

Friedrich-Alexander-Universität Erlangen-Nürnberg

**Lehrstuhl für Multimediakommunikation und  
Signalverarbeitung**

Prof. Dr.-Ing. André Kaup

Bachelorarbeit

**Analyse eines BM3D-basierten  
Super-Resolution-Ansatzes für Video  
Anwendungen**

von Christoph Gernhardt

April 2014

Betreuer: Prof. Dr. Clemens Forster,  
Michel Bätz



## **Bachelorarbeit**

für

**Herrn Christoph Gernhardt**

### **Analyse eines BM3D-basierten Super-Resolution-Ansatzes für Video-Anwendungen**

#### **Evaluation of a BM3D-based Super-Resolution Approach for Video Applications**

In der digitalen Signalverarbeitung existieren verschiedene Ansätze zur Erhöhung der Auflösung von Bildern und Videos. Reguläre Interpolationsstrategien können dabei allerdings keine hochfrequenten Anteile rekonstruieren und somit keine Details wiederherstellen. Im Gegensatz dazu gibt es Algorithmen, die sich mehrerer niedrigaufgelöster Bilder aus einer Videosequenz bedienen und mit Hilfe von subpixelgenauen Verschiebungen zwischen den einzelnen Frames und vorhandenem Alias Details bis zu einem gewissen Grad wiederherstellen können. Solche Verfahren werden als Super-Resolution bezeichnet.

Ein relativ neues Verfahren zur Anwendung für Super-Resolution ist das Block-Matching 3-D (BM3D). Dieses Verfahren nutzt die nicht-lokale Ähnlichkeit zwischen Bildblöcken aus und verzichtet auf eine explizite 1-zu-1 Registrierung zugunsten einer 1-zu-vielen Abbildung. Darüber hinaus kommt eine leistungsfähige Modellierung im Transformationsbereich zum Einsatz.

Herr Gernhardt erhält nun die Aufgabe die Fähigkeit zur Auflösungserhöhung einzelner Frames einer Videosequenz mittels eines BM3D-basierten Super-Resolution-Ansatzes zu untersuchen. Der Algorithmus soll zunächst auf synthetisch erzeugten Videosequenzen ausgewertet werden, die ausschließlich translatorische Bewegungen beinhalten. Anschließend sollen auch echte Videosequenzen und Medizindatensätze mit komplexeren Bewegungsmodellen betrachtet werden. Des Weiteren sollen die Parameter des Algorithmus an das jeweilige Szenario angepasst werden.

Im Rahmen dieser Arbeit soll eine Literaturrecherche zur Bestimmung des Standes der Technik durchgeführt werden. Auf eine saubere Dokumentation der Arbeit und des Quellcodes wird besonderer Wert gelegt.

Ausgabe: 04.11.2013



(Prof. Dr.-Ing. A. Kaup)



# Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

---

Ort, Datum

---

Unterschrift



# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>V</b>
<b>Abkürzungsverzeichnis</b>	<b>VII</b>
<b>Formelzeichen</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>3</b>
2.1 Definition Super-Resolution . . . . .	3
2.2 BM3D . . . . .	4
2.2.1 Versionen des BM3D . . . . .	4
2.2.2 Allgemeiner Aufbau . . . . .	6
<b>3 Methodik/Durchführung</b>	<b>11</b>
3.1 Der Super-Resolution Ansatz . . . . .	11
3.2 Aufbau des Versuchs . . . . .	12
3.3 Getestete Parameter . . . . .	14
3.3.1 Anzahl der Iterationen $k$ . . . . .	14
3.3.2 Stärke des Filters $\sigma$ . . . . .	14
3.3.3 Blockgröße $N_1$ . . . . .	15
3.3.4 Größe des Stapels $N_2$ . . . . .	15
3.3.5 Schwellwert für die Ähnlichkeit von Blöcken $\tau_{match}$ . . . . .	15
3.3.6 Verwendete Transformation . . . . .	15

---

3.4	Verwendete Sequenzen . . . . .	17
3.4.1	Synthetisch erzeugte Sequenzen . . . . .	17
3.4.2	Reale Videosequenzen . . . . .	17
3.4.3	Medizindaten . . . . .	18
3.5	Umsetzung . . . . .	18
3.6	Risiken bei der Durchführung und Auswertung . . . . .	19
<b>4</b>	<b>Ergebnisse</b>	<b>21</b>
4.1	Allgemeine Betrachtungen . . . . .	21
4.1.1	Standardparameter . . . . .	21
4.2	synthetisch erzeugte Sequenzen . . . . .	25
4.2.1	Translation . . . . .	25
4.2.2	Rotation . . . . .	26
4.2.3	Vergleich mit anderen Algorithmen . . . . .	27
4.3	Echte Videosequenzen . . . . .	28
4.3.1	Getestete Parameter . . . . .	28
4.3.2	Vergleich mit anderen Algorithmen . . . . .	29
4.4	Medizindaten . . . . .	30
4.4.1	Zeitlicher Verlauf . . . . .	30
4.4.2	Örtlicher Verlauf . . . . .	31
4.4.3	Vergleich mit anderen Algorithmen . . . . .	32
4.5	Auswertung und Qualität der Ergebnisse . . . . .	33
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>35</b>
5.1	Zusammenfassung . . . . .	35
5.2	Ausblick . . . . .	35
	<b>Abbildungsverzeichnis</b>	<b>36</b>
	<b>Tabellenverzeichnis</b>	<b>37</b>



Inhaltsverzeichnis

III

---

**Literaturverzeichnis**

**40**



# Kurzfassung

In der digitalen Videoverarbeitung gibt es zahlreiche Ansätze zur Auflösungserhöhung. Hierbei wird zwischen Interpolations- und Super-Resolution-Algorithmen unterschieden. Letztere zeichnen sich speziell durch die Rekonstruktion von hochfrequenten Details aus. In dieser Arbeit wird ein Super-Resolution-Ansatz vorgestellt, der auf dem BM3D-Filter basiert. Zusätzlich werden die Testergebnisse der Parameteranalyse präsentiert und ein Vergleich mit anderen Algorithmen zur Auflösungserhöhung durchgeführt. Abschließend werden Empfehlungen zur Konfiguration für verschiedene Sequenzarten gegeben.



# Abkürzungsverzeichnis

BM3D	Block-Matching 3D-Filter
BM3DDEB	Block-Matching 3D-Deblurring-Filter
BM3DSHARP	Block-Matching 3D-Sharpeneing-Filter
BM4D	Block-Matching 4D-Filter
Block-DCT	blockweise,diskrete Cosinus Transformation
DCT	diskrete Cosinus Transformation
DFT	diskrete Fourier Transformation
MSE	mittlere quadratische Abweichung
PSNR	Spitzen-Signal-Rausch-Verhältnis
SR-Ansatz	Super-Resolution-Ansatz
VBM3D	Video-Block-Matching 3D-Filter



## Formelzeichen und Symbole

$\underline{Y}(t)$	Sequenz mit niedriger Auflösung
$\underline{X}(t)$	hochauflösendes Bild
$M(t)$	Unterabtastung und Unschärfe
$\underline{V}(t)$	Rauschfaktor
$d(R, Q)$	Distanz zwischen Referenzblock und untersuchtem Block
$R$	Referenzblock
$Q$	untersuchter Block
$\Upsilon$	threshold-Operator
$T_{2D}$	zweidimensionale Transformation in den Frequenzbereich
$\lambda_{thr}$	threshold-Parameter, Eingabeparameter des BM3D
$\sigma$	Standardabweichung des Rauschens
	Stärke des Filters im SR-Ansatz
$N$	Kantenlänge eines Blocks (in Pixel)
$\tau_{match}$	Schwellwert für Ähnlichkeit von Blöcken
$\hat{Y}$	Abschätzungen des Stapels
$T_{3D}^{-1}$	dreidimensionale Transformation in den Zeitbereich
$T_{3D}$	dreidimensionale Transformation in den Frequenzbereich
$S$	Stapel, der ähnliche Blöcke enthält
$\omega$	Gewichtung der einzelnen Blöcke
$\kappa$	Anzahl der Koeffizienten eines Blockes im Frequenzbereich
$W$	Koeffizient des Wiener Filters
$T_m$	Frequenzband des niedrig aufgelösten Bildes

$T_M$	Frequenzband des hochauflösenden Bildes
$\beta$	Skalierungsfaktor
$T_{m-1}$	Transformation in den Frequenzbereich
$U_{m-1,m}$	Vergrößerung des Frequenzspektrums von $x_{m-1}^h \times x_{m-1}^v$ auf $x_m^h \times x_m^v$
$N_1$	Größe eines Blocks
$N_{schritt}$	Abstand zwischen untersuchten Blöcken
$N_2$	maximale Anzahl von Blöcken im Stapel
$\hat{y}$	endgültige Schätzung eines Blocks
$\Phi$	VBM3D-Filter
$P_{\perp}$	Projektionsoperator
$T_m^{-1}$	Transformation in den Zeitbereich
$k$	Anzahl der Iterationen
MSE	mittlere quadratische Abweichung
I	Maximalwert eines Pixels







# 1 Einleitung

In vielen Bereichen der Wissenschaft und des Alltages, wie zum Beispiel der bildgebenden Medizin, der Videoüberwachung oder auch der Mikroskopie, gibt es Bedarf an hochauflösenden Bildern. Allerdings sind die Kosten für Kamerasysteme mit sehr guter Auflösung relativ hoch und diese Systeme stoßen an die Grenzen der technischen Realisierbarkeit, da sich die Größe einzelner Pixel kaum noch verringern lässt.

