_i, y_j+1)\cdot J_{\alpha,\beta}\\ & +F_{Orig}(x_i+1, y_j+1)\cdot K_{\alpha,\beta} && +F_{Orig}(x_i+2, y_j+1)\cdot L_{\alpha,\beta}\\ & +F_{Orig}(x_i-1, y_j+2)\cdot M_{\alpha,\beta} && +F_{Orig}(x_i, y_j+2)\cdot N_{\alpha,\beta}\\ & +F_{Orig}(x_i+1, y_j+2)\cdot O_{\alpha,\beta} && +F_{Orig}(x_i+2, y_j+2)\cdot P_{\alpha,\beta} \end{aligned} \quad (4.21)$$

Auch in diesem Fall wird eine fehlerfreie Interpolation gefordert:

$$F_{Ipol}(x_i+\alpha, y_j+\beta) - F_{Orig}(x_i+\alpha, y_j+\beta) \stackrel{!}{=} 0 \ \ \forall \alpha, \beta \in [0:1[\quad (4.22)$$

Analog zum eindimensionalen Ansatz wird nun eine zweidimensionale Taylorentwicklung [BS91] vorgenommen.

$$\begin{aligned} F_{Taylor}(x_i+\alpha, y_j+\beta) &= F_{Orig}(x_i+\alpha, y_j+\beta) \\ & \quad F_{Orig}(x_i, y_j) \\ &+ F^{x}_{Orig}(x_i, y_j)\cdot\alpha + F^{y}_{Orig}(x_i, y_j)\cdot\beta \\ &+ F^{x\,x}_{Orig}(x_i, y_j)\cdot\alpha^2\cdot\tfrac{1}{2} \\ &+ F^{x\,y}_{Orig}(x_i, y_j)\cdot\alpha\cdot\beta + F^{y\,y}_{Orig}(x_i, y_j)\cdot\beta^2\cdot\tfrac{1}{2}\ldots \end{aligned} \quad (4.23)$$

Kombiniert man nun die Terme 4.21 und 4.22 und fügt die Taylorentwicklungen für die einzelnen Eingangspixel ein, so erhält man anschließend:

$$
\begin{aligned}
0 = \quad & F_{Orig}(x_i, y_j) \cdot \begin{bmatrix} A_{\alpha,\beta} & +B_{\alpha,\beta} & +C_{\alpha,\beta} & +D_{\alpha,\beta} \\ +E_{\alpha,\beta} & +F_{\alpha,\beta} & +G_{\alpha,\beta} & +H_{\alpha,\beta} \\ +I_{\alpha,\beta} & +J_{\alpha,\beta} & +K_{\alpha,\beta} & +L_{\alpha,\beta} \\ +M_{\alpha,\beta} & +N_{\alpha,\beta} & +O_{\alpha,\beta} & +P_{\alpha,\beta} \end{bmatrix} - 1 \\
+ \quad & F^{y}_{Orig}(x_i, y_j) \cdot \begin{bmatrix} -A_{\alpha,\beta} & -B_{\alpha,\beta} & -C_{\alpha,\beta} & -D_{\alpha,\beta} \\ +0 & +0 & +0 & +0 \\ +I_{\alpha,\beta} & +J_{\alpha,\beta} & +K_{\alpha,\beta} & +L_{\alpha,\beta} \\ +2M_{\alpha,\beta} & +2N_{\alpha,\beta} & +2O_{\alpha,\beta} & +2P_{\alpha,\beta} \end{bmatrix} - \beta \\
+\tfrac{1}{2} \quad & F^{x\,x}_{Orig}(x_i, y_j) \cdot \begin{bmatrix} A_{\alpha,\beta} & +0 & +C_{\alpha,\beta} & +4D_{\alpha,\beta} \\ +E_{\alpha,\beta} & +0 & +G_{\alpha,\beta} & +4H_{\alpha,\beta} \\ +I_{\alpha,\beta} & +0 & +K_{\alpha,\beta} & +4L_{\alpha,\beta} \\ +M_{\alpha,\beta} & +0 & +O_{\alpha,\beta} & +4P_{\alpha,\beta} \end{bmatrix} - \alpha \cdot \alpha \\
+ \quad & F^{x}_{Orig}(x_i, y_j) \cdot \begin{bmatrix} -A_{\alpha,\beta} & +0 & +C_{\alpha,\beta} & +2D_{\alpha,\beta} \\ -E_{\alpha,\beta} & +0 & +G_{\alpha,\beta} & +2H_{\alpha,\beta} \\ -I_{\alpha,\beta} & +0 & +K_{\alpha,\beta} & +2L_{\alpha,\beta} \\ -M_{\alpha,\beta} & +0 & +O_{\alpha,\beta} & +2P_{\alpha,\beta} \end{bmatrix} - \alpha \\
+ \quad & F^{x\,y}_{Orig}(x_i, y_j) \cdot \begin{bmatrix} A_{\alpha,\beta} & +0 & -C_{\alpha,\beta} & -2D_{\alpha,\beta} \\ +0 & +0 & +0 & +0 \\ -I_{\alpha,\beta} & +0 & +K_{\alpha,\beta} & +2L_{\alpha,\beta} \\ -2M_{\alpha,\beta} & +0 & +2O_{\alpha,\beta} & +4P_{\alpha,\beta} \end{bmatrix} - \alpha \cdot \beta \\
+\tfrac{1}{2} \quad & F^{y\,y}_{Orig}(x_i, y_j) \cdot \begin{bmatrix} A_{\alpha,\beta} & +B_{\alpha,\beta} & +C_{\alpha,\beta} & +D_{\alpha,\beta} \\ +0 & +0 & +0 & +0 \\ +I_{\alpha,\beta} & +J_{\alpha,\beta} & +K_{\alpha,\beta} & +L_{\alpha,\beta} \\ +4M_{\alpha,\beta} & +4N_{\alpha,\beta} & +4O_{\alpha,\beta} & +4P_{\alpha,\beta} \end{bmatrix} - \beta \cdot \beta \overset{!}{=} 0
\end{aligned}
\tag{4.24}
$$

An dieser Stelle wird die Abhängigkeit der Filteranforderungen vom Taylorentwurf deutlich. So erkennt man die Ausrichtung der Anforderungen in x- und y-Richtung sowie auch die Abhängigkeit vom Abstand. Der Abstand ist die Differenz der aktuellen Pixelposition zum Zentrum der Maske (siehe auch Formel 2.10).

Für eine beliebige Filtermaskengröße ergibt sich folgender Formalismus:

- DC-Erhalt: Summe aller Filtergewichte $(A + B + C \dots)$
- Rampenerhalt in x-Richtung:
  Einfache Gewichtung in Abhängigkeit vom x-Abstand $(-1A, 0, 1C, 2D \dots)$

- Rampenerhalt in y-Richtung:
  Einfache Gewichtung in Abhängigkeit vom y-Abstand $(-1A, 0, 1I, 2M \ldots)$
- Erhalt quadratischer Signale in x-Richtung:
  Quadratische Gewichtung in Abhängigkeit vom x-Abstand
  $((-1)^2A, 0, (1)^2C, (2)^2D)$
- Erhalt quadratischer Signale in y-Richtung:
  Quadratische Gewichtung in Abhängigkeit vom y-Abstand
  $((-1)^2A, 0, (1)^2I, (2)^2M)$
- Erhalt multiplikativer xy-Signale in xy-Richtung:
  Einfache Gewichtung in Abhängigkeit vom y-Abstand multipliziert mit der einfachen Gewichtung in Abhängigkeit vom x-Abstand
  $((-1) \cdot (-1)A, 0, (-1) \cdot (1)C, (-1) \cdot (2)D)$

Mit Hilfe dieser Vorschriften lassen sich die Filterbedingungen für verschiedenste Maskengrößen festlegen. Unabhängig von den zu Grunde liegenden Trainingssequenzen, wird die Einhaltung dieser Qualitätsvorschriften automatisch garantiert.

Zur Demonstration der Leistungsfähigkeit dieser Vorschriften soll im Folgenden angenommen werden, dass eine Interpolation lediglich auf zwei bzw. drei Stützstellen beruht. Im Fall von zwei Stellen (Pixeln) können zur Definition der Filterkoeffizienten auch nur zwei Bedingungen genommen werden (z. B. der DC Erhalt und der Erhalt von linearen Verläufen in x-Richtung).

$$\begin{aligned} 1 &= B(\alpha) + C(\alpha) \\ 0 &= \alpha - C(\alpha) \end{aligned} \tag{4.25}$$

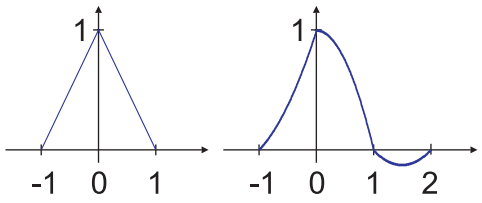
Somit ergibt sich:

$$\begin{aligned} B(\alpha) &= 1 - C(\alpha) = 1 - \alpha \\ C(\alpha) &= \alpha \end{aligned} \tag{4.26}$$

Dies entspricht der Konstruktionsvorschrift einer bilinearen Interpolation. Erweitert man die Interpolation um einen weiteren Koeffizienten, so ergibt sich die Möglichkeit auch quadratische Verläufe interpolieren zu können.

$$\begin{aligned} 1 &= B(\alpha) + C(\alpha) + D(\alpha) \\ \alpha &= C(\alpha) + 2D(\alpha) \\ \alpha^2 &= C(\alpha) + 4D(\alpha) \end{aligned} \tag{4.27}$$

$$\begin{aligned} B(\alpha) &= 1 - \tfrac{3}{2}\alpha + \tfrac{1}{2}\alpha^2 \\ C(\alpha) &= 2\alpha - \alpha^2 \\ D(\alpha) &= \tfrac{\alpha^2 - \alpha}{2} \end{aligned} \tag{4.28}$$



(a) Filterbreite 2 (b) Filterbreite 3

Abbildung 4.22: Verschiedene Impulsantworten, lediglich aus den Taylorbedingungen abgeleitet

Diese beiden Beispiele (siehe auch Abbildung 4.22) sollen lediglich die Leistungfähigkeit der Vorschriften darstellen. Für optimierte der Bildsituation angepasste Filter müssen diese Vorschriften mit in den MQF-Minimierungsprozess eingebunden werden.

4.3 Polynombasierte Interpolation

Bisherige adaptive Interpolationsverfahren erlauben nur eine begrenzte Anzahl an Interpolationsfaktoren. Polyphasenfilterdatenbänke verbessern dies, sind jedoch auf eine feste Anzahl an vortrainierten Faktoren beschränkt.
Auf der anderen Seite sind einfache statische Verfahren (wie z. B. HRS oder bilinear) in der Lage ohne erneutes Training beliebige Zwischenphasen zu berechnen. Dies ist dadurch bedingt, dass die Filterkoeffizienten über eine kontinuierliche Impulsantwort definiert sind.
In diesem Abschnitt wird der bisherige Polyphasenansatz (siehe Abschnitt 4.2.2) entsprechend erweitert, so dass eine kontinuierliche Interpolationsform angelernt werden kann.
Problematisch bei dem Training diskreter Filterwerte ist natürlich die Beschränkung auf eine feste Anzahl an Interpolationsphasen. Zusätzlich werden die Filterkoeffizienten Phase für Phase unabhängig voneinander angelernt. Somit existieren kleinere Abweichungen zwischen den Ergebnissen (siehe Abschnitt 4.2 und Abbildung 4.10).
Um den Trainingsprozess für alle Phasen zu vereinheitlichen und zu verbessern und quasi beliebige Phasen zur Verfügung zu stellen, wird im Folgenden ein polynombasiertes inhaltsadaptives Interpolationsverfahren vorgestellt.
Zur vereinfachten Darstellung wird von einem eindimensionalen Polynom ausgegangen. Um nicht mehr einzelne Filterkoeffizienten anzulernen, sondern die kontinuierliche Interpolationsform, muss die Gleichung 3.8 modifiziert werden. Im Folgenden ist es wichtig zu erkennen, dass die Werte des Polynoms an bestimmten Stützstellen die vorhin trainierten Filterkoeffizienten an einer bestimmten Phase sind.
Somit können durch die Beschreibung des Polynoms an beliebigen Stellen die Filterkoeffizienten berechnet werden. Definiert man die Position innerhalb der Interpolationsfunktion

mit $x_{0,\alpha}$ und den an dieser Stelle zu berechnenden Filterkoeffizienten mit $w_{0,\alpha}$, erhält man folgenden Zusammenhang:

$$poly(x) = a \cdot x^3 + b \cdot^2 + c \cdot x^1 + d \cdot x^0 \tag{4.29}$$

$$w_{0,\alpha} = poly(x_{0,\alpha}) = a \cdot x_{0,\alpha}^3 + b \cdot x_{0,\alpha}^2 + c \cdot x_{0,\alpha}^1 + d \cdot x_{0,\alpha}^0 \tag{4.30}$$

So können die Filterkoeffizienten an den einzelnen Positionen durch ein Polynom mit zugehörigen Polynomkoeffizienten beschrieben werden. Setzt man nun eine Beschreibung des Filterkoeffizienten wie in Formel 4.30 für alle Koeffizienten in die Formel 4.1 ein, ergibt sich:

$$\min \left\| \begin{bmatrix} SD_{11} & ... & SD_{19} \\ ... & & ... \\ SD_{n1} & ... & SD_{n9} \end{bmatrix} \cdot \begin{pmatrix} a \cdot x_{1,\alpha}^3 + b \cdot x_{1,\alpha}^2 + c \cdot x_{1,\alpha}^1 + d \cdot x_{1,\alpha}^0 \\ a \cdot x_{2,\alpha}^3 + b \cdot x_{2,\alpha}^2 + c \cdot x_{2,\alpha}^1 + d \cdot x_{2,\alpha}^0 \\ ... \\ a \cdot x_{9,\alpha}^3 + b \cdot x_{9,\alpha}^2 + c \cdot x_{9,\alpha}^1 + d \cdot x_{9,\alpha}^0 \end{pmatrix} - \begin{pmatrix} HD_{1,\alpha} \\ ... \\ HD_{n,\alpha} \end{pmatrix} \right\|_2^2 \tag{4.31}$$

Im Bereich des ersetzten Vektors $\vec{w}$ sind die Polynomkoeffizienten in jeder Zeile identisch, lediglich die abgetastete Position des Polynoms ist unterschiedlich. Es zeigt sich, dass man die Polynomkoeffizienten extrahieren und damit diesen Vektor auch als Matrix-Vektorprodukt aufschreiben kann.

$$\begin{aligned} &\min \left\| \begin{bmatrix} SD_{11} & ... & SD_{19} \\ ... & & ... \\ SD_{n1} & ... & SD_{n9} \end{bmatrix} \cdot \begin{bmatrix} x_{1,\alpha}^3 & x_{1,\alpha}^2 & x_{1,\alpha}^1 & x_{1,\alpha}^0 \\ x_{2,\alpha}^3 & x_{2,\alpha}^2 & x_{2,\alpha}^1 & x_{2,\alpha}^0 \\ ... & & & \\ x_{9,\alpha}^3 & x_{9,\alpha}^2 & x_{9,\alpha}^1 & x_{9,\alpha}^0 \end{bmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} - \begin{pmatrix} HD_{1,\alpha} \\ ... \\ HD_{n,\alpha} \end{pmatrix} \right\|_2^2 \\ &= \min \left\| [\mathbf{SD}] \cdot [\mathbf{X}_\alpha] \cdot \overrightarrow{poly} - \overrightarrow{HD_\alpha} \right\|_2^2 \end{aligned} \tag{4.32}$$

Somit erhält man eine Transformationsmatrix $[X_\alpha]$ mit deren Hilfe die Eingangspixel so umstrukturiert werden, dass sie zu den Polynomkoeffizienten passen.
Auf diese Weise ist es auch möglich die Matrix $[X_\alpha]$ mit den Trainingsdaten $[SD]$ zu multiplizieren, um einen einfachen MQF-Ansatz vorliegen zu haben:

$$\min \left\| [\mathbf{SDX}_\alpha] \cdot \overrightarrow{poly} - \overrightarrow{HD_\alpha} \right\|_2^2 \tag{4.33}$$

Diesen Ausdruck erweitert man auf die Trainingsdaten (SD und HD) für alle Phasen und erhält die gewünschten Koeffizienten des Interpolationspolynoms.

$$\begin{aligned} &\min \left\| \begin{bmatrix} \mathbf{SDX}_\alpha \\ \mathbf{SDX}_\beta \\ ... \end{bmatrix} \cdot \overrightarrow{poly} - \begin{pmatrix} HD_\alpha \\ HD_\beta \\ ... \end{pmatrix} \right\|_2^2 \\ &\Rightarrow \min \left\| [\mathbf{SDX}_{all}] \cdot \overrightarrow{poly} - \overrightarrow{HD_{all}} \right\|_2^2 \end{aligned} \tag{4.34}$$

Im Rahmen dieser Arbeit wird ein Polynom siebten Grades in x- und y-Richtung verwendet, so dass 64 freie Polynomkoeffizienten angelernt werden müssen. Die in Abbildung 4.23 dargestellte Impulsantwort verdeutlicht die verbesserte und vereinheitlichte Trainingsqualität. Zum einen können mit diesem Polynom nahezu beliebig viele Zwischenphasen bestimmt werden, zum anderen ist dieses Polynom kontinuierlich, so dass Artefakte durch unterschiedlich angelernte benachbarte Phasen systembedingt nicht möglich sind. Die hier dargestellte Impulsantwort entspricht derselben Klasse wie die diskrete Impulsantwort in Abbildung 4.10. Vergleicht man diese beiden Impulsantworten, so wird der Unterschied noch deutlicher.

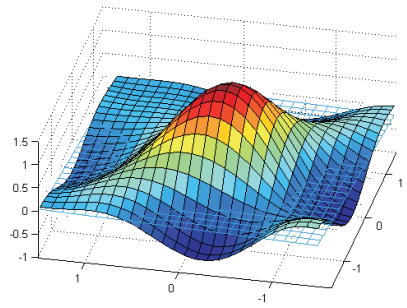


Abbildung 4.23: Impulsantwort des Polynoms für einen Faktor $\frac{8}{3}$

Vorüberlegung zur Einbringung von Nebenbedingungen in das Polynom

Auf Grund der im vorherigen Abschnitt (4.3) vorgestellten Vorteile eines Polynoms und den Verbesserungen bei der MQF-Minimierung, die sich durch die Einführung von Nebenbedingungen ergeben (siehe 4.2.4), wird in den folgenden zwei Abschnitten eine Kombination beider Vorschläge vorgenommen.
Die in Abschnitt 4.2.4 eingeführten Nebenbedingungen können jedoch nicht ohne weiteres für den Polynomansatz verwendet werden.
Dies ist schnell ersichtlich, wenn man bedenkt, dass das Filterpolynom aus mehreren Teilen höherer Ordnung besteht und bei einer Interpolation diese Teile lediglich verschoben zueinander addiert werden. Auf diese Weise verschwinden die einzelnen Anteile nicht vollständig. Für die Interpolation einer homogenen Fläche wäre es aber notwendig, dass die einzelnen höheren Anteile sich zu Null ergänzen. Da dies jedoch nicht möglich ist, besteht ein Polynom, welches dieser Nebenbedingung genügt, nur noch aus dem homogenen Teil (Polynom nullter Ordnung). Es stellt ein Mittelwertfilter dar, welches auf Grund seines Tiefpassverhaltens bei einer Interpolation von Bildern eine geringe Qualität erreicht.
Dies kann auch formal gezeigt werden. Geht man von der vereinfachten Polynomformel

4.29 aus, so würde sich bei einer Interpolation einer homogenen Ebene mit dem Wert q auf der Basis von vier Stützstellen Folgendes ergeben:

$$\begin{aligned} F_{Ipol}(x+\alpha) = &\ q \quad \cdot poly(x+\alpha-1) + q \cdot poly(x+\alpha) \\ + &\ q \quad \cdot poly(x+\alpha+1) + q \cdot poly(x+\alpha+2) \\ = &\ q \quad \cdot (\, a\,(x-1)^3 + b\,(x-1)^2 + c\,(x-1) + d \\ + &\ ax^3 + bx^2 + cx + d \\ + &\ a\,(x+1)^3 + b\,(x+1)^2 + c\,(x+1) + d \\ + &\ a\,(x+2)^3 + b\,(x+2)^2 + c\,(x+2) + d\,) \\ = &\ q \quad \cdot (4ax^3 + (4b+6a)\,x^2 + (4c+4\,b+18a))\,x + (8a+6b+2c+4d) \\ \stackrel{!}{=} &\ q \end{aligned} \tag{4.35}$$

Durch einen Koeffizientenvergleich wird deutlich, dass bei der relativ simplen Forderung eines fehlerfreien Erhalts von homogenen Flächen, das Polynom im Freiheitsgrad auf die Ordnung Null ($a = 0, b = 0, c = 0, d = 1/4$) eingeschränkt wird. Somit würde das Polynom nur aus einem Offset ($1/4$) bestehen und somit ein Rechteckfilter beschreiben.

Dass die im vorherigen Abschnitt beschriebenen Polynome, die nicht dieser Nebenbedingung gehorchen, dennoch eine relativ gute Interpolationsqualität erreichen und keine stark sichtbaren Artefakte produzieren, liegt daran, dass die Polynome durch die MQF-Minimierung und den hohen Freiheitsgrad sich an die Lösung annähern, auch wenn die Lösung (wie oben gezeigt) nicht absolut erreicht werden kann.

In dem Ausdruck 4.35 kann man erkennen, dass die Forderung nach einer fehlerfreien Interpolation höherer Polynome von der Anzahl frei wählbarer Variablen in den einzelnen Filterwerten abhängig ist. So führt im gegebenen Fall die Forderung $4ax^3 = 0$ zur Auslöschung von a. Danach führt der Ausdruck $4b + 6a$ zu einer Auslöschung von b. Dieser Prozess setzt sich bis zum letzten Koeffizienten d fort, so dass durch den letzten Koeffizientenvergleich von x^0 der Ausdruck $(8a + 6b + 2c + 4d) = 1$ dazu führt, dass $d = 1/4$ ist. Wenn beim höchsten Exponenten zwei oder mehr Koeffizienten zur Verfügung stehen, werden die weiteren Koeffizienten nicht automatisch ausgelöscht. Für die Interpolation einfacher Signale muss also bei Polynomen höheren Grades mehr als ein Koeffizient eine Auswirkung auf die Anteile höherer Ordnung haben. Um weitere Koeffizienten für die höheren Polynome vorliegen zu haben, muss das Polynom in mehrere Bereiche unterteilt werden. Dies wird im folgenden Abschnitt dargestellt.

Optimierung eines stückweise definierten Polynoms unter Nebenbedingungen

Wie im vorherigen Abschnitt erläutert, muss ein Polynom, welches den im Rahmen dieser Arbeit eingeführten Nebenbedingungen für eine verbesserte Bildqualität genügt, aus meh-

reren stückweise definierten Teilbereichen bestehen.
Deshalb werden im Rahmen dieser Arbeit die Grenzen des Gesamtpolynoms auf „-2“ bis „+2“ gesetzt und das Polynom in 4 Bereiche (jeweils in x- und y-Richtung) unterteilt, so dass insgesamt 16 verschiedene Bereiche entstehen.
Man erkennt in der Abbildung 4.24 sowohl die äußeren Grenzen, als auch die einzelnen Bereiche, die im Folgenden mit den römischen Ziffern I bis XVI adressiert werden.
Auf Grund des erhöhten Freiheitsgrades durch die Unterscheidung der verschiedenen Bereiche, wird der Grad der Polynome jeweils in x- und y-Richtung auf drei begrenzt. Pro

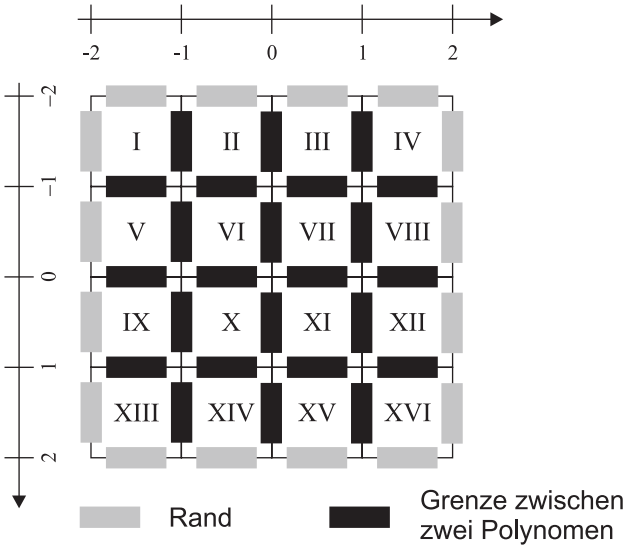


Abbildung 4.24: Stückweise Definition des Polynoms

Abschnitt erhält man also 16 Polynomkoeffizienten. Bei wiederum 16 Bereichen ergeben sich insgesamt 256 verschiedene Koeffizienten, die mit $a_{1...256}$ bezeichnet werden. Der Index des Koeffizienten berechnet sich aus dem Grad von x und y und dem aktuellen Bereich. Beispielsweise steht a_{23} für den II. Abschnitt $(+1 \cdot 16)$, dem Exponenten 1 in y-Richtung $(+1 \cdot 4)$ und dem Exponenten 2 in x-Richtung $(+2 \cdot 1)$. Somit ergibt sich für einen einzelnen Bereich (z. B. I) folgende Formel:

$$\begin{aligned} poly_I(x,y) &= a_{16} \cdot x^3y^3 + a_{15} \cdot^2 y^3 + a_{14} \cdot x^1y^3 + a_{13} \cdot x^0y^3 \\ &= a_{12} \cdot x^3y^2 + a_{11} \cdot^2 y^2 + a_{10} \cdot x^1y^2 + a_9 \cdot x^0y^2 \\ &= a_8 \cdot x^3y^1 + a_7 \cdot^2 y^1 + a_6 \cdot x^1y^1 + a_5 \cdot x^0y^1 \\ &= a_4 \cdot x^3y^0 + a_3 \cdot^2 y^0 + a_2 \cdot x^1y^0 + a_1 \cdot x^0y^0 \end{aligned} \tag{4.36}$$

Dieser kompliziert wirkende Formalismus ermöglicht zum einen eine einfachere Darstellung der nachfolgenden Formeln und zum anderen überhaupt erst die Möglichkeit die Formeln

mittels Softwarelösungen (Maple und Matlab) verarbeiten zu können. Da 256 Koeffizienten zueinander ins Verhältnis gesetzt werden müssen, ist eine manuelle Beschreibung der Koeffizienten nicht mehr möglich.
Um trotz der stückweisen Definition ein kontinuierliches Interpolationspolynom zu erhalten, müssen die Grenzen der einzelnen Bereiche miteinander kombiniert werden.
Hierzu müssen die Polynome an den Schnittgrenzen identisch sein, damit die Funktionswerte an dieser Stelle übereinstimmen. Beispielhaft sei dies an dem Übergang zweier benachbarter Bereiche (I und V) gezeigt. Die Grenze zeichnet sich dadurch aus, dass sie für alle x-Werte (zwischen -2 und -1) definiert ist, jedoch in y-Richtung konstant -1 ist. Auf Grund des konstanten Verlaufs in y-Richtung vereinfachen sich die Gleichungen zu:

$$\begin{aligned} 0 \quad & \stackrel{!}{=} poly_I(x,1) - poly_V(x,1) \\ & = (a_{12} - a_{68} - a_{16} + a_{72} - a_{76} + a_4 + a_{80} - a_8)\, x^3 \\ & + (-a_{15} + a_3 - a_7 + a_{11} + a_{79} - a_{67} - a_{75} + a_{71})\, x^2 \\ & + (-a_{14} + a_{78} - a_{66} + a_2 - a_{74} + a_{10} - a_6 + a_{70})\, x \\ & + (a - a_5 - a_{13} + a_{69} + a_9 + a_{77} - a_{65} - a_{73})\, x^0 \end{aligned} \tag{4.37}$$

Da diese Forderung für jedes x gültig sein muss, erhält man mit Hilfe eines Koeffizientenvergleiches der verschiedenen Exponenten von x vier Gleichungen. Auf diese Weise erreicht man eine Stetigkeit zwischen zwei benachbarten Bereichen. Des Weiteren müssen aber auch die Ableitungen benachbarter Polynome orthogonal zur Grenze identisch sein, um keine abrupten Spitzen in der Gesamtform zu erhalten. Für die oben genannte Grenze zwischen den Bereichen I und V ergibt sich eine Stetigkeitsforderung der Ableitungen der Bereiche in y-Richtung. Die Ableitung eines Bereiches in y-Richtung ergibt:

$$\begin{aligned} poly_I^y(x,y) \quad & = 3a_{16}x^3y^2 + 3\,a_{15}x^2y^2 + 3\,a_{14}xy^2 + 3\,a_{13}x^0y^2 \\ & + 2\,a_{12}x^3y + 2\,a_{11}x^2y + 2\,a_{10}xy + 2\,a_9x^0y \\ & + a_8x^3y^0 + a_7x^2y^0 + a_6xy^0 + a_5x^0y^0 \end{aligned} \tag{4.38}$$

Somit erhält man für den Übergang der Ableitung zwischen den Bereichen I und V:

$$\begin{aligned} 0 \quad & \stackrel{!}{=} poly_I^y(x,1) - poly_V^y(x,1) \\ & = (3\,a^{16} - a^{72} + a^8 + 2\,a^{76} - 2\,a^{12} - 3\,a^{80})\, x^3 \\ & + (a^7 - 2\,a^{11} + 3\,a^{15} - a^{71} + 2\,a^{75} - 3\,a^{79})\, x^2 \\ & + (a^6 - 2\,a^{10} + 3\,a^{14} - a^{70} + 2\,a^{74} - 3\,a^{78})\, x \\ & + (a^5 + 2\,a^{73} - 2\,a^9 - 3\,a^{77} + 3\,a^{13} - a^{69})\, x^0 \end{aligned} \tag{4.39}$$

Auch hier ergeben sich durch einen Koeffizientenvergleich vier Gleichungen. In Abbildung 4.24 sind insgesamt 24 Übergänge zwischen zwei Polynombereichen zu erkennen. Für jeden Übergang ergeben sich vier Gleichungen für die Stetigkeit und vier Gleichungen für die Stetigkeit der ersten Ableitung.

Zusätzlich sind noch 16 nach außen liegende Grenzen eingezeichnet. Für diese Grenzen werden Nebenbedingungen definiert, um die Impulsantwort am Rand auf Null zu setzen. Des Weiteren wird der zentrale Punkt der Impulsantwort $(x, y) = (0, 0)$ auf 1 gesetzt. Hierbei ergeben sich jeweils vier Gleichungen.
Insgesamt kommt man auf folgende Nebenbedingungen:

- 24 Forderungen für die Stetigkeit
- 24 Forderungen für die Stetigkeit der Ableitung
- 16 Forderungen für die äußere Grenze $(\pm 2, y)(x, \pm 2)$
- 1 Forderung für die Stelle $poly(0, 0)$

Dies ergibt $(24 + 24 + 16 + 1) \cdot 4 = 260$ Forderungen nur für die Verbindung der einzelnen Bereiche. Da diese Forderungen jedoch linear voneinander abhängig sind, reduziert sich die Anzahl der Forderungen auf 195. Weil die Anzahl der freien Koeffizienten bei 256 liegt, ist eine Optimierung von 61 freien Parametern erforderlich.
Abbildung 4.25 zeigt die Ergebnisse eines Trainings ohne Nebenbedingungen. Man erkennt deutlich die einzelnen Bereiche, da keine Stetigkeit gegeben ist.

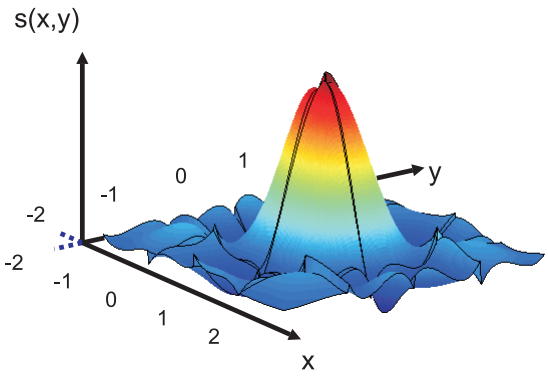


Abbildung 4.25: Impulsantwort eines stückweise definierten Polynoms ohne Nebenbedingungen

In Abbildung 4.26 sind die Ergebnisse eines Trainings mit Nebenbedingungen bezüglich der Stetigkeit zu erkennen.
Die Stetigkeit ist für den gesamten Bereich gegeben. Da keine Stetigkeitsforderung bezüglich der ersten Ableitung existiert, sind deutlich einzelne Spitzen zwischen den Bereichen zu sehen. Erst durch die Einführung der weiteren Nebenbedingungen bezüglich der Stetigkeit der ersten Ableitung und der Einführung von Forderungen für den Rand ergibt sich ein akzeptables Ergebnis (siehe Abbildung 4.27).

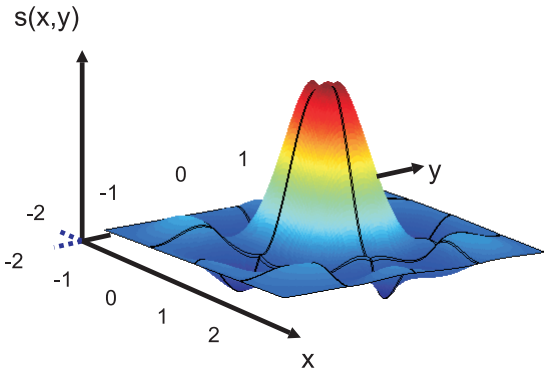


Abbildung 4.26: Impulsantwort eines stückweise definierten Polynoms mit Stetigkeitsbedingungen und Bedingungen für den Rand

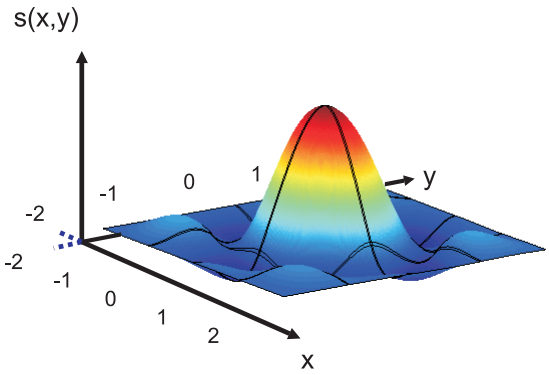
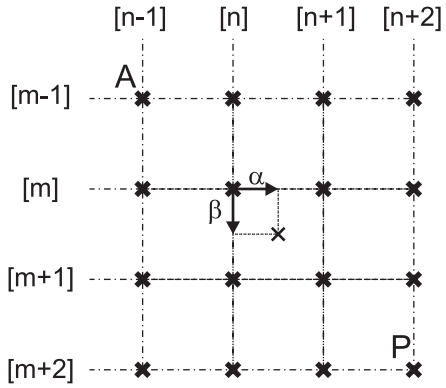
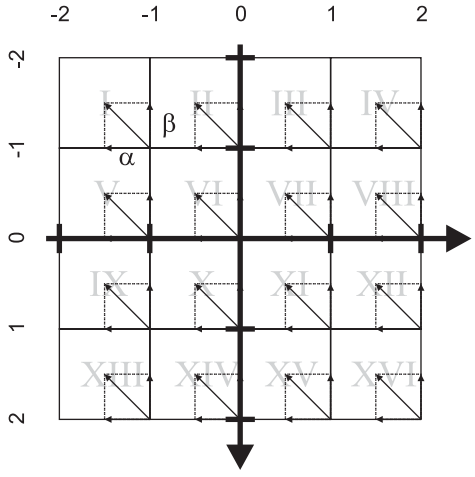


Abbildung 4.27: Impulsantwort eines stückweise definierten Polynoms mit Bedingungen zur Stetigkeit, Stetigkeit der 1. Ableitung und Bedingungen für den Rand

Zur Verbesserung der Bildqualität werden nun die Taylorbedingungen aus Formel 4.24 für das stückweise definierte Polynom transferiert. Hierzu werden für die 16 Filterkoeffizienten $A(\alpha,\beta), B(\alpha,\beta), \ldots P(\alpha,\beta)$ die entsprechenden 16 Polynombeschreibungen eingesetzt. Dabei muss der Zusammenhang zwischen dem Polynom und den tatsächlichen Pixelpositionen berücksichtigt werden. Abbildung 4.28 zeigt im oberen Bereich die Interpolationssituation einer bestimmten Zwischenposition (α,β) auf der Basis von 4x4 Eingangspixeln. Bei der Anwendung eines Interpolationspolynoms ergeben sich in Abhängigkeit der Phasen α und β die Filterkoeffizienten für die einzelnen Pixelwerte wie im unteren Bereich der Abbildung 4.28 dargestellt. Somit ergibt sich z. B. für den Erhalt von Geraden in



(a) Interpolation auf der Basis von 16 Eingangspixeln



(b) Polynome mit 16 Bereichen

Abbildung 4.28: Pixelpositionen im Eingangsbild und Phase des Ausgangspixels

x-Richtung folgender Ausdruck:

$$\begin{aligned}\alpha \stackrel{!}{=} \\ &-poly_I(-1-\alpha,-1-\beta)+poly_{III}(1-\alpha,-1-\beta)+2\cdot poly_{IV}(2-\alpha,-1-\beta)\\ &-poly_V(-1-\alpha,0-\beta)+poly_{VII}(1-\alpha,0-\beta)+poly_{VIII}(2-\alpha,0-\beta)\\ &-poly_{IX}(-1-\alpha,1-\beta)+poly_{XI}(1-\alpha,1-\beta)+poly_{XII}(2-\alpha,1-\beta)\\ &-poly_{XIII}(-1-\alpha,2-\beta)+poly_{XV}(1-\alpha,2-\beta)+poly_{XVI}(2-\alpha,2-\beta)\end{aligned} \tag{4.40}$$

Wie man in der Abbildung 4.28 erkennen kann, geht die Richtung der Phasen α und β negativ in die Positionsbestimmung des Interpolationspolynoms ein. Auf diese Weise verschiebt sich das Polynom positiv über die Eingangspixel. Durch einen Koeffizientenvergleich erhält man (für die Forderung 4.40) 16 Nebenbedingungen, die auf Grund ihrer Größe hier nur schematisch dargestellt werden:

$$\begin{aligned}(128\,a_{256}\ \ldots\ -a_1)\quad &\cdot\alpha^0\cdot\beta^0\stackrel{!}{=}0\cdot\alpha^0\cdot\beta^0\\ (-192\,a_{256}\ \ldots\ +a_2)\quad &\cdot\alpha^1\cdot\beta^0\stackrel{!}{=}1\cdot\alpha^1\cdot\beta^0\\ &\vdots\\ (2\,a_{256}+a_{240}-a_{208}+2\,a_{192}+a_{176}-a_{144}&\\ +2\,a_{128}+a_{112}-a_{80}+2\,a_{64}+a_{48}-a_{16})\quad &\cdot\alpha^3\cdot\beta^3\stackrel{!}{=}0\cdot\alpha^0\cdot\beta^0\end{aligned} \tag{4.41}$$

In die Polynomberechnung werden Nebenbedingungen für den Erhalt von DC und von linearen Verläufen in x- sowie y-Richtung integriert. Es ergeben sich insgesamt 308 Bedingungen, wovon jedoch lediglich 222 Bedingungen linear unabhängig sind. Auf Grund der Größe der Formeln (222 Bedingungen a 256 Koeffizienten) sei an dieser Stelle auf den Anhang B verwiesen. Abbildung 4.29 zeigt die Zusammenfassung aller linear unabhängigen Nebenbedingungen. Mit Hilfe dieser Nebenbedingungen erhält man optimierte Interpolationspolynome, die den Bildqualitätsbedingungen genügen und auf die jeweilige Klasse des vorliegenden Bildinhaltes optimiert sind. In der Abbildung 4.30 wird ein Ergebnis eines Trainings mit diesen Nebenbedingungen dargestellt. Auf Grund der kontinuierlichen Form des Interpolationstiefpasses können quasi beliebige Interpolationsfaktoren berücksichtigt werden und es wird durch die Anpassung ans Trainingsmaterial eine hohe Interpolationsgüte erreicht. Im nachfolgenden Kapitel 5 wird die Interpolationsqualität dieses Verfahrens eingehend untersucht.

