

Friedrich-Alexander-Universität Erlangen-Nürnberg

**Lehrstuhl für Multimediakommunikation und
Signalverarbeitung**

Prof. Dr.-Ing. André Kaup

Forschungspraktikum

**Erweiterung eines
Echtzeit-Extrapolations-Demonstrators**

von Michael Bergmann

Juni 2017

Betreuer: Dr.-Ing. Jürgen Seiler

Inhaltsverzeichnis

Kurzfassung	III
1 Einleitung	1
2 Grundlagen	3
3 Optimierungsverfahren	5
3.1 Umstrukturierte Iteration	5
3.2 Hybrid-Verfahren	7
4 Ergebnisse	9
4.1 Testumgebung	9
4.2 Laufzeit	10
4.3 Bildqualität	11
5 Schluss	13
A Anhang Kapitel	14
Abbildungsverzeichnis	17
Tabellenverzeichnis	17
Literaturverzeichnis	19

Kurzfassung

Für viele moderne Anwendungen wird hochaufgelöstes Bildmaterial benötigt. Da die Bildquellen häufig nur in niedrigen Auflösungen vorhanden sind, kommen Methoden zur Auflösungserhöhung zum Einsatz. Zur Vorführung einer solchen Methode, der Frequenzselektiven Rekonstruktion, steht am Lehrstuhl für Multimediakommunikation und Signalverarbeitung ein Echtzeit-Demonstrator zur Verfügung. Um die Bildrate dieses Demonstrators zu verbessern wurden in dieser Arbeit zwei Optimierungsansätze implementiert und evaluiert. Die Implementierung der iterativen Modellerstellung in der FSR wurde umstrukturiert, sowie ein bestehendes Hybridverfahren implementiert. Die Umstrukturierung des Programmcodes konnte hierbei keine Verbesserung gegenüber der bestehenden Implementierung erzielen. Mit dem eingeführten Hybridverfahren konnte eine Beschleunigung des Demonstrators um einen Faktor von 2,9 erzielt werden, allerdings sind die PSNR-Verluste von über 1dB deutlich höher als in vergleichbaren Untersuchungen.

Kapitel 1

Einleitung

Mit der Verbreitung hochauflösender Anzeigeräte steigt die Nachfrage nach ebenso hoch aufgelösten Bild- und Videodaten. Mache Daten liegen nur in niedriger Auflösung vor. Gründe hierfür sind beispielsweise die Kosten für hochauflösende Sensoren oder begrenzter Speicherplatz zur Zeit der Aufnahme. Um die Auflösung solcher Bilder zu verbessern stehen verschiedene Verfahren zur Verfügung.

In [1] wird ein solches Verfahren vorgestellt, das mithilfe einer nicht-regelmäßigen Maskierung und anschließenden Rekonstruktion des Bildes eine verbesserte Auflösung erzielt. Hierzu wird die Frequenzselektive Extrapolation eingesetzt. Am Lehrstuhl für Multimediakommunikation und Signalverarbeitung steht ein Demonstrator zur Verfügung, der dieses Verfahren in Echtzeit durchführt. Wegen des hohen Berechnungsaufwandes für die FSE können trotz leistungsfähiger Hardware nur wenige Bilder pro Sekunde erzielt werden.

In diesem Praktikum sollten verschiedene Ansätze zur Optimierung des Verfahrens und überprüft. Dabei wurden zwei Grundlegende Ansätze überprüft. Zunächst sollte der Programmcode umstrukturiert werden, um etwaige Ineffizienzen zu beseitigen. Des Weiteren wurde ein Hybrid-Verfahren wie in [2] vorgestellt implementiert. Hierbei wird das Bild in den YUV-Farbraum transformiert. Der Luminanzkanal enthält nun den größten Teil der Bildinformation und wird mit der FSE rekonstruiert, während auf die beiden Chrominanzkanäle eine schnellere lineare Extrapolation angewendet.

Dieser Bericht ist folgendermaßen strukturiert. In Kapitel 2 werden kurz die Grundlagen der Frequenzselektiven Extrapolation erläutert. Kapitel 3 stellt die implementierten Optimierungsverfahren vor. Anschließend werden in Kapitel 4 die Auswirkungen auf Laufzeit und Bildqualität dargestellt. Kapitel 5 schließt den Bericht ab.