Eine preisgünstigere Alternative bietet die nachträgliche Erhöhung der Auflösung mit Hilfe von Algorithmen. Es existieren zwar viele Ansätze zur Auflösungserhöhung, allerdings können herkömmliche Interpolationen keine hochfrequenten Anteile rekonstruieren. Super-Resolution-Ansätze können hingegen diese Details wiederherstellen und sind somit auch geeignet, um die erforderliche Auflösungserhöhung und Detailrekonstruktion zu erreichen.

Ein solcher Super-Resolution-Ansatz ist der Block-Matching 3D-Algorithmus, der nicht-lokale Ähnlichkeiten einzelner Bildblöcke ausnutzt um so effektiv die Auflösung zu erhöhen. Ziel dieser Arbeit soll es sein, für verschiedene Anwendungsszenarien geeignete Parametereinstellungen für ein bestmögliches Resultat zu finden.

In Kapitel 2 werden zunächst die theoretischen Grundlagen und die Funktionsweise des BM3D erläutert. Der SR-Ansatz und die durchgeführten Tests werden in Kapitel 3 vorgestellt. Anschließend werden in die Testergebnisse präsentiert sowie Empfehlungen zur Konfiguration des Algorithmus für verschiedene Sequenzen gegeben.



## 2 Grundlagen

### 2.1 Definition Super-Resolution

Im Gegensatz zu regulären Interpolationen können SR-Ansätze auch hochfrequente Anteile zu einem gewissen Grad wiederherstellen. [FREM04]

Es existiert keine allgemein gültige Definition für den Begriff Super-Resolution, da zwischen beispielbasierten und Multi-Frame SR-Ansätzen unterschieden werden muss. Ein beispielbasierter Ansatz versucht eine Erhöhung der Auflösung mit Hilfe von Datenbankinformationen zu erreichen. [TJE02]

Unter Multi-Frame Super-Resolution, wozu auch der hier vorgestellte Ansatz zählt, versteht man die Rekonstruktion eines hochaufgelösten Bildes aus einer Bildfolge mit niedriger Auflösung. Dabei werden subpixelgenaue Verschiebungen der Aufnahmen erkannt und die so gewonnenen zusätzlichen Details zur Rekonstruktion verwendet.

Erstmalig erwähnt wurde dieser Ansatz schon 1984 bei Tsai und Huang. [TH84]. Die Theorie zur Sub-Pixel-Verschiebung wurde von Ur und Gross vorgestellt [HD92].

Allgemein lassen sich diese Ansätze in 3 Schritte gliedern:

1. Registrierung

Hierbei werden alle Bilder mit niedriger Auflösung auf das Koordinatennetz der hochauflösenden Aufnahme abgebildet.

2. Interpolation

Die fehlenden Pixel des neuen Koordinatennetzes werden für jedes Bild mit niedriger Auflösung mittels des angewandten Algorithmuses berechnet.

3. Abschätzung/Fusion

Die einzelnen Interpolationen werden gewichtet und zu einem hochauflösenden Bild zusammengefügt.

Super-Resolution wird auch als inverses Problem bezeichnet: Die Bildfolge mit niedriger Auflösung wird als Ergebnis einer Unterabtastung und Trübung des hochauflösenden Bildes betrachtet, welches zusätzlich noch mit additivem Rauschen überlagert ist. Dieses Konzept wird in Gleichung 2.1 beschrieben.

$$\underline{Y}(t) = M(t) * \underline{X}(t) + \underline{V}(t) \quad (2.1)$$

Hierbei beschreibt  $\underline{Y}(t)$  die Bildfolge mit niedriger Auflösung,  $\underline{X}$  das unbekannte, hochauflösende Bild und  $\underline{V}$  den Rauschfaktor.  $M(t)$  beschreibt dabei eine Kombination aus Unterabtastung und Unschärfe verursacht durch das aufnehmende System. Ziel der Super-Resolution ist es,  $\underline{X}(t)$  zu rekonstruieren. [FREM04]

## 2.2 BM3D

### 2.2.1 Versionen des BM3D

Grundlage des vorgestellten SR-Ansatzes ist der "Block-Matching and 3D-Filtering"-Algorithmus (BM3D). Es existieren inzwischen verschiedene Versionen, um unterschiedlichen Anwendungsszenarien gerecht zu werden. Dieser Abschnitt soll einen kurzen

Überblick zu den verschiedenen Versionen geben, wobei die Wichtigsten hervorgehoben wurden.

- **BM3D**

Ursprünglich wurde dieser Algorithmus zum Entrauschen von Graustufenbildern verwendet.[KAVK] IM Jahr 2008 wurde auf Grundlage dieses Modells eine Methode zur Auflösungserhöhung erstellt.[AAVK08] Allerdings entspricht dieser nicht dem Super-Resolution-Prinzip, da nur ein Bild mit niedriger Auflösung als Grundlage dient und keine Details rekonstruiert werden können.

- **BM3DDEB**

Hierbei handelt es sich um eine Version des BM3D, die speziell zum Entfernen von Blurring-Artefakten entwickelt wurde [DFKE08a]

- **BM3DSHARP**

Dies ist eine Version des BM3D-Filters zum Entfernen von Rauschen und Blurring-Artefakten. [DFKE07]

- **CBM3D**

Der "Color-Block-Matching 3D"- Algorithmus (CBM3D) erweitert das o.g. BM3D-Modell auf Farbbilder. Hierbei wird das RGB-Bild in ein YCbYr-Bild umgewandelt. Anschließend wird der Gruppierungsschritt des BM3D nur auf dem Y-Kanal (Grundhelligkeit) angewendet. Die restlichen Arbeitsschritte werden allerdings für alle drei Kanäle durchgeführt. [DFKEa]

- **VBM3D**

Der "Video-Block-Matching 3D-Algorithmus" (VBM3D)überträgt das Prinzip des BM3D auf Videos und Bildfolgen.[DFE07] Ein Vorteil gegenüber dem einfach BM3D besteht darin, dass auch Informationen aus anderen Frames extrahiert werden können. Aus diesem Grund dient diese Version auch als Grundlage für den in dieser Arbeit vorgestellten SR-Ansatz.[DFKE08b]

- CVBM3D

Der "Color-Video-Block-Matching 3D-Algorithmus" (CVBM3D) entspricht in seiner Funktionsweise dem VBM3D, mit dem Unterschied, dass er Farbaufnahmen verarbeitet. Es gibt noch keinen wissenschaftlichen Beitrag zu seiner Eignung für Super Resolution. Aus diesem Grund wird für den in dieser Arbeit vorgestellten Ansatz der VBM3D verwendet.

Auch wenn es keine Version des BM3D ist, so muss an dieser Stelle noch auf den neu entwickelten Block-Matching-4D (BM4D) hingewiesen werden. [MKEF13] Dieser Algorithmus verarbeitet Volumendaten, wie zum Beispiel Computertomographie-Aufnahmen, während die bisher erwähnten Versionen auf dreidimensionale Aufnahmen angewendet werden.

## 2.2.2 Allgemeiner Aufbau

In diesem Abschnitt soll genauer auf den Aufbau des BM3D-Algorithmus eingegangen werden. Da sich alle Versionen des BM3D minimal unterscheiden, wird hier speziell der VBM3D-Algorithmus dargestellt werden, da dieser die Grundlage für den in dieser Arbeit vorgestellten SR-Ansatz bildet.

Man kann den BM3D-Algorithmus in 2 Stufen unterteilen; Stufe eins wird als hard thresholding bezeichnet, Stufe zwei besteht aus einem Wiener Filter.

### hard thresholding

- Gruppierung

Im ersten Schritt werden Abschätzungen für die einzelnen Blöcke jedes Bildes/Frames erstellt. Hierzu werden zunächst ähnliche Bildblöcke erkannt und in einem 3-dimensionalen Vektor gespeichert. Dieser Schritt nennt sich Gruppierung.



rung. Ob sich Blöcke ähneln wird mittels der  $l^2$ -Norm bestimmt:

$$d(R, Q) = \frac{\Upsilon(T_{2D}(R), \lambda_{thr} * \sigma * \sqrt{2 * \log(N^2)}) - \Upsilon(T_{2D}(Q), \lambda_{thr} * \sigma * \sqrt{2 * \log(N^2)})}{N} \quad (2.2)$$

Hierbei beschreibt  $d$  die Distanz der Blöcke,  $R$  entspricht dem Referenzblock und  $Q$  dem zu vergleichenden Block.  $T_{2D}$  entspricht einer Transformation in den Frequenzbereich (z.B. mittels Diskreter Cosinus Transformation) und  $N$  ist die Kantenlänge der Blöcke.  $\Upsilon$  ist der threshold-Operator. Er wird mit

$$\Upsilon(\lambda, \lambda_{thr}) = \begin{cases} \lambda, & \text{falls } |\lambda| > \lambda_{thr} \\ 0, & \text{sonst} \end{cases} \quad (2.3)$$

berechnet.