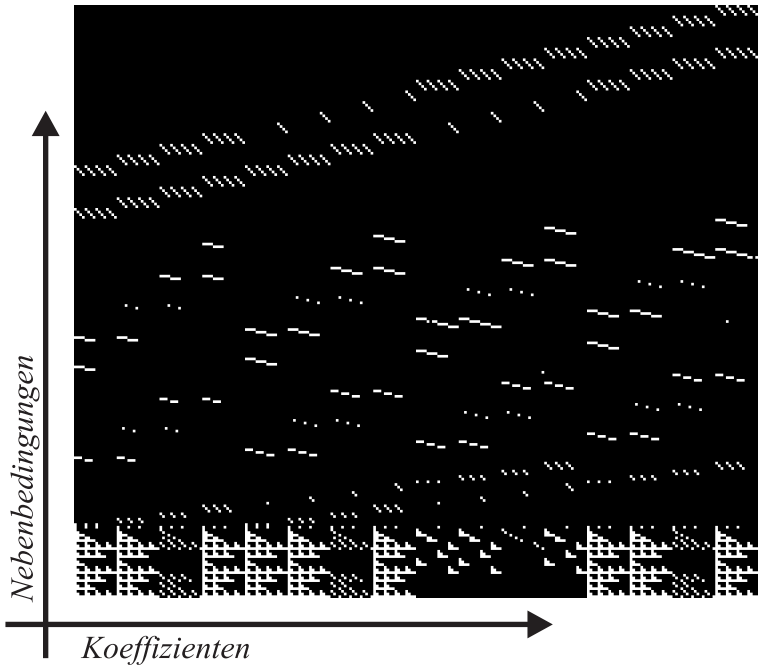


Abbildung 4.29: Graphische Darstellung der Nebenbedingungen des stückweise definierten Polynoms

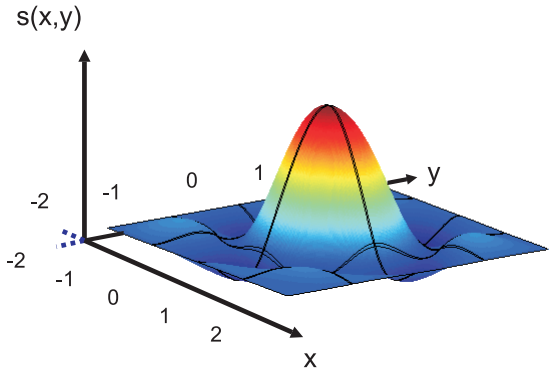
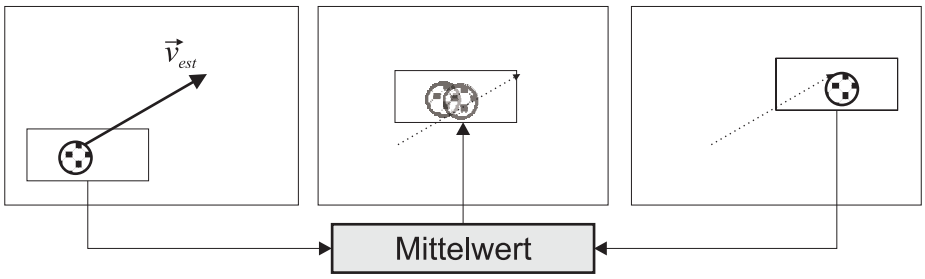
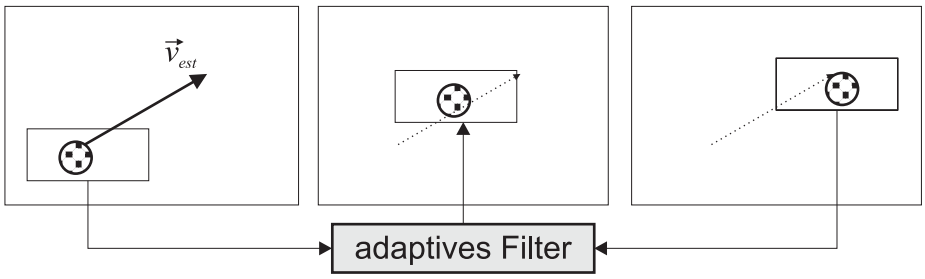


Abbildung 4.30: Impulsantwort eines stückweise definierten Polynoms mit den Stetigkeitsbedingungen (siehe Abbildung 4.27) und zusätzlich den Bedingungen bezüglich des Erhalts von DC, x- und y-Linearverläufen



(a) Einfache Mittelung



(b) Adaptive Filterung

Abbildung 4.31: Adaptive Zwischenbildinterpolation zur Kompensation von Fehlern des Bewegungsschätzers

4.4 Bewegungsfehleradaptive Zwischenbildinterpolation

Die klassifikationsbasierte Interpolation kann auch in die Zeitachse projiziert werden. So erhält man eine Zwischenbildinterpolation, die durch eine Klassifikation von Strukturen in zwei benachbarten Bildern optimale Filter zur zeitlichen Interpolation auswählt. Dies umfasst sowohl Bewegungsinformationen als auch Bildstrukturen.
Die Grundidee des bewegungsfehleradaptiven Verfahrens wird in der Abbildung 4.31 dargestellt. Bei einer einfachen bewegungsvektorgestützten Mittelung der Bildinhalte entstehen durch fehlerhafte Vektoren Doppelkonturen und Unschärfe. Durch eine lokale Analyse des Bewegungsvektorfehlers (in Form des unterschiedlich positionierten Balles im Analysefenster erkennbar) und angepassten Filterkoeffizienten kann der Versatz innerhalb des Analysefensters ausgeglichen werden.

Es gibt in der Literatur einige Ansätze zur zeitlichen Nutzung der klassifikationsbasierten Filterung. So wird in dem Patent von Kondo [KNF+01] für die örtliche Interpolation der Vorschlag gemacht, zwei 3x3 ADRC-Masken in aufeinanderfolgenden Bildern zu nutzen. Somit wird zwar die Klassenzahl stark vergrößert, es können jedoch Fehlklassifikationen

durch Rauschen o. Ä. reduziert werden.

Im Bereich des Deinterlacings wird eine einfache Klassifikation zur Auswahl der verschiedenen Deinterlacingverfahren vorgeschlagen [ZCd05].
An dieser Stelle soll jedoch dieses Verfahren zur Zwischenbildinterpolation benutzt werden. Der Vorteil bzw. die Grundidee einer klassifikationsbasierten Zwischenbildinterpolation liegt darin, dass durch die Klassifikation Unterschiede zwischen den Bildbereichen auf Verschiebungen bzw. Auf- und Verdeckung schließen lassen. Somit können optimale Filter trainiert werden, die eine optimale Interpolation ermöglichen. Auf diese Weise ist es möglich Fehler der Bewegungsschätzung zu kompensieren bzw. eine Bewegungsschätzung zu ersetzen.
Fehler in der Bewegungsschätzung treten trotz verbesserter Verfahren relativ häufig auf. Somit muss bei einer Zwischenbildinterpolation sichergestellt werden, dass derartige Fehler erkannt werden, damit keine stark sichtbaren Artefakte erzeugt werden. Aus dieser Klasse der fehlerkompensierenden Verfahren sind vor allem die gewichteten Medianfilter (siehe [Blu97] und [FS02]) zu nennen, die eine phasenrichtige Interpolation von Kanten, selbst bei Bewegungsschätzfehlern, ermöglichen, sowie Vorschläge ([dH00]) zur nachträglichen Überprüfung der Interpolationsdaten über Medianfilter. Der Unterschied der bisherigen Verfahren zu der hier vorgestellten Methode ist der Erhalt feinster Details selbst bei einer Fehlschätzung der vorliegenden Bewegung.

4.4.1 Aufbau

Die genaue Wirkweise der klassifikationsbasierten Zwischenbildinterpolation wird in Abbildung 4.32 gezeigt. Es wird eine Interpolation des Zwischenbildes an der Phase $n+\alpha$ angenommen. In den Bildern $n-1$ und n ist eine vertikale Kante dargestellt, die sich um $\vec{v}_{real} = 8$ Pixel bewegt. Die Bewegungsschätzung liefert jedoch ein fehlerhaftes Ergebnis von $\vec{v}_{est} = 6$ Pixel.

Eine Interpolation im Zwischenbild an der Stelle $\vec{x}$ basiert also auf fehlerhaften Positionen in den beiden Bildern $n-1$ und n. Durch eine Klassifikation der Bildinhalte (bei z. B. Kreuzmasken und angenommener binärer Klassifikation) kann der Fehler der Bewegungsschätzung (hier 2 Pixel) erkannt werden. Somit können optimale Filterkoeffizienten an dieser Stelle ebenfalls einen horizontalen Versatz von 2 Pixeln kompensieren.
Die Struktur dieses Verfahrens ähnelt dem örtlichen Ansatz. Abbildung 4.33 zeigt den groben Aufbau. Nach der bewegungsvektorgestützen Positionierung der Filtermasken wird eine Klassifikation der aktiven Bildinhalte vorgenommen. Auf der Basis dieser Klasse werden in Abhängigkeit der Interpolationsphase die optimalen Filterkoeffizienten aus einer Datenbank ausgelesen und mit diesen die Bildinhalte der Kreuzmasken gefiltert und das Ergebnis herausgeschrieben.

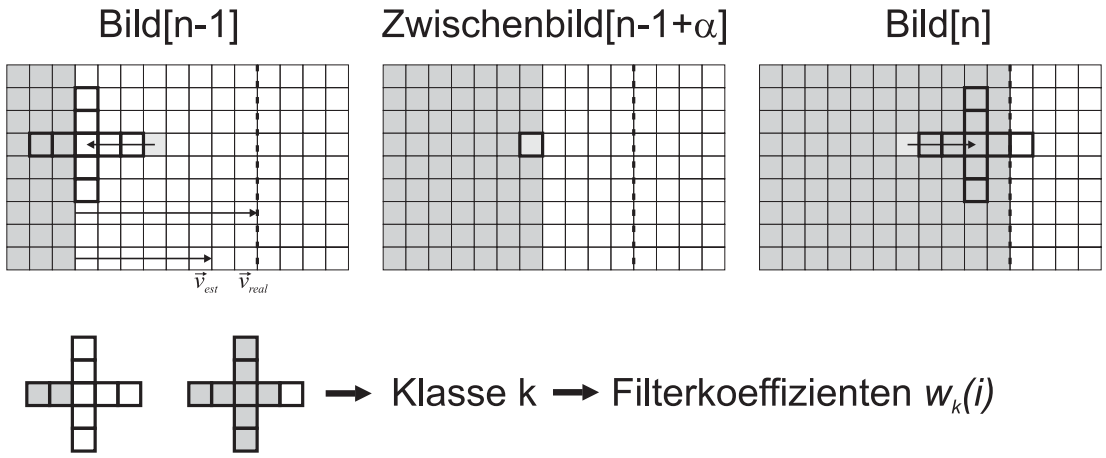


Abbildung 4.32: Grundidee einer klassifikationsbasierten Zwischenbildinterpolation

Kreuzmaske	Pixel	Klassen	reduz. Klassen (Symmetrie)
3x3	10	512	198
5x5	18	131.072	49.796
7x7	26	33.554.432	(nicht bekannt)

Tabelle 4.3: Einfluss der Größe der Kreuzmaske auf die Anzahl der Klassen pro Zeitphase

Weitere Details einer Zwischenbildinterpolation (wie die genaue Berechnung der Phase, die Berücksichtigung von Subpixelgenauigkeit, die Projektion der Bewegungsvektoren) werden hier nicht dargestellt, sondern vielmehr nur das grundsätzliche Prinzip vorgestellt. Diese Details sowie eine Übersicht über verschiedene Zwischenbildinterpolationsverfahren finden sich u. a. bei [FS02].

Klassifikation und Filterform

Für die zeitliche Interpolation sind theoretisch jegliche Filtermaskenformen denkbar. Bei Franzen [FS02] wird gezeigt, dass Kreuzmasken sich sehr gut für die Zwischeninterpolation eignen. Natürlich erfassen Rechteckmasken die Bildsituation genauer, jedoch haben Kreuzmasken bei gleicher Pixelanzahl eine größere Ausdehnung als Rechteckmasken.
Im Rahmen dieser Arbeit werden Kreuzmasken verwendet. Diese werden sowohl für die Klassifizierung als auch für die eigentliche Filterung genutzt.

Bei der Nutzung einer Binarisierung zur Klassifikation (wie bei der örtlichen Interpolation) ergeben Kreuzmasken mit einer Größe von $5x5$ in beiden Bildern insgesamt $2^{2\cdot(2\cdot4+1)}$

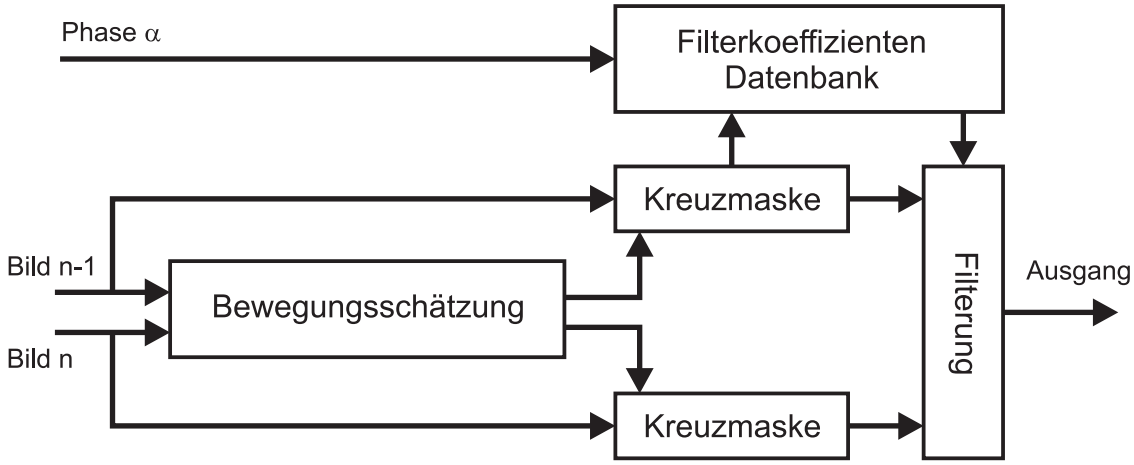


Abbildung 4.33: Struktur der klassifikationsbasierten Zwischenbildinterpolation

verschiedene Klassen. Durch die Berücksichtigung der schwarz/weiß Symmetrie ergibt sich eine Reduktion um den Faktor 2. Insgesamt können in diesem Fall 131.072 verschiedene Fälle unterschieden werden. Bei der Vergrößerung der Maske um ein Pixel in jede Richtung ergeben sich jedoch schon 33,6 Millionen verschiedene Klassen. Diese hohe Anzahl an Klassen kann nicht sinnvoll angelernt werden, so dass im Rahmen dieser Arbeit die Größe der Kreuzmasken auf maximal 5x5 festgelegt wird (siehe auch Tabelle 4.3).

Symmetrie

Um die relativ hohe Zahl der Klassen bei einer 5x5 Kreuzmaske zu reduzieren, werden verschiedene Symmetrien berücksichtigt. Es handelt sich um örtliche Rotationssymmetrien sowie zeitliche Symmetrien.
In Abbildung 4.34 wird die Adressierung der einzelnen Pixel dargestellt. Um Rotationssymmetrien ausnutzen zu können wird eine rotierende Auslesereihenfolge verwendet. Durch eine Modulo-Operation über die vier 45° Richtungspositionen sind die einzelnen Situationen ineinander überführbar. Die folgende Tabelle 4.4 stellt die Binarisierung zweier nahezu identischer Situationen dar, die lediglich um 45° gedreht zu einander stehen. Man erkennt, dass die Einsen nach links wandern und bei Überschreitung eines Viererblockes wieder rechts eingefügt werden. Somit ist diese Auslesereihenfolge sehr implementierungsgünstig. Bei der Rotation muss zusätzlich beachtet werden, dass ebenfalls eine identische Rotation der Filterkoeffizienten vorgenommen wird.
Eine weitere Symmetrieachse ist die Zeit. Durch eine Vertauschung der binarisierten Pixelsituation in den beiden Bildern (Vertauschen der kompletten Masken $1:9$ und $10:18$) erreicht man eine Umkehrung der Bildsituation in der Zeitachse. Dies bedeutet jedoch auch, dass die zeitliche Zwischenbildphase α zu $1-\alpha$ umgedreht wird und dass bei einer

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Situation 1	1	0	1	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0
Situation 2	1	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	1

Tabelle 4.4: Binarisierte Ergebnisse der Kreuzmasken bei gedrehten Bildsituationen

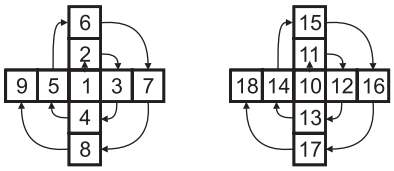


Abbildung 4.34: Auslesereihenfolge der Kreuzmasken

Verwendung dieser Symmetrie die Filterkoeffizienten der gespiegelten Phase benutzt werden.

4.4.2 Training

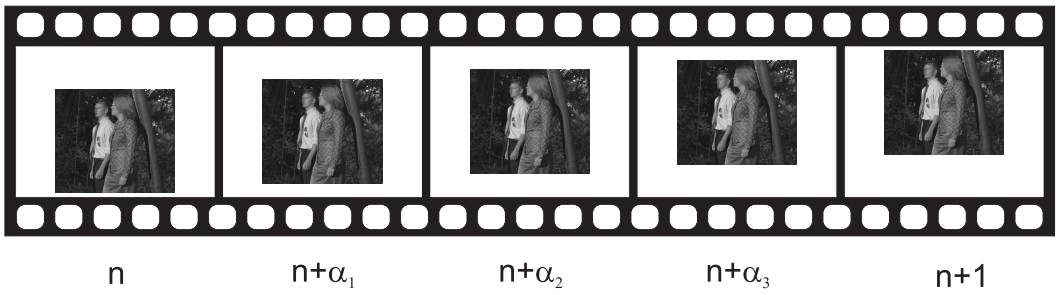
Ähnlich wie bei der örtlichen Interpolation müssen die Filterkoeffizienten vor der Anwendung angelernt werden. Wie beim örtlichen Ansatz werden diese mit Hilfe von hochaufgelösten und niedrig aufgelösten Sequenzen trainiert, wobei mit der Auflösung die Anzahl der Zwischenbilder gemeint ist. Um Filter für bestimmte Bewegungsfehler vorgeben zu können, werden künstliche Sequenzen erzeugt. In diesen Sequenzen werden statische Bildinhalte kontrolliert in x- und in y-Richtung verschoben. Auf diese Weise entstehen Sequenzen, mit denen jeder detektierbare Bewegungsfehler angelernt werden kann. Um den Einfluss der Bewegungsschätzung beim Training auszuschließen, wird im Training auf eine Bewegungsschätzung verzichtet und es werden lediglich Trainingssequenzen mit Bewegungen benutzt, die von den Masken detektiert werden können.

Auf der Basis der zeitlich „hochaufgelösten“ Trainingssequenz wird durch Verwerfen von Zwischenbildern eine zeitlich „niedrigaufgelöste“ Trainingssequenz erzeugt. Auf der Basis dieser beiden Sequenzen wird das Training analog zum örtlichen Prozess mit Hilfe einer Optimierung des mittleren quadratischen Fehlers vorgenommen.

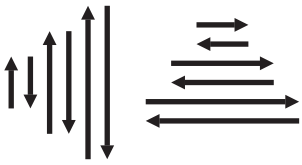
4.4.3 Verbesserungen

Glättung der Filterwerte

Bei der Anwendung der klassifikationsbasierten Zwischenbildinterpolation zeigt sich, dass die Genauigkeit der detektierten Bewegungsfehler durch Rauschen und Auf- und Verde-



(a) Verschiebung von Bildinhalten für eine vierfache Interpolation



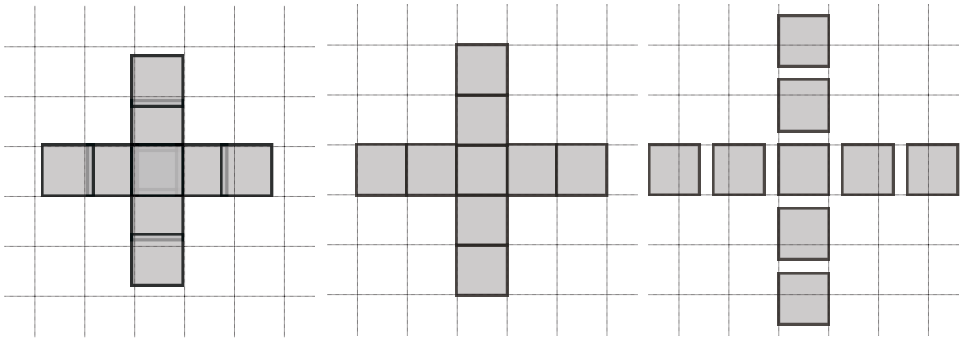
(b) Zu trainierende Bewegungsrichtungen

Abbildung 4.35: Erstellung von Trainingssequenzen für die Zwischenbildinterpolation

ckungsprobleme gestört werden kann. Im interpolierten Bild macht sich dies in Form von leichten Artefakten bemerkbar. Da für jedes Pixel eine eigene Klassifikation vorgenommen wird, können für benachbarte Pixel durch Störungen völlig unterschiedliche Klassen detektiert werden. Da die Bewegungsschätzung zumeist blockbasiert ist und bewegte Objekte in der Regel eine größere Ausdehnung besitzen, wird zur Verbesserung der Interpolation eine Glättung der Filterkoeffizienten vorgenommen. Dazu wird in einem lokalen Bereich (3x3) eine Mittelung über die empfohlenen Filterkoeffizienten vorgenommen. Auf diese Weise wird das Fehlerrauschen stark reduziert.

Aufweitung der Interpolationsmaske, Modifikation der Klassifikation

Wie im vorherigen Abschnitt dargestellt wird, ist die Ausdehnung der Filtermaske stark begrenzt. In vielen Fällen ist diese Ausdehnung nicht notwendig, in anderen reicht sie nicht aus. Nach Vorschlägen von Franzen [FTS01] wird an dieser Stelle ebenfalls eine Adaptierung der Maskenweite vorgenommen. Somit wird in Bereichen, in denen der Bewegungsvektor sehr wahrscheinlich fehlerfrei ist, die Maskenausdehnung durch Überabtastung verringert und in Bereichen, in denen die Bewegungsvektoren sehr wahrscheinlich fehlerhaft sind, durch eine Unterabtastung eine Vergrößerung der Maskengröße erreicht (siehe auch Abbildung 4.36). Als einfaches Maß für die Genauigkeit der Bewegungsvektoren kann der Fehlerwert des Bewegungsschätzers (z. B. die absolute oder quadratische



(a) Verkleinerte Maske (b) Keine Veränderung (c) Vergrößerte Maske

Abbildung 4.36: Unterschiedliche Ausdehnung der Maske

Differenz zwischen dem aktuellen Bildblock und dem durch die Bewegungsschätzung zugeordneten Bildblock) verwendet werden.

5 Simulationsergebnisse und Vergleich der Interpolationsverfahren

In diesem Kapitel wird die Qualität der klassifikationsbasierten Interpolationsverfahren (im Folgenden $Adapol_{Phase}$ und $Adapol_{Poly}$ genannt) im Detail analysiert. Zuerst wird auf die Auswirkungen der im vorherigen Kapitel vorgeschlagenen Optionen und Parameter eingegangen und anschließend ein direkter Vergleich der klassifikationsbasierten mit anderen üblichen Verfahren vorgenommen.
Im zweiten Abschnitt wird die in den Zeitbereich transferierte adaptive Interpolation (*Adapol*) mit alternativen Zwischenbildinterpolationsverfahren verglichen.
Bevor auf die einzelnen Ergebnisse eingegangen wird, folgt im nächsten Abschnitt zuerst eine Beschreibung der verschiedenen Bewertungsmethoden.

5.1 Bewertungsmethodik

Für die Bewertung der subjektiven wie objektiven Qualität wurden mehrere Untersuchungen und Messungen vorgenommen, die in den nächsten drei Unterabschnitten vorgestellt werden.

5.1.1 Testumgebung und Sequenzen

Die Testumgebung besteht algorithmusbedingt aus zwei Teilen. Zum einen das Trainieren von Filterkoeffizienten und zum anderen die Anwendung der Filterkoeffizienten.

Für das Training der Filterkoeffizienten wird zur besseren Vergleichbarkeit jeweils von derselben Trainingssequenz BIGSEQ ausgegangen (siehe auch Appendix A.2). Die Sequenz wurde für diese Arbeit aus einer Vielzahl verschiedener Einzelbilder zusammengestellt und beinhaltet somit eine sehr hohe Zahl an verschiedenen Bildmotiven. Daher eignet sie sich sehr gut zum Training von Filterkoeffizienten für eine Interpolation von natürlichem Bildmaterial. Die Bilder wurden in einer Art Collage zu einer Sequenz der Größe 720x576 (willkürlich gewählt) kombiniert. Diese Sequenz wird im Folgenden, wenn nicht gesondert

angegeben, immer als hochaufgelöste Version genutzt.
Auf der Basis dieser Sequenz werden zum Training unterschiedlicher Filterkoeffizientensätze niedrigaufgelöste Sequenzen erzeugt. Zur phasenkorrekten Dezimation wird ein bikubisches Interpolationsfilter in Kombination mit einem 11 Tap FIR Tiefpass verwendet (siehe auch Kapitel 4.2.2).
Für die Einbindung von Artefaktreduktionsmöglichkeiten bzw. Enhancementverfahren müssen die hochaufgelöste und die niedrigaufgelöste Sequenz modifiziert werden. Dieser Vorgang wird später im Detail beschrieben.

Die im Rahmen dieser Arbeit zu trainierenden und zu analysierenden Interpolationsfaktoren bewegen sich im Bereich der SD→HD-Konversion. Es wird sowohl der ganzzahlige Faktor 2 untersucht, sowie der nicht ganzzahlige Faktor $\frac{8}{3}$, der für eine Interpolation von 720 auf 1920 horizontale Bildpunkte steht.
Für jeden dieser Faktoren müssen die Trainingskoeffizienten gesondert angelernt werden. Um die Wirkung der einzelnen Verbesserungen besser nachvollziehen zu können, wird möglichst nur ein Optimierungsparamter verändert und die anderen Einflüsse werden konstant gehalten bzw. in ihrer Wirkung vermindert.
Nach dem Training wird mit Hilfe von objektiven und subjektiven Messungen die Qualität der trainierten Filterkoeffizienten an Hand der Sequenzen Football, Wheel, Lena und Testchart überprüft.

5.1.2 Objektive Bewertung

Als objektives Maß wird in vielen Fällen der mittlere quadratische Fehler zwischen einer Referenzsequenz und der interpolierten Sequenz verwendet. Sofern dieser Wert auf die Maximalaussteuerung (hier 255 bei 8 Bit) des Bildes bezogen wird, spricht man vom PSNR (Peak Signal to Noise Ratio):

$$PSNR[dB] = 10 \cdot \log \left[\frac{255^2}{\sum\limits_{x,y \in Bild} \left(F_{int}(x,y) - F_{ref}(x,y)\right)^2} \right] \tag{5.1}$$

Der PSNR Wert ist jedoch nur für starke Störungen und Unterschiede zwischen Sequenzen sinnvoll nutzbar. Für eine feine Unterscheidung von leichten Bildstörungen kann der PSNR nicht genutzt werden, zumal nur der akkumulierte Gesamtfehler, nicht jedoch die tatsächliche Sichtbarkeit der Störung beachtet wird.
Der Schwierigkeit PSNR-Werte für die Bewertung einer örtlichen Interpolation nutzen zu können, wird am folgenden Beispiel (Abbildung 5.1) deutlich: Die Abbildung zeigt eine Messung der PSNR Werte für die Testsequenzen für drei verschiedene Verfahren (klassifikationsbasierte Polyphaseninterpolation ohne Nebenbedingungen

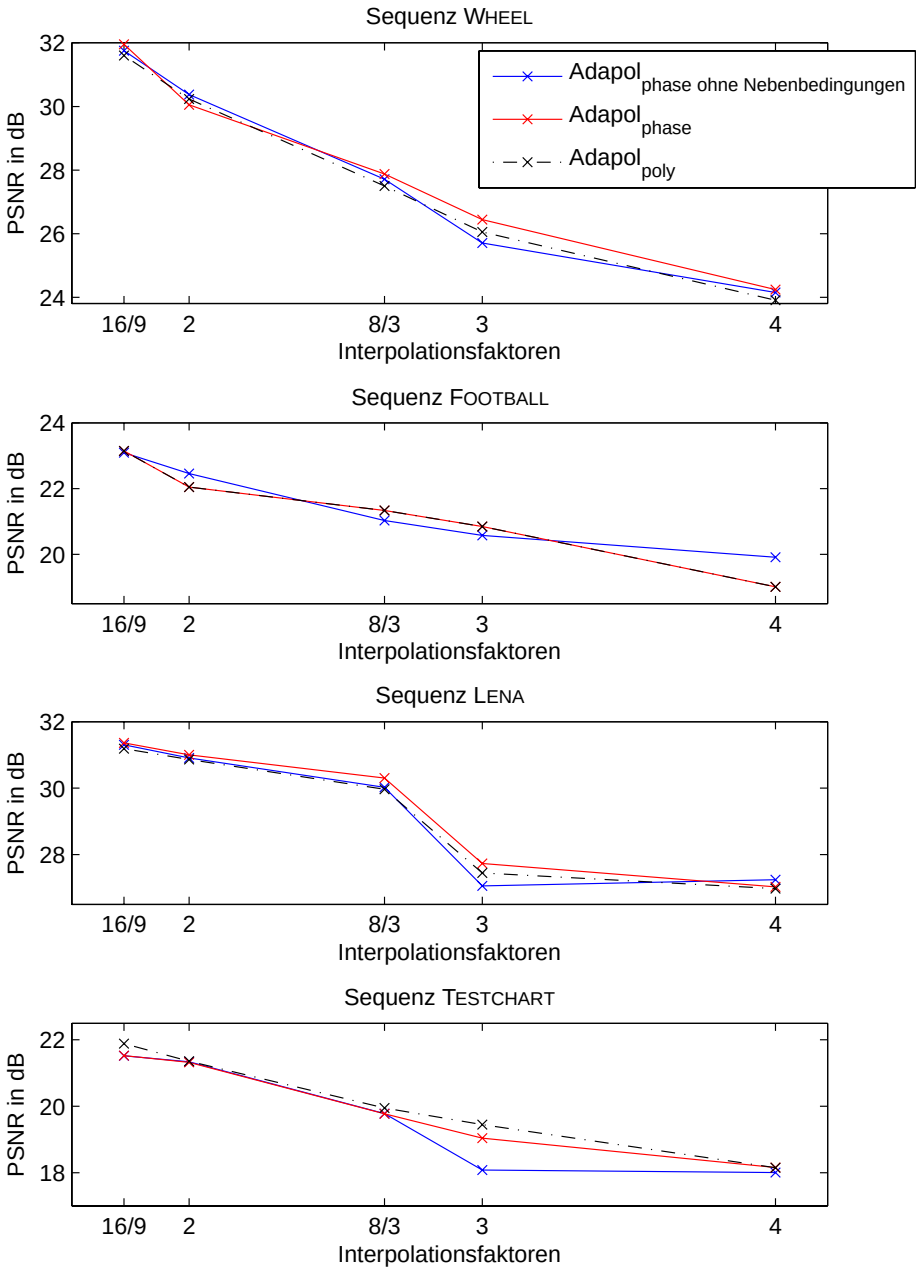
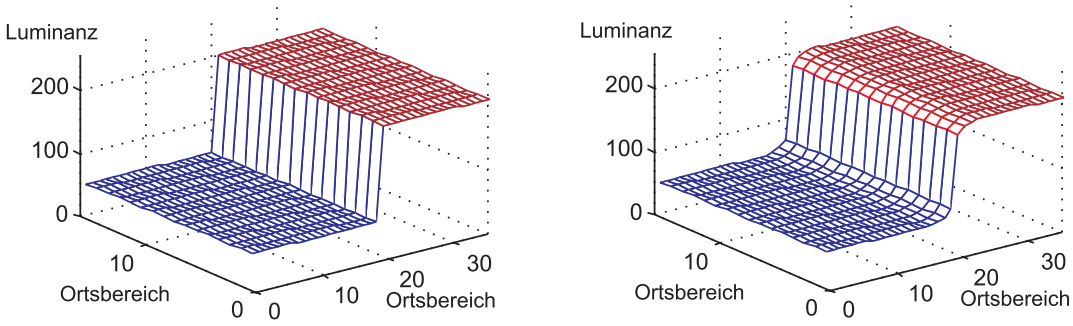


Abbildung 5.1: PSNR des Vergleiches Originalsequenz zu unterschiedlich interpolierten Sequenzen

$Adapol_{phase\,ohne\,Nebenbedingungen}$, mit Nebenbedingungen $Adapol_{phase}$, und die Nutzung eines Polynoms, welches ebenfalls mit Nebenbedingungen optimiert wurde $Adapol_{poly}$). Auf der x-Achse sind die Interpolationsfaktoren aufgeführt und in y-Richtung ist der jeweilige PSNR Wert zu erkennen. Um für die Messung eine Referenz zur Verfügung stehen zu haben, wurden die Testsequenzen (FOOTBALL (Fb), WHEEL (W), LENA (L) und TESTCHART (T)) vor der Interpolation um die jeweiligen Faktoren verkleinert und anschließend mit der Originalsequenz verglichen.
Bei Betrachtung der PSNR Werte fällt auf, dass die Unterschiede zwischen den einzelnen Sequenzen bzw. den Interpolationsfaktoren viel höher sind als die Unterschiede zwischen den eigentlichen Interpolationsmethoden.
Aus diesem Grund kann der PSNR nicht sinnvoll für eine Bewertung genutzt werden. Stattdessen werden im Folgenden neben der subjektiven Analyse weitere objektive Maße eingeführt.
Für die Qualität eines Interpolationsfilters sind neben den spektralen Eigenschaften vor allem die Kantensteilheit sowie die Überschwinger von großer Bedeutung. Um den Einfluss auf diese beiden Größen im Bildbereich messen zu können, wird die künstlich erzeugte Testsequenz TESTCHART genutzt. Es existieren zwei Testfelder (horizontal und vertikal) mit unterschiedlich stark gefilterten Helligkeitssprüngen (von 50 auf 200, bei 8 Bit). Abbildung 5.2 zeigt den Helligkeitsverlauf des ungefilterten (harte Kante) und des gefilterten Helligkeitssprungs (weiche Kante).



(a) ungefilterter Sprung, harte Kante (b) gefilterter Sprung, weiche Kante

Abbildung 5.2: Zwei Ausschnitte der Sequenz TESTCHART mit jeweils einem Helligkeitssprung

Um die Wirkung der Verfahren bzw. Filterkoeffizienten auf Kanten zu bestimmen, wird die Sequenz TESTCHART mit folgendem Messverfahren auf Steilheit und Überschwinger der vorliegenden Kanten untersucht. Bei der Analyse von Signalsprüngen ist es üblich nicht die

gesamte Sprunghöhe sondern stattdessen den Bereich von 10% bis 90% der Sprunghöhe zu analysieren. Mit den aus der Synthese der Testsequenz bekannten Helligkeitswerten $S_{low} = 50$ und $S_{high} = 200$ werden die Werte $S_{10\%}$ und $S_{90\%}$ berechnet und anschließend ihre jeweiligen Positionen subpixelgenau bestimmt. Mit den so gewonnenen Δt und ΔS wird die Kantensteilheit über ihren Quotienten $\frac{\Delta S}{\Delta t}$ näherungsweise bestimmt.

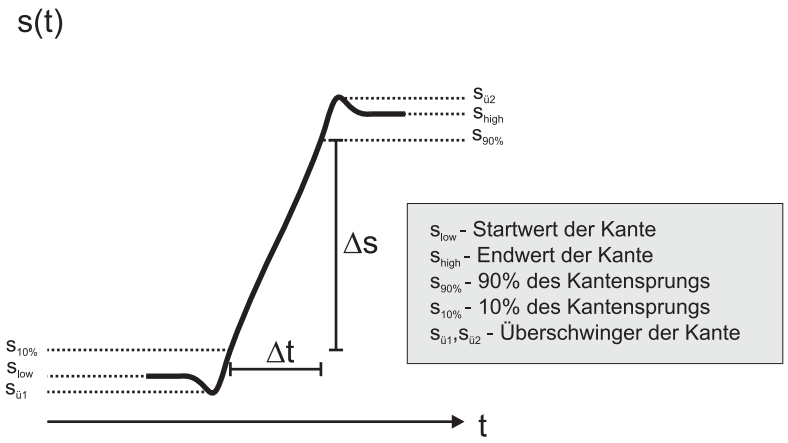


Abbildung 5.3: Messung der Steilheit von Kanten und der Größe von Überschwingern

Neben der Kantensteilheit werden auch die Amplituden der ersten Überschwinger $S_{\text{ü}1}$ und $S_{\text{ü}2}$ festgestellt und der am meisten abweichende Wert auf die Sprunghöhe normiert:

$$max(|S_{\text{ü}2} - S_{high}|, |S_{low} - S_{\text{ü}1}|) \tag{5.2}$$

Die in der Sequenz TESTCHART vorhandenen Kantensprünge ergeben eine Kantensteilheit von 150 Luminanzwerten / Pixel für den ungefilterten Sprung (im Folgenden harte Kante) und eine Steilheit von 71 Luminanzwerten / Pixel für den gefilterten Sprung (weiche Kante).