Kapitel 2

Grundlagen

Zur Auflösungserhöhung wird zunächst das niedrigaufgelöste Bild mit einer hochaufgelösten unregelmäßigen Maske verdeckt. Jedes Pixel wird hierzu in 4 Subpixel unterteilt und zufällig 3 der 4 Subpixel verdeckt. Anschließend wird mithilfe der FSE der verdeckte Bildbereich rekonstruiert.

Zur Rekonstruktion der von der Bildmaske verdeckten Gebiete wird im Demonstrator die FSE mit komplexwertigen Basisfunktionen [3] verwendet. Dabei wird das Bild in überlappende Blöcke aufgeteilt. In Abbildung 3.1 ist ein solcher Block dargestellt. Der Rand um das zu rekonstruierende Gebiet dient zur Vergrößerung des Unterstützungsbereichs und somit zur Verbesserung der Qualität.

Für die Rekonstruktion des Blocks wird nun aus Fourierbasisfunktionen

$$\phi_{(k,l)}[m, n] = e^{\frac{2\pi j}{M} km} e^{\frac{2\pi j}{N} ln} \quad (2.1)$$

iterativ ein Modell

$$g[m, n] = \sum_{(k,l) \in \mathcal{K}} c_{(k,l)} \phi_{(k,l)}[m, n] \quad (2.2)$$

erstellt. $\hat{c}_{(k,l)}$ bezeichnet die Expansionskoeffizienten und \mathcal{K} enthält die Indizes der bereits im Modell enthaltenen Basisfunktionen.

In jedem Iterationsschritt wird nun die Basisfunktion dem Modell hinzugefügt, welche die Differenz zwischen dem Modell und den bekannten Signalpunkten minimiert.

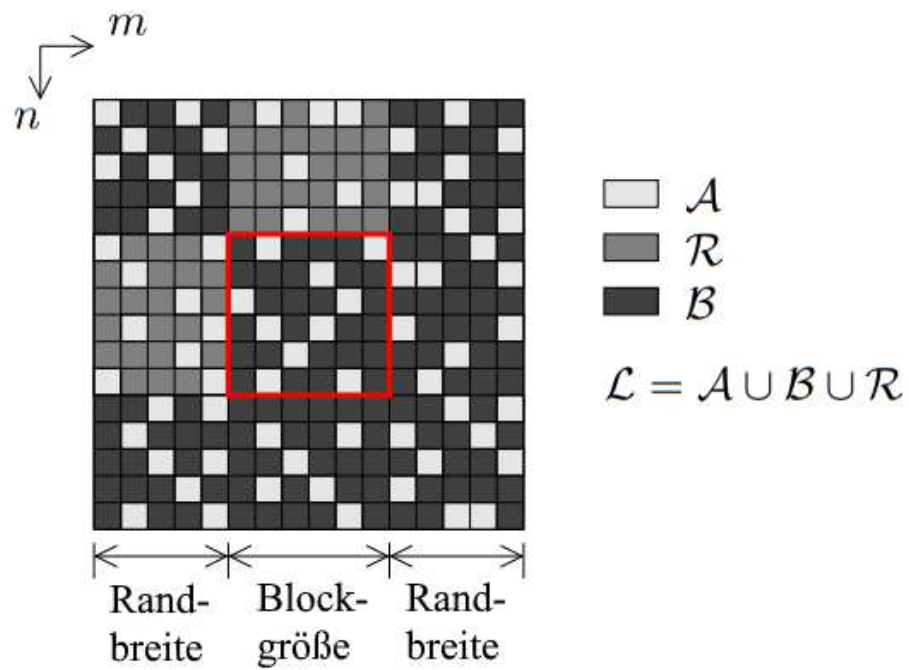


Abbildung 2.1: Ein zu rekonstruierender Block \mathcal{L} bestehend aus dem bekannten Unterstützungsg Gebiet \mathcal{A} , bereits rekonstruiertem Gebiet \mathcal{R} und unbekanntem Pixeln \mathcal{B} . Der rot umrahmte Bereich ist zu rekonstruieren. [4]

Kapitel 3

Optimierungsverfahren

Im Demonstrator sind zusätzlich zur Frequenzselektiven Rekonstruktion noch zusätzliche Aufgaben implementiert. Das zu verarbeitende Bild muss geladen, die Bildmaske erzeugt und das Bild maskiert werden. Erst danach wird die Rekonstruktion aufgerufen. Das rekonstruierte Bild muss ausgegeben, die Bildrate berechnet werden. Zudem müssen Nutzereingaben verarbeitet werden.