Für  $d(R, Q) < \tau_{match}$  ist Block  $Q$  ähnlich dem Referenzblock und wird in den Stapel aufgenommen. Wenn kein vergleichener Block dieses Kriterium erfüllt, befindet sich lediglich der Referenzblock im Stapel. [DFKEb],[DFE07]

Welche Blöcke auf Ähnlichkeit untersucht werden, hängt von den Parametern  $N_s$  und  $N_t$  ab. Hierbei beschreibt  $N_s$  das örtliche Suchfenster und  $N_t$  das zeitliche Suchfenster, welches die Anzahl der zu durchsuchenden Frames entlang der Zeitachse festlegt. Die maximale Anzahl an Blöcken im Stapel wird durch den Parameter  $N_2$  begrenzt.

Zwei Beispiele für die Blockfindung sind in Abbildung 2.1 angegeben. Der Referenzblock wird rot dargestellt, alle Blöcke, die das Kriterium der Ähnlichkeit erfüllen sind blau gefärbt. Je nach Distanz zum Referenzblock erfolgt die Wichtung der ähnlichen Blöcke, wobei eine geringere Distanz zu einer höheren Wichtungswert führt.

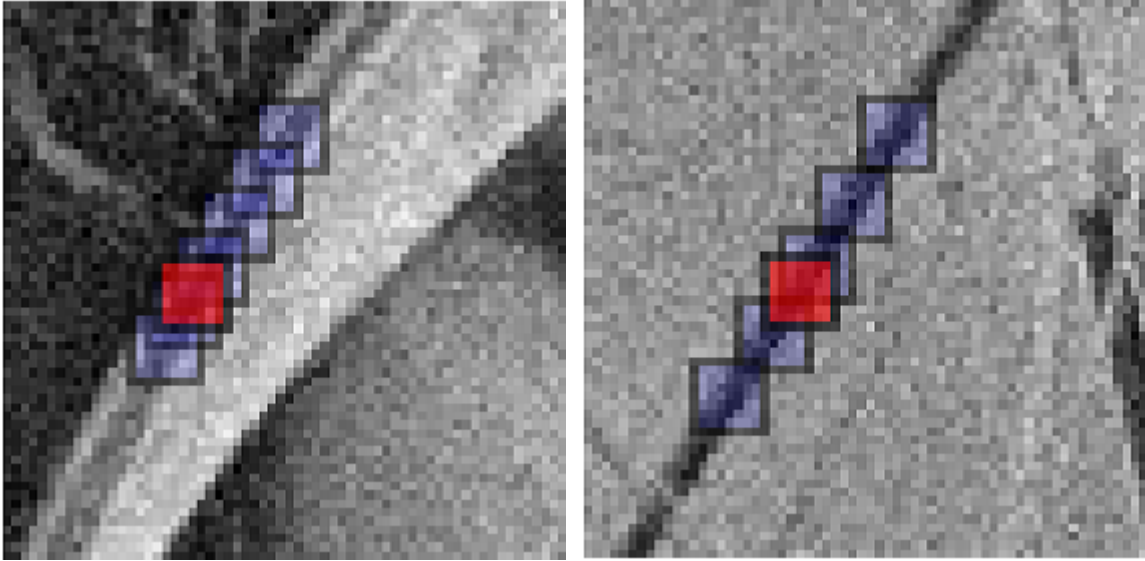


Abbildung 2.1: Beispiele für die Blocksuche [DFKEc]

- **hard thresholding**

Nach der Gruppierung folgt der eigentliche hard- thresholding-Schritt. Hierzu wird der Stapel zunächst in den Frequenzbereich transformiert. Die einzelnen Blöcke werden mittels einer zweidimensionalen Transformation (DCT, DST, bi-or) umgewandelt. Zusätzlich erfolgt eine Transformation entlang des Stapels, dazu wird beispielsweise ein Haar-Wavelet verwendet. In [Leb12] können Untersuchungen mit verschiedenen Transformationen verglichen werden.

$$\hat{Y} = T_{3D}^{-1}(\Upsilon(T_{3D}(S), \lambda_{thr3D} * \sigma * \sqrt{2\log N^2})) \quad (2.4)$$

$T_{3D}$  beschreibt die Transformation in den Frequenzbereich,  $T_{3D}^{-1}$  ist die Rücktransformation in den Zeitbereich.  $\lambda_{thr3D}$  ist der Thresholdparameter. Er gehört, wie  $\sigma$  zu den Eingabeparametern des BM3D.  $\hat{Y}$  enthält die Abschätzungen für die Blöcke aus dem Stapel  $S$ .

Diese Abschätzungen werden nun mittels der Gleichung

$$\omega = \begin{cases} \frac{1}{\kappa}, & \text{falls } \kappa \geq 1 \\ 1, & \text{sonst} \end{cases} \quad (2.5)$$

gewichtet.  $\kappa$  ist die Anzahl der Koeffizienten der Transformation eines Blockes.

- **Abschätzung**

Im letzten Schritt werden die so gewonnenen Blöcke fusioniert, um eine endgültige Schätzung  $\hat{y}$  des hochauflösenden Bild zu bekommen.

### Wiener Filter

Da auf diesen Teil des BM3D-Algorithmus im vorgestellten SR-Ansatz verzichtet wird, soll hier nur eine kurze Erläuterung folgen.

Der allgemeine Aufbau gleicht der hard-thresholding Stufe; Gruppierung und Abschätzung laufen ähnlich ab und das thresholding wird durch einen Wiener Filter ersetzt.

- **Gruppierung**

Statt von den gegebenen niedrig aufgelösten Bildern auszugehen, werden hier die Abschätzungen aus dem hard-thresholding-Schritt durchsucht.

- **Entrauschen**

An dieser Stelle kommt der Wiener Filter zum Einsatz; Zunächst werden die Koeffizienten des Wiener Filters ( $W$ ) mittels der Gleichung

$$W = \frac{|T_{3D}(S)|^2}{|T_{3D}(S)|^2 + \sigma^2} \quad (2.6)$$

berechnet.  $S$  entspricht dem Stapel der Blöcke aus dem Gruppierungsschritt. Anschließend erfolgt, wie beim hard thresholding, eine Gewichtung und die Fusion der Abschätzungen.

Die mathematischen Formeln wurden aus [DFKEc] adaptiert und im Sinne des VBM3D [DFE07] angepasst.



## 3 Methodik/Durchführung

### 3.1 Der Super-Resolution Ansatz

Als Grundlage für dieses Kapitel dient der Artikel [DFKE08b], in dem der untersuchte SR-Ansatz vorgestellt wird.

Als Grundannahme für diesen Ansatz gilt, dass jedes Frequenzband der niedrig aufgelösten Bilder  $T_m$  einen Teil des Frequenzbandes des hochauflösenderen Bildes  $T_M$  darstellt. Das Frequenzband kann durch Orthonormaltransformationen wie DCT, DFT oder auch einer Block-DCT ermittelt werden.

$T_m$  hat die Größe  $x_0^h \times x_0^v$  und es gelten die Bedingungen  $x_0^h < x_m^h$  und  $x_0^v < x_m^v$ .

Damit lässt sich Gleichung 2.1 für dieses Problem anpassen und ergibt die Gleichung

$$y_0 = T_0^{-1}(\beta_{0,m}^{-1} T_m(y_m)) \quad (3.1)$$

$y_0$  entspricht einem Bild mit niedriger Auflösung und  $y_m$  dem hochauflösenden Bild.  $\beta$  ist der Skalierungsfaktor. Er stellt sicher, dass die Mittelwerte der Frequenzbänder von  $y_0$  und  $y_m$  gleich groß sind. Der Skalierungsfaktor wird mit der Formel

$$\beta_{0,m} = \sqrt{\frac{x_m^h * x_m^v}{x_0^h * x_0^v}} \quad (3.2)$$

berechnet.

Ziel ist es, wie in Kapitel 2.1 erwähnt,  $y_M$  ermitteln. Ob es sich dabei um Auflösungserhöhung oder SR handelt, hängt von der Anzahl der Bilder mit niedriger Auflösung ab: Bei lediglich einem niedrig aufgelösten Bild handelt es sich um Auflösungserhöhung

[AAVK08], bei mehreren um Super Resolution [DFKE08b].