Da eine objektive Messung der Wirkung der Verfahren auf Texturen und hochfrequenten Anteilen nicht möglich ist, erfolgt die objektive Bewertung lediglich in Form einer Betrachtung zweier Testmuster und der Suche nach sichtbaren Artefakten. Die Testmuster bestehen aus allen horizontalen und vertikalen Frequenzen bis π. Sofern Artefakte nur bei einer bestimmten Frequenz bzw. Eingangssituation auftreten, werden sie in vielen Fällen durch dieses Testmuster sichtbar gemacht.

5.1.3 Subjektive Bewertung

Neben den objektiven Messgrößen ist vor allem die subjektive Messung der Interpolationsqualität durch eine genaue subjektive Bildanalyse notwendig. Auf Grund der großen Anzahl an unterschiedlichen Verfahren und Parametern, vor allem aber auch der in vielen Fällen nicht in einem Test zu berücksichtigen Größen (wie z. B. Flexibilität bei der Anwendung) wird auf einen subjektiven Test nach ITU-R 500 (siehe [ITU00]) verzichtet. Stattdessen werden die Einflüsse der Verbesserungen oder auch die Unterschiede in der Qualität zwischen den Verfahren an Hand ausgewählter Bildausschnitte vorgestellt. Einerseits wird die subjektive Bildschärfe analysiert und andererseits auch auf störende Sichtbarkeit von Überschwingern sowie Artefakten geachtet.
Auf Grund der mitunter sehr feinen Unterschiede in den Ergebnissen der einzelnen Verfahren, ist die Beurteilung der Bilder in gedruckter Form schwierig. Für die Analyse der natürlichen Bilder dieses Kapitels wird eine Betrachtung mit einem Bildschirm empfohlen.

Neben der Bewertung im Bildbereich wird an einigen Stellen auch die Beeinflussung des Eingangsspektrums als Indiz für die Bildqualität benutzt. Auf diese Weise lässt sich der Einfluss auf Wiederholspektren sowie die spektrale Erweiterung anschaulich visualisieren. Hierbei wird der Logarithmus der Absolutwerte einer zweidimensionalen Fouriertransformation der Bilder als Helligkeit dargestellt, da dies die vom menschlichen visuellen System wahrgenommene Filterwirkung der Interpolationsverfahren am besten darstellt.

5.2 Bewertung von Eigenschaften der klassifikationsbasierten Interpolation

In diesem Abschnitt werden die Auswirkungen der im Kapitel 4.2.3 vorgeschlagenen Optionen der klassifikationsbasierten Interpolation hinsichtlich der Interpolationsqualität untersucht.

5.2.1 Überprüfung der Anzahl der Trainingswerte

Auf Grund der unterschiedlichen Anzahl von Trainingswerten pro Bildklasse kann es bei der Erzeugung von Filterkoeffizienten passieren, dass nicht genügend Trainingswerte zum Anlernen einer Klasse vorhanden sind. Somit führt das Training entweder zu einer singulären Matrix oder aber jedoch zu einer Lösung, die eine lokale Lösung nur für diese Trainingsdaten erreicht.

Leicht modifizierte Bildsituationen können zu stark abweichenden Filterergebnissen führen. Dies ist in den Abbildungen 5.4 und 5.5 deutlich zu erkennen. Auf der linken Seite sind die Interpolationsergebnisse eines Trainingsprozesses zu erkennen, bei dem nicht ausreichendes Trainingsmaterial zur Verfügung stand. Es sind unabhängig von der Testsequenz sehr viele Artefakte zu erkennen. Durch eine Mindestanzahl der Trainingswerte (hier 1024 Werte, siehe auch Seite 52) kann bei ansonsten identischen Bedingungen eine starke Verbesserung der Filterkoeffizienten erreicht werden, da keine Artefakte mehr zu sehen sind.
Weil die nicht gelösten Filterkoeffizienten durch die Werte einer nichtadaptiven HRS Funktion ersetzt werden, ergibt sich natürlich - bei so wenig Trainingsmaterial - eine starke Reduktion der Bildschärfe.

Abbildung 5.6 zeigt deutlich den Rückgang der Kantensteilheit (siehe 5.1.2) durch die Überprüfung. Man erkennt bei 50 Trainingswerten dieselbe Kantensteilheit wie beim HRS, da viele Klassen mit Hilfe der HRS Funktion vorbesetzt werden. Aber schon ab ca. 2.500 Trainingswerten ist die Kantensteilheit der auf Mindestanzahl geprüften Filterkoeffizienten auf dem selben Niveau wie die unveränderte Version.
Die Überprüfung stellt also sicher, dass bei zu wenig Trainingsmaterial die Filterkoeffizienten ersetzt und somit keine Artefakte erzeugt werden. Sofern jedoch genügend Trainingsmaterial vorhanden ist, beeinträchtigt diese Forderung die Qualität der Filterkoeffizienten nicht.

5.2.2 Verifizierung der angelernten Daten

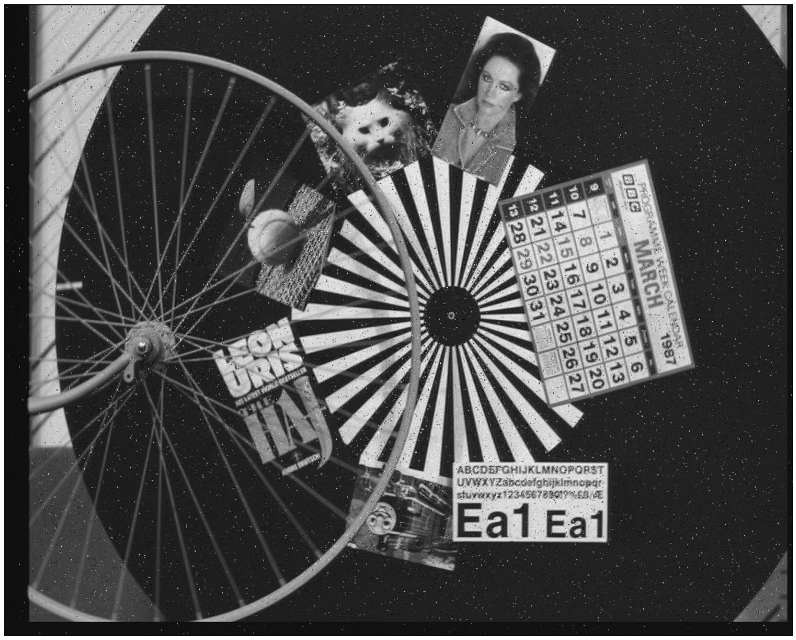
Eine weitere Analyse der Filterqualität findet durch eine nachträgliche Überprüfung der antrainierten Filterkoeffizienten statt(siehe auch Seite 55).

In der Abbildung 5.7 werden die Ergebnisse einer Interpolation mit derartig überprüften Filterkoeffizienten dargestellt. Man erkennt im direkten Vergleich zur unbehandelten Version (siehe Abbildung 5.4), dass die meisten der Interpolationsartefakte scheinbar verschwunden sind. Auf Grund der relativ hoch angesetzten Schwelle für Werte der Filterkoeffizienten, die noch als richtig gelten, werden einige Filterklassen, die nicht optimal angelernt sind, nicht ersetzt. So entstehen leichte Artefakte, wie auf der rechten Seite von 5.7 in den vergrößerten Ausschnitten zu erkennen ist. Im Gegensatz zur Überprüfung der Klassenanzahl, bleibt die subjektive Bildschärfe hier jedoch weitestgehend erhalten, da nur sehr wenige Klassen tatsächlich ersetzt werden.
Die gleichbleibende Bildschärfe lässt sich auch durch die gemessene Kantensteilheit (siehe Abbildung 5.8) stützen. Die Bildschärfe bleibt unabhängig von der Anzahl an Trainingsdaten konstant.

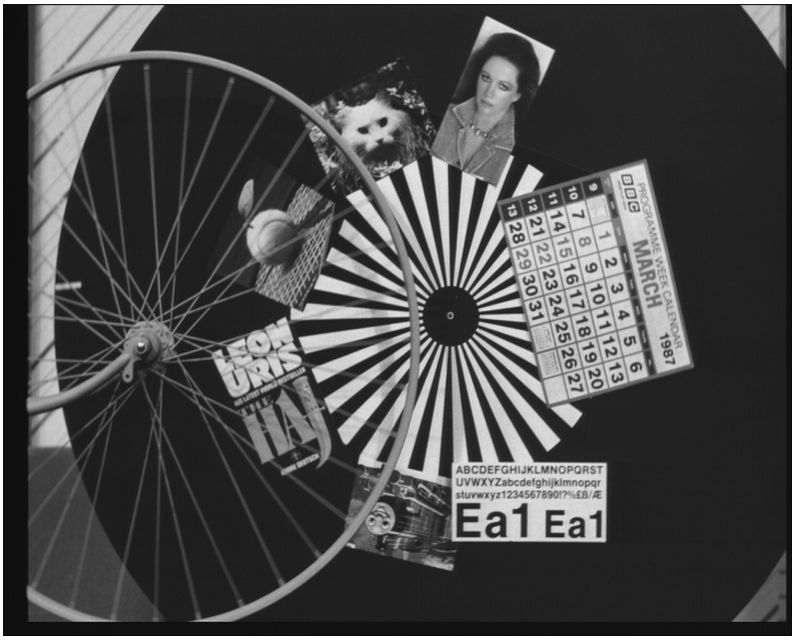
(a) Training ohne Überprüfung der Klassenanzahl

(b) Training mit Überprüfung der Klassenanzahl

Abbildung 5.4: Einfluss einer Mindestanzahl an Trainingswerten pro Klasse, Sequenz FOOTBALL, Interpolationsfaktor $\frac{8}{3}$, Training mit ∅ 50 Werten pro Klasse



(a) Training ohne Überprüfung der Klassenanzahl



(b) Training mit Überprüfung der Klassenanzahl

Abbildung 5.5: Einfluss einer Mindestanzahl an Trainingswerten pro Klasse, Sequenz WHEEL, Interpolationsfaktor $\frac{8}{3}$, Training mit ∅ 50 Werten pro Klasse

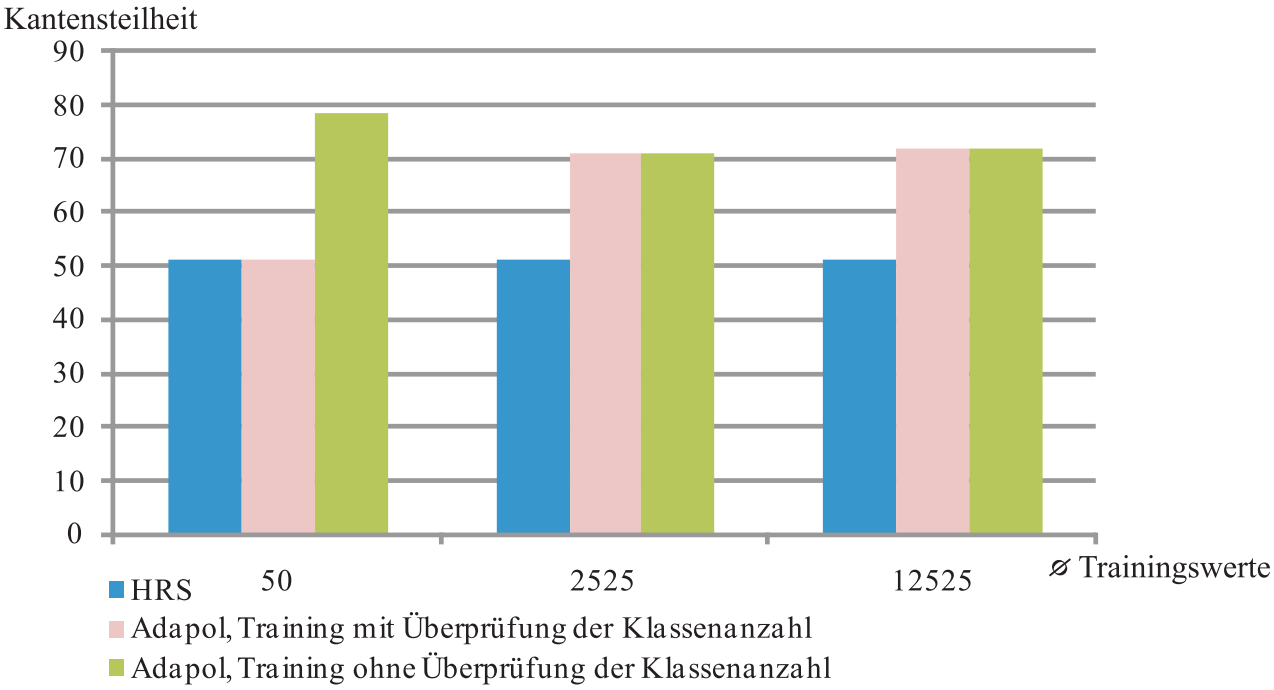
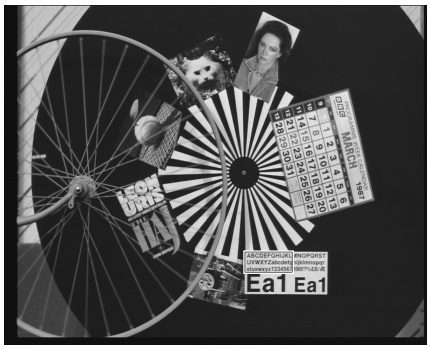


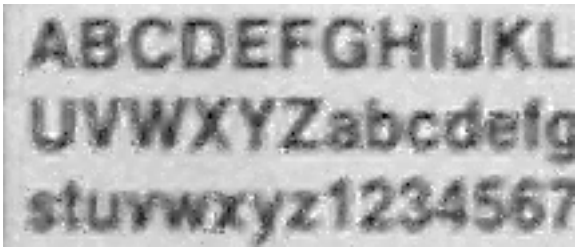
Abbildung 5.6: Einfluss einer Mindestanzahl an Trainingswerten pro Klasse auf die Kantensteilheit bei einem Interpolationsfaktor $\frac{8}{3}$ für unterschiedliche Anzahl an ⌀ Trainingswerten pro Klasse

(a) Sequenz FOOTBALL, Interpolationsfaktor $\frac{8}{3}$ Training mit ∅ 50 Werten pro Klasse

(b) Ausschnitt der Sequenz FOOTBALL, Interpolationsfaktor $\frac{8}{3}$ Training mit ∅ 50 Werten pro Klasse



(c) Sequenz WHEEL, Interpolationsfaktor $\frac{8}{3}$ Training mit ∅ 50 Werten pro Klasse



(d) Ausschnitt der Sequenz WHEEL, Interpolationsfaktor $\frac{8}{3}$ Training mit ∅ 50 Werten pro Klasse

Abbildung 5.7: Einfluss einer Filterkoeffizientenüberprüfung bei wenig Trainingsmaterial

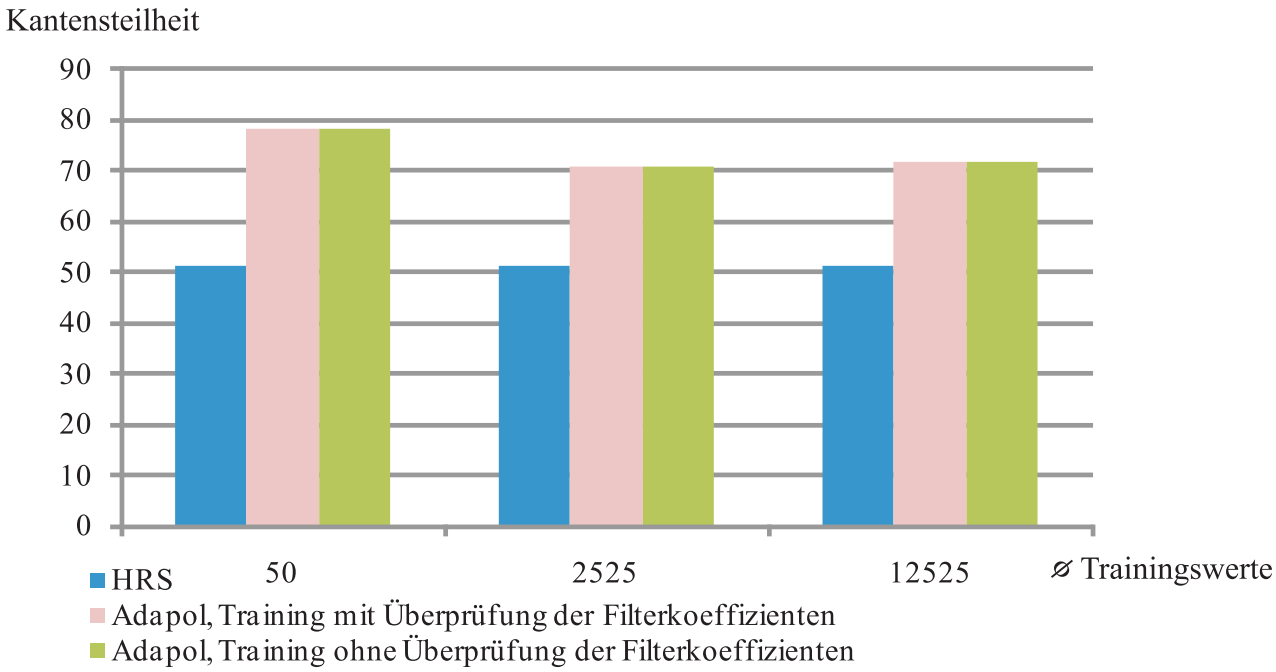


Abbildung 5.8: Einfluss einer nachträglichen Koeffizientenüberprüfung auf die Kantensteilheit bei einem Interpolationsfaktor $\frac{8}{3}$ für unterschiedliche Anzahl an ⌀ Trainingswerten pro Klasse

Beide bisher dargestellten Verbesserungen filtern Klassen heraus, die mit unzureichendem Trainingsmaterial angelernt wurden. Somit sind diese Forderungen sehr ähnlich und filtern viele identische Klassen heraus.
Trotz der Überschneidungen ergänzen sich beide Überprüfungen, wie am in Abbildung 5.9 gezeigten Beispiel deutlich zu sehen ist.
Die Artefakte durch unzureichendes Trainingsmaterial können von beiden Verfahren stark unterdrückt werden, wobei in beiden Ergebnissen noch kleinere Artefakte auftreten. An dieser Stelle kann man auch sehen, dass die Filterkoeffizientenüberprüfung zwar einen besseren Schärfeerhalt erreicht, jedoch viele Artefakte im Bild nicht vermeiden kann.
Erst durch die Kombination beider Forderungen werden diese noch weiter unterdrückt (hier z. B. an den kleinen Impulsstörungen im Auge erkennbar).

5.2.3 Einfluss der Maskengröße

Durch die Änderung der Maskengröße wird zum einen die Ausdehnung der Filter, zum anderen aber auch die Anzahl der Klassen drastisch verändert (siehe auch Seite 55).
Eine Vergrößerung bewirkt jedoch nur einen relativ moderaten Gewinn der Kantenschärfe (siehe auch Abbildung 5.10). Hauptsächlich macht sich dieser Gewinn bei sehr steilen Kanten bemerkbar. Bei vorgefilterten Kanten sind keine steileren Kanten festzustellen.

Bei Artefakten ist eine gegenläufige Entwicklung sichtbar. Die Störartefakte, die durch eine fehlende Unterdrückung der Wiederholspektren entstehen, können durch eine Vergrößerung der Masken zwar deutlich verringert werden (siehe Abbildung 5.11), bei einer genauen Analyse der Ergebnisse (z.B. am Bildschirm) erkennt man jedoch noch einige wenige Artefakte.
Durch eine weitere Vergrößerung der Maske (und damit auch der unterscheidbaren Eingangssituationen) wird das Verhältnis der Trainingsdaten zu den anzulernenden Klassen immer kleiner, und ein fehlerfreies Anlernen der Klassen wird immer schwieriger. In Abbildung 5.12 verbessert sich im sehr hochfrequenten Bereich zwar die Interpolationsqualität beim Sprung von 3 x 3 auf 4 x 3, bei einer noch größeren Maske (5 x 3) steigt jedoch wieder die Unschärfe. Auch im niederfrequenten Bereich ist eine Zunahme von Artefakten durch die Vergrößerung der Maskengröße festzustellen.
Für eine 4 x 4 Maske ist die Anzahl an Klassen doppelt so hoch wie bei einer 5 x 3 Maske (siehe Tabelle 4.1). In diesem Fall reicht der Trainingsprozess mit der hier verwendeten Sequenz BIGSEQ nicht mehr aus, um benutzbare Filterkoeffizienten zu erstellen.
Betrachtet man diese Erkenntnisse bezogen auf den ebenfalls exponentiellen Mehraufwand an Speicher (siehe Tabelle 4.1) und den quadratisch zunehmenden Mehraufwand an Rechenoperationen, ist eine Vergrößerung der Maske unter jetzigen zur Verfügung stehenden Rechenressourcen nicht zu empfehlen.

(a) Ohne zusätzliche Fehlerbehandlung

(b) Verbesserung durch Mindestklassen

(c) Verbesserung durch Koeffizientenüberprüfung

(d) Verbesserung durch Kombination von Mindestklassen und Koeffizientenüberprüfung

Abbildung 5.9: Ausschnitt der Sequenz LENA, Interpolationsfaktor $\frac{8}{3}$ Training mit $\varnothing$ 50 Werten pro Klasse

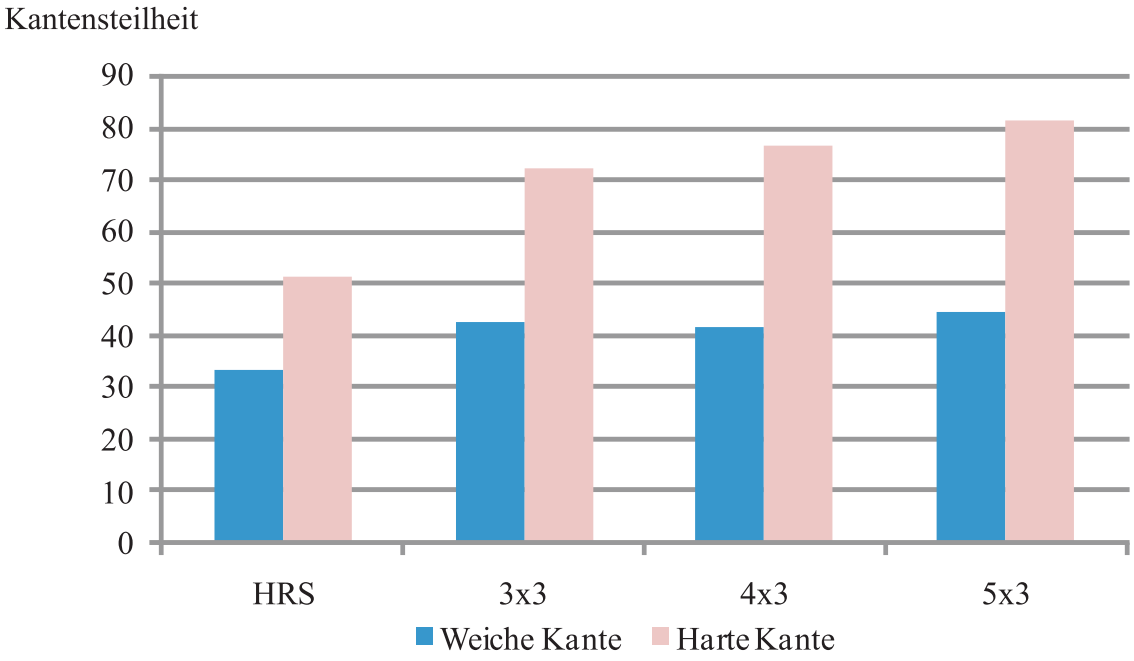
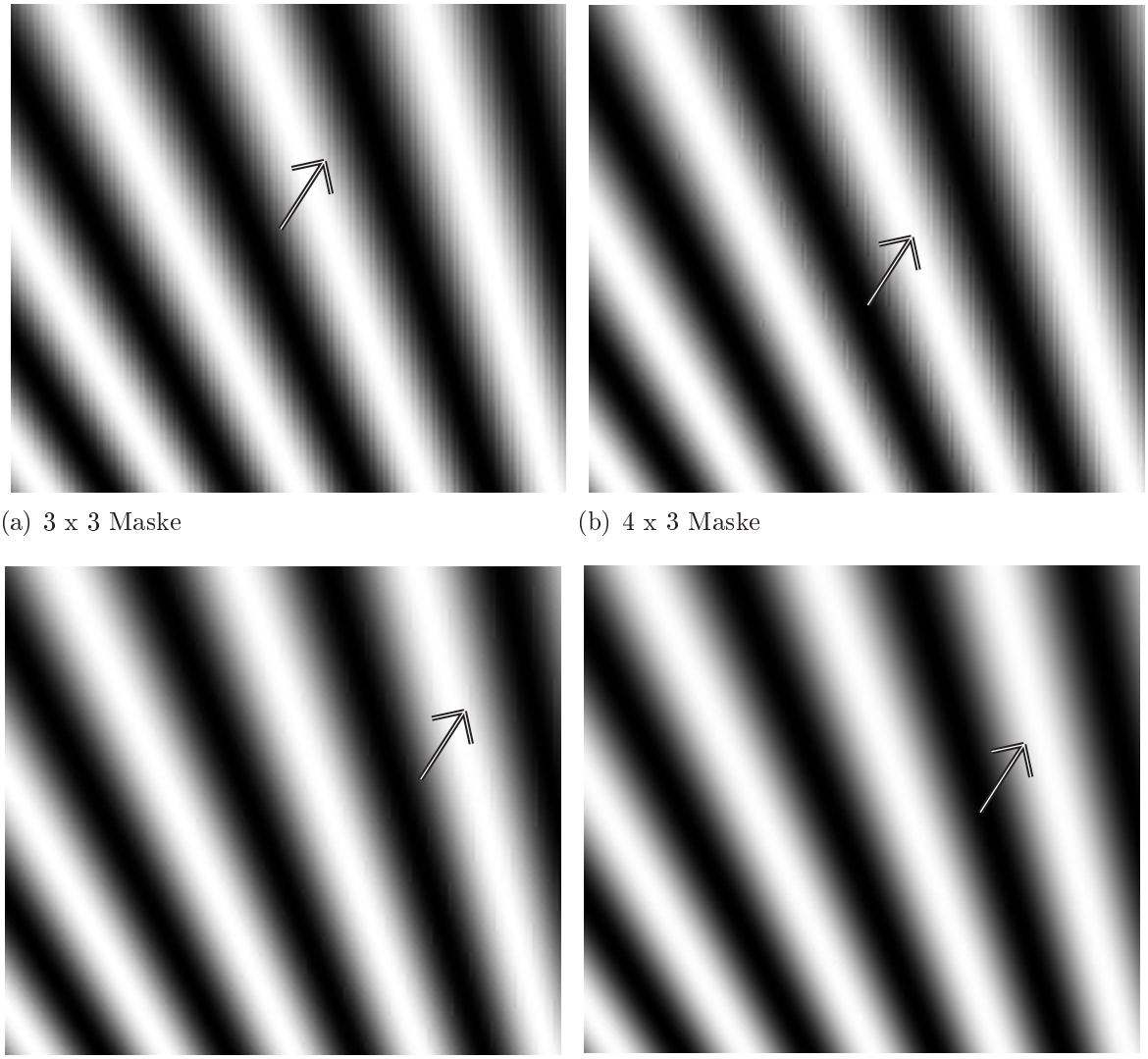


Abbildung 5.10: Einfluss der Maskengrößen auf die Bildschärfe



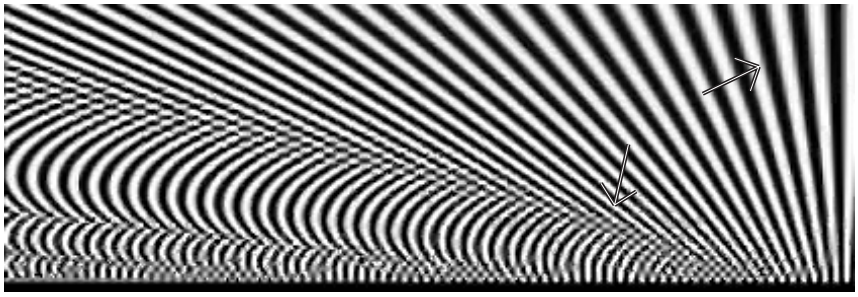
(a) 3 x 3 Maske

(b) 4 x 3 Maske

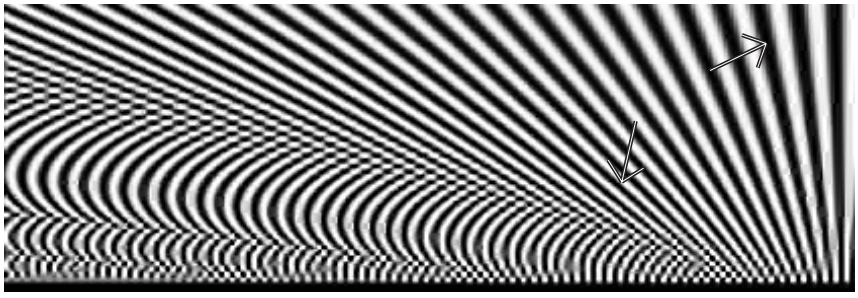
(c) 5 x 3 Maske

(d) 3 x 3 Maske, jedoch mit Taylornebenbedingungen

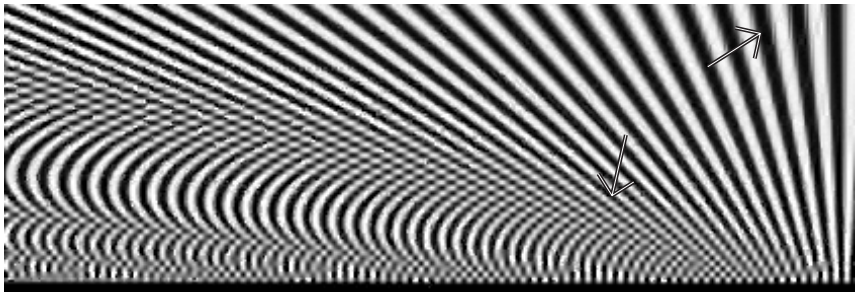
Abbildung 5.11: Ausschnitt der Sequenz TESTCHART, Interpolationsfaktor $\frac{8}{3}$



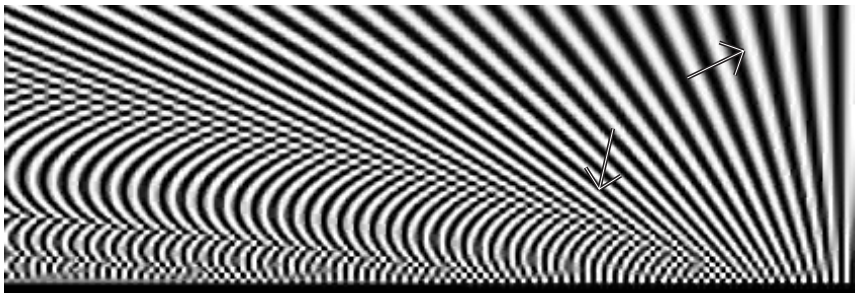
(a) 3 x 3 Maske



(b) 4 x 3 Maske



(c) 5 x 3 Maske



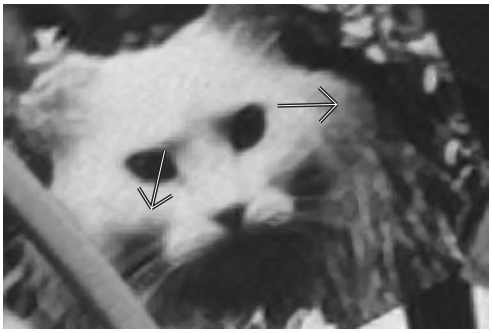
(d) 3 x 3 Maske, jedoch mit Taylornebenbedingungen

Abbildung 5.12: Ausschnitt der Sequenz TESTCHART, Interpolationsfaktor $\frac{8}{3}$

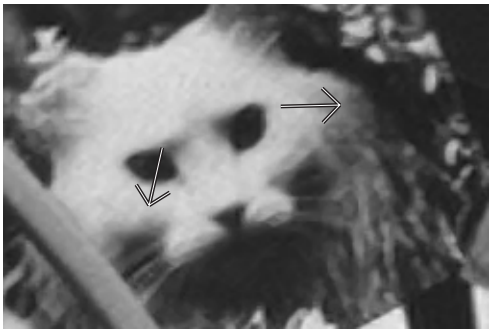
5.2.4 Einführung von Nebenbedingungen

Durch eine Vergrößerung der Filtermasken erreicht man zwar eine verbesserte Unterdrückung von Störspektren, benötigt hierfür jedoch einen großen Aufwand durch die erhöhte Klassenzahl.
Um eine ausreichende Unterdrückung der Störspektren zu erreichen bieten sich die im Rahmen dieser Arbeit eingeführten Taylornebenbedingungen an (siehe auch Seite 63).

Um auch bei einer 3x3 Maske einen ausreichenden Freiheitsgrad zu haben, werden im Rahmen dieser Arbeit lediglich die ersten beiden Taylornebenbedingungen genutzt (nullte und erste Ordnung für die x- und y-Richtung), da hiermit schon eine starke Reduktion der Artefakte möglich ist. In den Abbildungen 5.11(d) und 5.12(d) sieht man die Reduktion der Streifenartefakte. Vor allem in den niederfrequenten Bereichen ergibt sich ein deutlich homogenerer Bildeindruck.
Aber auch bei Betrachtung von natürlichen Bildausschnitten (siehe Abbildung 5.13) kann man den Gewinn durch die Einführung der Taylornebenbedingungen erkennen. Auf der linken Seite sind die ursprünglichen Ergebnisse und auf der rechten Seite die Ergebnisse bei identischen Verhältnissen, jedoch unter Berücksichtigung der Nebenbedingungen zu sehen. Der Unterschied wird vor allem bei Betrachtung der Bilder auf einem Bildschirm deutlich. Man bemerkt in den Bildausschnitten, die mit den Filterkoeffizienten mit Nebenbedingungen interpoliert wurden, einen leichten subjektiven Schärfeverlust. Die Filter haben nur eine begrenzte Ausdehnung (z. B. 3 x 3) und der Lösungsraum wird durch die Nebenbedingungen deutlich verkleinert. Da nun die scharfen Lösungen, die in homogenen Bereichen zu Artefakten führen, ausgeschlossen werden, ergibt sich eine etwas unschärfere Lösung. Diese - subjektiv empfunden - leicht reduzierte Bildschärfe kann auch durch die Bestimmung der Kantensteilheit bestätigt werden (Abbildung 5.14). Bei sehr starken Kanten erreichen die Filterkoeffizienten mit und ohne Nebenbedingungen dieselbe Performance. Bei weichen Kanten ist jedoch eine reduzierte Schärfung zu erkennen. Dies zeigt die Grenzen der Optimierbarkeit bei einer 3 x 3 Maske. Da die weichen Kanten eine größere Sprungausdehnung als 3 Pixel haben, können systembedingt entweder Artefakte mit den Taylornebenbedingungen unterdrückt werden, oder eine leichte Bildschärfung unter Zulassung der Artefakte erreicht werden.
Neben dem im vorherigen Kapitel dargestellten Ursprung der Artefakte im Spektrum durch unzureichende Unterdrückung von Wiederholspektren, zeigt die Abbildung 5.15 die Zusammenhänge der Bildung der Artefakte mit der vorliegenden Filterausdehnung und dem gewünschten Schärfegrad im Ortsbereich.
Auf der linken Seite ist ein Helligkeitssprung mit geringer Ausdehnung dargestellt. Bei der Filterausdehnung von 3 Koeffizienten wäre nun ein Kantenverlauf wünschenswert, wie er mit „geschärfte Kante“ dargestellt ist. Unter Berücksichtigung der Taylornebenbedingungen ergibt sich jedoch ein nicht so scharfer Kantenverlauf. Wird nun ein längerer



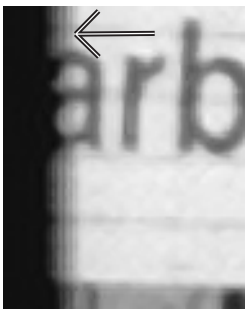
(a) Ausschnitt der Sequenz WHEEL ohne Nebenbedingungen



(b) Ausschnitt der Sequenz WHEEL mit Nebenbedingungen

(c) Ausschnitt der Sequenz LENA ohne Nebenbedingungen

(d) Ausschnitt der Sequenz LENA mit Nebenbedingungen



(e) Ausschnitt der Sequenz FOOTBALL ohne Nebenbedingungen

(f) Ausschnitt der Sequenz FOOTBALL mit Nebenbedingungen

Abbildung 5.13: Verschiedene Ausschnitte von Sequenzen mit natürlichem Bildinhalt, jeweils mit und ohne Taylornebenbedingungen interpoliert, Interpolationsfaktor $\frac{8}{3}$, Training mit $\varnothing$ 12525 Werten pro Klasse

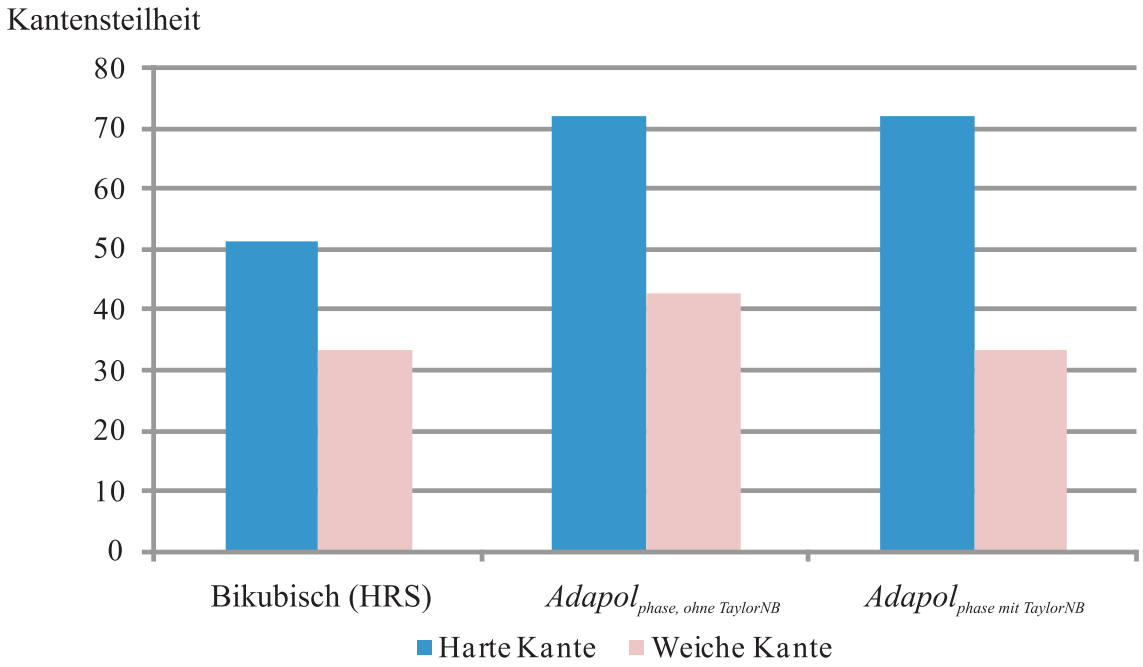
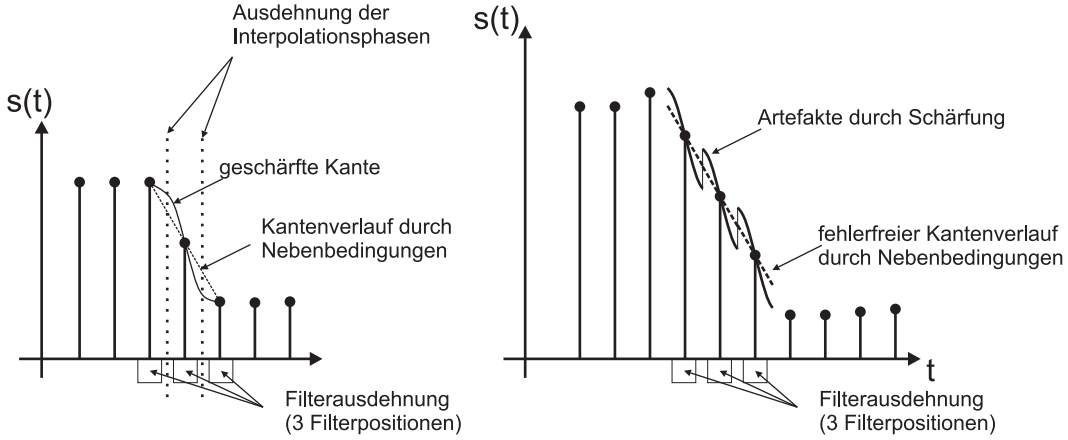


Abbildung 5.14: Einfluss der Nebenbedingungen auf die Bildschärfe

Kantenverlauf interpoliert, wie es auf der rechten Seite dargestellt ist, so werden auf Grund der begrenzten Filterausdehnung die vermuteten Enden des Kantenverlaufs in die Nähe approximiert. Bei dem langen Kantenverlauf wiederholt sich die Interpolationssituation der linken Seite mehrmals. Somit führen die besonders scharfen Filterkoeffizienten zu einem unruhigen Signalverlauf mit kleinen Sprüngen. Ein nicht so scharfer Kantenverlauf (durch Nebenbedingungen) führt zu einem glatten Verlauf der ausgedehnten Kante.



(a) kurze Kante (b) länger ausgedehnte Kante

Abbildung 5.15: Örtliche Darstellung der Zusammenhänge Filterausdehnung, Artefakte und Bildschärfe

5.2.5 Globale Klassifikation

Für eine verbesserte Interpolationsqualität macht es Sinn, die Filterkoeffizienten auf das Eingangsmaterial anzupassen. Da das Trainingsmaterial neben den bisher genannten Bedingungen und Überprüfungen einen großen Einfluss auf die Filterkoeffizienten hat, werden zwei Filterkoeffizientensätze mit modifiziertem Trainingsmaterial erzeugt.
Um die leichten Abstriche in der Bildschärfe durch die Taylornebenbedingungen zu kompensieren wird zum einen ein Enhancementkoeffizientensatz erzeugt. Hierzu wird die niedrigaufgelöste Trainingssequenz unverändert gelassen und die hochaufgelöste Trainingssequenz mit einem Peakingalgorithmus [dH00] zusätzlich geschärft.
Um gestörtes Bildmaterial qualitativ hochwertig interpolieren zu können wird die niedrigaufgelöste Trainingssequenz mit 30 dB additivem weißen Rauschen versehen und die hochaufgelöste Sequenz mit einem 3x3 Medianfilter von zu feinen (nicht anlernbaren) Details gefiltert.

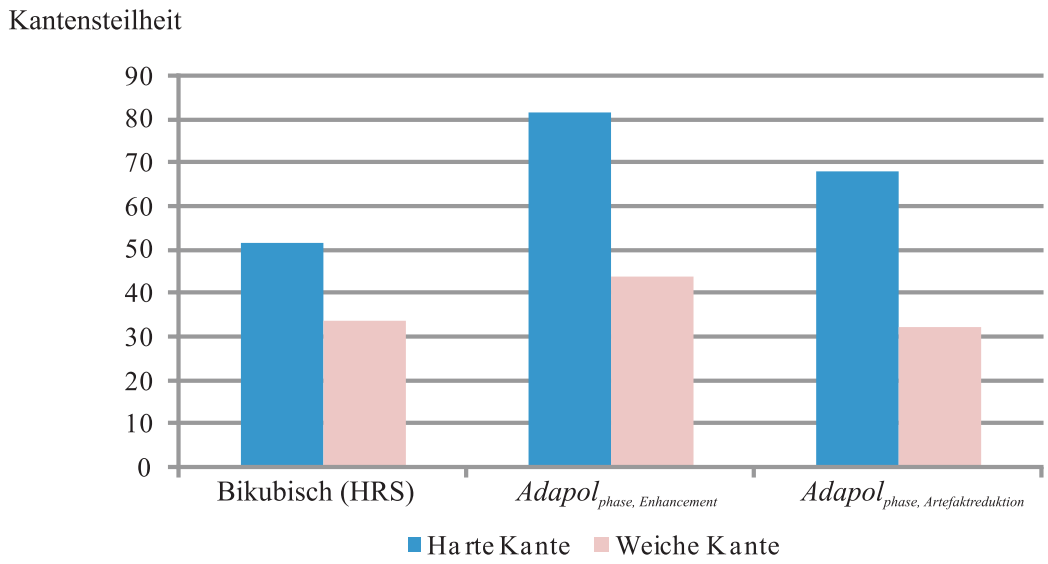
Im Folgenden wird der Einfluss dieser beiden Datensätze (Enhancement- und Artefaktreduktionskoeffizienten) auf unterschiedliches Eingangsmaterial dargestellt. Abbildung 5.17 zeigt die Interpolationsergebnisse der Sequenz Lena. Die Enhancementversion erreicht bei artefaktfreiem Eingangsmaterial eine sehr hohe Interpolationsgüte. Das interpolierte Bild ist scharf und beinhaltet viele Details. Die Interpolation mit Artefaktreduktionskoeffizienten erhält zwar auch die wesentlichen Kanten im Bild (siehe z. B. die Hutkrempe oder die Ränder der Pupille), viele kleinere Details werden jedoch unterdrückt. Zum besseren Vergleich ist noch eine bikubische HRS Interpolation dargestellt. Hier sind zum einen Treppenartefakte im Bereich der Hutkrempe und ein insgesamt schlechterer Schärfeeindruck (siehe auch hier z. B. die Hutkrempe oder die Ränder der Pupille) als bei den beiden adaptiven Verfahren zu erkennen.

In der Abbildung 5.18 ist die Qualität bei artefaktbehaftetem Eingangsmaterial zu sehen (Sequenz Lena mit 30 dB Rauschen). Das Rauschen ist so stark, dass es schon in der Eingangssequenz deutlich bemerkbar ist. Bei der Interpolation mit Enhancementkoeffizienten werden diese Artefakte enorm verstärkt, so dass die Artefakte störend wirken. Die nichtadaptive Variante (bikubische HRS Interpolation) verstärkt das Rauschen zwar nicht, im Vergleich zum Eingangsbild wird jedoch keine nennenswerte Rauschreduktion erreicht.

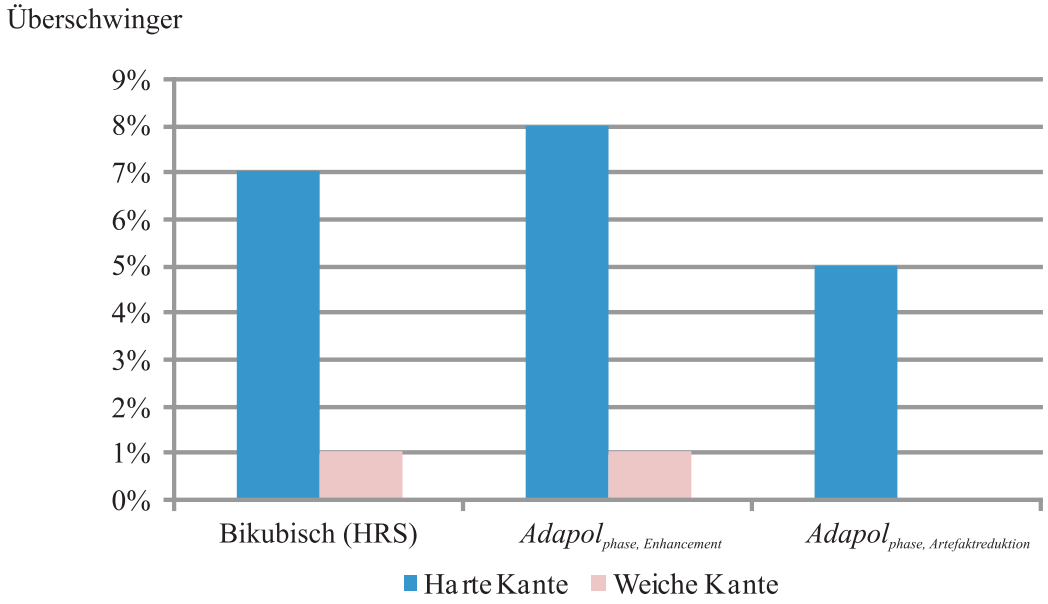
Eine deutliche Verbesserung der Bildqualität stellt sich durch die Verwendung des Artefaktreduktionskoeffizientensatzes ein. Die Rauschstörungen sind merklich reduziert. Gleichzeitig werden die wichtigsten Konturen scharf erhalten (Hutkrempe und Pupillenränder).

Die verstärkte Schärfung durch den Enhancementdatensatz im Gegensatz zum „normalen“ Ergebnis lässt sich auch in einer gestiegenen Kantensteilheit messen (siehe Abbildung 5.16). Sowohl bei einer harten als auch bei einer weichen Kante sind Steigerungen festzustellen. Die Ergebnisse der Artefaktreduktionskoeffizienten sind fast identisch mit denen des „normalen“ Trainings. Dies lässt sich damit erklären, dass bei der Messung der Kantensteilheit eine einzelne ausgeprägte Kante gemessen wird. Somit wird der Detailverlust, der zur Artefaktunterdrückung genutzt wird, an dieser Stelle nicht berücksichtigt.
Der Einfluss auf die Überschwinger ist relativ moderat. Trotz der signifikanten Verbesserung der Kantensteilheit beim Enhancement, steigt der Überschwinger nur von 7% auf 8%.

Somit existieren zwei optimierte Datensätze. Der eine Datensatz wird für qualitativ hochwertiges Material eingesetzt und erreicht dabei eine sehr hohe Interpolationsgüte. Der andere Datensatz eignet sich für eine Anwendung von artefaktbehaftetem Eingangsmaterial, bei dem die Interpolation zum größten Teil aus einem Erhalt der signifikanten Konturen bei gleichzeitiger Unterdrückung der Störungen besteht.



(a) Kantensteilheit



(b) Überschwinger

Abbildung 5.16: Einfluss des Trainingsmaterials auf die Kantensteilheit

(a) Originalausschnitt

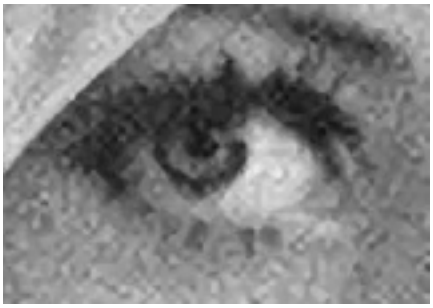
(b) adaptive Interpolation mit Enhancementkoeffizienten

(c) adaptive Interpolation mit Artefaktreduktionskoeffizienten

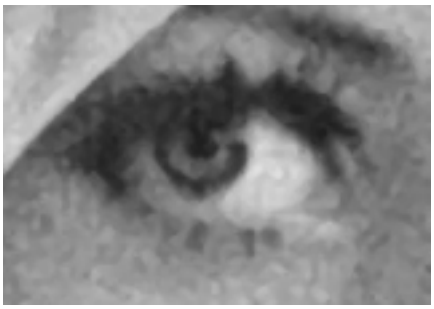
(d) bikubische HRS Interpolation

Abbildung 5.17: Interpolation der Sequenz LENA ohne Rauschen, mit Taylornebenbedingungen trainierte Koeffizienten, Interpolationsfaktor $\frac{8}{3}$, Training mit $\varnothing$ 12525 Werten pro Klasse

(a) Originalausschnitt



(b) adaptive Interpolation mit Enhancementkoeffizienten



(c) adaptive Interpolation mit Artefaktreduktionskoeffizienten

(d) bikubische HRS Interpolation

Abbildung 5.18: Interpolation der Sequenz LENA mit 30 dB Rauschen, mit Taylornebenbedingungen trainierte Koeffizienten, Interpolationsfaktor $\frac{8}{3}$, Training mit $\varnothing$ 12525 Werten pro Klasse

5.3 Vergleich der klassifikationsbasierten Interpolation mit Standardverfahren

Das im Rahmen dieser Arbeit vorgestellte klassifikationsbasierte Polyphasen Interpolationsverfahren ($Adapol_{phase}$, Abschnitt 4.2), sowie die polynombasierte klassifikationsbasierte Interpolation ($Adapol_{poly}$) aus Abschnitt 4.3 werden mit anderen üblichen Standardverfahren (wie Bilinear, HRS, Diamond und NEDI, siehe Kapitel 2) hinsichtlich ihrer Bildqualität verglichen.

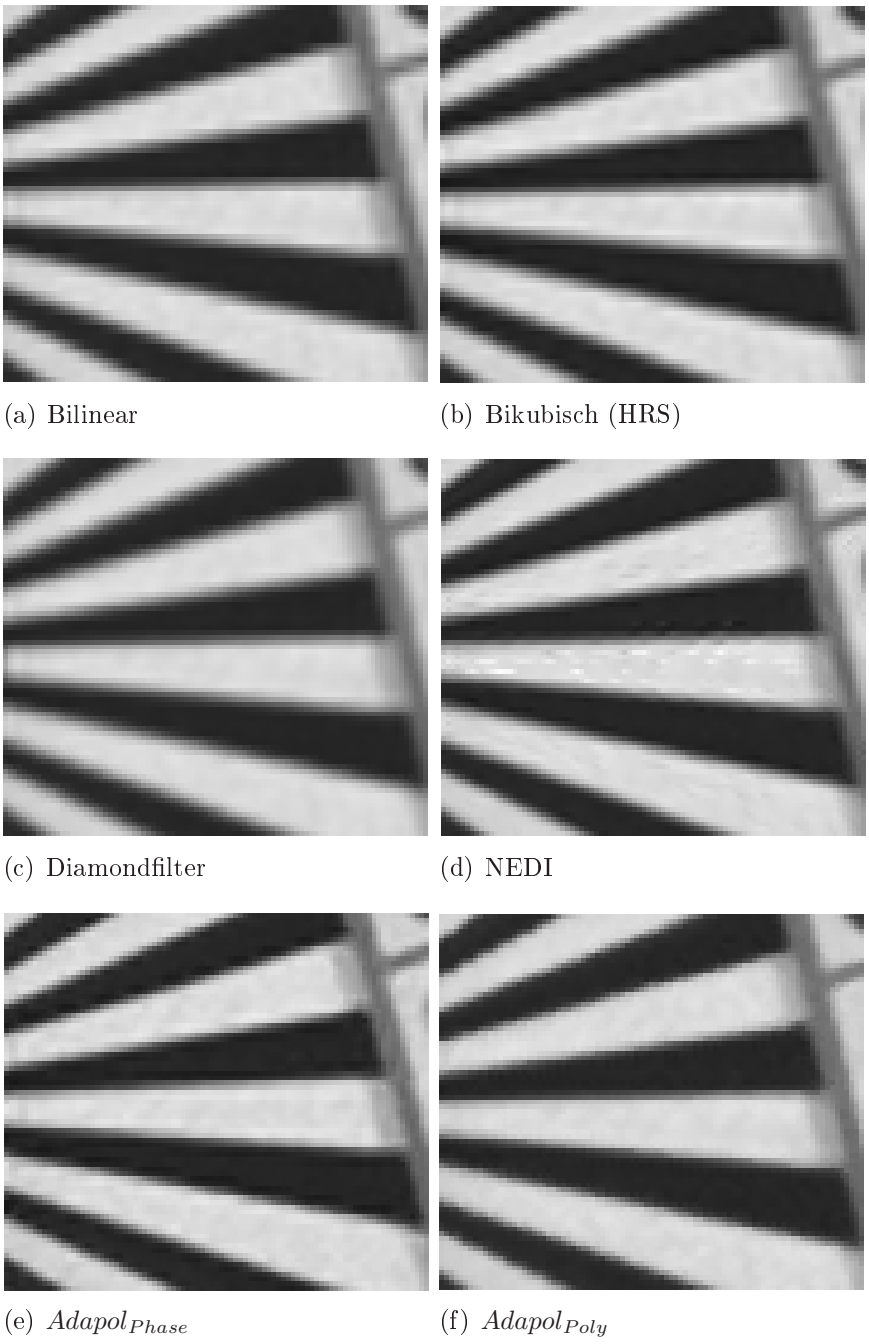
Auf Grund der begrenzten Möglichkeit des adaptiven Interpolationsverfahrens NEDI, wird der Vergleich auf der Basis des Interpolationsfaktors 2 durchgeführt.
Eine weitere technische Einschränkung ist ein Unterschied in der Phasenlage der Verfahren. Bis auf die Verfahren Diamond Shape und NEDI verwenden die restlichen Verfahren die Überlegungen zu den örtlichen Phasenverhältnissen aus Kapitel 2.1.2. Die beiden Verfahren Diamond Shape und NEDI nutzen das Originalraster und fügen an der Stelle zwischen zwei Originalpixelpositionen einen Wert ein. Grundsätzlich führt diese Vorgehensweise zu etwas schärferen Kanten als eine komplette Neuberechnung, da nur jeder zweite Pixel neu berechnet werden muss. Auf diese Weise werden aber die zu bewertenden Verfahren nicht bevorzugt, es werden eher die Referenzverfahren Diamond Shape und NEDI bevorteilt.

Eine qualitativ hochwertige Interpolation muss sowohl Kanten scharf interpolieren, als auch einen natürlichen Bildeindruck erzeugen und dabei möglichst keine Artefakte vorweisen. An Hand der nächsten Abbildungen wird der Einfluss der verschiedenen adaptiven und nichtadaptiven Verfahren auf die Bildqualität vorgestellt.

Die Interpolation starker kontrastreicher SW Kanten ist in Abbildung 5.19 dargestellt. Man erkennt in dem Ausschnitt der Sequenz Wheel verschieden ausgerichtete Kanten. Die nichtadaptiven linearen Verfahren wie Bilinear und HRS erreichen hier erwartungsgemäß ein sehr schlechtes Ergebnis. Es sind starke Treppenartefakte sichtbar. Das Diamond Shape Verfahren vermindert durch die verbesserte Unterdrückung der Wiederholspektren diesen Effekt, produziert dabei jedoch eine sehr große Unschärfe.
Die bildinhaltsabhängige Interpolation NEDI erreicht hier die besten Ergebnisse. Es sind keine Treppenstufen oder Artefakte erkennbar. Außerdem sind die Kanten verhältnismäßig scharf.
Die im Rahmen dieser Arbeit beschriebenen Verfahren $Adapol_{Phase}$ und $Adapol_{Poly}$ erreichen auch eine Reduktion der Treppenartefakte bei guter Bildschärfe. $Adapol_{Poly}$ hat einen etwas ruhigeren Kantenverlauf als $Adapol_{Phase}$.

Im direkten Vergleich zum Kantenverlauf von NEDI wird jedoch sichtbar, dass NEDI die Kanten am besten (noch glatter) interpoliert. Das Verfahren NEDI trainiert seine Filterkoeffizienten durch Betrachtung der Umgebung des zu interpolierenden Punktes. Bei derart ausgeprägten Kanten wird dementsprechend die vorherrschende Kantenrichtung perfekt erkannt. Bei detaillierteren Bildausschnitten ist das gemittelte Training über die Kantenrichtungen in der Nachbarschaft nicht vorteilhaft, da sich die Kantenrichtung zu schnell ändert.
Abbildung 5.19 zeigt einen anderen Ausschnitt der Sequenz WHEEL. Es ist ein Ausschnitt eines Kalenders zu sehen. Auch hier ist der Nachteil nichtadaptiver Verfahren deutlich zu erkennen (Treppenartefakte und Unschärfe).
Bei den Ergebnissen von NEDI zeigt sich hier sehr deutlich, dass zwar die Kanten im Kalenderblatt sauber interpoliert werden, der detaillierte Bereich der Zahlen wird jedoch nur unzureichend interpoliert (siehe vor allem die 12 und 13).
$Adapol_{Phase}$ und $Adapol_{Poly}$ erreichen hier deutlich schärfere und detailreichere Ergebnisse. Bei einem direkten Vergleich ist eine etwas bessere Interpolation von Detailsignalen bei $Adapol_{Phase}$ zu erkennen und eine etwas saubere Interpolation von Kanten beim Polynom.

Diese Ergebnisse finden sich auch bei der Betrachtung anderer Testsequenzen wieder, wie in Abbildung 5.21 (Ausschnitt der Sequenz LENA) zu erkennen ist. Dieser Ausschnitt beinhaltet alle wesentlichen Elemente zum Test eines Interpolationsverfahrens. Es sind feine Details wie die Wimpern und die Pupille als auch signifikante Kanten wie die Hutkrempe im Bild erhalten. In diesem Bild sind die Treppenartefakte lediglich bei Verwendung einer bilinearen Interpolation zu erkennen. Die bikubische HRS Interpolation erreicht schon ein akzeptables Ergebnis. Aus diesem Grund ist beim Diamond-Shape Verfahren kein visueller Vorteil im Vergleich zu den anderen nichtadaptiven Verfahren zu erkennen.
In diesem Bildauschnitt wird auch deutlich, dass das adaptive Verfahren NEDI den Eindruck eines Gemäldes verursacht. Die feinen Details werden ausgelöscht, dabei jedoch stärkere Kanten erhalten. So wirkt das Bild wie mit einem breiteren Pinsel nachgemalt (in Zugrichtung des „Pinsels“ sind die Kanten scharf, aber es können keine feinen Details gezeichnet werden).
Die Interpolation mit einem adaptiven Polynom und die Interpolation mit einer adaptiven Filterdatenbank erreichen ähnliche Ergebnisse. Beide Bildausschnitte erscheinen scharf und detailliert. Selbst die feinen Strukturen (wie z. B. die Wimpern) sind gut zu erkennen.
Neben diesen subjektiven Bildbeispielen kann man den Schärfegewinn auch an Hand der Messung der Kantensteilheit erkennen. In Abbildung 5.22 ist die gemessene Kantensteilheit für alle vorgestellten Verfahren angegeben.
Die bisherigen Ergebnisse zum Schärfeeindruck werden hierdurch vollständig bestätigt. Die nichtadaptiven Verfahren erreichen bei einer scharfen Kante die schlechtesten Ergebnisse.

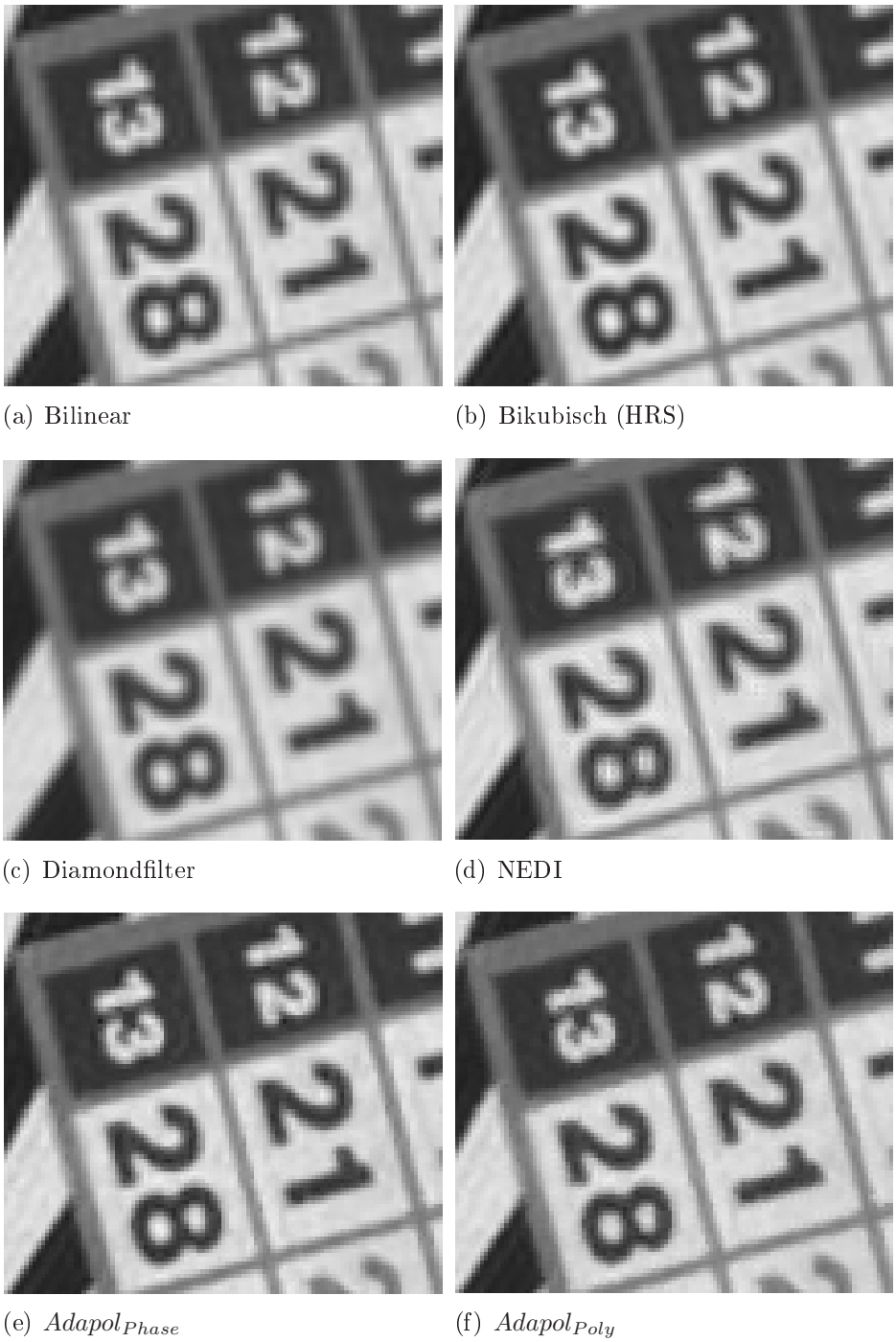


(a) Bilinear (b) Bikubisch (HRS)

(c) Diamondfilter (d) NEDI

(e) $Adapol_{Phase}$ (f) $Adapol_{Poly}$

Abbildung 5.19: Zweifache Interpolation eines Ausschnittes der schwarz/weiß Kanten der Sequenz WHEEL

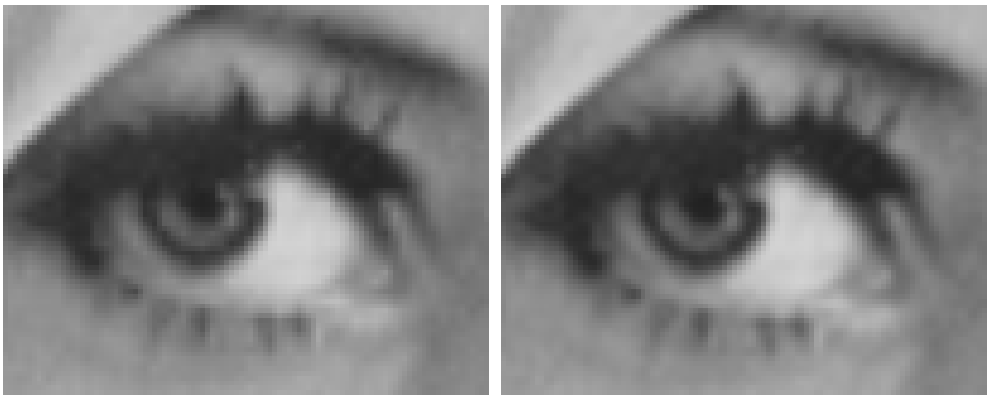


(a) Bilinear (b) Bikubisch (HRS)

(c) Diamondfilter (d) NEDI

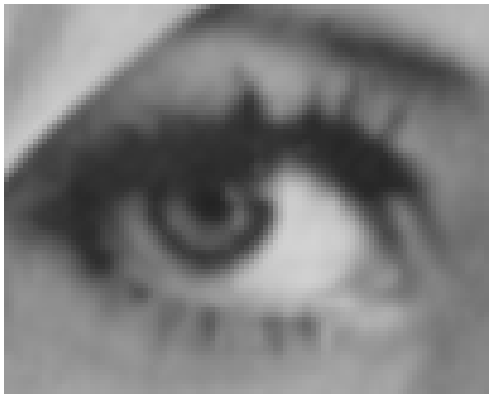
(e) $Adapol_{Phase}$ (f) $Adapol_{Poly}$

Abbildung 5.20: Zweifache Interpolation eines Ausschnittes des Kalenderblattes der Sequenz WHEEL



(a) Bilinear

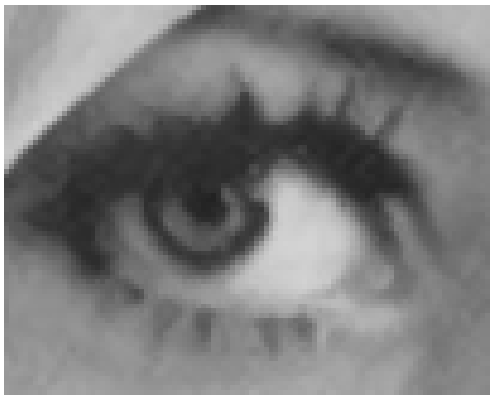
(b) Bikubisch (HRS)



(c) Diamondfilter

(d) NEDI

(e) $Adapol_{Phase}$



(f) $Adapol_{Poly}$

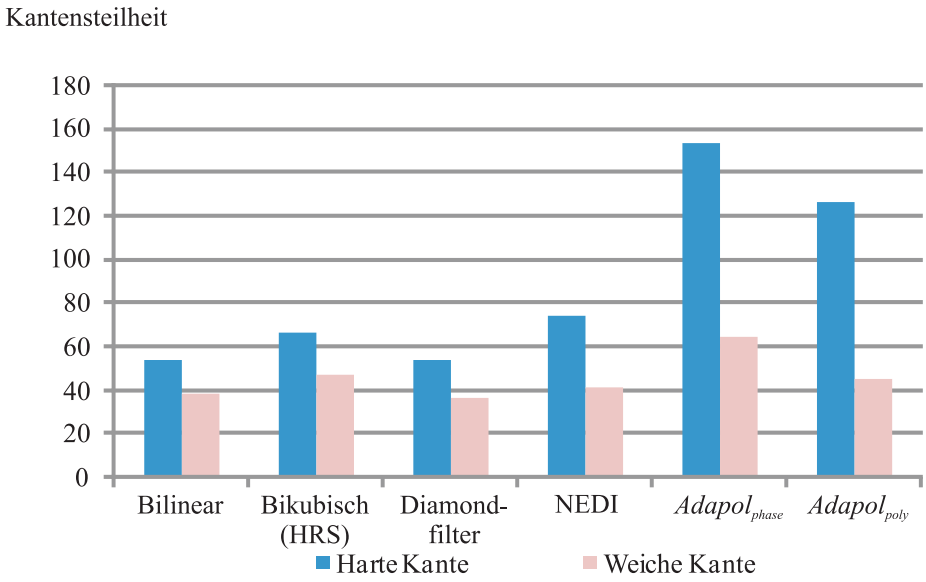
Abbildung 5.21: Zweifache Interpolation eines Ausschnittes der Sequenz Lena

Der größte Wert der nichtadaptiven Verfahren liegt, bedingt durch die hochfrequente Impulsantwort, bei der HRS Interpolation. Hierbei werden auch die größten Überschwinger erzeugt. Man erkennt an diesen Messwerten die Fähigkeit der adaptiven klassifikationsbasierten Verfahren, harte Kanten weitestgehend zu erhalten. Die Steigerung gegenüber dem hochfrequenten HRS ist zwar beim NEDI Verfahren nur gering, was sich aber mit den Bildeindrücken deckt. Bei der Interpolation gewinnt NEDI vor allem durch die vollständige Unterdrückung der Treppenartefakte, wobei der Bildeindruck (Kantensteilheit) nur wenig schärfer wird, als beim HRS.

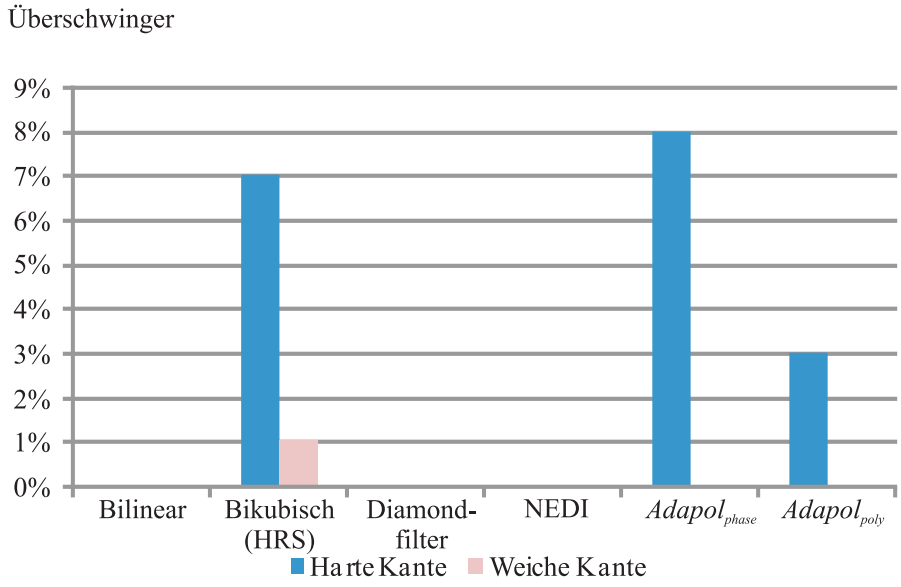
Eine enorme Steigerung der Kantensteilheit wird durch die Verwendung der beiden im Rahmen dieser Arbeit vorgestellten Verfahren $Adapol_{phase}$ und $Adapol_{poly}$ erreicht. Der ursprüngliche Signalhub von 150 Luminanzwerten bei der harten ungefilterten Kante sowie der Signalhub der gefilterten Kante von 71 werden trotz der Interpolation um den Faktor 2 nahezu vollständig rekonstruiert. Es ergeben sich als Kantensteilheit die Werte 153 und 67.
Bei sehr weichen Kanten ist jedoch sehr schnell eine Abnahme der Kantensteilheit festzustellen. Dies ist zwar durch die begrenzte Filterausdehnung bedingt, aber auch für einen natürlichen Bildeindruck vorteilhaft, da sehr weiche Verläufe nicht künstlich geschärft werden dürfen.

Man kann auch erkennen, dass das hier verwendete Polynom nicht dieselbe Schärfeperformance erreicht wie die Polyphasenrealisierung. Die gemessenen Kantenwerte, aber auch die Bildeindrücke der Polynomrealisierung zeigen eine etwas geringere Schärfe als die Ergebnisse der Polyphasenrealisierung. Dies deutet daraufhin, dass der Freiheitsgrad des Polynoms (4. Grad in x- und y-Richtung) nicht ausreichend ist, um den Verlauf der Interpolationsform absolut fehlerfrei abzubilden.