Um ein besseres Resultat zu erzielen, soll die Skalierung in mehreren Schritten erfolgen. Dabei wird in jedem Schritt das aktuelle Bild der Größe  $x_m^h \times x_m^v$  auf die Größe  $x_{m+1}^h \times x_{m+1}^v$  gebracht. Es muss beachtet werden, dass der Skalierungsfaktor  $\beta$  gemäß Gleichung 3.2 angepasst wird. Das Ergebnis der vorherigen Stufe dient jeweils als Grundlage für die nächste Stufe, bis die gewünschte Auflösung erreicht wird.

## 3.2 Aufbau des Versuchs

Das Blockschaltbild für den SR-Ansatz ist in Abbildung 3.1 dargestellt. Die innere Schleife entspricht den einzelnen Iterationen und die äußere Schleife dem Übergang zwischen den einzelnen Stufen.

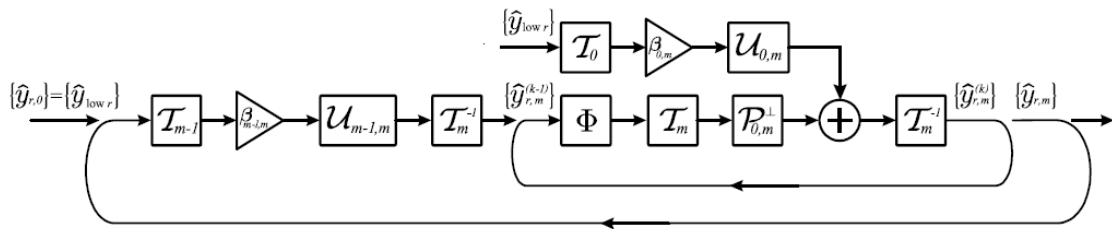


Abbildung 3.1: Blockschaltbild des SR-Ansatzes [DFKE08b]

Im ersten Durchlauf der äußeren Schleife ist die Sequenz der niedrig aufgelösten Bilder der Input. Falls mehrere Stufen vorgesehen sind, so ist als Input das Ergebnis der vorherigen Stufe zu wählen.

Zunächst wird die Bildsequenz in den Frequenzbereich transformiert ( $\mathcal{T}_{m-1}$ ). Der Index  $m-1$  beschreibt dabei die Größe der zweidimensionalen Transformation.

Es folgt eine Multiplikation mit dem Skalierungsfaktor  $\beta$  um, wie in Kapitel 3.1 erwähnt, den Mittelwert der Transformation nach der folgenden Auflösungserhöhung konstant zu halten. Die Berechnung von  $\beta$  ist aus Gleichung 3.2 bekannt und muss lediglich an

die Dimensionen der einzelnen Stufen angepasst werden.

$U_{m-1,m}$  entspricht der Vergrößerung des Spektrums auf die gewünschte Auflösung  $x_m^h \times x_m^v$ . Je nach gewählter Transformation ( $T_{m-1}$ ) gestaltet sich dieser Schritt sehr unterschiedlich. Während bei der DCT lediglich Spalten und Zeilen mit dem Wert 0 an das Bild angehängt werden, müssen bei der Block-DCT Spalten und Zeilen gemäß Abbildung 3.2 eingefügt werden. Der dunkle Bereich der Abbildung zeigt das ursprüngliche Spektrum, hellgrau und weiß das neue Spektrum nach ein- bzw. mehrfacher Vergrößerung.

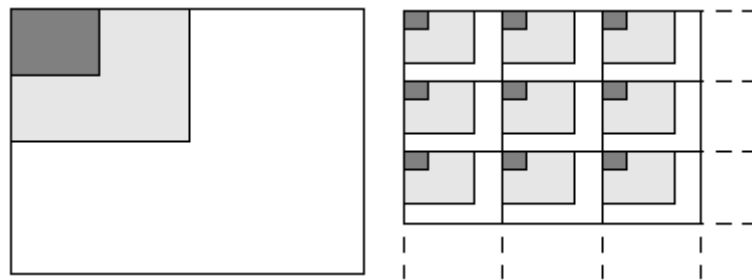


Abbildung 3.2: Veranschaulichung der Vergrößerung des Spektrums für DCT (links) und Block - DCT (rechts) [DFKE08b]

Nachdem die Vergrößerung des Spektrums durchgeführt wurde, erfolgt eine Rücktransformation in den Zeitbereich  $T_m^{-1}$ . Die Größenänderung des Spektrums wurde durch Anpassung des Index von  $m - 1$  zu  $m$  berücksichtigt.

$\Phi$  ist das Symbol für den VBM3D-Filter. Er wird ohne den Wiener Filter verwendet, da von rauschfreien Bildern ausgegangen wird. Soll eine Sequenz betrachtet werden, die mit Rauschen überlagert ist, so muss vor der Durchführung dieses Ansatzes ein Entrauschen durchgeführt werden. Die Übersicht der verwendeten Standard-Eingabeparameter kann in Tabelle 4.1 eingesehen werden. Eine Übersicht der untersuchten Parameter befindet sich in Kapitel 3.3 .

Nach einer erneuten Transformation  $T_m$  in den Frequenzbereich kommt der Projektionsoperator  $P_{\perp}$  zum Einsatz. Hiermit werden alle Koeffizienten von  $T_m$ , die schon in  $T_{m-1}$  existierten, zu Null gesetzt. Dieser Schritt ist nötig, um die Originalinformationen aus den niedrig aufgelösten Bild zu erhalten und nur die fehlenden Informationen zu rekonstruieren.

Die Originaldaten aus der Ausgangssequenz werden auf die neue Größe skaliert und zu dem Ergebnis des Projektionsoperators hinzuaddiert. Unabhängig von der aktuellen Stufe des Prozesses wird in jedem Fall die Originalsequenz verwendet.

Abschließend wird eine Rücktransformation  $T_m^{-1}$  in den Zeitbereich durchgeführt.

## 3.3 Getestete Parameter

### 3.3.1 Anzahl der Iterationen $k$

Die Anzahl der Iterationen gibt an, wie oft die in Abbildung 3.1 dargestellte innere Schleife durchlaufen wird. Die Erwartungshaltung ist, dass mit steigender Anzahl an Iterationen auch die Qualität des Ergebnisses steigt.

### 3.3.2 Stärke des Filters $\sigma$

$\sigma$  gehört zu den Eingabeparametern des BM3D. Der BM3D wurde ursprünglich zum Entrauschen eingesetzt. Dabei beschrieb dieser Parameter die erwartete Standardabweichung des Rauschens. Da in diesem Ansatz von rauschfreien Bildern ausgegangen wird, regelt dieser Parameter die Stärke des Filters.

Da bei mehreren Iterationen mit dem selben Wert für  $\sigma$  Unschärfen auftreten, wird er bei jeder Iteration um den Faktor  $\Delta_{\sigma}$  verringert.



### 3.3.3 Blockgröße $N_1$

Die Blockgröße  $N_1$  ist einer der wichtigsten Eingabeparameter. Er beschreibt die Kantenlänge der untersuchten Blöcke. An ihn sind auch weitere Eingabeparameter gekoppelt: Mit steigender Größe muss auch die Schrittweite  $N_{schritt}$  angepasst werden. Für alle durchgeführten Versuche wurde die Schrittweite auf  $N_{schritt} = \frac{N_1}{2}$  festgelegt. Aus Gleichung 2.2 ist zudem ersichtlich, dass auch die Blockdistanz  $d$  von der Blockgröße abhängig ist. Gemäß den Angaben aus [DFKE08b] wird die Blockgröße nach jeder Stufe verringert.

### 3.3.4 Größe des Stapels $N_2$

Die Größe des Stapels definiert die maximale Anzahl ähnlicher Blöcke im Stapel. Der Wert für  $N_2$  wird in der Regel hoch gewählt, um wirklich alle Blöcke, die die Ähnlichkeitsbedingung erfüllen, zu erfassen.

### 3.3.5 Schwellwert für die Ähnlichkeit von Blöcken $\tau_{match}$

$\tau_{match}$  legt fest, ab welchen Wert für  $d(R,Q)$  Blöcke als ähnlich gelten und in den Stapel aufgenommen werden.

### 3.3.6 Verwendete Transformation

In "Image and video super resolution via spatially adaptive blockmatching filtering" [DFKE08b] werden mehrere mögliche Transformationen vorgeschlagen. In dieser Arbeit werden speziell die diskrete Cosinus Transformation (DCT) und die blockweise, diskrete Cosinus Transformation (Block-DCT) untersucht. Diese Transformationen unterscheiden sich wesentlich bei der durchgeführten Spektrumvergrößerung. Um vergleichbare Ergebnisse zu erhalten, wurden die Vergrößerungsstufen angepasst. In der ersten Stufe wird die Auflösung verdoppelt und in der zweiten Stufe wird sie um den Faktor 1,5 erhöht. Diese Werte sind durch die Größen der Transformationsblöcke der Block-DCT

bedingt. Bei jeder Spektrumvergrößerung  $U_{m-1,m}$  wird die Blockgröße erhöht. Bei der Erhöhung von 1x1 auf 2x2 wird so die Auflösung verdoppelt, bei der Erhöhung von 2x2 auf 3x3 steigt sie um den Faktor 1,5.