Ein Training mit höherem Freiheitsgrad ist mit der jetzigen Struktur nicht möglich. Der Trainingsalgorithmus und die Erstellung der Nebenbedingungen ist zwar im Grad frei parametrisierbar, jedoch führt ein Training mit höherer Ordnung zu Ungenauigkeiten der Filterkoeffizienten. Diese sind auf schlecht konditionierte Matrizen durch Rechenungenauigkeit zurückzuführen. Bei dem vierten Grad existieren 256 Koeffizienten. Durch eine Erhöhung auf den 5. Grad ergeben sich jedoch schon 400 Koeffizienten. Da innerhalb des Trainingsprozesses die Werte transponiert und mit sich selber multipliziert werden, steigt an dieser Stelle der Bedarf an Rechengenauigkeit. Für eine Erhöhung des Freiheitsgrades ist eine Neuorganisation des Trainingsprozesses ein möglicher Lösungsansatz.
Trotz der etwas geringeren Schärfe im Vergleich zur Polyphasenrealisierung sind die Schärfewerte der Polynomrealisierung absolut betrachtet beachtlich. Berücksichtigt man nun noch die Möglichkeit, mit diesem einmal angelernten Datensatz beliebige Interpolationsfaktoren nutzen zu können, wird der Vorteil einer Polynomrealisierung deutlich. In Ab-



(a) Kantensteilheit



(b) Überschwinger

Abbildung 5.22: Kantensteilheit verschiedener Verfahren bei einer zweifachen Interpolation

(a) Originalausschnitt

(b) Faktor $\frac{16}{9}$

(c) Faktor $\frac{2}{1}$

(d) Faktor $\frac{8}{3}$

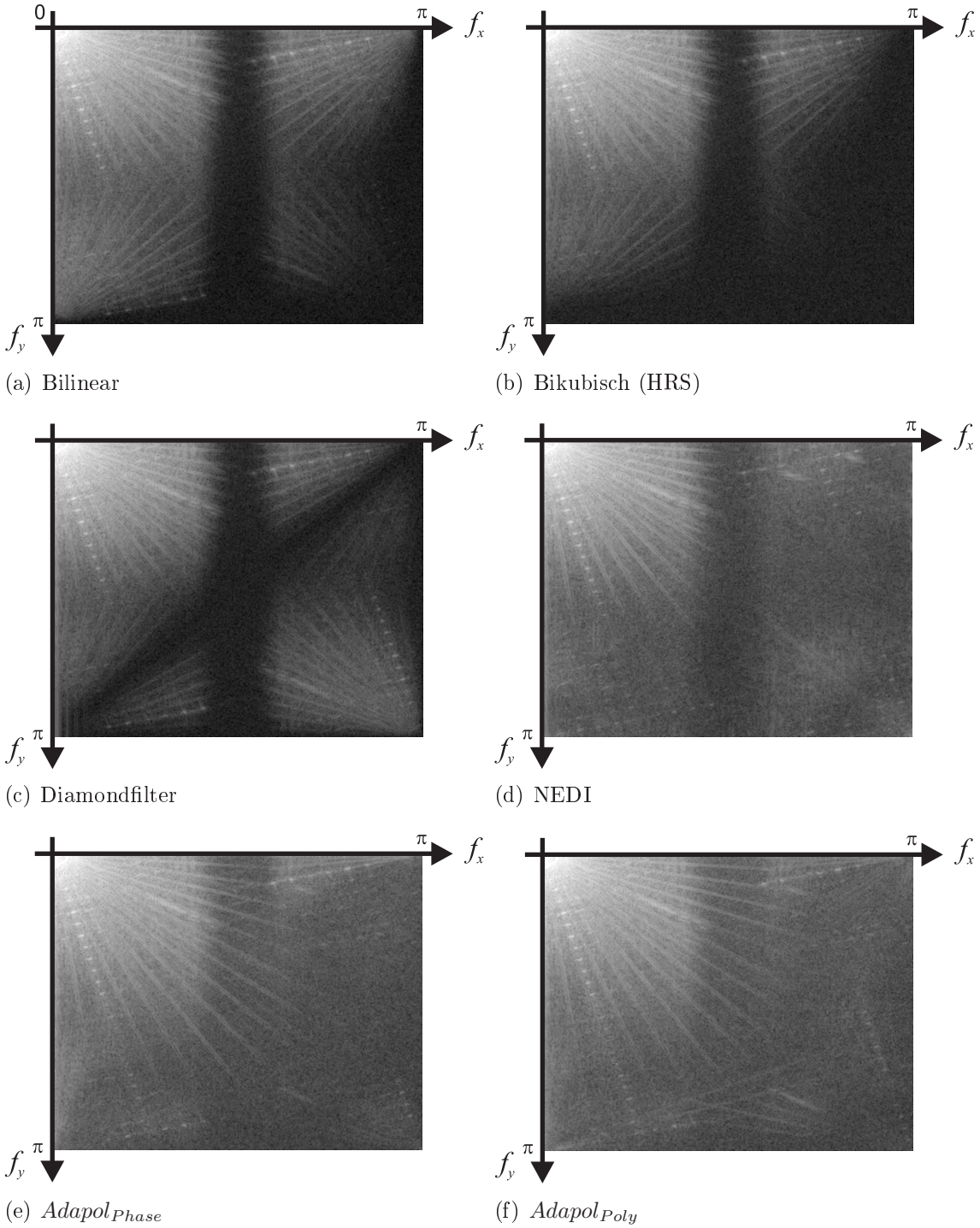
(e) Faktor $\frac{3}{1}$

Abbildung 5.23: Interpolationsergebnisse eines Ausschnittes der Sequenz LENA bei Verwendung eines antrainierten Datensatzes mit $Adapol_{Poly}$

bildung 5.23 kann man unterschiedlich vergrößerte Bildausschnitte aus der Sequenz LENA erkennen, die mit Hilfe eines einmalig angelernten Polynomdatensatzes interpoliert wurden.
Bei allen Vergrößerungsfaktoren ist eine deutliche Schärfe zu erkennen. Somit überragt die $Adapol_{Poly}$ Interpolation die anderen Verfahren durch seine hervorragende Vielseitigkeit bei gleichzeitig sehr guter Performance.

Beim Vergleich der daraus resultierenden Spektren (hier am Beispiel der Sequenz WHEEL zu sehen) werden die Unterschiede zwischen den einzelnen Verfahren nochmals verdeutlicht. Abbildung 5.24 zeigt die horizontalen und vertikalen Frequenzen von 0 bis π der um den Faktor 2 vergrößerten Sequenz WHEEL.
Bei der bilinearen Interpolation sind die Zwischenspektren nur sehr schwach unterdrückt. Die HRS Interpolation erreicht eine etwas bessere Unterdrückung der Spektren bei einem etwas besseren Erhalt des Grundspektrums.
Das Diamondfilter ähnelt dem bilinearen Filter, da die Störspektren nur sehr schlecht unterdrückt werden. Beide Filter sind sich sehr ähnlich, da sie grundsätzlich auf der Faltung zweier Rechtecke basieren. Lediglich durch die Drehung des zweiten Rechtecks beim Diamondfilter werden in den rein vertikalen bzw. horizontalen Frequenzbereichen um π die niederfrequenten Anteile des Störspektrums stärker unterdrückt. Andere Bereiche (wie z. B. $(f_x = \pi, f_y = \pi)$) werden jedoch weniger gedämpft.
Beim Vergleich dieser drei nicht adaptiven Verfahren zu den folgenden drei adaptiven Verfahren fällt gerade im ungenutzten Spektralbereich ein gestiegenes Rauschen auf (sichtbar an der gestiegenen Grundhelligkeit). Dies ist auf die Inhaltsadaptivität und die örtlich ausgerichteten Filter zurückzuführen.
Auf der anderen Seite erreichen die adaptiven Verfahren ein deutlich besseres Ergebnis. Sie erweitern alle das Hauptspektrum und unterdrücken die Störspektren besser als die nicht adaptiven Verfahren. NEDI erreicht im Vergleich zu den nichtadaptiven Verfahren schon eine Erweiterung der spektralen Anteile. Man kann im direkten Vergleich zur HRS Interpolation eine Verlängerung der Strahlen erkennen. Diese Strahlen entsprechen den Fourierreihen der stark ausgeprägten SW Kanten in der Sequenz WHEEL.

Die adaptiven Verfahren $Adapol_{Poly}$ und $Adapol_{Phase}$ erreichen eine spektrale Erweiterung dieser Strahlen bis weit in den Bereich der Wiederholspektren (von $\frac{\pi}{2}$ bis π) hinein. Der Unterschied zwischen beiden Verfahren ist nur gering. Einige Richtungen werden von $Adapol_{Poly}$ weiter verlängert, andere von $Adapol_{Phase}$. Auffallend ist das unterschiedliche Verhalten in den rein vertikalen und horizontalen Frequenzen. Hier unterdrückt $Adapol_{Poly}$ die Anteile der Wiederholspektren ein wenig besser. Da hier maßgeblich die Anteile für Treppenartefakte existieren, deckt sich die bessere spektrale Unterdrückung mit dem Eindruck einer leicht verbesserten Interpolation von Kanten bei der Verwendung von $Adapol_{Poly}$ (siehe hierzu auch Abbildung 5.19).



(a) Bilinear

(b) Bikubisch (HRS)

(c) Diamondfilter

(d) NEDI

(e) $Adapol_{Phase}$

(f) $Adapol_{Poly}$

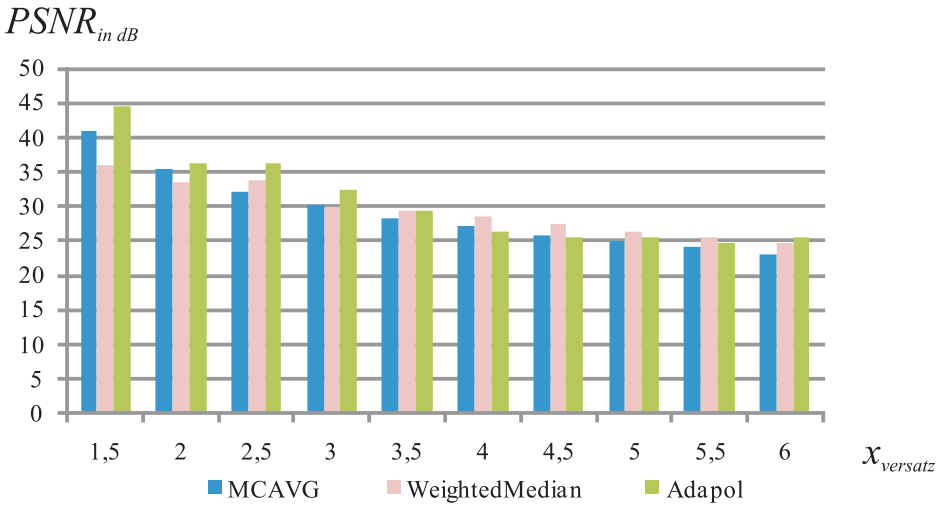
Abbildung 5.24: Spektrum der Interpolationsergebnisse der Sequenz WHEEL, Faktor 2

5.4 Simulationsergebnisse und Vergleich der zeitlichen Interpolationsverfahren

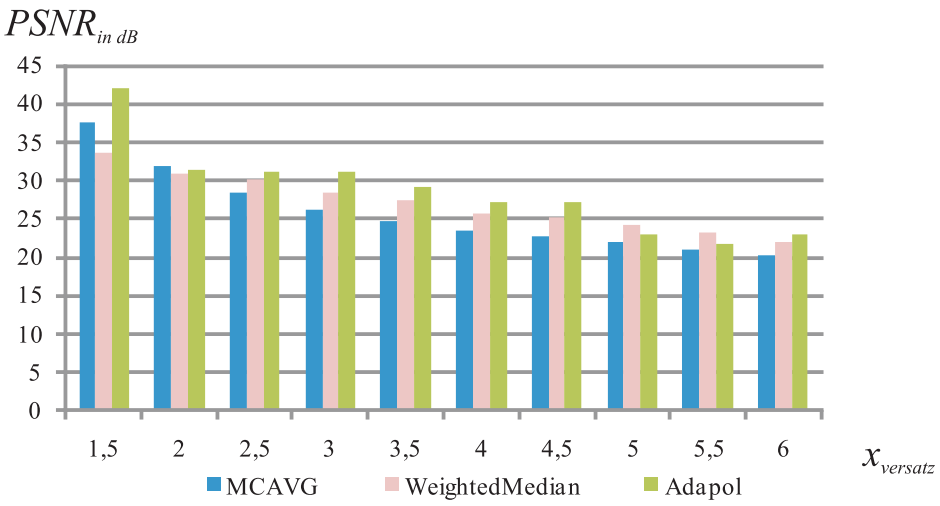
In diesem Abschnitt wird der Vorschlag einer klassifikationsbasierten zeitlichen Zwischenbildinterpolation (im Folgenden *Adapol* genannt) aus Abschnitt 4.4 in seiner Qualität beurteilt. Hierzu wird ein Vergleich mit einer einfachen bewegungskompensierten Mittelwertbildung (Motion Compensated Average, MCAVG) und mit einem Verfahren, welches auf gewichteten Medianfiltern (WeightedMedian) basiert, vorgenommen. Die Ausdehnung des Interpolationsfilters (Kreuzmaske) bei MCAVG ist $1x1$ und sowohl beim klassifikationsbasierten Verfahren als auch beim WeightedMedian $5x5$.
Beispiel der Simulationen ist eine Zwischenbildinterpolation von 24 auf 120 Bilder in der Sekunde. Dieser Fall tritt z. B. bei der Darstellung von Kinomaterial auf einem 120 Hz LCD auf. Dies entspricht den zeitlichen Zwischenphasen $\left\{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}\right\}$.

5.4.1 Objektiver Vergleich

Ein rein objektiver Vergleich mittels PSNR (siehe Abschnitt 5.1.2) ist auch bei der Zwischenbildinterpolation kein perfektes Maß, jedoch wird dieser Wert bei der Bewertung zeitlicher Verfahren oft genutzt und kann als Indiz für grobe Tendenzen genutzt werden.
Um den PSNR nutzen zu können, benötigt man Referenzsequenzen, die sowohl in hoher als auch in niedriger Auflösung vorliegen. Des Weiteren sollte möglichst der Einfluss der Qualität des Bewegungsschätzers verringert werden. Aus diesem Grund wird die PSNR Bewertung der Zwischenbildinterpolation mit künstlich erzeugten Sequenzen vorgenommen, für die die Bewegung genau bekannt ist. Hierzu wird ein Standbild einer Sequenz (z. B. Wheel) genommen und in horizontaler und auch diagonaler Richtung bewegt. Auf diese Weise können die Referenz- und die Testsequenzen einfach erstellt werden, und es kann das Verhalten der einzelnen Verfahren auf vorgegebene Bewegungsschätzfehler untersucht werden.
In Abbildung 5.25 sind die PSNR-Werte für verschiedene horizontale Verschiebungen dargestellt. Im oberen Bereich sind die Werte für eine Zwischenbildposition bei $\alpha = \frac{1}{5} = 0,2$ und im unteren Bereich die Werte bei $\alpha = \frac{2}{5} = 0,4$ zu sehen. Man erkennt in beiden Abbildungen deutlich, dass gerade für den Bereich einer kleinen Bewegung von einem bis vier Pixeln die klassifikationsbasierte Zwischenbildinterpolation die anderen Verfahren übertrifft. Ab ungefähr vier Pixel überragt WeightedMedian die Werte vom MCAVG und vom klassifikationsbasierten Ansatz.
Dies lässt sich damit erklären, dass das klassifikationsbasierte Verfahren bei Bewegungsfehlern, die in der Ausdehnung der Maske liegen, den Fehler detektieren und kompensieren kann. WeightedMedian erreicht bei noch größeren Bewegungsfehlern die beste Performance, da im Filterentwurf der Medianfilter Bedingungen eingebunden worden sind, die bei



(a) Phase $\alpha = \frac{1}{5}$



(b) Phase $\alpha = \frac{2}{5}$

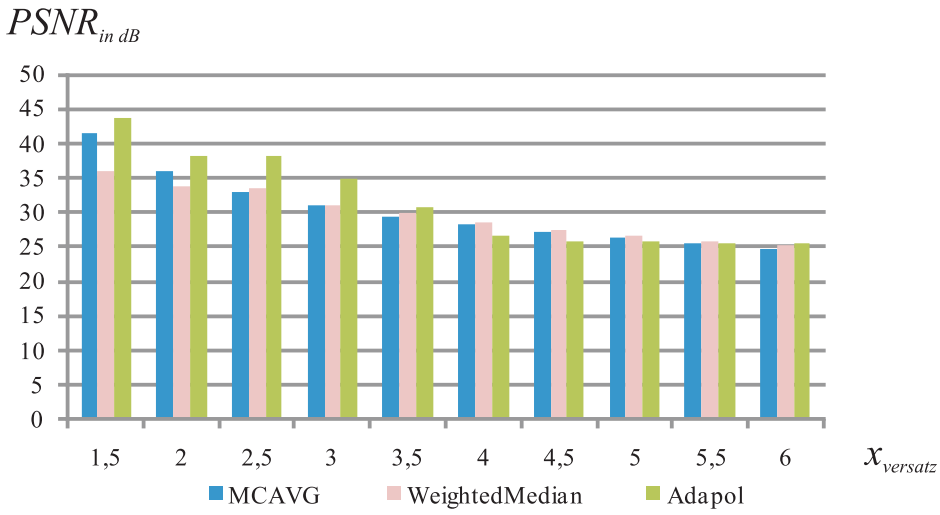
Abbildung 5.25: PSNR-Werte der Zwischenbildinterpolation für ein horizontal verschobenes Bild der Sequenz WHEEL

nicht kompensierbaren Bewegungsfehlern die Informationen des zeitlich am nächst gelegenen Bildes übernimmt und keine lineare Mittelung der beiden unterschiedlichen Informationen vornimmt.
Abbildung 5.26 zeigt für die Verschiebung eines Bildes der Sequenz FOOTBALL ähnliche Ergebnisse. Bis zu einem Versatz von ungefähr vier Pixel erreicht auch hier *Adapol* sehr gute Ergebnisse und wird dann von der gewichteten Medianfilterung überholt.

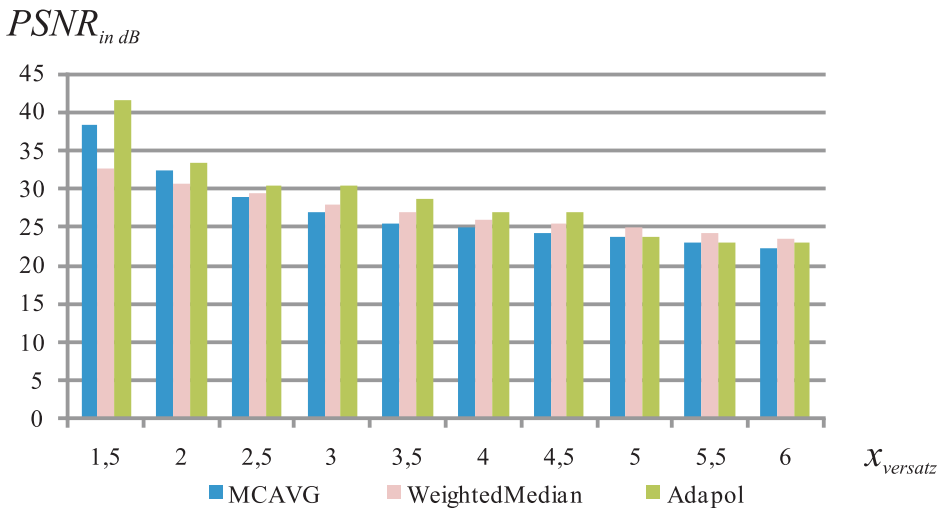
5.4.2 Subjektiver Vergleich

In diesem Abschnitt werden die PSNR Ergebnisse des vorherigen Abschnittes mit Bildbeispielen unterstützt und die Anwendbarkeit des Verfahrens bei natürlichen Bewegungsvektoren untersucht.
Bei einer Bewegung, die innerhalb der Kreuzmasken über eine Binarisierung detektierbar ist, ergeben sich bei einem globalen Bewegungsschätzungsfehler für die klassifikationsbasierte Zwischenbildinterpolation sehr gute PSNR Werte (siehe Abschnitt 5.4.1). Durch die Klassifikation der Fehler werden Filterkoeffizienten ausgewählt, die den Fehler selbst für Detailsignale ausgleichen können.
In den folgenden Abbildungen 5.27 und 5.28 sind Ausschnitte der Interpolationsergebnisse bei einem globalen Bewegungsschätzungfehler zu sehen (2 Pixel horizontal). Das MCAVG-Verfahren erzeugt durch den Bewegungsschätzungsfehler Doppelkonturen und Unschärfe. WeightedMedian erreicht trotz des Bewegungsschätzungsfehler eine phasengenaue Interpolation der Kanten. Auf Grund des Fehlers werden jedoch die Details zerstört (erkennbar an der weggefilterten Rasenstruktur und den eingeschränkt lesbaren Textbanden in der Sequenz FOOTBALL sowie den Inhalten des Kalenderblattes der Sequenz WHEEL). Die klassifikationsbasierte Zwischenbildinterpolation erhält jedoch sowohl einzelne Kanten als auch feine Details.
Der betrachtete globale Bewegungsvektorfehler ist notwendig, um die Effekte der einzelnen Verfahren auf Fehler untersuchen zu können. Die Wahrscheinlichkeit des Auftretens eines globalen Bewegungsvektorfehlers ist sehr gering und spiegelt nicht die tatsächlichen Probleme einer Zwischenbildinterpolation wieder. Aus diesem Grund werden die Verfahren zusätzlich mit tatsächlichen Bewegungsvektoren eines parallel-prädiktiven Bewegungsschätzers [PK04] getestet.

In den folgenden Bildausschnitten sind jeweils die Ausschnitte der benachbarten Originalbilder, sowie die Ergebnisse des MCAVG, WeightedMedian und *Adapol* zu sehen.
In Abbildung 5.29 ist ein Ausschnitt der Sequenz WHEEL zu erkennen. Durch einen Bewegungsschätzungsfehler ist eine Speiche doppelt vorhanden. Gerade beim MCAVG ist dies deutlich zu erkennen. Die klassifikationsbasierte Zwischenbildinterpolation erreicht eine Verbesserung durch eine Reduktion der Artefakte. Die gewichtete Medianfilterung erzielt jedoch eine noch besseres Ergebnis, da die Artefakte noch weiter reduziert werden.

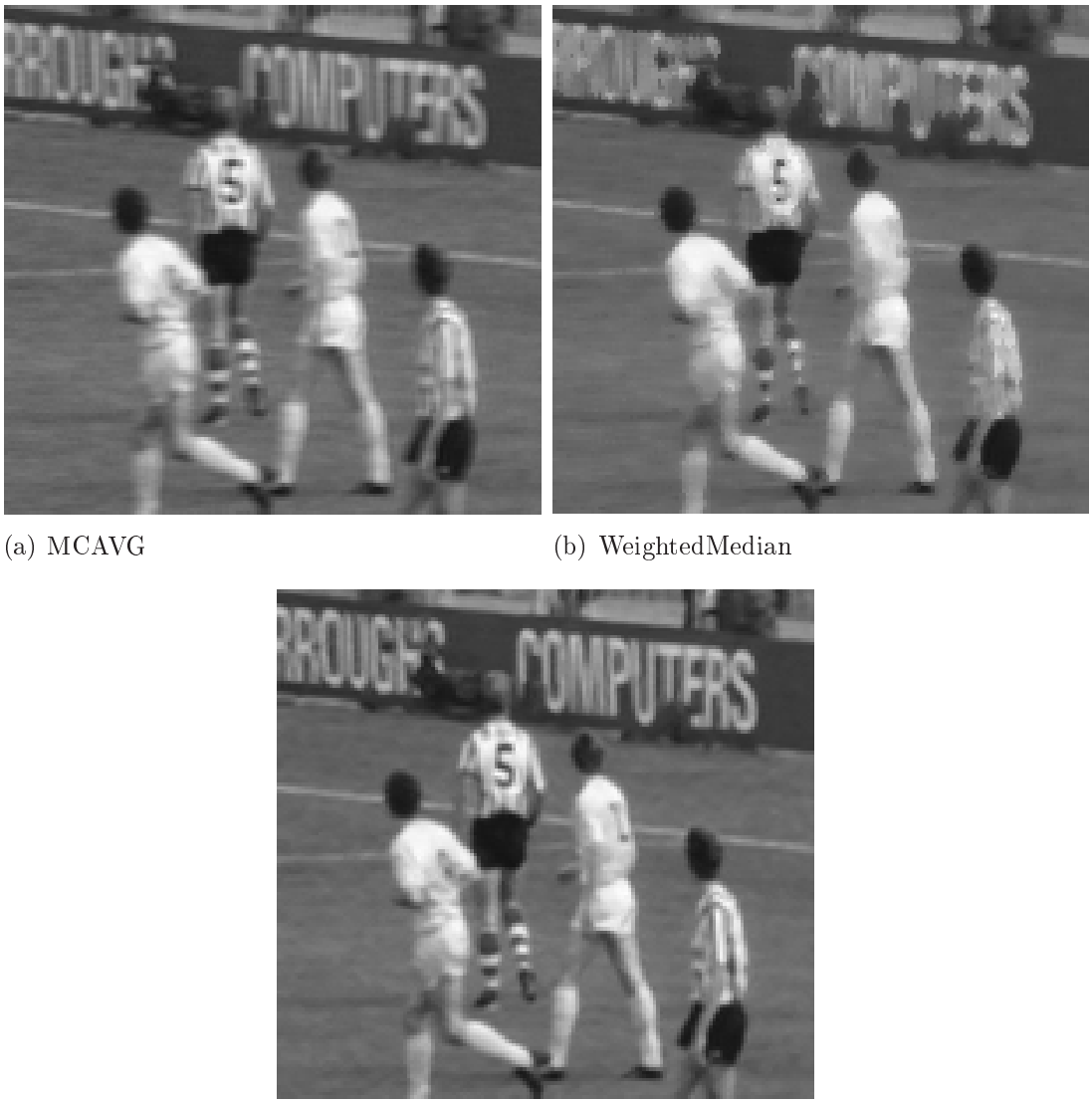


(a) Phase $\alpha = \frac{1}{5}$



(b) Phase $\alpha = \frac{2}{5}$

Abbildung 5.26: PSNR-Werte der Zwischenbildinterpolation für ein horizontal verschobenes Bild der Sequenz FOOTBALL

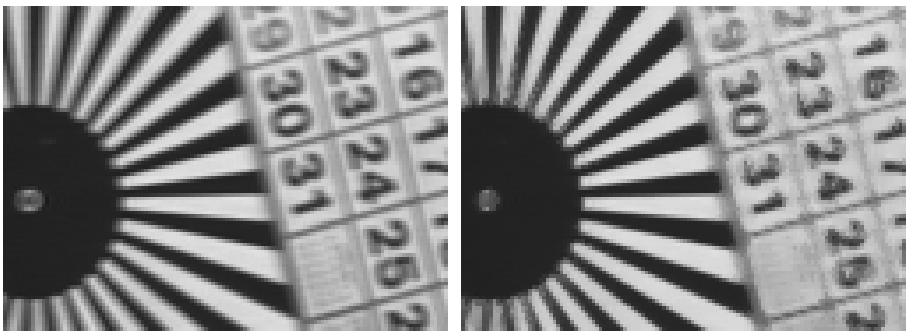


(a) MCAVG

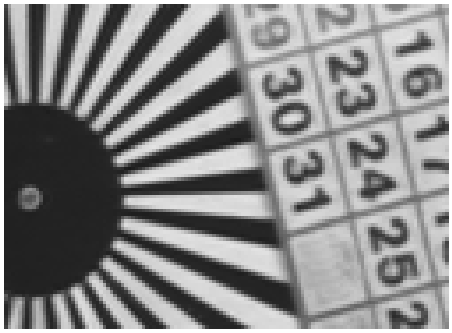
(b) WeightedMedian

(c) Adapol

Abbildung 5.27: Interpolationsergebnisse eines Bildes der Sequenz WHEEL bei fehlerhaftem Bewegungsvektor (hier zwei Pixel horizontal), Interpolationsphase $\alpha = \frac{2}{5}$



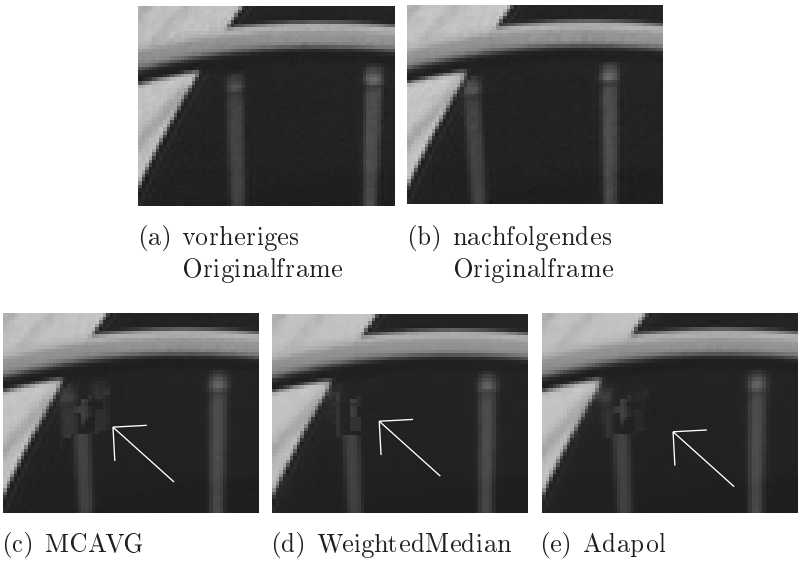
(a) MCAVG (b) WeightedMedian



(c) Adapol

Abbildung 5.28: Interpolationsergebnisse eines Bildes der Sequenz FOOTBALL bei fehlerhaftem Bewegungsvektor (hier zwei Pixel horizontal), Interpolationsphase $\alpha = \frac{2}{5}$

In Abbildung 5.30 ist ein ähnliches Verhalten zu erkennen. Die klassifikationsbasierte



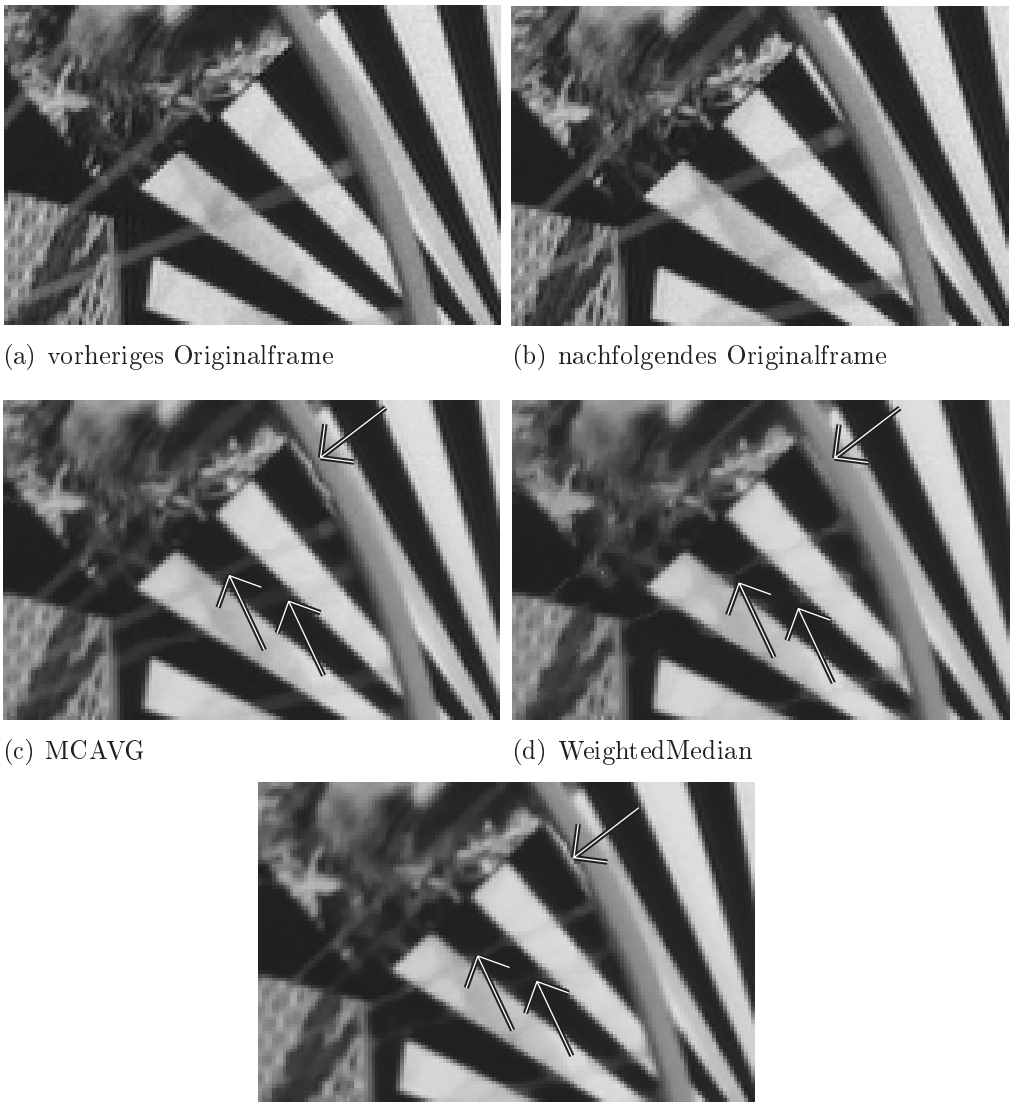
(a) vorheriges Originalframe (b) nachfolgendes Originalframe

(c) MCAVG (d) WeightedMedian (e) Adapol

Abbildung 5.29: Interpolationsergebnisse eines Bildes der Sequenz WHEEL bei geschätzten Bewegungsvektoren, Interpolationsphase $\alpha = \frac{2}{5}$

Interpolation und die gewichtete Medianfilterung erreichen beide eine Reduktion der Störartefakte, aber auch hier ist die Unterdrückung bei der gewichteten Medianfilterung deutlicher. Auch in der Abbildung 5.31 ist dieses Verhalten wiederzuerkennen. Auf Grund der eingeschränkten Filterausdehnung kann keine vollständige Kompensierung der Bewegung erfolgen. Durch eine Aufweitung der Filtermaske (siehe Abbildung 5.31(f)) um den Faktor $1,5$ wird zwar eine weitere Unterdrückung der Bildinhalte erreicht, aber eine saubere Rekonstruktion wie bei der gewichteten Medianfilterung ist somit nicht möglich. Zusätzlich wird durch die Aufweitung der Interpolationsmaske die Unschärfe im Bild vergrößert.
Die Abbildung 5.32 zeigt die Vorteile der klassifikationsbasierten Interpolation, da die Bewegungsfehler in diesem Bild innerhalb der Maskengröße liegen. Durch die Benutzung des MCAVG und der WeightedMedian-Filterung wird die Hand des Fußballspielers soweit aufgeweicht, dass sie fast nicht mehr sichtbar ist. Bei der klassifikationsbasierten Interpolation hingegen wird sie vollständig und scharf abgebildet.

Zusammenfassend kann man sagen, dass die bewegungskompensierte Interpolation in der jetzigen durch den Aufwand begrenzten Struktur nicht die Qualität der gewichteten Medianfilterung erreicht. Andererseits kann man das Potential des Verfahrens erkennen, bei Bewegungsfehlern trotzdem Detailsignale fehlerfrei interpolieren zu können.



(a) vorheriges Originalframe

(b) nachfolgendes Originalframe

(c) MCAVG

(d) WeightedMedian

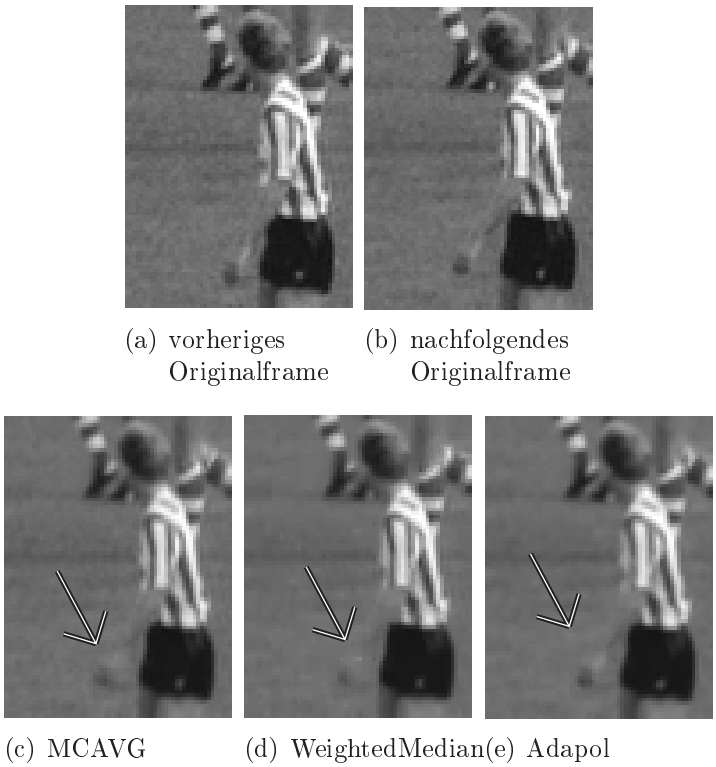
(e) Adapol

Abbildung 5.30: Interpolationsergebnisse eines Bildes der Sequenz WHEEL bei geschätzten Bewegungsvektoren, Interpolationsphase $\alpha = \frac{2}{5}$

(a) vorheriges Originalframe (b) nachfolgendes Originalframe

(c) MCAVG (d) WeightedMedian (e) Adapol (f) Adapol mit Aufweitung 1,5

Abbildung 5.31: Interpolationsergebnisse eines Bildes der Sequenz FOOTBALL bei geschätzten Bewegungsvektoren, Interpolationsphase $\alpha = \frac{2}{5}$



(a) vorheriges Originalframe (b) nachfolgendes Originalframe

(c) MCAVG (d) WeightedMedian (e) Adapol

Abbildung 5.32: Interpolationsergebnisse eines Bildes der Sequenz FOOTBALL bei geschätzten Bewegungsvektoren, Interpolationsphase $\alpha = \frac{2}{5}$

Bei der Interpolation von Zwischenbildern kann das Verfahren bei einer Maskenausdehnung von 5 Pixeln nur Bewegungsfehler von einem bis vier Pixeln kompensieren. Eine einfache Aufweitung der Masken kann in den problematischen Sequenzen nur eine geringe Verbesserung erreichen. Die bei der Zwischenbildinterpolation stark sichtbaren Artefakte durch eine fehlerhafte Bewegungsschätzung liegen sehr oft im Bereich von 10-20 Bildpunkten. Um dies mit der klassifikationsbasierten Interpolation erfassen zu können, müssten größere Filtermasken benutzt werden. Aus den in Kapitel 4.4 abgeschätzten Klassenzahlen wird deutlich, dass die vorhandene Klassifikation auf der Basis einer Binarisierung der Kreuzmasken eine zu hohe Anzahl an Klassen erzeugt. Es ist zwar sinnvoll durch die Binarisierung Informationen über die örtliche Struktur in beiden Bildern zu erhalten, die wichtigste Information steckt jedoch in dem klassifizierten Bewegungsunterschied.
Auf Grund des Potentials des Verfahrens macht es Sinn, alternative Klassifikationsverfahren zu untersuchen, die eine Detektion des Bewegungsvektorfehlers mit weniger Klassen ermöglichen. Hierbei sind zum einen hierarchische Ansätze denkbar, die auf verkleinerten Bildern zuerst große Bewegungsfehler vorklassifizieren und dann anschließend eine Feinklassifikation vorgenommen wird.
Zum anderen bieten sich Möglichkeiten in einer Autokorrelationsfunktion (AKF) von größeren Kreuzmasken, die auf horizontal und vertikal abgeleiteten Kantenbildern erfolgt. Eine horizontale und vertikale Maximumsdetektion ergibt daraufhin den jeweiligen Bewegungsfehler. Erste Untersuchungen bezüglich der Bewegungsdetektion zeigen hier Potential; eine ausführliche Untersuchung bezüglich Aufwand und vor allem Stabilität bei verrauschtem Bildmaterial fehlt jedoch noch.

6 Zusammenfassung und Ausblick

Die Formatkonversion von Bildern und Bildsequenzen für hochauflösende Displays stellt einen hohen Anspruch an die zu verwendenden Interpolationsverfahren. So müssen unterschiedlichste Eingangsformate und -qualitäten berücksichtigt und vor allem eine hohe Interpolationsgüte erreicht werden.
Nichtlineare adaptive Verfahren ermöglichen zwar in vielen Fällen eine im Vergleich zu linearen Verfahren gestiegene Interpolationsqualität, tendieren jedoch zur Erzeugung von Interpolationsartefakten und benötigen zumeist einen erhöhten Realisierungsaufwand.
Die klassifikationsbasierte Interpolation schließt die Lücke und kombiniert die Vorteile von linearen Filtern (geringer Aufwand, geringe Artefaktverstärkung) mit dem Potential einer erhöhten Qualität durch eine nichtlineare Verarbeitung.

In dieser Arbeit wurden zwei klassifikationsbasierte örtliche Interpolationsverfahren vorgestellt. Das eine Verfahren basiert auf einer Filterdatenbank, in der für jede Bildsituation und zu berücksichtigende Zwischenphase optimierte Filterdaten abgelegt sind. Das andere Verfahren besitzt für jede Bildsituation jeweils eine optimierte kontinuierliche Interpolationsform. Letzteres Verfahren hat den Vorteil weitgehend beliebige Interpolationsfaktoren zu ermöglichen, ohne ein erneutes Training vornehmen zu müssen oder die Datenbank auswechseln zu müssen.
Es wurden verschiedene Optimierungen für diese Verfahren dargestellt. Neben Kriterien zur Beurteilung der Güte der trainierten Filterkoeffizienten und Austauschstrategien für nicht optimal angelernte Filterkoeffizienten, wurde auch die notwendige Anzahl an Trainingsdaten bei Modifikation von Parametern wie z. B. Filtergröße und Interpolationsfaktor untersucht. Auch der Einfluss dieser Parameter sowie der Einfluss des Inhalts der Trainingssequenzen auf die resultierende Qualität der Filterkoeffizienten wurde dargestellt. Des Weiteren wurden Nebenbedingungen hergeleitet, die grundlegende Bildstrukturen fehlerfrei erhalten können, und diese in den Trainingsprozess integriert. Somit werden Artefakte durch fehlerhafte Filterkoeffizienten systembedingt ausgeschlossen.

Diese beiden örtlichen Verfahren übertreffen die in der Literatur bisher dargestellten üblichen nichtadaptiven und adaptiven Verfahren in der Bildschärfe und Bildqualität. Hinzu kommt die Möglichkeit durch die Bereitstellung mehrerer unterschiedlicher Filterkoeffizientensätze die Interpolation an die Qualität des Eingangsmaterials anzupassen. Sonst zusätzlich notwendige Bildverarbeitungsschritte wie Rausch-, Artefaktreduktion

oder Bildschärfeanhebung können durch die Wahl der passenden Koeffizienten beim Interpolationsprozess ersetzt bzw. unterstützt werden. Berücksichtigt man zusätzlich den geringen Aufwand dieser Verfahren, so wird der Gewinn durch eine klassifikationsbasierte Verarbeitung deutlich.
Man sieht jedoch auch die Grenzen dieser örtlichen Interpolation. Es gibt verständlicherweise eine starke Wechselwirkung zwischen der Maskengröße und der realisierbaren Kantensteilheit. Mit Hilfe der Nebenbedingungen, können zwar sehr scharfe Kantenübergänge trainiert werden, ohne dass starke Artefakte entstehen, eine perfekte Behandlung vor allem der ausgedehnteren Kantenübergänge wird jedoch auf Grund der begrenzten Maskengröße nicht erreicht. Die Ergebnisse größerer Masken zeigen zwar eine deutliche Steigerung der Interpolationsqualität, können jedoch auf Grund nicht optimal angelernter Filterklassen und somit einer erhöhten Anzahl an Artefakten leider nicht verwendet werden. Bei der hier verwendeten Klassifizierung mittels einer Binarisierung des lokalen Bildinhaltes hat die Maskengröße einen exponentiellen Einfluss auf die Anzahl der Klassen und somit auch auf jeglichen Aufwand (Speicher, notwendiges Trainingsmaterial etc.). Der Speicheraufwand beim Training einer 3x3 Maske liegt bei ca. 730KB, bei einer Erweiterung auf eine 5x5 Maske ist der notwendige Speicher schon bei 325GB. Dies zeigt deutlich die Grenzen des Verfahrens.

Auf Grund der Vorteile einer klassifikationsbasierten Verarbeitung wird zusätzlich ein hierauf basierendes zeitliches Zwischenbildinterpolationsverfahren vorgestellt. Dies ermöglicht durch eine einfache Klassifikation Bewegungsfehler adaptiv erkennen und ausgleichen zu können.
Hierbei ist sogar bei Bewegungsfehlern der Erhalt feinster Detailsignale möglich. Dies unterscheidet das Verfahren von den bisher üblichen bewegungsfehlerkompensierenden Zwischenbildinterpolationen, da diese Detailsignale bei Bewegungsfehlern nicht in der selben Qualität erhalten können. Das Verfahren ist jedoch durch die Klassifikationsmethode auf sehr kleine Filtermasken beschränkt, da ansonsten zu viele Klassen entstehen. Im direkten Vergleich mit Standardverfahren zur zeitlichen Zwischenbildinterpolation kann der Vorteil einer adaptiven Verarbeitung nicht deutlich herausgestellt werden. Dies ist maßgeblich dadurch bedingt, dass das Verfahren lediglich im Bereich weniger Pixel fehlerkorrigierend agiert und hierbei auch die anderen Interpolationsverfahren überragt, jedoch bei größeren Bewegungsfehlern keine Kompensation erfolgen kann und Bewegungsfehlerartefakte entstehen können. Hier erreichen nichtlineare Verfahren, wie die gewichteten Medianfilter, bessere Ergebnisse. Da die nichtlineare Medianfilterung bei nicht kompensierbaren Bewegungsfehlern keine Mittelung von unterschiedlichen Bildinhalten vornimmt, sondern die zeitlich am nächsten liegenden Bildinhalte kopiert, werden somit Artefakte durch Doppelkonturen vermieden.
Auf Grund der Optimierung mit Trainingsdaten ergeben für das hier vorgestellte Verfahren die linearen Filterkoeffizienten bei nicht kompensierbaren Bewegungen eine

Bildmittelung. Der nichtlineare Fallbackmodus einer Übernahme des am nächst liegenden Bildinhaltes müsste bei der klassifikationsbasierten Zwischenbildinterpolation als nachträgliche nichtlineare Veränderung der Filterkoeffizienten vorgenommen werden, um ähnliche Ergebnisse erreichen zu können.

Sowohl bei der örtlichen als auch der zeitlichen klassifikationsbasierten Interpolation liegt die Beschränkung der Anwendungsqualität in den begrenzten Maskengrößen. Die Filtergröße an sich ist jedoch nicht der begrenzende Faktor, sondern die aus der Klassifizierung entstehende Anzahl an Klassen.
Im Rahmen der Forschungsarbeiten für diese Arbeit wurden verschiedene örtliche und zeitliche klassenreduzierte Bildklassifikatoren implementiert, aus Aufwandsgründen jedoch nicht weiter optimiert oder analysiert. Erste Ergebnisse zeigen hier jedoch Potential, so dass eine eingehendere Untersuchung von alternativen Bildklassifikatoren, die für eine örtliche und/oder zeitliche Anwendung mit adaptiv ausgewählten linearen Filtern verwendet werden können, ein möglicher Ansatzpunkt für weitere Forschungsarbeiten darstellt.

Darüber hinaus ist die Wahl einer alternativen Bestimmung der optimalen Filterkoeffizienten denkbar. Im Gegensatz zu der im Rahmen dieser Arbeit verwendeten Minimierung des quadratischen Fehlers mit Nebenbedingungen bieten sich hier gerade die nichtlinearen Optimierungsverfahren an, da diese auch Bildqualitätsforderungen berücksichtigen können, die sich nicht durch lineare Gleichungen beschreiben lassen.

Literaturverzeichnis

[ABA01] Atkins, C.B., C.A. Bouman und J.P. Allebach: *Optimal image scaling using pixel classification.* In: *Image Processing, 2001. Proceedings. 2001 International Conference on*, Band 3, Seiten 864–867vol.3, 7-10 Oct. 2001.

[Atk98] Atkins, C. Brian: *Classification-based methods in optimal image interpolation.* Doktorarbeit, Faculty of Purdue University, December 1998.

[Blu97] Blume, Holger: *Nichtlineare fehlertolerante Interpolation von Zwischenbildern.* Fortschritt-Berichte, Reihe 10, Band 503, Universität Dortmund, VDI-Verlag, Düsseldorfh, 1997.

[Bro06] Brockhaus: *Der Große Brockhaus in einem Band.* Brockhaus, Mannheim, January 2006.

[BS91] Bronstein, I.N. und K.A. Semendjajew: *Taschenbuch der Mathematik.* B.G. Teubner Verlag, 1991.

[CR83] Crochiere, Ronald E. und Lawrence R. Rabiner: *Multirate Digital Signal Processing.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1983.

[dH00] Haan, Gerald de: *Video Processing for multimedia systems.* University Press, 2000.

[Dol06] Dolar, Carsten: *Auswirkungen der zeitlich-örtlichen LC Display-Auflösung auf die Qualität der Bewegtbildwiedergabe.* In: *FKTG-Jahrestagung 2006*, Potsdam, Deutschland, 2006.

[EBU02] EBU: *Technical Information I34-2002.* Technischer Bericht, European Broadcasting Union, 2002.

[FBS98] Franzen, O., H. Blume und H. Schröder: *FIR-Filter Design with Spatial and Frequency Design Constraints using Evolution Strategies.* Signal Processing Elsevier, 68(3):295–306, August 1998.

[FJP02] Freeman, W.T., T.R. Jones und E.C. Pasztor: *Example-Based Super-Resolution.* In: *IEEE Computer Graphics and Applications*, Band 22, April 2002.

[FS02] Franzen, O. und H. Schröder: *Nonlinear Polyphase Image Rate Conversion*. In: *Proceedings EUSIPCO, Toulouse, Frankreich*, Band 2, Seiten S. 522–525, 3.–6. September 2002.

[FTS01] Franzen, O., C. Tuschen und H. Schröder: *Intermediate Image Interpolation using Polyphase Weighted Median Filters*. In: *Proceedings of the IS&T SPIE Electronic Imaging, San Jose, USA*, Band 4304, Seiten S. 306–317, 20.–26. Januar 2001.

[Göt06] Götze, Jürgen: *Methoden der Informationstechnik. Skriptum zur Vorlesung*. Technischer Bericht, Universität Dortmund, Arbeitsgebiet Datentechnik, 2006.

[HA78] Hou, Hsieh und H. Andrews: *Cubic splines for image interpolation and digital filtering*. IEEE Transactions on Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], 26(6):508–517, Dec 1978.

[HCH03] Huang, Chin-Hui, Mei-Juan Chen und Ching-Ting Hsu: *Fast Edge-Oriented Image Interpolation Algorithm*. In: *Proceedings of 2003 Workshop on Consumer Electronics*. IEEE Consumer Electronics Society, Taipei Chapter IEEE Circuits and Systems Society, Taipei Chapter, 27. - 28. November 2003.

[HdH07] Hu, H. und G. de Haan: *Simultaneous Coding Artifact Red uction and Sharpness Enhancement*. In: *Digest of the ICCE*, 2007.

[Hen03] Hentschel, Christian: *Generic method for 2D image resizing with non-separable filters*. In: *International Conference on Consumer Electronics*, Seiten 1653–1656, 2003.

[HJO+01] Hertzmann, A., C. E. Jacobs, N. Oliver, B. Curless und D. H. Salesin: *Image Analogies*. In: *SIGGRAPH 2001 Conference Proceedings*, Seiten 327–340, August 12-16 2001. Los Angeles, California.

[HS07] Hentschel, Christian und Stefan Schiemenz: *Bildskalierer mit nichtlinearer Kantenanschärfung*. In: *ITG-Fachbericht 199, ITG/FKTG-Fachtagung Elektronische Medien, 12. Dortmunder Fernsehseminar*, 2007.

[ITU00] ITU: *Recommendation ITU-R BT.500-10 Methodology for the Subjective Assessment of the Quality of Television Pictures*. Technischer Bericht, ITU, 2000.

[Iva94] Ivanov, K.V.: *Ein neues Verfahren zur Konversion von Fernsehbildsignalen für die progressive Wiedergabe*. In: *FKT-Magazin*, Band 48, 1994.

[KFC01] Kondo, Tetsujiro, Yasuhiro Fujimori und Sugata Ghosal James J. Carrig: *Method and apparatus for adaptive filter tap selection according to a class*, February 2001. United States Patent US 6,192,161 B1, 20.02,2001.

[KK95] Kondo, Tetsujiro und Kunio Kawaguchi: *Adaptive dynamic range encoding method and apparatus*, August 1995. United States Patent US 5,444,487, 22.08,1995.

[KLL96] Kuo, Chung J., Ching Liao und Ching C. Lin: *Adaptive Interpolation Techniques for Scanning Rate Conversions.* IEEE Transactions on Circuits and Systems for Video Technology, 6(3):317–321, June 1996.

[KNF+01] Kondo, T., Y. Node, T. Fujiwara, und Y. Okumura: *Picture conversion apparatus, picture conversion method, learning apparatus and learning method*, November 2001. US patent: no. 6,323,905.

[LE07] Lenke, Sebastian und Oliver Erdler: *Detail- und Texturwiedergabe bei der SDTV -> HDTV Interpolation.* In: *12. Dortmunder Fernsehseminar Elektronische Medien*, Seiten 145–150, Dortmund, März 20-21 2007 2007.

[LH74] Lawson, C. L. und R. J. Hanson: *Solving Least Squares.* Prentice Hall, 1974.

[Li00] Li, Xin: *Edge directed statistical inference and its applications to image processing.* Doktorarbeit, Princeton University, November 2000.

[LO01] Li, Xin und M.T. Orchard: *New edge directed interpolation.* IEEE Transactions on Image Processing, 10:1521–1527, 10-13 Sept. 2001.

[LPB+00] Lee, Ho Young, Jin Woo Park, Tae Min Bae, Sang Um Choi und Yeong Ho Ha: *Adaptive scan rate upconversion system based on human visual characteristics.* IEEE Transactions on Consumer Electronics, 46(4):999–1006, November 2000. modified ELA.

[LZd03] Leitao, J.A., M. Zhao und G. deHaan: *Content-adaptive video up-scaling for high definition displays.* In: *IVCP 2003 Proc. IIS&T/SPIE Electronic Imaging 2003*, Band 5022, Santa Clara, CA, USA, January 2003. SPIE.

[ML08] Mösle, Frank und Sebastian Lenke: *Klassifikationsbasierte SD-HD-Interpolation.* 21. Tagung der FKTG, München, Germany, 26.05.2008 - 29.05.2008, 2008.

[PK04] Piastowski, Patrick und Guido Kohlmeyer: *Neues Verfahren zur kombinierten örtlich/zeitlichen Rauschreduktion und verbesserte Bewegungsschätzung für neue Display-Technologien.* In: *21. Tagung der FKTG Koblenz*, 2004.

[PPK03] Park, Sung C., Min K. Park und Moon G. Kang: *Super-resolution image reconstruction: a technical overview.* Signal Processing Magazine, IEEE, 20(3):21–36, 2003.

[Pro00] Proakis, John: *Digital Communications.* McGraw-Hill Science, Engineering, Math, August 2000.

[Rec94] Rechenberg, Ingo: *Evolutionsstrategien'94.* Frommann-Holzboog Verlag, Stuttgart, Germany, 1994.

[RG83] Reininger, R. und J. Gibson: *Distributions of the Two-Dimensional DCT Coefficients for Images.* IEEE Transactions on Communications, 31:835–839, 1983.

[RJM96] Robert J. Marks, II: *Alternating projections onto convex sets.* Seiten 476–501, 1996.

[SB00] Schröder, H. und H. Blume: *Mehrdimensionale Signalverarbeitung – Band 2: Architekturen und Anwendungen für Bilder und Bildsequenzen.* B.G. Teubner, 2000.

[SR06] Shi, Jiazheng und S.E. Reichenbach: *Image interpolation by two-dimensional parametric cubic convolution.* IEEE Transactions on Image Processing, 15(7):1857–1870, July 2006.

[SS96] Schultz, R. R. und R. L. Stevenson: *Extraction of High-Resolution Frames from Video Sequences.* IEEE Transactions on image processing, 5(6):996, June 1996.

[TOS92] Tekalp, A.M., M.K. Ozkan und M.I. Sezan: *High-resolution image reconstruction from lower-resolution imagesequences and space-varying image restoration.* In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Band 3, Seiten 169 – 172, 23-26 March 1992.

[ZCd05] Zhao, M., C. Ciuhu und G. deHaan: *Classification based data mixing for hybrid de-interlacing techniques.* In: *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*, September 2005.

[ZLd02] Zhao, M., J. A. Leitão und G. deHaan: *Towards an Overview of Spatial Up-conversion Techniques.* In: *Proceedings of ISCE'02*, 2002.

A Verwendete Sequenzen

A.1 Testsequenzen

Die in dieser Arbeit verwendeten Testsequenzen werden im Folgenden in alphabetischer Reihenfolge aufgelistet. Für jede Sequenz wird ein Bildbeispiel gegeben. Des Weiteren erfolgt eine kurze Beschreibung der wichtigsten Eigenschaften der Sequenzen.

Football

Die Sequenz Football zeigt einen Zoom auf ein Fußballspiel. In der Bildszene bewegen sich einige Spieler nach links (siehe Abbildung A.1). Im oberen Bildbereich sind viele stark ausgesprägte Kanten vorhanden (z. B. Treppen, Strommasten, Banden). Interessant für eine Interpolation sind zum einen die feinen Strukturen wie z. B. die Fußballspieler und der Rasen, aber auch die kontrastreichen Kanten stellen eine hohe Herausforderung für Interpolationsverfahren dar.

Abbildung A.1: Testsequenz Football

Lena

Die Sequenz Lena besteht aus einem Porträtfoto. Neben den großen fast homogenen Flächen mit Hauttönen sind auch feine Kantenverläufe in den Haaren, der Hutkrempe sowie den Augen vorhanden (siehe Abbildung A.2). Dieses Bild eignet sich auf Grund der Hautpartien sehr gut um zu testen, ob der natürliche Bildeindruck durch das Verfahren verfälscht wird.

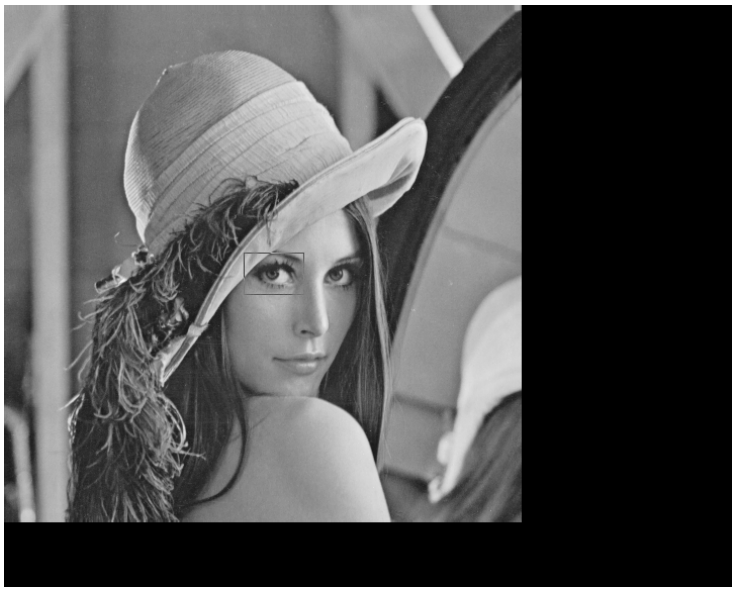


Abbildung A.2: Testsequenz Lena

TestChart

Die Sequenz TestChart ist eine künstlich erzeugte Testsequenz. Sie beinhaltet zum einen zwei Bereiche, in denen alle horizontalen und vertikalen Frequenzen getestet werden. Zusätzlich sind verschieden steile Kantenübergänge (Luminanzübergang 50 auf 200) vorhanden (siehe Abbildung A.3). Die Sequenz ist vor allem für das Testskript zur Bestimmung der Kantensteilheit wichtig, aber auch in den Frequenzsweeps kann die Erzeugung von Artefakten überprüft werden.

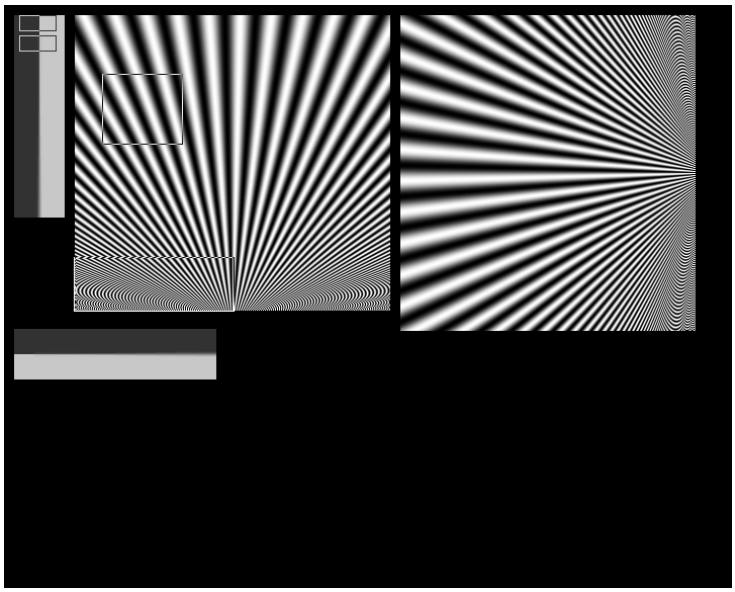


Abbildung A.3: Testsequenz TestChart

Wheel

Die Sequenz Wheel besteht im Wesentlichen aus zwei linksdrehenden Scheiben. Im Hintergrund dreht sich eine schwarze Platte, auf der mehrere Objekte befestigt sind. Im Vordergrund dreht sich ein Speichenrad. Durch die sich überlappenden Drehbereiche ist eine Bewegungsschätzung sehr schwierig.
Die Sequenz besteht zudem aus sehr kontrastreichen scharfen Kanten und vielen Detailsignalen, so dass sie ideal zum Test von Interpolationsalgorithmen ist (siehe Abbildung A.4).

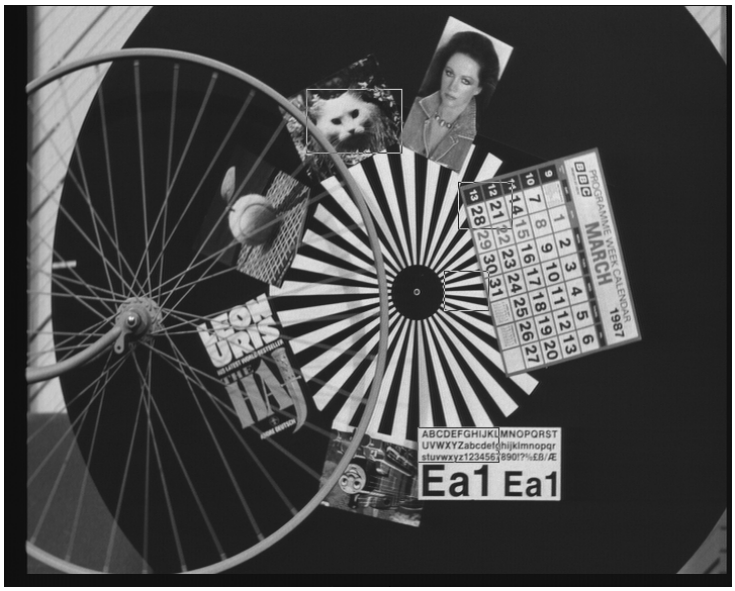


Abbildung A.4: Testsequenz Wheel

A.2 Trainingssequenz

BIGSEQ

Die Sequenz BIGSEQ wird im Rahmen dieser Arbeit als Trainingssequenz verwendet und besteht aus einer Vielzahl unterschiedlicher Bilder aus mehreren Bilddatenbanken. Ziel der Zusammenstellung ist es, möglichst viele typische Elemente von natürlichen Bildern in einer Sequenz zusammenzufassen.

(a) Frame 0 (b) Frame 4

Abbildung A.5: Trainingssequenz BIGSEQ

B Nebenbedingungen für das stückweise definierte Polynom

Bei der Kombination der Nebenbedingungen für die Bildqualität mit der Nutzung eines Interpolationspolynoms für den Optimerungsprozess ergeben sich auf Grund der hohen Zahl der Variablen (256) und Gleichungen (308) schwer zu überschaubare Formeln, die deswegen nur hier im Anhang angegeben werden.

1. $a_4 - 2\,a_8 + 4\,a_{12} - 8\,a_{16} = 0$
2. $a_3 - 2\,a_7 + 4\,a_{11} - 8\,a_{15} = 0$
3. $a_2 - 2\,a_6 + 4\,a_{10} - 8\,a_{14} = 0$
4. $a - 2\,a_5 + 4\,a_9 - 8\,a_{13} = 0$
5. $a_{20} - 2\,a_{24} + 4\,a_{28} - 8\,a_{32} = 0$
6. $a_{19} - 2\,a_{23} + 4\,a_{27} - 8\,a_{31} = 0$
7. $a_{18} - 2\,a_{22} + 4\,a_{26} - 8\,a_{30} = 0$
8. $a_{17} - 2\,a_{21} + 4\,a_{25} - 8\,a_{29} = 0$
9. $a_{36} - 2\,a_{40} + 4\,a_{44} - 8\,a_{48} = 0$
10. $a_{35} - 2\,a_{39} + 4\,a_{43} - 8\,a_{47} = 0$
11. $a_{34} - 2\,a_{38} + 4\,a_{42} - 8\,a_{46} = 0$
12. $a_{33} - 2\,a_{37} + 4\,a_{41} - 8\,a_{45} = 0$
13. $a_{52} - 2\,a_{56} + 4\,a_{60} - 8\,a_{64} = 0$
14. $a_{51} - 2\,a_{55} + 4\,a_{59} - 8\,a_{63} = 0$
15. $a_{50} - 2\,a_{54} + 4\,a_{58} - 8\,a_{62} = 0$
16. $a_{49} - 2\,a_{53} + 4\,a_{57} - 8\,a_{61} = 0$
17. $a_4 - a_8 + a_{12} - a_{16} - a_{68} + a_{72} - a_{76} + a_{80} = 0$
18. $a_3 - a_7 + a_{11} - a_{15} - a_{67} + a_{71} - a_{75} + a_{79} = 0$
19. $a_2 - a_6 + a_{10} - a_{14} - a_{66} + a_{70} - a_{74} + a_{78} = 0$
20. $a - a_5 + a_9 - a_{13} - a_{65} + a_{69} - a_{73} + a_{77} = 0$
21. $a_{20} - a_{24} + a_{28} - a_{32} - a_{84} + a_{88} - a_{92} + a_{96} = 0$
22. $a_{19} - a_{23} + a_{27} - a_{31} - a_{83} + a_{87} - a_{91} + a_{95} = 0$
23. $a_{18} - a_{22} + a_{26} - a_{30} - a_{82} + a_{86} - a_{90} + a_{94} = 0$
24. $a_{17} - a_{21} + a_{25} - a_{29} - a_{81} + a_{85} - a_{89} + a_{93} = 0$
25. $a_{36} - a_{40} + a_{44} - a_{48} - a_{100} + a_{104} - a_{108} + a_{112} = 0$
26. $a_{35} - a_{39} + a_{43} - a_{47} - a_{99} + a_{103} - a_{107} + a_{111} = 0$
27. $a_{34} - a_{38} + a_{42} - a_{46} - a_{98} + a_{102} - a_{106} + a_{110} = 0$
28. $a_{33} - a_{37} + a_{41} - a_{45} - a_{97} + a_{101} - a_{105} + a_{109} = 0$
29. $a_{52} - a_{56} + a_{60} - a_{64} - a_{116} + a_{120} - a_{124} + a_{128} = 0$
30. $a_{51} - a_{55} + a_{59} - a_{63} - a_{115} + a_{119} - a_{123} + a_{127} = 0$
31. $a_{50} - a_{54} + a_{58} - a_{62} - a_{114} + a_{118} - a_{122} + a_{126} = 0$
32. $a_{49} - a_{53} + a_{57} - a_{61} - a_{113} + a_{117} - a_{121} + a_{125} = 0$
33. $a_{68} - a_{132} = 0$
34. $a_{67} - a_{131} = 0$
35. $a_{66} - a_{130} = 0$
36. $a_{65} - a_{129} = 0$
37. $a_{84} - a_{148} = 0$
38. $a_{83} - a_{147} = 0$
39. $a_{82} - a_{146} = 0$
40. $a_{81} - a_{145} = 0$
41. $a_{100} - a_{164} = 0$
42. $a_{99} - a_{163} = 0$
43. $a_{98} - a_{162} = 0$

44. $a_{97} - a_{161} = 0$

45. $a_{116} - a_{180} = 0$

46. $a_{115} - a_{179} = 0$

47. $a_{114} - a_{178} = 0$

48. $a_{113} - a_{177} = 0$

49. $a_{132} + a_{136} + a_{140} + a_{144} - a_{196} - a_{200} - a_{204} - a_{208} = 0$

50. $a_{131} + a_{135} + a_{139} + a_{143} - a_{195} - a_{199} - a_{203} - a_{207} = 0$

51. $a_{130} + a_{134} + a_{138} + a_{142} - a_{194} - a_{198} - a_{202} - a_{206} = 0$

52. $a_{129} + a_{133} + a_{137} + a_{141} - a_{193} - a_{197} - a_{201} - a_{205} = 0$

53. $a_{148} + a_{152} + a_{156} + a_{160} - a_{212} - a_{216} - a_{220} - a_{224} = 0$

54. $a_{147} + a_{151} + a_{155} + a_{159} - a_{211} - a_{215} - a_{219} - a_{223} = 0$

55. $a_{146} + a_{150} + a_{154} + a_{158} - a_{210} - a_{214} - a_{218} - a_{222} = 0$

56. $a_{145} + a_{149} + a_{153} + a_{157} - a_{209} - a_{213} - a_{217} - a_{221} = 0$

57. $a_{164} + a_{168} + a_{172} + a_{176} - a_{228} - a_{232} - a_{236} - a_{240} = 0$

58. $a_{163} + a_{167} + a_{171} + a_{175} - a_{227} - a_{231} - a_{235} - a_{239} = 0$

59. $a_{162} + a_{166} + a_{170} + a_{174} - a_{226} - a_{230} - a_{234} - a_{238} = 0$

60. $a_{161} + a_{165} + a_{169} + a_{173} - a_{225} - a_{229} - a_{233} - a_{237} = 0$

61. $a_{180} + a_{184} + a_{188} + a_{192} - a_{244} - a_{248} - a_{252} - a_{256} = 0$

62. $a_{179} + a_{183} + a_{187} + a_{191} - a_{243} - a_{247} - a_{251} - a_{255} = 0$

63. $a_{178} + a_{182} + a_{186} + a_{190} - a_{242} - a_{246} - a_{250} - a_{254} = 0$

64. $a_{177} + a_{181} + a_{185} + a_{189} - a_{241} - a_{245} - a_{249} - a_{253} = 0$

65. $a_{196} + 2\,a_{200} + 4\,a_{204} + 8\,a_{208} = 0$

66. $a_{195} + 2\,a_{199} + 4\,a_{203} + 8\,a_{207} = 0$

67. $a_{194} + 2\,a_{198} + 4\,a_{202} + 8\,a_{206} = 0$

68. $a_{193} + 2\,a_{197} + 4\,a_{201} + 8\,a_{205} = 0$

69. $a_{212} + 2\,a_{216} + 4\,a_{220} + 8\,a_{224} = 0$

70. $a_{211} + 2\,a_{215} + 4\,a_{219} + 8\,a_{223} = 0$

71. $a_{210} + 2\,a_{214} + 4\,a_{218} + 8\,a_{222} = 0$

72. $a_{209} + 2\,a_{213} + 4\,a_{217} + 8\,a_{221} = 0$

73. $a_{228} + 2\,a_{232} + 4\,a_{236} + 8\,a_{240} = 0$

74. $a_{227} + 2\,a_{231} + 4\,a_{235} + 8\,a_{239} = 0$

75. $a_{226} + 2\,a_{230} + 4\,a_{234} + 8\,a_{238} = 0$

76. $a_{225} + 2\,a_{229} + 4\,a_{233} + 8\,a_{237} = 0$

77. $a_{244} + 2\,a_{248} + 4\,a_{252} + 8\,a_{256} = 0$

78. $a_{243} + 2\,a_{247} + 4\,a_{251} + 8\,a_{255} = 0$

79. $a_{242} + 2\,a_{246} + 4\,a_{250} + 8\,a_{254} = 0$

80. $a_{241} + 2\,a_{245} + 4\,a_{249} + 8\,a_{253} = 0$

81. $a_{13} - 2\,a_{14} + 4\,a_{15} - 8\,a_{16} = 0$

82. $a_{9} - 2\,a_{10} + 4\,a_{11} - 8\,a_{12} = 0$

83. $a_{5} - 2\,a_{6} + 4\,a_{7} - 8\,a_{8} = 0$

84. $a - 2\,a_{2} + 4\,a_{3} - 8\,a_{4} = 0$

85. $a_{77} - 2\,a_{78} + 4\,a_{79} - 8\,a_{80} = 0$

86. $a_{73} - 2\,a_{74} + 4\,a_{75} - 8\,a_{76} = 0$
87. $a_{69} - 2\,a_{70} + 4\,a_{71} - 8\,a_{72} = 0$
88. $a_{65} - 2\,a_{66} + 4\,a_{67} - 8\,a_{68} = 0$
89. $a_{141} - 2\,a_{142} + 4\,a_{143} - 8\,a_{144} = 0$
90. $a_{137} - 2\,a_{138} + 4\,a_{139} - 8\,a_{140} = 0$
91. $a_{133} - 2\,a_{134} + 4\,a_{135} - 8\,a_{136} = 0$
92. $a_{129} - 2\,a_{130} + 4\,a_{131} - 8\,a_{132} = 0$
93. $a_{205} - 2\,a_{206} + 4\,a_{207} - 8\,a_{208} = 0$
94. $a_{201} - 2\,a_{202} + 4\,a_{203} - 8\,a_{204} = 0$
95. $a_{197} - 2\,a_{198} + 4\,a_{199} - 8\,a_{200} = 0$
96. $a_{193} - 2\,a_{194} + 4\,a_{195} - 8\,a_{196} = 0$
97. $a_{13} - a_{14} + a_{15} - a_{16} - a_{29} + a_{30} - a_{31} + a_{32} = 0$
98. $a_{9} - a_{10} + a_{11} - a_{12} - a_{25} + a_{26} - a_{27} + a_{28} = 0$
99. $a_{5} - a_{6} + a_{7} - a_{8} - a_{21} + a_{22} - a_{23} + a_{24} = 0$
100. $a - a_{2} + a_{3} - a_{4} - a_{17} + a_{18} - a_{19} + a_{20} = 0$
101. $a_{77} - a_{78} + a_{79} - a_{80} - a_{93} + a_{94} - a_{95} + a_{96} = 0$
102. $a_{73} - a_{74} + a_{75} - a_{76} - a_{89} + a_{90} - a_{91} + a_{92} = 0$
103. $a_{69} - a_{70} + a_{71} - a_{72} - a_{85} + a_{86} - a_{87} + a_{88} = 0$
104. $a_{65} - a_{66} + a_{67} - a_{68} - a_{81} + a_{82} - a_{83} + a_{84} = 0$
105. $a_{141} - a_{142} + a_{143} - a_{144} - a_{157} + a_{158} - a_{159} + a_{160} = 0$
106. $a_{137} - a_{138} + a_{139} - a_{140} - a_{153} + a_{154} - a_{155} + a_{156} = 0$
107. $a_{133} - a_{134} + a_{135} - a_{136} - a_{149} + a_{150} - a_{151} + a_{152} = 0$
108. $a_{129} - a_{130} + a_{131} - a_{132} - a_{145} + a_{146} - a_{147} + a_{148} = 0$
109. $a_{205} - a_{206} + a_{207} - a_{208} - a_{221} + a_{222} - a_{223} + a_{224} = 0$
110. $a_{201} - a_{202} + a_{203} - a_{204} - a_{217} + a_{218} - a_{219} + a_{220} = 0$
111. $a_{197} - a_{198} + a_{199} - a_{200} - a_{213} + a_{214} - a_{215} + a_{216} = 0$
112. $a_{193} - a_{194} + a_{195} - a_{196} - a_{209} + a_{210} - a_{211} + a_{212} = 0$
113. $a_{29} - a_{45} = 0$
114. $a_{25} - a_{41} = 0$
115. $a_{21} - a_{37} = 0$
116. $a_{17} - a_{33} = 0$
117. $a_{93} - a_{109} = 0$
118. $a_{89} - a_{105} = 0$
119. $a_{85} - a_{101} = 0$
120. $a_{81} - a_{97} = 0$
121. $a_{157} - a_{173} = 0$
122. $a_{153} - a_{169} = 0$
123. $a_{149} - a_{165} = 0$
124. $a_{145} - a_{161} = 0$
125. $a_{221} - a_{237} = 0$
126. $a_{217} - a_{233} = 0$
127. $a_{213} - a_{229} = 0$
128. $a_{209} - a_{225} = 0$