## 3.4 Verwendete Sequenzen

### 3.4.1 Synthetisch erzeugte Sequenzen

Das Erzeugen von synthetischen Sequenzen dient dazu, genauere Informationen hinsichtlich des Einflusses bestimmter Bewegungsarten auf die Qualität der Ergebnisse zu untersuchen.

#### Sequenz mit translatorischen Bewegungen

Konkret wurde eine Videosequenz mit ausschließlich translatorischen Bewegungen erzeugt. Sie dient dazu, den Einfluss von Sub-Pixel-Verschiebungen auf das Ergebnis zu untersuchen.

Die erzeugte Bildfolge besteht aus 9 Bildern. Die erste Aufnahme entspricht dem Originalbild. Die weiteren Aufnahmen sind horizontal bzw. vertikal um jeweils einen Pixel verschoben. Anschließend wurde die Sequenz getrübt und verkleinert.

#### Sequenz mit Rotationsbewegung

Es wurde eine Bildfolge erstellt, in der die Aufnahme jeweils um  $1^\circ$  im Vergleich zum vorherigen Bild gedreht wurde. Der Drehwinkel wurde bewusst hoch gewählt, um einen möglichen Effekt auf das Ergebnis zu verstärken. Auch hier wurde die Sequenz anschließend getrübt und verkleinert.

### 3.4.2 Reale Videosequenzen

Es wurden die Sequenzen 'foreman' aus dem Originalartikel und 'Race Horses' aus dem Datenpool des Lehrstuhls ausgewählt. Es wurden jeweils die ersten 75 Frames betrachtet. Diese Sequenzen bilden den Kern der Parameteruntersuchung.

### 3.4.3 Medizindaten

Als Beispiel für medizinische Daten dient eine Computertomographie des Herzens. Da diese Aufnahmen vierdimensional sind, der BM3D aber nur drei Dimensionen untersuchen kann, wird zwischen zeitlichen und örtlichen Verlauf unterschieden. Der zeitliche Verlauf beschreibt die Veränderung einer bestimmten Schicht der Computertomographie über den Zeitraum des Scans. Der örtliche Verlauf entspricht einem Durchlauf der Schichten an einem festen Zeitpunkt  $t$ .

## 3.5 Umsetzung

Die Videosequenzen wurden mit *MATLAB*® eingelesen und mit Hilfe eines Gauss-Filters der Größe  $3 \times 3$  getrübt. Abschließend wurde eine Unterabtastung auf die Hälfte der ursprünglichen Kantenlängen durchgeführt. Die Trübung wurde durchgeführt, um einen Sensor mit geringerer Auflösung zu simulieren. Mittels des vorgestellten SR-Ansatzes wird die so entstehende Sequenz auf die ursprüngliche Auflösung rekonstruiert und mit der Originalaufnahme verglichen.

In der Regel wird Super-Resolution zur Auflösungserhöhung angewendet. Um zu untersuchen, inwiefern der hier vorgestellte Ansatz zur Rekonstruktion geeignet ist und um eine Vergleichbarkeit zu erreichen, sind diese Schritte nötig, um eine qualitative Bewertung der Endergebnisse durchführen zu können.

Als Vergleichskriterium dient das Spitzen-Signal-Rausch-Verhältnis (PSNR).

### Das Spitzen-Signal-Rausch-Verhältnis (PSNR)

Das PSNR wird mit Hilfe der Formel

$$PSNR = 10 * \lg\left(\frac{I^2}{MSE}\right)dB \quad (3.3)$$

berechnet, wobei MSE die mittlere quadratische Abweichung beschreibt. Das PSNR wird in Dezibel (dB) angegeben. Die mittlere quadratische Abweichung wird mit Hilfe

der Formel

$$MSE = \frac{1}{m * n} * \sum_{j=0}^m \sum_{k=0}^n (A(j, k) - B(j, k))^2 \quad (3.4)$$

ermittelt.  $m$  und  $n$  entsprechen den Dimensionen der Bilder  $A$  und  $B$ . Eine der Aufnahmen ist dabei das Referenzbild und die Andere die Rekonstruktion. Sind beide Bilder identisch, gilt  $PSNR \rightarrow \infty$ .  $I$  ist der Maximalwert, den ein einzelner Pixel annehmen kann. Bei einer Bit-Tiefe von 8 Bit entspricht der Maximalwert  $I = 255$ . [Mö10] In dieser Arbeit wird der PSNR-Wert über das gesamte Bild berechnet, während in Artikel [DFKE08b] ein Randbereich von 15 Pixel ignoriert wird.

### 3.6 Risiken bei der Durchführung und Auswertung

Das größte Risiko besteht in der Korrelation zwischen den einzelnen Parametern. Die Parameter wurden getrennt voneinander untersucht. Dies kann dazu führen, dass der Einfluss einzelner Parameter nicht in vollem Umfang zu ermitteln ist, da zur Untersuchung eines Parameters alle Anderen als konstant betrachtet werden.

Bei den Medizindaten besteht, wie in 3.4.3 erwähnt, das Problem, dass die CT-Aufnahmen vierdimensional sind. Der VBM3D-Algorithmus kann jedoch nur 3 Dimensionen verarbeiten. Dadurch gehen möglicherweise Informationen verloren, die zur Rekonstruktion notwendig gewesen wären.

Medizindaten werden im Rohdatenformat RAW gespeichert. Im Gegensatz zu realen Videosequenzen mit 8-Bit-Codierung sind diese mit 16 Bit codiert. Um Medizindaten trotzdem mit dem hier vorgestellten Algorithmus verarbeiten zu können, ist eine Formatierung erforderlich, durch die zusätzliche Informationen verloren gehen können.

Es besteht Möglichkeit, dass ein optischer Vergleich der einzelnen Sequenzen zu einer anderen Bewertung führt, als die PSNR-Werte.



## 4 Ergebnisse

### 4.1 Allgemeine Betrachtungen

#### 4.1.1 Standardparameter

Um eine Vergleichbarkeit der Ergebnisse zu gewährleisten, wurden die Standardparameter aus Tabelle 4.1 definiert. Sie entsprechen im Wesentlichen den Angaben aus [DFKE08b].

Die Parameter  $k$  und  $\sigma$  wurden lediglich für die echten Videosequenzen getestet, da sie einen allgemeinen Einfluss, unabhängig von spezifischen Bewegungen, auf das Ergebnis haben. Außerdem wurde in nur einer Stufe die Auflösung erhöht. Bei der Analyse der Blockgröße wurde der Parameter  $N_1$  in der zweiten Vergrößerungsstufe auf  $\frac{N_1}{2}$  verringert. Bei nicht ganzzahligen Werten für  $N_1$  in der zweiten Stufe wurde das Ergebnis abgerundet.

Parameter	Wert
Anzahl der Stufen	2
Erhöhung der Kantenlängen je Stufe	2
Anzahl der Iterationen $k$	15
Stärke des Filters $\sigma$	60 (1.Stufe) 35 (2.Stufe)
Verringerung $\Delta_\sigma$	2.5 (1.Stufe) 1.5 (2.Stufe)
Blockgröße $N_1$	12 (1.Stufe) 8 (2.Stufe)
Schrittweite $N_{schritt}$	$\frac{N_1}{2}$
Stapelgröße $N_2$	128
Schwellwert Ähnlichkeit $\tau_{match}$	4500
zeitliches Suchfenster $N_t$	9
verwendete Transformation	DCT

Tabelle 4.1: Verwendete Standardparameter

### Anzahl der Iterationen $k$

Um passende Werte aufzuzeichnen wurden die Standardparameter zusätzlich angepasst:  $\sigma = 60$        $\Delta_\sigma = 1.5$        $N_1 = 8$ .

Wie in Tabelle 4.2 zu erkennen ist, hat die Anzahl der Iterationen nur einen geringen Einfluss auf das Endergebnis. Mit jeder zusätzlichen Iteration steigt auch die Berechnungszeit der jeweiligen Stufe. Da der Umfang jeder Iteration, bei der Verwendung der DCT als Transformation, innerhalb einer Stufe nahezu identisch ist, kann von einer direkten Proportionalität ausgegangen werden. Aus diesem Grund ist es sinnvoll einen Wert von  $k = 15$  zu verwenden, da er ein gutes Verhältnis von Berechnungsdauer und Ergebnis bietet.



k	PSNR foreman	PSNR Race Horses
1	31,742	29,053
2	31,738	29,057
3	31,744	29,059
4	31,745	29,060
5	31,746	29,062
10	31,747	29,074
15	31,744	29,087
20	31,750	29,102
25	31,760	29,126
30	<b>31,788</b>	<b>29,151</b>

Tabelle 4.2: Anzahl der Iterationen k

### Stärke des Filters $\sigma$

Um den Einfluss dieses Parameters zu untersuchen wurde die Anzahl der Iterationen auf 15 festgelegt. Die Ergebnisse sind in Tabelle 4.3 dargestellt. Es zeigen sich leichte Schwankungen der PSNR-Werte, jedoch kein massiver Einfluss auf das Endergebnis. Das beste Ergebnis wurde für  $\sigma = 25$  erzielt. Auffällig ist auch ein zweites Maximum bei  $\sigma = 65$ . Da in [DFKE08b] angeraten wurde,  $\sigma$  kontinuierlich zu verringern, empfiehlt es sich, die vorgeschlagene Standardkonfiguration von  $\sigma = 60$  in der ersten Stufe und  $\sigma = 35$  für die zweite Stufe zu übernehmen. Eine Orientierung an den Standardwerten für  $\Delta_\sigma$  ist ebenfalls sinnvoll.