129. $a_{45}+a_{46}+a_{47}+a_{48}-a_{61}-a_{62}-a_{63}-a_{64}=0$

130. $a_{41}+a_{42}+a_{43}+a_{44}-a_{57}-a_{58}-a_{59}-a_{60}=0$

131. $a_{37}+a_{38}+a_{39}+a_{40}-a_{53}-a_{54}-a_{55}-a_{56}=0$

132. $a_{33}+a_{34}+a_{35}+a_{36}-a_{49}-a_{50}-a_{51}-a_{52}=0$

133. $a_{109}+a_{110}+a_{111}+a_{112}-a_{125}-a_{126}-a_{127}-a_{128}=0$

134. $a_{105}+a_{106}+a_{107}+a_{108}-a_{121}-a_{122}-a_{123}-a_{124}=0$

135. $a_{101}+a_{102}+a_{103}+a_{104}-a_{117}-a_{118}-a_{119}-a_{120}=0$

136. $a_{97}+a_{98}+a_{99}+a_{100}-a_{113}-a_{114}-a_{115}-a_{116}=0$

137. $a_{173}+a_{174}+a_{175}+a_{176}-a_{189}-a_{190}-a_{191}-a_{192}=0$

138. $a_{169}+a_{170}+a_{171}+a_{172}-a_{185}-a_{186}-a_{187}-a_{188}=0$

139. $a_{165}+a_{166}+a_{167}+a_{168}-a_{181}-a_{182}-a_{183}-a_{184}=0$

140. $a_{161}+a_{162}+a_{163}+a_{164}-a_{177}-a_{178}-a_{179}-a_{180}=0$

141. $a_{237}+a_{238}+a_{239}+a_{240}-a_{253}-a_{254}-a_{255}-a_{256}=0$

142. $a_{233}+a_{234}+a_{235}+a_{236}-a_{249}-a_{250}-a_{251}-a_{252}=0$

143. $a_{229}+a_{230}+a_{231}+a_{232}-a_{245}-a_{246}-a_{247}-a_{248}=0$

144. $a_{225}+a_{226}+a_{227}+a_{228}-a_{241}-a_{242}-a_{243}-a_{244}=0$

145. $a_{61}+2\,a_{62}+4\,a_{63}+8\,a_{64}=0$

146. $a_{57}+2\,a_{58}+4\,a_{59}+8\,a_{60}=0$

147. $a_{53}+2\,a_{54}+4\,a_{55}+8\,a_{56}=0$

148. $a_{49}+2\,a_{50}+4\,a_{51}+8\,a_{52}=0$

149. $a_{125}+2\,a_{126}+4\,a_{127}+8\,a_{128}=0$

150. $a_{121}+2\,a_{122}+4\,a_{123}+8\,a_{124}=0$

151. $a_{117}+2\,a_{118}+4\,a_{119}+8\,a_{120}=0$

152. $a_{113}+2\,a_{114}+4\,a_{115}+8\,a_{116}=0$

153. $a_{189}+2\,a_{190}+4\,a_{191}+8\,a_{192}=0$

154. $a_{185}+2\,a_{186}+4\,a_{187}+8\,a_{188}=0$

155. $a_{181}+2\,a_{182}+4\,a_{183}+8\,a_{184}=0$

156. $a_{177}+2\,a_{178}+4\,a_{179}+8\,a_{180}=0$

157. $a_{253}+2\,a_{254}+4\,a_{255}+8\,a_{256}=0$

158. $a_{249}+2\,a_{250}+4\,a_{251}+8\,a_{252}=0$

159. $a_{245}+2\,a_{246}+4\,a_{247}+8\,a_{248}=0$

160. $a_{241}+2\,a_{242}+4\,a_{243}+8\,a_{244}=0$

161. $a_{81}=0$

162. $a_{97}=0$

163. $a_{145}=0$

164. $a_{161}=0$

165. $a_{14}-2\,a_{15}+3\,a_{16}-a_{30}+2\,a_{31}-3\,a_{32}=0$

166. $a_{10}-2\,a_{11}+3\,a_{12}-a_{26}+2\,a_{27}-3\,a_{28}=0$

167. $a_{6}-2\,a_{7}+3\,a_{8}-a_{22}+2\,a_{23}-3\,a_{24}=0$

168. $a_{2}-2\,a_{3}+3\,a_{4}-a_{18}+2\,a_{19}-3\,a_{20}=0$

169. $a_{78}-2\,a_{79}+3\,a_{80}-a_{94}+2\,a_{95}-3\,a_{96}=0$

170. $a_{74}-2\,a_{75}+3\,a_{76}-a_{90}+2\,a_{91}-3\,a_{92}=0$

171. $a_{70} - 2\,a_{71} + 3\,a_{72} - a_{86} + 2\,a_{87} - 3\,a_{88} = 0$
172. $a_{66} - 2\,a_{67} + 3\,a_{68} - a_{82} + 2\,a_{83} - 3\,a_{84} = 0$
173. $a_{142} - 2\,a_{143} + 3\,a_{144} - a_{158} + 2\,a_{159} - 3\,a_{160} = 0$
174. $a_{138} - 2\,a_{139} + 3\,a_{140} - a_{154} + 2\,a_{155} - 3\,a_{156} = 0$
175. $a_{134} - 2\,a_{135} + 3\,a_{136} - a_{150} + 2\,a_{151} - 3\,a_{152} = 0$
176. $a_{130} - 2\,a_{131} + 3\,a_{132} - a_{146} + 2\,a_{147} - 3\,a_{148} = 0$
177. $a_{206} - 2\,a_{207} + 3\,a_{208} - a_{222} + 2\,a_{223} - 3\,a_{224} = 0$
178. $a_{202} - 2\,a_{203} + 3\,a_{204} - a_{218} + 2\,a_{219} - 3\,a_{220} = 0$
179. $a_{198} - 2\,a_{199} + 3\,a_{200} - a_{214} + 2\,a_{215} - 3\,a_{216} = 0$
180. $a_{194} - 2\,a_{195} + 3\,a_{196} - a_{210} + 2\,a_{211} - 3\,a_{212} = 0$
181. $a_{30} - a_{46} = 0$
182. $a_{26} - a_{42} = 0$
183. $a_{22} - a_{38} = 0$
184. $a_{18} - a_{34} = 0$
185. $a_{94} - a_{110} = 0$
186. $a_{90} - a_{106} = 0$
187. $a_{86} - a_{102} = 0$
188. $a_{82} - a_{98} = 0$
189. $a_{158} - a_{174} = 0$
190. $a_{154} - a_{170} = 0$
191. $a_{150} - a_{166} = 0$
192. $a_{146} - a_{162} = 0$
193. $a_{222} - a_{238} = 0$
194. $a_{218} - a_{234} = 0$
195. $a_{214} - a_{230} = 0$
196. $a_{210} - a_{226} = 0$
197. $a_{46} + 2\,a_{47} + 3\,a_{48} - a_{62} - 2\,a_{63} - 3\,a_{64} = 0$
198. $a_{42} + 2\,a_{43} + 3\,a_{44} - a_{58} - 2\,a_{59} - 3\,a_{60} = 0$
199. $a_{38} + 2\,a_{39} + 3\,a_{40} - a_{54} - 2\,a_{55} - 3\,a_{56} = 0$
200. $a_{34} + 2\,a_{35} + 3\,a_{36} - a_{50} - 2\,a_{51} - 3\,a_{52} = 0$
201. $a_{110} + 2\,a_{111} + 3\,a_{112} - a_{126} - 2\,a_{127} - 3\,a_{128} = 0$
202. $a_{106} + 2\,a_{107} + 3\,a_{108} - a_{122} - 2\,a_{123} - 3\,a_{124} = 0$
203. $a_{102} + 2\,a_{103} + 3\,a_{104} - a_{118} - 2\,a_{119} - 3\,a_{120} = 0$
204. $a_{98} + 2\,a_{99} + 3\,a_{100} - a_{114} - 2\,a_{115} - 3\,a_{116} = 0$
205. $a_{174} + 2\,a_{175} + 3\,a_{176} - a_{190} - 2\,a_{191} - 3\,a_{192} = 0$
206. $a_{170} + 2\,a_{171} + 3\,a_{172} - a_{186} - 2\,a_{187} - 3\,a_{188} = 0$
207. $a_{166} + 2\,a_{167} + 3\,a_{168} - a_{182} - 2\,a_{183} - 3\,a_{184} = 0$
208. $a_{162} + 2\,a_{163} + 3\,a_{164} - a_{178} - 2\,a_{179} - 3\,a_{180} = 0$
209. $a_{238} + 2\,a_{239} + 3\,a_{240} - a_{254} - 2\,a_{255} - 3\,a_{256} = 0$
210. $a_{234} + 2\,a_{235} + 3\,a_{236} - a_{250} - 2\,a_{251} - 3\,a_{252} = 0$

211. $a_{230} + 2\,a_{231} + 3\,a_{232} - a_{246} - 2\,a_{247} - 3\,a_{248} = 0$

212. $a_{226} + 2\,a_{227} + 3\,a_{228} - a_{242} - 2\,a_{243} - 3\,a_{244} = 0$

213. $a_8 - 2\,a_{12} + 3\,a_{16} - a_{72} + 2\,a_{76} - 3\,a_{80} = 0$

214. $a_7 - 2\,a_{11} + 3\,a_{15} - a_{71} + 2\,a_{75} - 3\,a_{79} = 0$

215. $a_6 - 2\,a_{10} + 3\,a_{14} - a_{70} + 2\,a_{74} - 3\,a_{78} = 0$

216. $a_5 - 2\,a_9 + 3\,a_{13} - a_{69} + 2\,a_{73} - 3\,a_{77} = 0$

217. $a_{24} - 2\,a_{28} + 3\,a_{32} - a_{88} + 2\,a_{92} - 3\,a_{96} = 0$

218. $a_{23} - 2\,a_{27} + 3\,a_{31} - a_{87} + 2\,a_{91} - 3\,a_{95} = 0$

219. $a_{22} - 2\,a_{26} + 3\,a_{30} - a_{86} + 2\,a_{90} - 3\,a_{94} = 0$

220. $a_{21} - 2\,a_{25} + 3\,a_{29} - a_{85} + 2\,a_{89} - 3\,a_{93} = 0$

221. $a_{40} - 2\,a_{44} + 3\,a_{48} - a_{104} + 2\,a_{108} - 3\,a_{112} = 0$

222. $a_{39} - 2\,a_{43} + 3\,a_{47} - a_{103} + 2\,a_{107} - 3\,a_{111} = 0$

223. $a_{38} - 2\,a_{42} + 3\,a_{46} - a_{102} + 2\,a_{106} - 3\,a_{110} = 0$

224. $a_{37} - 2\,a_{41} + 3\,a_{45} - a_{101} + 2\,a_{105} - 3\,a_{109} = 0$

225. $a_{56} - 2\,a_{60} + 3\,a_{64} - a_{120} + 2\,a_{124} - 3\,a_{128} = 0$

226. $a_{55} - 2\,a_{59} + 3\,a_{63} - a_{119} + 2\,a_{123} - 3\,a_{127} = 0$

227. $a_{54} - 2\,a_{58} + 3\,a_{62} - a_{118} + 2\,a_{122} - 3\,a_{126} = 0$

228. $a_{53} - 2\,a_{57} + 3\,a_{61} - a_{117} + 2\,a_{121} - 3\,a_{125} = 0$

229. $a_{72} - a_{136} = 0$

230. $a_{71} - a_{135} = 0$

231. $a_{70} - a_{134} = 0$

232. $a_{69} - a_{133} = 0$

233. $a_{88} - a_{152} = 0$

234. $a_{87} - a_{151} = 0$

235. $a_{86} - a_{150} = 0$

236. $a_{85} - a_{149} = 0$

237. $a_{104} - a_{168} = 0$

238. $a_{103} - a_{167} = 0$

239. $a_{102} - a_{166} = 0$

240. $a_{101} - a_{165} = 0$

241. $a_{120} - a_{184} = 0$

242. $a_{119} - a_{183} = 0$

243. $a_{118} - a_{182} = 0$

244. $a_{117} - a_{181} = 0$

245. $a_{136} + 2\,a_{140} + 3\,a_{144} - a_{200} - 2\,a_{204} - 3\,a_{208} = 0$

246. $a_{135} + 2\,a_{139} + 3\,a_{143} - a_{199} - 2\,a_{203} - 3\,a_{207} = 0$

247. $a_{134} + 2\,a_{138} + 3\,a_{142} - a_{198} - 2\,a_{202} - 3\,a_{206} = 0$

248. $a_{133} + 2\,a_{137} + 3\,a_{141} - a_{197} - 2\,a_{201} - 3\,a_{205} = 0$

249. $a_{152} + 2\,a_{156} + 3\,a_{160} - a_{216} - 2\,a_{220} - 3\,a_{224} = 0$

250. $a_{151} + 2\,a_{155} + 3\,a_{159} - a_{215} - 2\,a_{219} - 3\,a_{223} = 0$

251. $a_{150} + 2\,a_{154} + 3\,a_{158} - a_{214} - 2\,a_{218} - 3\,a_{222} = 0$

252. $a_{149} + 2\,a_{153} + 3\,a_{157} - a_{213} - 2\,a_{217} - 3\,a_{221} = 0$

253. $a_{168} + 2\,a_{172} + 3\,a_{176} - a_{232} - 2\,a_{236} - 3\,a_{240} = 0$

254. $a_{167} + 2\,a_{171} + 3\,a_{175} - a_{231} - 2\,a_{235} - 3\,a_{239} = 0$

255. $a_{166} + 2\,a_{170} + 3\,a_{174} - a_{230} - 2\,a_{234} - 3\,a_{238} = 0$

256. $a_{165} + 2\,a_{169} + 3\,a_{173} - a_{229} - 2\,a_{233} - 3\,a_{237} = 0$

257. $a_{184} + 2\,a_{188} + 3\,a_{192} - a_{248} - 2\,a_{252} - 3\,a_{256} = 0$

258. $a_{183} + 2\,a_{187} + 3\,a_{191} - a_{247} - 2\,a_{251} - 3\,a_{255} = 0$

259. $a_{182} + 2\,a_{186} + 3\,a_{190} - a_{246} - 2\,a_{250} - 3\,a_{254} = 0$

260. $a_{181} + 2\,a_{185} + 3\,a_{189} - a_{245} - 2\,a_{249} - 3\,a_{253} = 0$

261. $a_{208} + a_{48} + a_{224} + a_{16} + a_{112} + a_{32} + a_{64} + a_{256} + a_{144} + a_{192} + a_{80} + a_{96} + a_{176} + a_{128} + a_{160} + a_{240} = 0$

262. $-3\,a_{80} + a_{79} + a_{31} + a_{15} + a_{143} + a_{159} - 3\,a_{16} + a_{63} + a_{47} + a_{223} + a_{175} + 3\,a_{176} + 3\,a_{48} - 3\,a_{144} + 6\,a_{128} + a_{127} + a_{255} + 6\,a_{256} + a_{95} + a_{207} - 3\,a_{208} + 6\,a_{64} + a_{191} + 6\,a_{192} + a_{239} + 3\,a_{240} + 3\,a_{112} + a_{111} = 0$

263. $3\,a_{80} - 2\,a_{79} + a_{30} - 2\,a_{15} + a_{14} + a_{142} - 2\,a_{143} + a_{158} + 3\,a_{16} + a_{62} + 4\,a_{63} + a_{46} + 2\,a_{47} + a_{222} + a_{174} + 2\,a_{175} + 3\,a_{176} + 3\,a_{48} + 3\,a_{144} + 12\,a_{128} + 4\,a_{127} + a_{126} + a_{78} + a_{254} + 4\,a_{255} + 12\,a_{256} + a_{94} + a_{206} - 2\,a_{207} + 3\,a_{208} + 12\,a_{64} + a_{190} + 4\,a_{191} + 12\,a_{192} + a_{238} + 2\,a_{239} + 3\,a_{240} + 3\,a_{112} + 2\,a_{111} + a_{110} = 0$

264. $-a_{80} + a_{79} + a_{157} + a_{29} + a_{141} + a_{15} - a_{14} - a_{142} + a_{143} + a_{13} - a_{16} + a_{61} + 2\,a_{62} + 4\,a_{63} + a_{45} + a_{46} + a_{47} + a_{221} + a_{173} + a_{174} + a_{175} + a_{176} + a_{48} - a_{144} + 8\,a_{128} + 4\,a_{127} + 2\,a_{126} + a_{125} - a_{78} + a_{253} + 2\,a_{254} + 4\,a_{255} + 8\,a_{256} + a_{77} + a_{205} - a_{206} + a_{207} - a_{208} + 8\,a_{64} + a_{189} + 2\,a_{190} + 4\,a_{191} + 8\,a_{192} + a_{237} + a_{238} + a_{239} + a_{240} + a_{93} + a_{112} + a_{111} + a_{110} + a_{109} = 0$

265. $a_{76} + a_{28} + a_{156} - 3\,a_{32} + a_{12} + 3\,a_{160} - 3\,a_{16} + a_{60} + a_{44} + a_{220} + 6\,a_{224} + a_{172} + 3\,a_{176} + a_{140} - 3\,a_{48} + 3\,a_{144} + a_{124} + a_{252} + 6\,a_{256} + a_{92} + a_{204} + 6\,a_{208} - 3\,a_{64} + a_{188} + 3\,a_{192} + a_{236} + 6\,a_{240} + a_{108} = 0$

266. $-3\,a_{76} + a_{75} + a_{155} - 3\,a_{31} - 3\,a_{12} + a_{11} - 3\,a_{15} + 3\,a_{143} + 3\,a_{159} + 9\,a_{16} + a_{59} + 6\,a_{60} - 3\,a_{63} + a_{43} + 3\,a_{44} - 3\,a_{47} + a_{219} + 6\,a_{223} + a_{171} + 3\,a_{172} + 3\,a_{175} + 9\,a_{176} - 3\,a_{140} - 9\,a_{48} - 9\,a_{144} + 6\,a_{124} + a_{123} + a_{251} + 6\,a_{252} + 6\,a_{255} + 36\,a_{256} + a_{91} + a_{27} + a_{203} - 3\,a_{204} + 6\,a_{207} - 18\,a_{208} - 18\,a_{64} + a_{187} + 6\,a_{188} + 3\,a_{191} + 18\,a_{192} + a_{235} + 3\,a_{236} + 6\,a_{239} + 18\,a_{240} + a_{139} + 3\,a_{108} + a_{107} = 0$

267. $3\,a_{76} - 2\,a_{75} + a_{154} - 3\,a_{30} + 3\,a_{12} - 2\,a_{11} + 6\,a_{15} - 3\,a_{14} + 3\,a_{142} - 6\,a_{143} + 3\,a_{158} - 9\,a_{16} + a_{58} + 4\,a_{59} + 12\,a_{60} - 3\,a_{62} - 12\,a_{63} + a_{42} + 2\,a_{43} + 3\,a_{44} - 3\,a_{46} - 6\,a_{47} + 6\,a_{222} + a_{170} + 2\,a_{171} + 3\,a_{172} + 3\,a_{174} + 6\,a_{175} + 9\,a_{176} + 3\,a_{140} - 9\,a_{48} + 9\,a_{144} + a_{218} + 12\,a_{124} + 4\,a_{123} + a_{122} + a_{74} + 4\,a_{251} + 12\,a_{252} + 6\,a_{254} + 24\,a_{255} + 72\,a_{256} + a_{138} + a_{90} - 2\,a_{203} + 3\,a_{204} + 6\,a_{206} - 12\,a_{207} + 18\,a_{208} + a_{202} - 36\,a_{64} + a_{186} + 4\,a_{187} + 12\,a_{188} + 3\,a_{190} + 12\,a_{191} + 36\,a_{192} + a_{250} + a_{10} + a_{234} + 2\,a_{235} + 3\,a_{236} + 6\,a_{238} + 12\,a_{239} + 18\,a_{240} - 2\,a_{139} + 3\,a_{108} + 2\,a_{107} + a_{106} + a_{26} = 0$

268. $-a_{76} + a_{75} + a_{25} + a_{153} + 3\,a_{157} - 3\,a_{29} + 3\,a_{141} - a_{12} + a_{11} - 3\,a_{15} + 3\,a_{14} - 3\,a_{142} + 3\,a_{143} - 3\,a_{13} + 3\,a_{16} + a_{57} + 2\,a_{58} + 4\,a_{59} + 8\,a_{60} - 3\,a_{61} - 6\,a_{62} - 12\,a_{63} + a_{41} + a_{42} + a_{43} + a_{44} - 3\,a_{45} - 3\,a_{46} - 3\,a_{47} + a_{9} + a_{217} + 6\,a_{221} + a_{169} + a_{170} + a_{171} + a_{172} + 3\,a_{173} + 3\,a_{174} + 3\,a_{175} + 3\,a_{176} - a_{140} - 3\,a_{48} - 3\,a_{144} + a_{137} + 8\,a_{124} + 4\,a_{123} + 2\,a_{122} + a_{121} - a_{74} + a_{249} + 4\,a_{251} + 8\,a_{252} + 6\,a_{253} + 12\,a_{254} + 24\,a_{255} + 48\,a_{256} - a_{138} + a_{73} + a_{89} + a_{203} - a_{204} + 6\,a_{205} - 6\,a_{206} + 6\,a_{207} - 6\,a_{208} + a_{201} - a_{202} - 24\,a_{64} + a_{185} +$

$2\,a_{186} + 4\,a_{187} + 8\,a_{188} + 3\,a_{189} + 6\,a_{190} + 12\,a_{191} + 24\,a_{192} + 2\,a_{250} - a_{10} + a_{233} + a_{234} + a_{235} + a_{236} + 6\,a_{237} + 6\,a_{238} + 6\,a_{239} + 6\,a_{240} + a_{139} + a_{108} + a_{107} + a_{106} + a_{105} = 0$

269. $a_{72} + a_{24} - 2\,a_{28} + a_{152} + 2\,a_{156} + 3\,a_{32} - 2\,a_{12} + a_{136} + 3\,a_{160} + 3\,a_{16} + a_{56} - 2\,a_{60} + a_{40} - 2\,a_{44} + a_{8} + a_{216} + 4\,a_{220} + 12\,a_{224} + a_{168} + 2\,a_{172} + 3\,a_{176} + 2\,a_{140} + 3\,a_{48} + 3\,a_{144} + a_{120} + a_{248} + 4\,a_{252} + 12\,a_{256} + a_{88} + 4\,a_{204} + 12\,a_{208} + a_{200} + 3\,a_{64} + a_{184} + 2\,a_{188} + 3\,a_{192} + a_{232} + 4\,a_{236} + 12\,a_{240} + a_{104} = 0$

270. $-3\,a_{72} + a_{71} + a_{23} + a_{151} + 2\,a_{155} + a_{135} + 3\,a_{31} + 6\,a_{12} - 2\,a_{11} + 3\,a_{15} - 3\,a_{136} + 3\,a_{143} + 3\,a_{159} - 9\,a_{16} + a_{55} + 6\,a_{56} - 2\,a_{59} - 12\,a_{60} + 3\,a_{63} + a_{39} + 3\,a_{40} - 2\,a_{43} - 6\,a_{44} + 3\,a_{47} + a_{7} - 3\,a_{8} + a_{215} + 4\,a_{219} + 12\,a_{223} + a_{167} + 3\,a_{168} + 2\,a_{171} + 6\,a_{172} + 3\,a_{175} + 9\,a_{176} - 6\,a_{140} + 9\,a_{48} - 9\,a_{144} + 6\,a_{120} + a_{119} + a_{247} + 6\,a_{248} + 4\,a_{251} + 24\,a_{252} + 12\,a_{255} + 72\,a_{256} + a_{87} - 2\,a_{27} + 4\,a_{203} - 12\,a_{204} + 12\,a_{207} - 36\,a_{208} - 3\,a_{200} + a_{199} + 18\,a_{64} + a_{183} + 6\,a_{184} + 2\,a_{187} + 12\,a_{188} + 3\,a_{191} + 18\,a_{192} + a_{231} + 3\,a_{232} + 4\,a_{235} + 12\,a_{236} + 12\,a_{239} + 36\,a_{240} + 2\,a_{139} + 3\,a_{104} + a_{103} = 0$

271. $3\,a_{72} - 2\,a_{71} + a_{150} + 2\,a_{154} + a_{22} - 2\,a_{135} + 3\,a_{30} + a_{134} - 6\,a_{12} + 4\,a_{11} - 6\,a_{15} + 3\,a_{14} + 3\,a_{136} + 3\,a_{142} - 6\,a_{143} + 3\,a_{158} + 9\,a_{16} + a_{54} + 4\,a_{55} + 12\,a_{56} - 2\,a_{58} - 8\,a_{59} - 24\,a_{60} + 3\,a_{62} + 12\,a_{63} + a_{38} + 2\,a_{39} + 3\,a_{40} - 2\,a_{42} - 4\,a_{43} - 6\,a_{44} + 3\,a_{46} + 6\,a_{47} + a_{6} - 2\,a_{7} + 3\,a_{8} + a_{214} + 12\,a_{222} + a_{166} + 2\,a_{167} + 3\,a_{168} + 2\,a_{170} + 4\,a_{171} + 6\,a_{172} + 3\,a_{174} + 6\,a_{175} + 9\,a_{176} + 6\,a_{140} + 9\,a_{48} + 9\,a_{144} + 4\,a_{218} + 12\,a_{120} + 4\,a_{119} + a_{118} + a_{70} + a_{246} + 4\,a_{247} + 12\,a_{248} + 16\,a_{251} + 48\,a_{252} + 12\,a_{254} + 48\,a_{255} + 144\,a_{256} + 2\,a_{138} + a_{86} - 8\,a_{203} + 12\,a_{204} + 12\,a_{206} - 24\,a_{207} + 36\,a_{208} + a_{198} + 3\,a_{200} + 4\,a_{202} - 2\,a_{199} + 36\,a_{64} + a_{182} + 4\,a_{183} + 12\,a_{184} + 2\,a_{186} + 8\,a_{187} + 24\,a_{188} + 3\,a_{190} + 12\,a_{191} + 36\,a_{192} + 4\,a_{250} - 2\,a_{10} + a_{230} + 2\,a_{231} + 3\,a_{232} + 4\,a_{234} + 8\,a_{235} + 12\,a_{236} + 12\,a_{238} + 24\,a_{239} + 36\,a_{240} - 4\,a_{139} + 3\,a_{104} + 2\,a_{103} + a_{102} - 2\,a_{26} = 0$

272. $-a_{72} + a_{71} - 2\,a_{25} + a_{149} + 2\,a_{153} + 3\,a_{157} + a_{37} + a_{21} + a_{135} + 3\,a_{29} + 3\,a_{141} + a_{133} - a_{134} + 2\,a_{12} - 2\,a_{11} + 3\,a_{15} - 3\,a_{14} - a_{136} - 3\,a_{142} + 3\,a_{143} + 3\,a_{13} - 3\,a_{16} + a_{53} + 2\,a_{54} + 4\,a_{55} + 8\,a_{56} - 2\,a_{57} - 4\,a_{58} - 8\,a_{59} - 16\,a_{60} + 3\,a_{61} + 6\,a_{62} + 12\,a_{63} + a_{38} + a_{39} + a_{40} - 2\,a_{41} - 2\,a_{42} - 2\,a_{43} - 2\,a_{44} + 3\,a_{45} + 3\,a_{46} + 3\,a_{47} + a_{5} - a_{6} + a_{7} - a_{8} - 2\,a_{9} + a_{213} + 4\,a_{217} + 12\,a_{221} + a_{165} + a_{166} + a_{167} + a_{168} + 2\,a_{169} + 2\,a_{170} + 2\,a_{171} + 2\,a_{172} + 3\,a_{173} + 3\,a_{174} + 3\,a_{175} + 3\,a_{176} - 2\,a_{140} + 3\,a_{48} - 3\,a_{144} + 2\,a_{137} + 8\,a_{120} + 4\,a_{119} + 2\,a_{118} + a_{117} - a_{70} + a_{245} + 2\,a_{246} + 4\,a_{247} + 8\,a_{248} + 4\,a_{249} + 16\,a_{251} + 32\,a_{252} + 12\,a_{253} + 24\,a_{254} + 48\,a_{255} + 96\,a_{256} - 2\,a_{138} + a_{69} + a_{85} + 4\,a_{203} - 4\,a_{204} + 12\,a_{205} - 12\,a_{206} + 12\,a_{207} - 12\,a_{208} - a_{198} - a_{200} + 4\,a_{201} - 4\,a_{202} + a_{199} + a_{197} + 24\,a_{64} + a_{181} + 2\,a_{182} + 4\,a_{183} + 8\,a_{184} + 2\,a_{185} + 4\,a_{186} + 8\,a_{187} + 16\,a_{188} + 3\,a_{189} + 6\,a_{190} + 12\,a_{191} + 24\,a_{192} + 8\,a_{250} + 2\,a_{10} + a_{229} + a_{230} + a_{231} + a_{232} + 4\,a_{233} + 4\,a_{234} + 4\,a_{235} + 4\,a_{236} + 12\,a_{237} + 12\,a_{238} + 12\,a_{239} + 12\,a_{240} + 2\,a_{139} + a_{104} + a_{103} + a_{102} + a_{101} = 0$

273. $a_{68} - a_{24} + a_{28} + a_{152} + a_{156} + a_{20} - a_{32} + a_{132} + a_{12} + a_{136} + a_{160} - a_{16} - a_{56} + a_{60} - a_{40} + a_{44} + a_{4} - a_{8} + 2\,a_{216} + 4\,a_{220} + 8\,a_{224} + a_{228} + a_{168} + a_{172} + a_{176} + a_{140} - a_{48} + a_{144} + a_{180} + a_{116} + 2\,a_{248} + 4\,a_{252} + 8\,a_{256} + a_{36} + a_{84} + 4\,a_{204} + 8\,a_{208} + 2\,a_{200} - a_{64} + a_{184} + a_{188} + a_{192} + a_{148} + a_{244} + a_{52} + a_{164} + a_{212} + 2\,a_{232} + 4\,a_{236} + 8\,a_{240} + a_{100} + a_{196} = 0$

274. $a_{19} - 3\,a_{68} + a_{67} - a_{23} + a_{151} + a_{155} + a_{131} + a_{135} - a_{31} - 3\,a_{132} - 3\,a_{12} + a_{11} + a_{3} - a_{15} - 3\,a_{136} + a_{143} + a_{159} + 3\,a_{16} - a_{55} - 6\,a_{56} + a_{59} + 6\,a_{60} - a_{63} - a_{39} - 3\,a_{40} + a_{43} + 3\,a_{44} -$

$a_{47} - 3\,a_4 - a_7 + 3\,a_8 + 2\,a_{215} + 4\,a_{219} + 8\,a_{223} + a_{227} + 3\,a_{228} + a_{163} + a_{167} + 3\,a_{168} + a_{171} + 3\,a_{172} + a_{175} + 3\,a_{176} - 3\,a_{140} - 3\,a_{48} + a_{51} - 3\,a_{144} + a_{179} + 6\,a_{180} + a_{115} + 6\,a_{116} + 2\,a_{247} + 12\,a_{248} + 4\,a_{251} + 24\,a_{252} + 8\,a_{255} + 48\,a_{256} + a_{35} + 3\,a_{36} + a_{83} + a_{27} + 4\,a_{203} - 12\,a_{204} + 8\,a_{207} - 24\,a_{208} - 6\,a_{200} + 2\,a_{199} - 6\,a_{64} + a_{183} + 6\,a_{184} + a_{187} + 6\,a_{188} + a_{191} + 6\,a_{192} + a_{147} + a_{243} + 6\,a_{244} + 6\,a_{52} + 3\,a_{164} + a_{211} + 2\,a_{231} + 6\,a_{232} + 4\,a_{235} + 12\,a_{236} + 8\,a_{239} + 24\,a_{240} + a_{139} + 3\,a_{100} + a_{99} + a_{195} - 3\,a_{196} = 0$

275. $3\,a_{68} - 2\,a_{67} + a_{150} + a_{154} - a_{22} - 2\,a_{131} + a_{130} - 2\,a_{135} + 3\,a_{132} - a_{30} + a_{134} + 3\,a_{12} - 2\,a_{11} - 2\,a_3 + 2\,a_{15} - a_{14} + 3\,a_{136} + a_{142} - 2\,a_{143} + a_{158} - 3\,a_{16} - a_{54} - 4\,a_{55} - 12\,a_{56} + a_{58} + 4\,a_{59} + 12\,a_{60} - a_{62} - 4\,a_{63} - a_{38} - 2\,a_{39} - 3\,a_{40} + a_{42} + 2\,a_{43} + 3\,a_{44} - a_{46} - 2\,a_{47} + 3\,a_4 - a_6 + 2\,a_7 - 3\,a_8 + 2\,a_{214} + 8\,a_{222} + a_{226} + 2\,a_{227} + 3\,a_{228} + a_{162} + 2\,a_{163} + a_{166} + 2\,a_{167} + 3\,a_{168} + a_{170} + 2\,a_{171} + 3\,a_{172} + a_{174} + 2\,a_{175} + 3\,a_{176} + 3\,a_{140} - 3\,a_{48} + a_{50} + 4\,a_{51} + 3\,a_{144} + a_{18} + 4\,a_{218} + a_{178} + 4\,a_{179} + 12\,a_{180} + 4\,a_{115} + a_{114} + 12\,a_{116} + a_{66} + 2\,a_{246} + 8\,a_{247} + 24\,a_{248} + 16\,a_{251} + 48\,a_{252} + 8\,a_{254} + 32\,a_{255} + 96\,a_{256} + a_{138} + 2\,a_{35} + 3\,a_{36} + a_{82} - 8\,a_{203} + 12\,a_{204} + 8\,a_{206} - 16\,a_{207} + 24\,a_{208} + 2\,a_{198} + 6\,a_{200} + 4\,a_{202} - 4\,a_{199} - 12\,a_{64} + a_{182} + 4\,a_{183} + 12\,a_{184} + a_{186} + 4\,a_{187} + 12\,a_{188} + a_{190} + 4\,a_{191} + 12\,a_{192} + a_{146} + 4\,a_{243} + 12\,a_{244} + 12\,a_{52} + 4\,a_{250} + a_{10} + a_{34} + a_2 + 3\,a_{164} + a_{210} + 2\,a_{230} + 4\,a_{231} + 6\,a_{232} + 4\,a_{234} + 8\,a_{235} + 12\,a_{236} + 8\,a_{238} + 16\,a_{239} + 24\,a_{240} - 2\,a_{139} + a_{242} + 3\,a_{100} + 2\,a_{99} + a_{98} + a_{26} + a_{194} - 2\,a_{195} + 3\,a_{196} = 0$

276. $-a_{68} + a_{67} + a_{25} + a_{149} + a_{153} + a_{157} - a_{37} + a + a_{129} + a_{131} - a_{21} - a_{130} + a_{135} - a_{132} - a_{29} + a_{141} + a_{133} - a_{134} - a_{12} + a_{11} + a_3 - a_{15} + a_{14} - a_{136} - a_{142} + a_{143} - a_{13} + a_{16} - a_{53} - 2\,a_{54} - 4\,a_{55} - 8\,a_{56} + a_{57} + 2\,a_{58} + 4\,a_{59} + 8\,a_{60} - a_{61} - 2\,a_{62} - 4\,a_{63} - a_{38} - a_{39} - a_{40} + a_{41} + a_{42} + a_{43} + a_{44} - a_{45} - a_{46} - a_{47} - a_4 - a_5 + a_6 - a_7 + a_8 + a_9 + 2\,a_{213} + 4\,a_{217} + 8\,a_{221} + a_{226} + a_{227} + a_{228} + a_{162} + a_{163} + a_{165} + a_{166} + a_{167} + a_{168} + a_{169} + a_{170} + a_{171} + a_{172} + a_{173} + a_{174} + a_{175} + a_{176} - a_{140} + a_{209} - a_{48} + a_{17} + 2\,a_{50} + 4\,a_{51} - a_{144} + a_{241} + a_{137} + 2\,a_{178} + 4\,a_{179} + 8\,a_{180} + 4\,a_{115} + 2\,a_{114} + a_{113} + 8\,a_{116} + a_{49} + a_{177} - a_{66} + 2\,a_{245} + 4\,a_{246} + 8\,a_{247} + 16\,a_{248} + 4\,a_{249} + 16\,a_{251} + 32\,a_{252} + 8\,a_{253} + 16\,a_{254} + 32\,a_{255} + 64\,a_{256} - a_{138} + a_{33} + a_{35} + a_{36} + a_{65} + a_{81} + 4\,a_{203} - 4\,a_{204} + 8\,a_{205} - 8\,a_{206} + 8\,a_{207} - 8\,a_{208} - 2\,a_{198} - 2\,a_{200} + 4\,a_{201} - 4\,a_{202} + 2\,a_{199} + 2\,a_{197} - 8\,a_{64} + a_{181} + 2\,a_{182} + 4\,a_{183} + 8\,a_{184} + a_{185} + 2\,a_{186} + 4\,a_{187} + 8\,a_{188} + a_{189} + 2\,a_{190} + 4\,a_{191} + 8\,a_{192} + 4\,a_{243} + 8\,a_{244} + 8\,a_{52} + 8\,a_{250} - a_{10} + a_{34} - a_2 + a_{225} + a_{164} + 2\,a_{229} + 2\,a_{230} + 2\,a_{231} + 2\,a_{232} + 4\,a_{233} + 4\,a_{234} + 4\,a_{235} + 4\,a_{236} + 8\,a_{237} + 8\,a_{238} + 8\,a_{239} + 8\,a_{240} + a_{161} + a_{139} + 2\,a_{242} + a_{100} + a_{99} + a_{98} + a_{97} - a_{194} + a_{195} - a_{196} + a_{193} + a_{145} = 0$

277. $a_{48} + 2\,a_{256} + 2\,a_{128} - a_{80} - a_{144} + 2\,a_{64} + a_{112} + 2\,a_{192} + a_{176} - a_{16} - a_{208} + a_{240} = 0$

278. $-a_{15} + 12\,a_{192} + a_{175} + 12\,a_{256} - a_{207} + 2\,a_{191} + a_{239} + 3\,a_{208} + 3\,a_{176} + 2\,a_{255} - a_{143} + 12\,a_{128} + 2\,a_{63} + 12\,a_{64} + a_{47} + 3\,a_{240} + 3\,a_{48} + a_{111} + 3\,a_{112} - a_{79} + 3\,a_{80} + 3\,a_{16} + 3\,a_{144} + 2\,a_{127} = 0$

279. $-3\,a_{80} + 2\,a_{79} + 2\,a_{15} - a_{14} - a_{142} + 2\,a_{143} - 3\,a_{16} + 2\,a_{62} + 8\,a_{63} + a_{46} + 2\,a_{47} + a_{174} + 2\,a_{175} + 3\,a_{176} + 3\,a_{48} - 3\,a_{144} + 24\,a_{128} + 8\,a_{127} + 2\,a_{126} - a_{78} + 2\,a_{254} + 8\,a_{255} + 24\,a_{256} - a_{206} + 2\,a_{207} - 3\,a_{208} + 24\,a_{64} + 2\,a_{190} + 8\,a_{191} + 24\,a_{192} + a_{238} + 2\,a_{239} + 3\,a_{240} + 3\,a_{112} + 2\,a_{111} + a_{110} = 0$

280. $a_{80} - a_{79} - a_{141} - a_{15} + a_{14} + a_{142} - a_{143} - a_{13} + a_{16} + 2\,a_{61} + 4\,a_{62} + 8\,a_{63} + a_{45} + a_{46} + a_{47} + a_{173} + a_{174} + a_{175} + a_{176} + a_{48} + a_{144} +$

$16\,a_{128} + 8\,a_{127} + 4\,a_{126} + 2\,a_{125} + a_{78} + 2\,a_{253} + 4\,a_{254} + 8\,a_{255} + 16\,a_{256} - a_{77} - a_{205} + a_{206} - a_{207} + a_{208} + 16\,a_{64} + 2\,a_{189} + 4\,a_{190} + 8\,a_{191} + 16\,a_{192} + a_{237} + a_{238} + a_{239} + a_{240} + a_{112} + a_{111} + a_{110} + a_{109} = 0$

281. $-6\,a_{208} + a_{236} + a_{108} + 2\,a_{124} - 3\,a_{48} + 6\,a_{192} + 3\,a_{176} + 2\,a_{188} + a_{44} - a_{140} + 12\,a_{256} - a_{204} + 6\,a_{240} + 2\,a_{252} - 3\,a_{144} + 2\,a_{60} - a_{76} - a_{12} - 6\,a_{64} + a_{172} + 3\,a_{16} = 0$

282. $3\,a_{76} - a_{75} + 3\,a_{12} - a_{11} + 3\,a_{15} - 3\,a_{143} - 9\,a_{16} + 2\,a_{59} + 12\,a_{60} - 6\,a_{63} + a_{43} + 3\,a_{44} - 3\,a_{47} + a_{171} + 3\,a_{172} + 3\,a_{175} + 9\,a_{176} + 3\,a_{140} - 9\,a_{48} + 9\,a_{144} + 12\,a_{124} + 2\,a_{123} + 2\,a_{251} + 12\,a_{252} + 12\,a_{255} + 72\,a_{256} - a_{203} + 3\,a_{204} - 6\,a_{207} + 18\,a_{208} - 36\,a_{64} + 2\,a_{187} + 12\,a_{188} + 6\,a_{191} + 36\,a_{192} + a_{235} + 3\,a_{236} + 6\,a_{239} + 18\,a_{240} - a_{139} + 3\,a_{108} + a_{107} = 0$