Ein Augenmerk bei der Wahl von  $\Delta_\sigma$  sollte aber auf der Anzahl der Iterationen liegen. Bei wenigen Iterationen sollte der Faktor größer gewählt werden, bei einer hohen Anzahl eher gering. Wird der Wert für  $\Delta_\sigma$  zu gering gewählt, besteht die Gefahr, dass Unschärfe auftritt. [DFKE08b]

$\sigma$	$\Delta_\sigma$	PSNR foreman	PSNR Race Horses
15	0,5	33,033	27,659
25	1,5	<b>33,041</b>	<b>27,662</b>
35	1,5	33,020	27,657
45	1,5	33,008	27,653
55	2,5	33,013	27,654
65	3,5	33,018	27,656
75	3,5	33,007	27,652
85	3,5	33,000	27,650
95	4,5	33,003	27,651

Tabelle 4.3: Stärke des Filters

### Vergleich von DCT und Block-DCT

Wie in Tabelle 4.4 zu erkennen ist, sind die PSNR-Werte für die Block-DCT etwas höher als für die DCT. Dieser Gewinn geht allerdings deutlich auf Kosten der Berechnungszeit. Auffällig ist, dass speziell in den Randbereichen die Rekonstruktion wesentlich schlechtere Ergebnisse liefert, als im zentralen Bereich der Bilder. Die Unterschiede der PSNR-Werte unter Vernachlässigung der rRandbereiche der Block-DCT zu den Angaben aus [DFKE08b] sind den Unterschieden der hier definierten Standardparameter zu den Angaben aus dem Artikel geschuldet.

Sequenz	$\text{PSNR}_{DCT}$	$\text{PSNR}_{Block-DCT}$
foreman (inkl. Randbereich)	29,414	29,522
foreman (exkl. Randbereich)	31,780	32,490

Tabelle 4.4: Vergleich DCT , Block-DCT (PSNR in dB)

## 4.2 synthetisch erzeugte Sequenzen

### 4.2.1 Translation

#### **Blockgröße** $N_1$

Bei translatorischen Bewegungen wurde das beste Ergebnis bei einer Blockgröße von 5x5 Pixeln erreicht, wie in Tabelle 4.5 abzulesen ist. Eine höhere Blockgröße führt zu Verlusten bei den PSNR-Werten. Auch kleinere Blockgrößen liefern schlechtere Ergebnisse. Das kann daran liegen, dass in der 2. Vergrößerungsstufe die Blockgröße sehr klein wird ( z.B: 2x2 Pixel ) und deshalb weniger Blöcke das Kriterium der Ähnlichkeit erfüllen. Wird die Blockgröße zu hoch gewählt, so lassen sich die Sub-Pixel-Verschiebungen schwerer erfassen.

#### **Stapelgröße** $N_2$

Wie in Tabelle 4.5 ersichtlich, hat die Stapelgröße nahezu keinen Effekt auf das Endergebnis. Lediglich für sehr kleine Werte (z.B:  $N_2 = 10$ ) ist der PSNR-Wert geringer. Dieses Ergebnis lässt sich damit erklären, dass im Stapel nie mehr als 20 ähnliche Blöcke gespeichert werden. Die Größe des Stapels hatte daher in den durchgeführten Untersuchungen auch kaum Einfluss auf die Geschwindigkeit der Berechnungen. Wird das Suchgebiet vergrößert ( $N_s$  bzw.  $N_t$  erhöht) oder wird der Schwellwert-Parameter verringert, wird sich auch die Anzahl der Blöcke im Stapel erhöhen. Aus diesem Grund kann die Blockgröße weiterhin hoch gewählt werden, wenn genügend Arbeitsspeicher zur Verfügung steht.

#### **Schwellwert** $\tau_{match}$

Eine Veränderung des Ähnlichkeitskriteriums  $\tau_{match}$  hat ebenfalls nur geringe Auswirkungen auf das Endergebnis. Für diesen Versuch wurde die Stapelgröße auf  $N_2 = 128$

gemäß der Standardkonfiguration aus [DFKE08b] festgelegt. Die Werte in Tabelle 4.5 zeigen den höchsten PSNR-Wert bei  $\tau_{match} = 1000$ .

$N_1$	$PSNR_{N_1}$	$N_2$	$PSNR_{N_2}$	$\tau_{match}$	$PSNR_{\tau_{match}}$
4	32,224	10	32,671	1000	<b>32,753</b>
5	<b>33,033</b>	20	<b>32,734</b>	2000	32,720
6	32,924	30	<b>32,734</b>	3000	32,731
8	32,850	40	<b>32,734</b>	3500	32,735
10	32,775	50	<b>32,734</b>	4000	32,741
12	32,742	60	<b>32,734</b>	4500	32,744
14	32,736	100	<b>32,734</b>	5000	32,742
22	32,625	150	<b>32,734</b>	6000	32,741

Tabelle 4.5: Translation: Blockgröße  $N_1$ , Stapelgröße  $N_2$  und Schwellwert  $\tau_{match}$  (PSNR in dB)

## 4.2.2 Rotation

### Blockgröße $N_1$

Auch bei der Untersuchung von Rotationsbewegungen entsteht der gleiche Effekt wie bei der Translatorischen Bewegung: Für die Blockgröße 5x5 wird das beste Ergebnis erzielt. Für steigende Werte von  $N_1$  ist auch hier eine Verringerung des PSNR-Wertes erkennbar, auch wenn diese geringer ausfällt als erwartet.

### Stapelgröße $N_2$

Die Veränderung der Stapelgröße hatte auch hier keinen messbaren Einfluss auf das Endergebnis. Die Differenz zwischen den Werten für  $N_2 = 10$  und  $N_2 = 20$  ist allerdings fast doppelt so hoch wie bei den translatorischen Bewegungen.

**Schwellwert**  $\tau_{match}$ 

Wie schon bei den translatorischen Bewegungen hat der Schwellwert  $\tau_{match}$  auch bei der Rotation kaum einen Einfluss auf das Ergebnis. Wie in den Ergebnissen aus 4.2.1 liegt auch hier das Maximum bei einem sehr kleinen Wert von ( $\tau_{match} = 1000$ ).

$N_1$	$PSNR_{N_1}$	$N_2$	$PSNR_{N_2}$	$\tau_{match}$	$PSNR_{\tau_{match}}$
4	31,529	10	31,331	1000	<b>31,452</b>
5	<b>31,725</b>	20	<b>31,448</b>	2000	31,450
6	31,690	30	<b>31,448</b>	3000	31,448
8	31,592	40	<b>31,448</b>	3500	31,448
10	31,536	50	<b>31,448</b>	4000	31,447
12	31,496	60	<b>31,448</b>	4500	31,447
14	31,461	100	<b>31,448</b>	5000	31,446
22	31,397	150	<b>31,448</b>	6000	31,445

Tabelle 4.6: Rotation: Blockgröße  $N_1$ , Stapelgröße  $N_2$  und Schwellwert  $\tau_{match}$  (PSNR in dB)

### 4.2.3 Vergleich mit anderen Algorithmen

Wie in Tabelle 4.7 ersichtlich, schneidet der vorgestellte Ansatz im Vergleich zu anderen Algorithmen bei den synthetisch erzeugten Sequenzen deutlich besser ab und bestätigt auch die Annahmen aus [DFKE08b]. Der hier vorgestellte Algorithmus kann translatorische Bewegungen besser rekonstruieren als Rotationen. Diese Annahme ist zulässig, da für beide Sequenzen das gleiche Bild als Grundlage verwendet wurde und auch Trübung sowie Verkleinerung identisch waren.

Sequenz	<b>nearest neighbour</b>	<b>bilinear</b>	<b>vorgestellt</b>
Translation	29,643	30,421	<b>33,033</b>
Rotation	28,921	29,582	<b>31,725</b>

Tabelle 4.7: synthetische Sequenzen: Vergleich mit anderen Algorithmen (PSNR in dB)

## 4.3 Echte Videosequenzen

### 4.3.1 Getestete Parameter

#### Blockgröße $N_1$

Bei den realen Videosequenzen liefert eine Blockgröße von 8x8 Pixeln bei beiden getesteten Dateien das beste Resultat (vgl. Tabelle 4.8). Allerdings sind auch hier, ähnlich wie bei den synthetisch erzeugten Sequenzen, die Unterschiede eher gering. Mit zunehmender Blockgröße sinkt der PSNR-Wert geringfügig. Speziell bei der Sequenz "Horse Race" sind die PSNR-Werte niedrig.

$N_1$	<b>PSNR foreman</b>	<b>PSNR Race Horses</b>
4	31,889	28,784
5	31,886	29,195
6	31,893	29,116
8	<b>31,968</b>	<b>29,253</b>
10	31,884	29,229
12	31,826	29,203
14	31,840	29,173
22	31,796	29,107

Tabelle 4.8: reale Sequenzen: Blockgröße  $N_1$  (PSNR in dB)

**Stapelgröße  $N_2$** 

Wie in Tabelle 4.9 zu erkennen ist, hat auch bei den realen Videosequenzen die Stapelgröße  $N_2$  kaum einen Einfluss auf das Endergebnis. Lediglich für einen Wert  $N_2 < 20$  ist das Ergebnis abweichend. Dieser Umstand lässt sich wie bei den synthetischen Sequenzen mit einer maximalen Anzahl von 20 ähnlichen Blöcken erklären.

$N_2$	PSNR foreman	PSNR Horses
10	31,763	29,159
20	<b>31,803</b>	<b>29,162</b>
30	<b>31,803</b>	<b>29,162</b>
40	<b>31,803</b>	<b>29,162</b>
50	<b>31,803</b>	<b>29,162</b>
60	<b>31,803</b>	<b>29,162</b>
100	<b>31,803</b>	<b>29,162</b>
150	<b>31,803</b>	<b>29,162</b>

Tabelle 4.9: reale Sequenzen: Stapelgröße  $N_2$  (PSNR in dB)**4.3.2 Vergleich mit anderen Algorithmen**

Im Vergleich zu anderen Algorithmen (Tabelle 4.10) wird auch bei den realen Videosequenzen mit dem vorgestellten Ansatz ein besserer PSNR-Wert erreicht.

Die Differenz ist im Vergleich zum "k-nearest neighbour"-Ansatz im Video "foreman" mit einem Wert von 2,834 dB größer als bei der Sequenz "Race Horse" (1,702 dB).

Sequenz	nearest neighbour	bilinear	vorgestellt
Horse Race	27,551	27,712	29,253
foreman	29,134	29,916	31,968

Tabelle 4.10: reale Sequenzen: Vergleich mit anderen Algorithmen (PSNR in dB)

## 4.4 Medizindaten

### 4.4.1 Zeitlicher Verlauf

#### **Blockgröße** $N_1$

Die Ergebnisse, die in Tabelle 4.11 dargestellt sind, lassen keine eindeutige Tendenz erkennen, ob ein linearer Zusammenhang zwischen Blockgröße und Endergebnis besteht. Bei kleineren Werten bis  $N_1 = 12$  schwanken die Resultate. Erst für Werte von  $N_1 \geq 12$  steigt der PSNR-Wert an. Der höchste Wert wird bei der maximalen Blockgröße erreicht. Da bei allen Werten für  $N_1$  hohe PSNR-Werte erreicht werden, fällt der Unterschied kaum ins Gewicht. Auch der optische Vergleich liefert hier bei allen Messungen gute Resultate.

#### **Stapelgröße** $N_2$

Bei der Stapelgröße ist im Vergleich zu den bisherigen Ergebnissen ein identischer Verlauf zu beobachten. Lediglich für  $N_2 = 10$  weicht der Wert ab. Da auch verschiedene Werte für den Schwellwert  $\tau_{match}$  getestet wurden, lässt sich für  $N_2$  nur unter Verwendung eines ausgedehnten Suchbereiches ( $N_s$  und  $N_t$ ) ein anderes Ergebnis erzielen.

#### **Schwellwert** $\tau_{match}$

Eine Veränderung des Schwellwertes für die Ähnlichkeit von Blöcken bewirkt keine effektive Veränderung des Endergebnisses. Die leichten Abweichungen in der dritten Nachkommastelle haben keinen Einfluss auf die optische Qualität.



$N_1$	$PSNR_{N_1}$	$N_2$	$PSNR_{N_2}$	$\tau_{match}$	$PSNR_{\tau_{match}}$
4	42,766	10	48,355	1000	48,846
5	48,591	20	<b>48,847</b>	2000	48,847
6	47,586	30	<b>48,847</b>	3000	48,847
8	48,314	40	<b>48,847</b>	3500	<b>48,848</b>
10	48,295	50	<b>48,847</b>	4000	<b>48,848</b>
12	48,575	60	<b>48,847</b>	4500	48,847
14	48,720	100	<b>48,847</b>	5000	<b>48,848</b>
22	<b>48,895</b>	150	48,847	6000	<b>48,848</b>

Tabelle 4.11: Herz-CT, zeitlicher Verlauf: Blockgröße  $N_1$ , Stapelgröße  $N_2$  und Schwellwert  $\tau_{match}$  (PSNR in dB)

## 4.4.2 Örtlicher Verlauf

### Blockgröße $N_1$

Genau wie beim zeitlichen Verlauf liefert auch hier der höchste Wert für  $N_1$  das beste Ergebnis. Auffällig ist das Ergebnis für  $N_1 = 4$  (Tabelle 4.12). Beim zeitlichen Verlauf war der Unterschied zu den anderen Ergebnissen wesentlich deutlicher als hier.

### Stapelgröße $N_2$

Der geringe Einfluss der Stapelgröße ist in Tabelle 4.12 klar zu erkennen. Wie bei den anderen Messungen zuvor existiert auch hier lediglich für  $N_2 = 10$  ein abweichendes Ergebnis. Auch optisch sind keine Unterschiede zwischen den Ergebnissen erkennbar.

$N_1$	$PSNR_{N_1}$	$N_2$	$PSNR_{N_2}$
4	46,441	10	48,854
5	48,832	20	<b>49,152</b>
6	48,073	30	<b>49,152</b>
8	48,783	40	<b>49,152</b>
10	48,827	50	<b>49,152</b>
12	48,966	60	<b>49,152</b>
14	49,077	100	<b>49,152</b>
22	<b>49,203</b>	150	<b>49,152</b>

Tabelle 4.12: Herz-CT, örtlicher Verlauf: Blockgröße  $N_1$  und Stapelgröße  $N_2$  (PSNR in dB)

#### 4.4.3 Vergleich mit anderen Algorithmen

Der Vergleich mit anderen Algorithmen fällt für die Medizindaten am deutlichsten aus. Eine mittlere Differenz von 10,685 dB gegenüber nearest-neighbour und 5,385 dB im Vergleich zu bilinearer Interpolation sind wesentlich höher, als bei den anderen getesteten Sequenzen (vgl. Tabelle 4.13). Auch optisch ist das Ergebnis durchaus zufriedenstellend. Die Rekonstruktionen mit Hilfe des vorgestellten Ansatzes konnten Details wesentlich besser wiederherstellen. Die rekonstruierten Aufnahmen weisen allerdings eine leichte Unschärfe auf.

Sequenz	nearest neighbour	bilinear	vorgestellt
zeitlicher Verlauf	37,854	43,355	48,895
örtlicher Verlauf	38,874	43,973	49,203

Tabelle 4.13: Medizindaten: Vergleich mit anderen Algorithmen (PSNR in dB)

## 4.5 Auswertung und Qualität der Ergebnisse

Die zentrale Erkenntnis aus den vorgestellten Messwerten ist, dass alle hier untersuchten Parameter nur einen geringen Einfluss auf die Qualität der Ergebnisse haben. Verbesserungen von 0,1 dB sind visuell kaum zu erfassen. Die starke Korrelation der einzelnen Parameter erschwert eine separate Untersuchung zusätzlich.

Bei der Anzahl an Iterationen muss eine Balance zwischen Ergebnis und Berechnungszeit gefunden werden. Mit zunehmender Anzahl an Iterationen steigt auch der PSNR-Wert. Allerdings erhöht sich auch proportional dazu die Berechnungszeit.

Die Wahl der Filterstärke und des Faktors  $\Delta_\sigma$  sollte an die Gesamtzahl der Vergrößerungsstufen und die Anzahl der Iterationen angepasst werden. Bei zweistufiger Auflösungserhöhung sind Werte von  $\sigma = 55$  und  $\sigma = 25$  sinnvoll. Der Verringerungsfaktor  $\Delta_\sigma$  sollte sich an der Anzahl der Iterationen orientieren.

Die Auswahl der Transformation gestaltet sich schwierig: die Block-DCT liefert bessere PSNR-Werte, allerdings auf Kosten der Berechnungszeit. Steht im verwendeten System weniger Arbeitsspeicher zur Verfügung, ist die Wahl der DCT vernünftig, weil sonst erhebliche Berechnungszeiten in Kauf genommen werden müssen.