283. $-3\,a_{76} + 2\,a_{75} - 3\,a_{12} + 2\,a_{11} - 6\,a_{15} + 3\,a_{14} - 3\,a_{142} + 6\,a_{143} + 9\,a_{16} + 2\,a_{58} + 8\,a_{59} + 24\,a_{60} - 6\,a_{62} - 24\,a_{63} + a_{42} + 2\,a_{43} + 3\,a_{44} - 3\,a_{46} - 6\,a_{47} + a_{170} + 2\,a_{171} + 3\,a_{172} + 3\,a_{174} + 6\,a_{175} + 9\,a_{176} - 3\,a_{140} - 9\,a_{48} - 9\,a_{144} + 24\,a_{124} + 8\,a_{123} + 2\,a_{122} - a_{74} + 8\,a_{251} + 24\,a_{252} + 12\,a_{254} + 48\,a_{255} + 144\,a_{256} - a_{138} + 2\,a_{203} - 3\,a_{204} - 6\,a_{206} + 12\,a_{207} - 18\,a_{208} - a_{202} - 72\,a_{64} + 2\,a_{186} + 8\,a_{187} + 24\,a_{188} + 6\,a_{190} + 24\,a_{191} + 72\,a_{192} + 2\,a_{250} - a_{10} + a_{234} + 2\,a_{235} + 3\,a_{236} + 6\,a_{238} + 12\,a_{239} + 18\,a_{240} + 2\,a_{139} + 3\,a_{108} + 2\,a_{107} + a_{106} = 0$

284. $a_{76} - a_{75} - 3\,a_{141} + a_{12} - a_{11} + 3\,a_{15} - 3\,a_{14} + 3\,a_{142} - 3\,a_{143} + 3\,a_{13} - 3\,a_{16} + 2\,a_{57} + 4\,a_{58} + 8\,a_{59} + 16\,a_{60} - 6\,a_{61} - 12\,a_{62} - 24\,a_{63} + a_{41} + a_{42} + a_{43} + a_{44} - 3\,a_{45} - 3\,a_{46} - 3\,a_{47} - a_9 + a_{169} + a_{170} + a_{171} + a_{172} + 3\,a_{173} + 3\,a_{174} + 3\,a_{175} + 3\,a_{176} + a_{140} - 3\,a_{48} + 3\,a_{144} - a_{137} + 16\,a_{124} + 8\,a_{123} + 4\,a_{122} + 2\,a_{121} + a_{74} + 2\,a_{249} + 8\,a_{251} + 16\,a_{252} + 12\,a_{253} + 24\,a_{254} + 48\,a_{255} + 96\,a_{256} + a_{138} - a_{73} - a_{203} + a_{204} - 6\,a_{205} + 6\,a_{206} - 6\,a_{207} + 6\,a_{208} - a_{201} + a_{202} - 48\,a_{64} + 2\,a_{185} + 4\,a_{186} + 8\,a_{187} + 16\,a_{188} + 6\,a_{189} + 12\,a_{190} + 24\,a_{191} + 48\,a_{192} + 4\,a_{250} + a_{10} + a_{233} + a_{234} + a_{235} + a_{236} + 6\,a_{237} + 6\,a_{238} + 6\,a_{239} + 6\,a_{240} - a_{139} + a_{108} + a_{107} + a_{106} + a_{105} = 0$

285. $-a_{72} + 2\,a_{12} - a_{136} - 3\,a_{16} + 2\,a_{56} - 4\,a_{60} + a_{40} - 2\,a_{44} - a_8 + a_{168} + 2\,a_{172} + 3\,a_{176} - 2\,a_{140} + 3\,a_{48} - 3\,a_{144} + 2\,a_{120} + 2\,a_{248} + 8\,a_{252} + 24\,a_{256} - 4\,a_{204} - 12\,a_{208} - a_{200} + 6\,a_{64} + 2\,a_{184} + 4\,a_{188} + 6\,a_{192} + a_{232} + 4\,a_{236} + 12\,a_{240} + a_{104} = 0$

286. $3\,a_{72} - a_{71} - a_{135} - 6\,a_{12} + 2\,a_{11} - 3\,a_{15} + 3\,a_{136} - 3\,a_{143} + 9\,a_{16} + 2\,a_{55} + 12\,a_{56} - 4\,a_{59} - 24\,a_{60} + 6\,a_{63} + a_{39} + 3\,a_{40} - 2\,a_{43} - 6\,a_{44} + 3\,a_{47} - a_7 + 3\,a_8 + a_{167} + 3\,a_{168} + 2\,a_{171} + 6\,a_{172} + 3\,a_{175} + 9\,a_{176} + 6\,a_{140} + 9\,a_{48} + 9\,a_{144} + 12\,a_{120} + 2\,a_{119} + 2\,a_{247} + 12\,a_{248} + 8\,a_{251} + 48\,a_{252} + 24\,a_{255} + 144\,a_{256} - 4\,a_{203} + 12\,a_{204} - 12\,a_{207} + 36\,a_{208} + 3\,a_{200} - a_{199} + 36\,a_{64} + 2\,a_{183} + 12\,a_{184} + 4\,a_{187} + 24\,a_{188} + 6\,a_{191} + 36\,a_{192} + a_{231} + 3\,a_{232} + 4\,a_{235} + 12\,a_{236} + 12\,a_{239} + 36\,a_{240} - 2\,a_{139} + 3\,a_{104} + a_{103} = 0$

287. $-3\,a_{72} + 2\,a_{71} + 2\,a_{135} - a_{134} + 6\,a_{12} - 4\,a_{11} + 6\,a_{15} - 3\,a_{14} - 3\,a_{136} - 3\,a_{142} + 6\,a_{143} - 9\,a_{16} + 2\,a_{54} + 8\,a_{55} + 24\,a_{56} - 4\,a_{58} - 16\,a_{59} - 48\,a_{60} + 6\,a_{62} + 24\,a_{63} + a_{38} + 2\,a_{39} + 3\,a_{40} - 2\,a_{42} - 4\,a_{43} - 6\,a_{44} + 3\,a_{46} + 6\,a_{47} - a_6 + 2\,a_7 - 3\,a_8 + a_{166} + 2\,a_{167} + 3\,a_{168} + 2\,a_{170} + 4\,a_{171} + 6\,a_{172} + 3\,a_{174} + 6\,a_{175} + 9\,a_{176} - 6\,a_{140} + 9\,a_{48} - 9\,a_{144} + 24\,a_{120} + 8\,a_{119} + 2\,a_{118} - a_{70} + 2\,a_{246} + 8\,a_{247} + 24\,a_{248} + 32\,a_{251} + 96\,a_{252} + 24\,a_{254} + 96\,a_{255} + 288\,a_{256} - 2\,a_{138} + 8\,a_{203} - 12\,a_{204} - 12\,a_{206} + 24\,a_{207} - 36\,a_{208} - a_{198} - 3\,a_{200} - 4\,a_{202} + 2\,a_{199} + 72\,a_{64} + 2\,a_{182} + 8\,a_{183} + 24\,a_{184} + 4\,a_{186} + 16\,a_{187} + 48\,a_{188} + 6\,a_{190} + 24\,a_{191} + 72\,a_{192} + 8\,a_{250} + 2\,a_{10} + a_{230} + 2\,a_{231} + 3\,a_{232} + 4\,a_{234} + 8\,a_{235} + 12\,a_{236} + 12\,a_{238} + 24\,a_{239} +$

$$36\,a_{240}+4\,a_{139}+3\,a_{104}+2\,a_{103}+a_{102}=0$$

288. $$a_{72}-a_{71}+a_{37}-a_{135}-3\,a_{141}-a_{133}+a_{134}-2\,a_{12}+2\,a_{11}-3\,a_{15}+3\,a_{14}+a_{136}+3\,a_{142}-3\,a_{143}-3\,a_{13}+3\,a_{16}+2\,a_{53}+4\,a_{54}+8\,a_{55}+16\,a_{56}-4\,a_{57}-8\,a_{58}-16\,a_{59}-32\,a_{60}+6\,a_{61}+12\,a_{62}+24\,a_{63}+a_{38}+a_{39}+a_{40}-2\,a_{41}-2\,a_{42}-2\,a_{43}-2\,a_{44}+3\,a_{45}+3\,a_{46}+3\,a_{47}-a_5+a_6-a_7+a_8+2\,a_9+a_{165}+a_{166}+a_{167}+a_{168}+2\,a_{169}+2\,a_{170}+2\,a_{171}+2\,a_{172}+3\,a_{173}+3\,a_{174}+3\,a_{175}+3\,a_{176}+2\,a_{140}+3\,a_{48}+3\,a_{144}-2\,a_{137}+16\,a_{120}+8\,a_{119}+4\,a_{118}+2\,a_{117}+a_{70}+2\,a_{245}+4\,a_{246}+8\,a_{247}+16\,a_{248}+8\,a_{249}+32\,a_{251}+64\,a_{252}+24\,a_{253}+48\,a_{254}+96\,a_{255}+192\,a_{256}+2\,a_{138}-a_{69}-4\,a_{203}+4\,a_{204}-12\,a_{205}+12\,a_{206}-12\,a_{207}+12\,a_{208}+a_{198}+a_{200}-4\,a_{201}+4\,a_{202}-a_{199}-a_{197}+48\,a_{64}+2\,a_{181}+4\,a_{182}+8\,a_{183}+16\,a_{184}+4\,a_{185}+8\,a_{186}+16\,a_{187}+32\,a_{188}+6\,a_{189}+12\,a_{190}+24\,a_{191}+48\,a_{192}+16\,a_{250}-2\,a_{10}+a_{229}+a_{230}+a_{231}+a_{232}+4\,a_{233}+4\,a_{234}+4\,a_{235}+4\,a_{236}+12\,a_{237}+12\,a_{238}+12\,a_{239}+12\,a_{240}-2\,a_{139}+a_{104}+a_{103}+a_{102}+a_{101}=0$$

289. $$-a_{68}-a_{132}-a_{12}-a_{136}+a_{16}-2\,a_{56}+2\,a_{60}-a_{40}+a_{44}-a_4+a_8+a_{228}+a_{168}+a_{172}+a_{176}-a_{140}-a_{48}-a_{144}+2\,a_{180}+2\,a_{116}+4\,a_{248}+8\,a_{252}+16\,a_{256}+a_{36}-4\,a_{204}-8\,a_{208}-2\,a_{200}-2\,a_{64}+2\,a_{184}+2\,a_{188}+2\,a_{192}+2\,a_{244}+2\,a_{52}+a_{164}+2\,a_{232}+4\,a_{236}+8\,a_{240}+a_{100}-a_{196}=0$$

290. $$3\,a_{68}-a_{67}-a_{131}-a_{135}+3\,a_{132}+3\,a_{12}-a_{11}-a_3+a_{15}+3\,a_{136}-a_{143}-3\,a_{16}-2\,a_{55}-12\,a_{56}+2\,a_{59}+12\,a_{60}-2\,a_{63}-a_{39}-3\,a_{40}+a_{43}+3\,a_{44}-a_{47}+3\,a_4+a_7-3\,a_8+a_{227}+3\,a_{228}+a_{163}+a_{167}+3\,a_{168}+a_{171}+3\,a_{172}+a_{175}+3\,a_{176}+3\,a_{140}-3\,a_{48}+2\,a_{51}+3\,a_{144}+2\,a_{179}+12\,a_{180}+2\,a_{115}+12\,a_{116}+4\,a_{247}+24\,a_{248}+8\,a_{251}+48\,a_{252}+16\,a_{255}+96\,a_{256}+a_{35}+3\,a_{36}-4\,a_{203}+12\,a_{204}-8\,a_{207}+24\,a_{208}+6\,a_{200}-2\,a_{199}-12\,a_{64}+2\,a_{183}+12\,a_{184}+2\,a_{187}+12\,a_{188}+2\,a_{191}+12\,a_{192}+2\,a_{243}+12\,a_{244}+12\,a_{52}+3\,a_{164}+2\,a_{231}+6\,a_{232}+4\,a_{235}+12\,a_{236}+8\,a_{239}+24\,a_{240}-a_{139}+3\,a_{100}+a_{99}-a_{195}+3\,a_{196}=0$$

291. $$-3\,a_{68}+2\,a_{67}+2\,a_{131}-a_{130}+2\,a_{135}-3\,a_{132}-a_{134}-3\,a_{12}+2\,a_{11}+2\,a_3-2\,a_{15}+a_{14}-3\,a_{136}-a_{142}+2\,a_{143}+3\,a_{16}-2\,a_{54}-8\,a_{55}-24\,a_{56}+2\,a_{58}+8\,a_{59}+24\,a_{60}-2\,a_{62}-8\,a_{63}-a_{38}-2\,a_{39}-3\,a_{40}+a_{42}+2\,a_{43}+3\,a_{44}-a_{46}-2\,a_{47}-3\,a_4+a_6-2\,a_7+3\,a_8+a_{226}+2\,a_{227}+3\,a_{228}+a_{162}+2\,a_{163}+a_{166}+2\,a_{167}+3\,a_{168}+a_{170}+2\,a_{171}+3\,a_{172}+a_{174}+2\,a_{175}+3\,a_{176}-3\,a_{140}-3\,a_{48}+2\,a_{50}+8\,a_{51}-3\,a_{144}+2\,a_{178}+8\,a_{179}+24\,a_{180}+8\,a_{115}+2\,a_{114}+24\,a_{116}-a_{66}+4\,a_{246}+16\,a_{247}+48\,a_{248}+32\,a_{251}+96\,a_{252}+16\,a_{254}+64\,a_{255}+192\,a_{256}-a_{138}+2\,a_{35}+3\,a_{36}+8\,a_{203}-12\,a_{204}-8\,a_{206}+16\,a_{207}-24\,a_{208}-2\,a_{198}-6\,a_{200}-4\,a_{202}+4\,a_{199}-24\,a_{64}+2\,a_{182}+8\,a_{183}+24\,a_{184}+2\,a_{186}+8\,a_{187}+24\,a_{188}+2\,a_{190}+8\,a_{191}+24\,a_{192}+8\,a_{243}+24\,a_{244}+24\,a_{52}+8\,a_{250}-a_{10}+a_{34}-a_2+3\,a_{164}+2\,a_{230}+4\,a_{231}+6\,a_{232}+4\,a_{234}+8\,a_{235}+12\,a_{236}+8\,a_{238}+16\,a_{239}+24\,a_{240}+2\,a_{139}+2\,a_{242}+3\,a_{100}+2\,a_{99}+a_{98}-a_{194}+2\,a_{195}-3\,a_{196}=0$$

292. $$a_{68}-a_{67}-a_{37}-a-a_{129}-a_{131}+a_{130}-a_{135}+a_{132}-a_{141}-a_{133}+a_{134}+a_{12}-a_{11}-a_3+a_{15}-a_{14}+a_{136}+a_{142}-a_{143}+a_{13}-a_{16}-2\,a_{53}-4\,a_{54}-8\,a_{55}-16\,a_{56}+2\,a_{57}+4\,a_{58}+8\,a_{59}+16\,a_{60}-2\,a_{61}-4\,a_{62}-8\,a_{63}-a_{38}-a_{39}-a_{40}+a_{41}+a_{42}+a_{43}+a_{44}-a_{45}-a_{46}-a_{47}+a_4+a_5-a_6+a_7-a_8-a_9+a_{226}+a_{227}+a_{228}+a_{162}+a_{163}+a_{165}+a_{166}+a_{167}+a_{168}+a_{169}+a_{170}+a_{171}+a_{172}+a_{173}+a_{174}+a_{175}+a_{176}+a_{140}-a_{48}+4\,a_{50}+8\,a_{51}+a_{144}+2\,a_{241}-a_{137}+4\,a_{178}+8\,a_{179}+16\,a_{180}+8\,a_{115}+4\,a_{114}+2\,a_{113}+16\,a_{116}+2\,a_{49}+2\,a_{177}+$$

$a_{66} + 4\,a_{245} + 8\,a_{246} + 16\,a_{247} + 32\,a_{248} + 8\,a_{249} + 32\,a_{251} + 64\,a_{252} + 16\,a_{253} + 32\,a_{254} + 64\,a_{255} + 128\,a_{256} + a_{138} + a_{33} + a_{35} + a_{36} - a_{65} - 4\,a_{203} + 4\,a_{204} - 8\,a_{205} + 8\,a_{206} - 8\,a_{207} + 8\,a_{208} + 2\,a_{198} + 2\,a_{200} - 4\,a_{201} + 4\,a_{202} - 2\,a_{199} - 2\,a_{197} - 16\,a_{64} + 2\,a_{181} + 4\,a_{182} + 8\,a_{183} + 16\,a_{184} + 2\,a_{185} + 4\,a_{186} + 8\,a_{187} + 16\,a_{188} + 2\,a_{189} + 4\,a_{190} + 8\,a_{191} + 16\,a_{192} + 8\,a_{243} + 16\,a_{244} + 16\,a_{52} + 16\,a_{250} + a_{10} + a_{34} + a_2 + a_{225} + a_{164} + 2\,a_{229} + 2\,a_{230} + 2\,a_{231} + 2\,a_{232} + 4\,a_{233} + 4\,a_{234} + 4\,a_{235} + 4\,a_{236} + 8\,a_{237} + 8\,a_{238} + 8\,a_{239} + 8\,a_{240} + a_{161} - a_{139} + 4\,a_{242} + a_{100} + a_{99} + a_{98} + a_{97} + a_{194} - a_{195} + a_{196} - a_{193} = 0$

293. $a_{176} + 2\,a_{256} + a_{192} + a_{160} + a_{144} + 2\,a_{208} - a_{64} + 2\,a_{240} - a_{48} - a_{32} + 2\,a_{224} - a_{16} = 0$

294. $3\,a_{16} + a_{191} - 6\,a_{64} + a_{143} + a_{159} - a_{15} + 2\,a_{239} + 2\,a_{223} - a_{47} - a_{63} - 3\,a_{48} + a_{175} + 6\,a_{240} + 6\,a_{192} + 3\,a_{176} + 2\,a_{207} - a_{31} + 2\,a_{255} - 6\,a_{208} - 3\,a_{144} + 12\,a_{256} = 0$

295. $-a_{30} + 2\,a_{15} - a_{14} + a_{142} - 2\,a_{143} + a_{158} - 3\,a_{16} - a_{62} - 4\,a_{63} - a_{46} - 2\,a_{47} + 2\,a_{222} + a_{174} + 2\,a_{175} + 3\,a_{176} - 3\,a_{48} + 3\,a_{144} + 2\,a_{254} + 8\,a_{255} + 24\,a_{256} + 2\,a_{206} - 4\,a_{207} + 6\,a_{208} - 12\,a_{64} + a_{190} + 4\,a_{191} + 12\,a_{192} + 2\,a_{238} + 4\,a_{239} + 6\,a_{240} = 0$

296. $a_{157} - a_{29} + a_{141} - a_{15} + a_{14} - a_{142} + a_{143} - a_{13} + a_{16} - a_{61} - 2\,a_{62} - 4\,a_{63} - a_{45} - a_{46} - a_{47} + 2\,a_{221} + a_{173} + a_{174} + a_{175} + a_{176} - a_{48} - a_{144} + 2\,a_{253} + 4\,a_{254} + 8\,a_{255} + 16\,a_{256} + 2\,a_{205} - 2\,a_{206} + 2\,a_{207} - 2\,a_{208} - 8\,a_{64} + a_{189} + 2\,a_{190} + 4\,a_{191} + 8\,a_{192} + 2\,a_{237} + 2\,a_{238} + 2\,a_{239} + 2\,a_{240} = 0$

297. $3\,a_{176} - a_{28} + 3\,a_{32} + 2\,a_{252} + 3\,a_{64} + 3\,a_{160} + 2\,a_{236} + a_{156} + 12\,a_{224} + 12\,a_{208} + 2\,a_{220} + a_{172} + 12\,a_{256} + a_{140} + 3\,a_{16} + 12\,a_{240} - a_{60} - a_{12} - a_{44} + 2\,a_{204} + 3\,a_{48} + 3\,a_{144} + 3\,a_{192} + a_{188} = 0$

298. $a_{155} + 3\,a_{31} + 3\,a_{12} - a_{11} + 3\,a_{15} + 3\,a_{143} + 3\,a_{159} - 9\,a_{16} - a_{59} - 6\,a_{60} + 3\,a_{63} - a_{43} - 3\,a_{44} + 3\,a_{47} + 2\,a_{219} + 12\,a_{223} + a_{171} + 3\,a_{172} + 3\,a_{175} + 9\,a_{176} - 3\,a_{140} + 9\,a_{48} - 9\,a_{144} + 2\,a_{251} + 12\,a_{252} + 12\,a_{255} + 72\,a_{256} - a_{27} + 2\,a_{203} - 6\,a_{204} + 12\,a_{207} - 36\,a_{208} + 18\,a_{64} + a_{187} + 6\,a_{188} + 3\,a_{191} + 18\,a_{192} + 2\,a_{235} + 6\,a_{236} + 12\,a_{239} + 36\,a_{240} + a_{139} = 0$

299. $a_{154} + 3\,a_{30} - 3\,a_{12} + 2\,a_{11} - 6\,a_{15} + 3\,a_{14} + 3\,a_{142} - 6\,a_{143} + 3\,a_{158} + 9\,a_{16} - a_{58} - 4\,a_{59} - 12\,a_{60} + 3\,a_{62} + 12\,a_{63} - a_{42} - 2\,a_{43} - 3\,a_{44} + 3\,a_{46} + 6\,a_{47} + 12\,a_{222} + a_{170} + 2\,a_{171} + 3\,a_{172} + 3\,a_{174} + 6\,a_{175} + 9\,a_{176} + 3\,a_{140} + 9\,a_{48} + 9\,a_{144} + 2\,a_{218} + 8\,a_{251} + 24\,a_{252} + 12\,a_{254} + 48\,a_{255} + 144\,a_{256} + a_{138} - 4\,a_{203} + 6\,a_{204} + 12\,a_{206} - 24\,a_{207} + 36\,a_{208} + 2\,a_{202} + 36\,a_{64} + a_{186} + 4\,a_{187} + 12\,a_{188} + 3\,a_{190} + 12\,a_{191} + 36\,a_{192} + 2\,a_{250} - a_{10} + 2\,a_{234} + 4\,a_{235} + 6\,a_{236} + 12\,a_{238} + 24\,a_{239} + 36\,a_{240} - 2\,a_{139} - a_{26} = 0$

300. $-a_{25} + a_{153} + 3\,a_{157} + 3\,a_{29} + 3\,a_{141} + a_{12} - a_{11} + 3\,a_{15} - 3\,a_{14} - 3\,a_{142} + 3\,a_{143} + 3\,a_{13} - 3\,a_{16} - a_{57} - 2\,a_{58} - 4\,a_{59} - 8\,a_{60} + 3\,a_{61} + 6\,a_{62} + 12\,a_{63} - a_{41} - a_{42} - a_{43} - a_{44} + 3\,a_{45} + 3\,a_{46} + 3\,a_{47} - a_9 + 2\,a_{217} + 12\,a_{221} + a_{169} + a_{170} + a_{171} + a_{172} + 3\,a_{173} + 3\,a_{174} + 3\,a_{175} + 3\,a_{176} - a_{140} + 3\,a_{48} - 3\,a_{144} + a_{137} + 2\,a_{249} + 8\,a_{251} + 16\,a_{252} + 12\,a_{253} + 24\,a_{254} + 48\,a_{255} + 96\,a_{256} - a_{138} + 2\,a_{203} - 2\,a_{204} + 12\,a_{205} - 12\,a_{206} + 12\,a_{207} - 12\,a_{208} + 2\,a_{201} - 2\,a_{202} + 24\,a_{64} + a_{185} + 2\,a_{186} + 4\,a_{187} + 8\,a_{188} + 3\,a_{189} + 6\,a_{190} + 12\,a_{191} + 24\,a_{192} + 4\,a_{250} + a_{10} + 2\,a_{233} + 2\,a_{234} + 2\,a_{235} + 2\,a_{236} + 12\,a_{237} + 12\,a_{238} + 12\,a_{239} + 12\,a_{240} + a_{139} = 0$

301. $-a_{24} + 2\,a_{28} + a_{152} + 2\,a_{156} - 3\,a_{32} + 2\,a_{12} + a_{136} + 3\,a_{160} - 3\,a_{16} - a_{56} + 2\,a_{60} - a_{40} + 2\,a_{44} - a_8 + 2\,a_{216} + 8\,a_{220} + 24\,a_{224} + a_{168} + 2\,a_{172} + 3\,a_{176} + 2\,a_{140} - 3\,a_{48} + 3\,a_{144} + 2\,a_{248} + 8\,a_{252} + 24\,a_{256} +$

$8\,a_{204} + 24\,a_{208} + 2\,a_{200} - 3\,a_{64} + a_{184} + 2\,a_{188} + 3\,a_{192} + 2\,a_{232} + 8\,a_{236} + 24\,a_{240} = 0$

302. $-a_{23} + a_{151} + 2\,a_{155} + a_{135} - 3\,a_{31} - 6\,a_{12} + 2\,a_{11} - 3\,a_{15} - 3\,a_{136} + 3\,a_{143} + 3\,a_{159} + 9\,a_{16} - a_{55} - 6\,a_{56} + 2\,a_{59} + 12\,a_{60} - 3\,a_{63} - a_{39} - 3\,a_{40} + 2\,a_{43} + 6\,a_{44} - 3\,a_{47} - a_7 + 3\,a_8 + 2\,a_{215} + 8\,a_{219} + 24\,a_{223} + a_{167} + 3\,a_{168} + 2\,a_{171} + 6\,a_{172} + 3\,a_{175} + 9\,a_{176} - 6\,a_{140} - 9\,a_{48} - 9\,a_{144} + 2\,a_{247} + 12\,a_{248} + 8\,a_{251} + 48\,a_{252} + 24\,a_{255} + 144\,a_{256} + 2\,a_{27} + 8\,a_{203} - 24\,a_{204} + 24\,a_{207} - 72\,a_{208} - 6\,a_{200} + 2\,a_{199} - 18\,a_{64} + a_{183} + 6\,a_{184} + 2\,a_{187} + 12\,a_{188} + 3\,a_{191} + 18\,a_{192} + 2\,a_{231} + 6\,a_{232} + 8\,a_{235} + 24\,a_{236} + 24\,a_{239} + 72\,a_{240} + 2\,a_{139} = 0$

303. $a_{150} + 2\,a_{154} - a_{22} - 2\,a_{135} - 3\,a_{30} + a_{134} + 6\,a_{12} - 4\,a_{11} + 6\,a_{15} - 3\,a_{14} + 3\,a_{136} + 3\,a_{142} - 6\,a_{143} + 3\,a_{158} - 9\,a_{16} - a_{54} - 4\,a_{55} - 12\,a_{56} + 2\,a_{58} + 8\,a_{59} + 24\,a_{60} - 3\,a_{62} - 12\,a_{63} - a_{38} - 2\,a_{39} - 3\,a_{40} + 2\,a_{42} + 4\,a_{43} + 6\,a_{44} - 3\,a_{46} - 6\,a_{47} - a_6 + 2\,a_7 - 3\,a_8 + 2\,a_{214} + 24\,a_{222} + a_{166} + 2\,a_{167} + 3\,a_{168} + 2\,a_{170} + 4\,a_{171} + 6\,a_{172} + 3\,a_{174} + 6\,a_{175} + 9\,a_{176} + 6\,a_{140} - 9\,a_{48} + 9\,a_{144} + 8\,a_{218} + 2\,a_{246} + 8\,a_{247} + 24\,a_{248} + 32\,a_{251} + 96\,a_{252} + 24\,a_{254} + 96\,a_{255} + 288\,a_{256} + 2\,a_{138} - 16\,a_{203} + 24\,a_{204} + 24\,a_{206} - 48\,a_{207} + 72\,a_{208} + 2\,a_{198} + 6\,a_{200} + 8\,a_{202} - 4\,a_{199} - 36\,a_{64} + a_{182} + 4\,a_{183} + 12\,a_{184} + 2\,a_{186} + 8\,a_{187} + 24\,a_{188} + 3\,a_{190} + 12\,a_{191} + 36\,a_{192} + 8\,a_{250} + 2\,a_{10} + 2\,a_{230} + 4\,a_{231} + 6\,a_{232} + 8\,a_{234} + 16\,a_{235} + 24\,a_{236} + 24\,a_{238} + 48\,a_{239} + 72\,a_{240} - 4\,a_{139} + 2\,a_{26} = 0$

304. $2\,a_{25} + a_{149} + 2\,a_{153} + 3\,a_{157} - a_{37} - a_{21} + a_{135} - 3\,a_{29} + 3\,a_{141} + a_{133} - a_{134} - 2\,a_{12} + 2\,a_{11} - 3\,a_{15} + 3\,a_{14} - a_{136} - 3\,a_{142} + 3\,a_{143} - 3\,a_{13} + 3\,a_{16} - a_{53} - 2\,a_{54} - 4\,a_{55} - 8\,a_{56} + 2\,a_{57} + 4\,a_{58} + 8\,a_{59} + 16\,a_{60} - 3\,a_{61} - 6\,a_{62} - 12\,a_{63} - a_{38} - a_{39} - a_{40} + 2\,a_{41} + 2\,a_{42} + 2\,a_{43} + 2\,a_{44} - 3\,a_{45} - 3\,a_{46} - 3\,a_{47} - a_5 + a_6 - a_7 + a_8 + 2\,a_9 + 2\,a_{213} + 8\,a_{217} + 24\,a_{221} + a_{165} + a_{166} + a_{167} + a_{168} + 2\,a_{169} + 2\,a_{170} + 2\,a_{171} + 2\,a_{172} + 3\,a_{173} + 3\,a_{174} + 3\,a_{175} + 3\,a_{176} - 2\,a_{140} - 3\,a_{48} - 3\,a_{144} + 2\,a_{137} + 2\,a_{245} + 4\,a_{246} + 8\,a_{247} + 16\,a_{248} + 8\,a_{249} + 32\,a_{251} + 64\,a_{252} + 24\,a_{253} + 48\,a_{254} + 96\,a_{255} + 192\,a_{256} - 2\,a_{138} + 8\,a_{203} - 8\,a_{204} + 24\,a_{205} - 24\,a_{206} + 24\,a_{207} - 24\,a_{208} - 2\,a_{198} - 2\,a_{200} + 8\,a_{201} - 8\,a_{202} + 2\,a_{199} + 2\,a_{197} - 24\,a_{64} + a_{181} + 2\,a_{182} + 4\,a_{183} + 8\,a_{184} + 2\,a_{185} + 4\,a_{186} + 8\,a_{187} + 16\,a_{188} + 3\,a_{189} + 6\,a_{190} + 12\,a_{191} + 24\,a_{192} + 16\,a_{250} - 2\,a_{10} + 2\,a_{229} + 2\,a_{230} + 2\,a_{231} + 2\,a_{232} + 8\,a_{233} + 8\,a_{234} + 8\,a_{235} + 8\,a_{236} + 24\,a_{237} + 24\,a_{238} + 24\,a_{239} + 24\,a_{240} + 2\,a_{139} = 0$

305. $a_{24} - a_{28} + a_{152} + a_{156} - a_{20} + a_{32} + a_{132} - a_{12} + a_{136} + a_{160} + a_{16} + a_{56} - a_{60} + a_{40} - a_{44} - a_4 + a_8 + 4\,a_{216} + 8\,a_{220} + 16\,a_{224} + 2\,a_{228} + a_{168} + a_{172} + a_{176} + a_{140} + a_{48} + a_{144} + a_{180} + 4\,a_{248} + 8\,a_{252} + 16\,a_{256} - a_{36} + 8\,a_{204} + 16\,a_{208} + 4\,a_{200} + a_{64} + a_{184} + a_{188} + a_{192} + a_{148} + 2\,a_{244} - a_{52} + a_{164} + 2\,a_{212} + 4\,a_{232} + 8\,a_{236} + 16\,a_{240} + 2\,a_{196} = 0$

306. $-a_{19} + a_{23} + a_{151} + a_{155} + a_{131} + a_{135} + a_{31} - 3\,a_{132} + 3\,a_{12} - a_{11} - a_3 + a_{15} - 3\,a_{136} + a_{143} + a_{159} - 3\,a_{16} + a_{55} + 6\,a_{56} - a_{59} - 6\,a_{60} + a_{63} + a_{39} + 3\,a_{40} - a_{43} - 3\,a_{44} + a_{47} + 3\,a_4 + a_7 - 3\,a_8 + 4\,a_{215} + 8\,a_{219} + 16\,a_{223} + 2\,a_{227} + 6\,a_{228} + a_{163} + a_{167} + 3\,a_{168} + a_{171} + 3\,a_{172} + a_{175} + 3\,a_{176} - 3\,a_{140} + 3\,a_{48} - a_{51} - 3\,a_{144} + a_{179} + 6\,a_{180} + 4\,a_{247} + 24\,a_{248} + 8\,a_{251} + 48\,a_{252} + 16\,a_{255} + 96\,a_{256} - a_{35} - 3\,a_{36} - a_{27} + 8\,a_{203} - 24\,a_{204} + 16\,a_{207} - 48\,a_{208} - 12\,a_{200} + 4\,a_{199} + 6\,a_{64} + a_{183} + 6\,a_{184} + a_{187} + 6\,a_{188} + a_{191} + 6\,a_{192} + a_{147} + 2\,a_{243} + 12\,a_{244} - 6\,a_{52} + 3\,a_{164} + 2\,a_{211} + 4\,a_{231} + 12\,a_{232} + 8\,a_{235} + 24\,a_{236} + 16\,a_{239} + 48\,a_{240} + a_{139} + 2\,a_{195} - 6\,a_{196} = 0$

307. $a_{150} + a_{154} + a_{22} - 2\,a_{131} + a_{130} - 2\,a_{135} + 3\,a_{132} + a_{30} + a_{134} - 3\,a_{12} + 2\,a_{11} + 2\,a_3 - 2\,a_{15} + a_{14} + 3\,a_{136} + a_{142} - 2\,a_{143} + a_{158} +$

$3\,a_{16}+a_{54}+4\,a_{55}+12\,a_{56}-a_{58}-4\,a_{59}-12\,a_{60}+a_{62}+4\,a_{63}+a_{38}+2\,a_{39}+3\,a_{40}-a_{42}-2\,a_{43}-3\,a_{44}+a_{46}+2\,a_{47}-3\,a_4+a_6-2\,a_7+3\,a_8+4\,a_{214}+16\,a_{222}+2\,a_{226}+4\,a_{227}+6\,a_{228}+a_{162}+2\,a_{163}+a_{166}+2\,a_{167}+3\,a_{168}+a_{170}+2\,a_{171}+3\,a_{172}+a_{174}+2\,a_{175}+3\,a_{176}+3\,a_{140}+3\,a_{48}-a_{50}-4\,a_{51}+3\,a_{144}-a_{18}+8\,a_{218}+a_{178}+4\,a_{179}+12\,a_{180}+4\,a_{246}+16\,a_{247}+48\,a_{248}+32\,a_{251}+96\,a_{252}+16\,a_{254}+64\,a_{255}+192\,a_{256}+a_{138}-2\,a_{35}-3\,a_{36}-16\,a_{203}+24\,a_{204}+16\,a_{206}-32\,a_{207}+48\,a_{208}+4\,a_{198}+12\,a_{200}+8\,a_{202}-8\,a_{199}+12\,a_{64}+a_{182}+4\,a_{183}+12\,a_{184}+a_{186}+4\,a_{187}+12\,a_{188}+a_{190}+4\,a_{191}+12\,a_{192}+a_{146}+8\,a_{243}+24\,a_{244}-12\,a_{52}+8\,a_{250}-a_{10}-a_{34}-a_2+3\,a_{164}+2\,a_{210}+4\,a_{230}+8\,a_{231}+12\,a_{232}+8\,a_{234}+16\,a_{235}+24\,a_{236}+16\,a_{238}+32\,a_{239}+48\,a_{240}-2\,a_{139}+2\,a_{242}-a_{26}+2\,a_{194}-4\,a_{195}+6\,a_{196}=0$

308. $-a_{25}+a_{149}+a_{153}+a_{157}+a_{37}-a+a_{129}+a_{131}+a_{21}-a_{130}+a_{135}-a_{132}+a_{29}+a_{141}+a_{133}-a_{134}+a_{12}-a_{11}-a_3+a_{15}-a_{14}-a_{136}-a_{142}+a_{143}+a_{13}-a_{16}+a_{53}+2\,a_{54}+4\,a_{55}+8\,a_{56}-a_{57}-2\,a_{58}-4\,a_{59}-8\,a_{60}+a_{61}+2\,a_{62}+4\,a_{63}+a_{38}+a_{39}+a_{40}-a_{41}-a_{42}-a_{43}-a_{44}+a_{45}+a_{46}+a_{47}+a_4+a_5-a_6+a_7-a_8-a_9+4\,a_{213}+8\,a_{217}+16\,a_{221}+2\,a_{226}+2\,a_{227}+2\,a_{228}+a_{162}+a_{163}+a_{165}+a_{166}+a_{167}+a_{168}+a_{169}+a_{170}+a_{171}+a_{172}+a_{173}+a_{174}+a_{175}+a_{176}-a_{140}+2\,a_{209}+a_{48}-a_{17}-2\,a_{50}-4\,a_{51}-a_{144}+2\,a_{241}+a_{137}+2\,a_{178}+4\,a_{179}+8\,a_{180}-a_{49}+a_{177}+4\,a_{245}+8\,a_{246}+16\,a_{247}+32\,a_{248}+8\,a_{249}+32\,a_{251}+64\,a_{252}+16\,a_{253}+32\,a_{254}+64\,a_{255}+128\,a_{256}-a_{138}-a_{33}-a_{35}-a_{36}+8\,a_{203}-8\,a_{204}+16\,a_{205}-16\,a_{206}+16\,a_{207}-16\,a_{208}-4\,a_{198}-4\,a_{200}+8\,a_{201}-8\,a_{202}+4\,a_{199}+4\,a_{197}+8\,a_{64}+a_{181}+2\,a_{182}+4\,a_{183}+8\,a_{184}+a_{185}+2\,a_{186}+4\,a_{187}+8\,a_{188}+a_{189}+2\,a_{190}+4\,a_{191}+8\,a_{192}+8\,a_{243}+16\,a_{244}-8\,a_{52}+16\,a_{250}+a_{10}-a_{34}+a_2+2\,a_{225}+a_{164}+4\,a_{229}+4\,a_{230}+4\,a_{231}+4\,a_{232}+8\,a_{233}+8\,a_{234}+8\,a_{235}+8\,a_{236}+16\,a_{237}+16\,a_{238}+16\,a_{239}+16\,a_{240}+a_{161}+a_{139}+4\,a_{242}-2\,a_{194}+2\,a_{195}-2\,a_{196}+2\,a_{193}+a_{145}=0$