Während bei den synthetisch erzeugten und den realen Videosequenzen eine kleinere Blockgröße zu besseren Ergebnissen führte, ist bei den Medizindaten ein gegensätzliches Verhalten erkennbar. Aus diesem Grund ist für reale Sequenzen eine Blockgröße von  $N_1 = 5$  zu empfehlen. Für Medizindaten sollte  $N_1 = 22$  die besten Ergebnisse liefern. Um eine verlässliche Aussage für Medizindaten treffen zu können, sollten weitere Tests mit anderen Sequenzen durchgeführt werden, um die Annahme zu bestätigen.

Die Stapelgröße hat, abgesehen von kleinen Werten  $N_2 < 20$ , für alle getesteten Sequenzen keinen Einfluss auf das Endergebnis. Da die Wahl dieser Größe keinen Einfluss auf die Berechnungsdauer hat, sollte trotzdem ein hoher Wert für  $N_2$  gewählt werden, sofern genügend Speicherkapazität zur Verfügung steht.

Auch eine Variation des Schwellwertes  $\tau_{match}$  erbrachte keine deutlichen Unterschiede bei den PSNR-Werten. Deshalb kann für  $\tau_{match}$  der Standardwert  $\tau_{match} = 4500$  beibe-

halten werden.

Auffällig ist bei allen Sequenzen, dass speziell der Randbereich schlecht rekonstruiert werden kann. Das liegt an der Tatsache, dass das örtliche Suchfenster  $N_s$  wesentlich kleiner wird und so auch weniger ähnliche Blöcke ermittelt werden können. Berechnet man die PSNR-Werte nur für den zentralen Bereich, also bezieht die Randbereiche nicht in die Berechnung ein, so erhöht sich der PSNR-Wert deutlich. Dabei bleiben die Tendenzen aus den angegebenen Tabellen erhalten.

Der optische Vergleich mit den Originalsequenzen weist für alle Testsequenzen Unschärfe auf. Um diesen Effekt zu verringern, könnte gegebenenfalls noch ein Deblurring durchgeführt werden.

# 5 Zusammenfassung und Ausblick

## 5.1 Zusammenfassung

In dieser Arbeit wurde der SR-Ansatz aus [DFKE08b] umgesetzt. Des Weiteren wurde der Einfluss bestimmter Parameter wie Blockgröße , Stapelgröße , Stärke des Filters und Anzahl der Iterationen auf das Endergebnis untersucht. Um den Einfluss dieser Größen besser beurteilen zu können, wurden die Versuche an verschiedenen Daten getestet.

Bei allen Versuchen konnten durch die Variation der Parameter nur geringe Verbesserungen des Endergebnisses erreicht werden. Dennoch wurden für alle Videoarten, im Vergleich zu anderen Algorithmen, bessere PSNR-Werte erzielt. Speziell bei den Medizindaten wurden gute Ergebnisse erreicht. Da die untersuchten CT-Daten ursprünglich vier-dimensional waren und zusätzlich in ein VBM3D-kompatibles Format umgewandelt werden mussten, sind weitere Tests mit anderen Datensätzen nötig, um ein endgültiges Urteil über die Qualität der Ergebnisse treffen zu können.

## 5.2 Ausblick

Um noch detailliertere Aussagen über eine optimale Konfiguration der Parameter für bestimmte Dateitypen treffen zu können, ist der Einfluss weiterer Parameter zu untersuchen. Zusätzliche Tests zur genaueren Bestimmung der Korrelation der einzelnen Parameter sollten dabei helfen, noch bessere Ergebnisse zu erzielen. Da für diese Arbeit ausschließlich der VBM3D als Filter im SR-Ansatz verwendet wurde, könnten zusätzli-

che Versuche auch mit dem CVBM3D durchgeführt werden. Speziell für die Medizindaten wäre die Implementierung eines auf dem BM4D basierenden SR-Ansatzes relevant, da dieser Volumendaten ohne Beschränkung auf einen bestimmten Verlauf (örtlich oder zeitlich) verarbeiten kann.

# Abbildungsverzeichnis

2.1	Beispiele für die Blocksuche [DFKEc]	8
3.1	Blockschaltbild des SR-Ansatzes [DFKE08b]	12
3.2	Vernschaulichung der Vergrößerung des Spektrums für DCT (links) und Block - DCT (rechts) [DFKE08b]	13





# Tabellenverzeichnis

4.1	Verwendete Standardparameter . . . . .	22
4.2	Anzahl der Iterationen $k$ . . . . .	23
4.3	Stärke des Filters . . . . .	24
4.4	Vergleich DCT , Block-DCT (PSNR in dB) . . . . .	24
4.5	Translation: Blockgröße $N_1$ , Stapelgröße $N_2$ und Schwellwert $\tau_{match}$ (PSNR in dB) . . . . .	26
4.6	Rotation: Blockgröße $N_1$ , Stapelgröße $N_2$ und Schwellwert $\tau_{match}$ (PSNR in dB) . . . . .	27
4.7	synthetische Sequenzen: Vergleich mit anderen Algorithmen (PSNR in dB) . . . . .	28
4.8	reale Sequenzen: Blockgröße $N_1$ (PSNR in dB) . . . . .	28
4.9	reale Sequenzen: Stapelgröße $N_2$ (PSNR in dB) . . . . .	29
4.10	reale Sequenzen: Vergleich mit anderen Algorithmen (PSNR in dB) . . . . .	29
4.11	Herz-CT, zeitlicher Verlauf: Blockgröße $N_1$ , Stapelgröße $N_2$ und Schwellwert $\tau_{match}$ (PSNR in dB) . . . . .	31
4.12	Herz-CT,örtlicher Verlauf: Blockgröße $N_1$ und Stapelgröße $N_2$ (PSNR in dB) . . . . .	32
4.13	Medizindaten: Vergleich mit anderen Algorithmen (PSNR in dB) . . . . .	32



## Literaturverzeichnis

- [AAVK08] A.DANIELYAN ; A.FOI ; V.KATKOVNIK ; K.EGIAZARIAN: Image Upsampling via spatially adaptive Block-Matching Filtering. In: *Proc. IEEE Int. Conf. Digital Signal Process. (DSP)* (2008), July, S. 1–5
- [DFE07] DABOV, K. ; FOI, A. ; EGIAZARIAN, K.: ideo denoising by sparse 3D transform-domain collaborative filtering. In: *European Signal Processing Conference (EUSIPCO)* (2007), September
- [DFKEa] DABOV, K. ; FOI, A. ; KATKOVNIK, V. ; EGIAZARIAN, K.: Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space.
- [DFKEb] DABOV, K. ; FOI, A. ; KATKOVNIK, V. ; EGIAZARIAN, K.: Image denoising with block-matching and 3D filtering.
- [DFKEc] DABOV, K. ; FOI, A. ; KATKOVNIK, V. ; EGIAZARIAN, K.: Image denoising with block-matching and 3D filtering.
- [DFKE07] DABOV, K. ; FOI, A. ; KATKOVNIK, V. ; EGIAZARIAN, K.: Joint image sharpening and denoising by 3D transform-domain collaborative filtering. In: *Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process* (2007)
- [DFKE08a] DABOV, K. ; FOI, A. ; KATKOVNIK, V. ; EGIAZARIAN, K.: Image restoration by sparse 3D transform-domain collaborative filtering. In: *Proc SPIE Electronic Imaging* (2008), Januar

- [DFKE08b] DANIELYAN, A. ; FOI, A. ; KATKOVNIK, V. ; EGIAZARIAN, K.: Image and video super-resolution via spatially adaptive blockmatching filtering. In: *Proceedings of International workshop on local and non-local approximation image processing (LNLA)* (2008)
- [FREM04] FARSIU, S. ; ROBINSON, D. ; ELAD, M. ; MILANFAR, P.: Advances and Challenges in Super-Resolution. (2004)
- [HD92] H.UR ; D.GROSS: Improved resolution from sub-pixel shifted pictures. (1992)
- [KAVK] K.DABOV ; A.FOI ; V.KATKOVNIK ; K.EGIAZARIAN: Image Denoising by sparse 3D transform-domain collaborative filtering. In: *IEEE Trans. on Image Processing Vol 16.* , August
- [Leb12] LEBRUN, M.: An Analysis and Implementation of the BM3D Image Denoising Method. In: *Image Processing On Line* (2012), S. 175–213
- [MKEF13] MAGGIONI, M. ; KATKOVNIK, V. ; EGIAZARIAN, K. ; FOI, A.: A Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction. (2013), Januar
- [Mö10] MÖLLER, Prof. Dr.-Ing. S.: *Qualität von Video-Übertragungssystemen.* Springer-Verlag Berlin Heidelberg, 2010. – 97 – 121 S. – ISBN 978–3–642–11547–9
- [TH84] TSAI, R.Y. ; HUANG, T.S.: Multipleframe image restoration und registration. In: *Advances in Computer Vision and Image Processing* (1984)
- [TJE02] T.FREEMAN ; JONES, T.R. ; E.C.PASZTOR: Example Based Super-Resolution. In: *IEEE Computer Graphics and Application* (2002)