Um die Laufzeit zu verbessern ist es somit nötig zu analysieren welche Abschnitte im Programmcode den größten Rechenaufwand verursachen. Dabei ergab sich, dass der Großteil des Rechenaufwandes in der Iteration der FSE besteht, woraufhin die zwei folgenden Ansätze zur Optimierung untersucht wurden.

3.1 Umstrukturierte Iteration

Ein Ansatz zur Verbesserung der Bildrate war die Umstrukturierung der zentralen Iteration, um die Anzahl der in jedem Iterationsschritt zu durchlaufenden Schleifen zu reduzieren. Im ursprünglichen Algorithmus wird zunächst mit einer zweidimensionalen Schleife über das Residuum jene Fourierbasisfunktion bestimmt, die das Residuum zwischen Modell und bekannten Bildpunkten minimiert. Dann wird diese Funktion dem Modell hinzugefügt. Danach wird das Residuum in einer weiteren zweidimensionalen Schleife neu berechnet. Algorithmus 1 stellt die Struktur der alten Iteration dar.

Algorithmus 1 Alte Iteration

```
for Anzahl der Iterationen do
  for  $(x, y) \in$  Residuum do
    Finde Fourierbasisfunktion, welche Residuum minimiert
  end for
  Füge Fourierbasisfunktion dem Modell hinzu
  for  $(x, y) \in$  Residuum do
    Berechne Residuum neu
  end for
end for
```

Algorithmus 2 Umstrukturierte Iteration

```
for  $(x, y) \in$  Residuum do
  Finde Fourierbasisfunktion, welche Residuum minimiert
end for
Füge Fourierbasisfunktion dem Modell hinzu
for Anzahl der Iterationen - 1 do
  for  $(x, y) \in$  Residuum do
    Berechne Residuum neu
    Finde Fourierbasisfunktion, welche Residuum minimiert
  end for
  Füge Fourierbasisfunktion  $\phi_{(u,v)}[m, n]$  dem Modell hinzu
end for
```

Die umstrukturierte Iteration verschiebt die Neuberechnung des Residuums an den Beginn des Iterationschritts wodurch die beiden zweidimensionalen Schleifen zusammengefasst werden können. Dazu muss der erste Iterationsschritt vor der Schleife verarbeitet werden, damit die erste Fourierbasisfunktion für die Neuberechnung des Residuums bekannt ist. Zusätzlich müssen zwei Zwischenvariablen eingeführt werden, da in

der zusammengeführten Schleife gleichzeitig nach der nächsten Basisfunktion gesucht und auf Grundlage der Basisfunktion aus dem letzten Iterationsritt das Residuum neu berechnet wird. In Algorithmus 2 ist der Aufbau der umgestellten Iteration abgebildet.

3.2 Hybrid-Verfahren

Der zweite Ansatz zur Optimierung basierte auf dem in [2] vorgestellten Hybridverfahren. Zur Rekonstruktion eines Farbbildes muss die FSE dreimal durchlaufen werden, jeweils für die drei Farbkanäle. Das vorgestellte Verfahren beschleunigt den Vorgang indem nur noch ein Kanal mit der FSE extrapoliert wird.

Dazu wird ausgenutzt, dass der größte Teil des Informationsgehalts im Luminanzkanal eines Bildes enthalten ist, in den Chrominanzkanälen dagegen nur einen kleinen Anteil. Daher können die Chrominanzkanäle ohne große Verluste bei der Qualität mit einem schnelleren Verfahren extrapoliert werden, in diesem Fall einer linearen Extrapolation. Das maskierte niedrigaufgelöste Bild $\mathbf{I}_{\mathbf{S},\mathbf{RGB}}$ wird zunächst in den YUV-Farbraum transformiert. Der Luminanzkanal $\mathbf{I}_{\mathbf{S},\mathbf{Y}}$ wird dann mit der FSE, die Chrominanzkanäle $\mathbf{I}_{\mathbf{S},\mathbf{U}}$ und $\mathbf{I}_{\mathbf{S},\mathbf{V}}$ mit einem linearen Verfahren extrapoliert. Die extrapolierten Bilder $\mathbf{I}_{\mathbf{R},\mathbf{Y}}$, $\mathbf{I}_{\mathbf{R},\mathbf{U}}$ und $\mathbf{I}_{\mathbf{R},\mathbf{V}}$ werden dann für die Wiedergabe wieder in den RGB-Farbraum zurücktransformiert. Dieses Verfahren ist in Abbildung 3.1 schematisch dargestellt

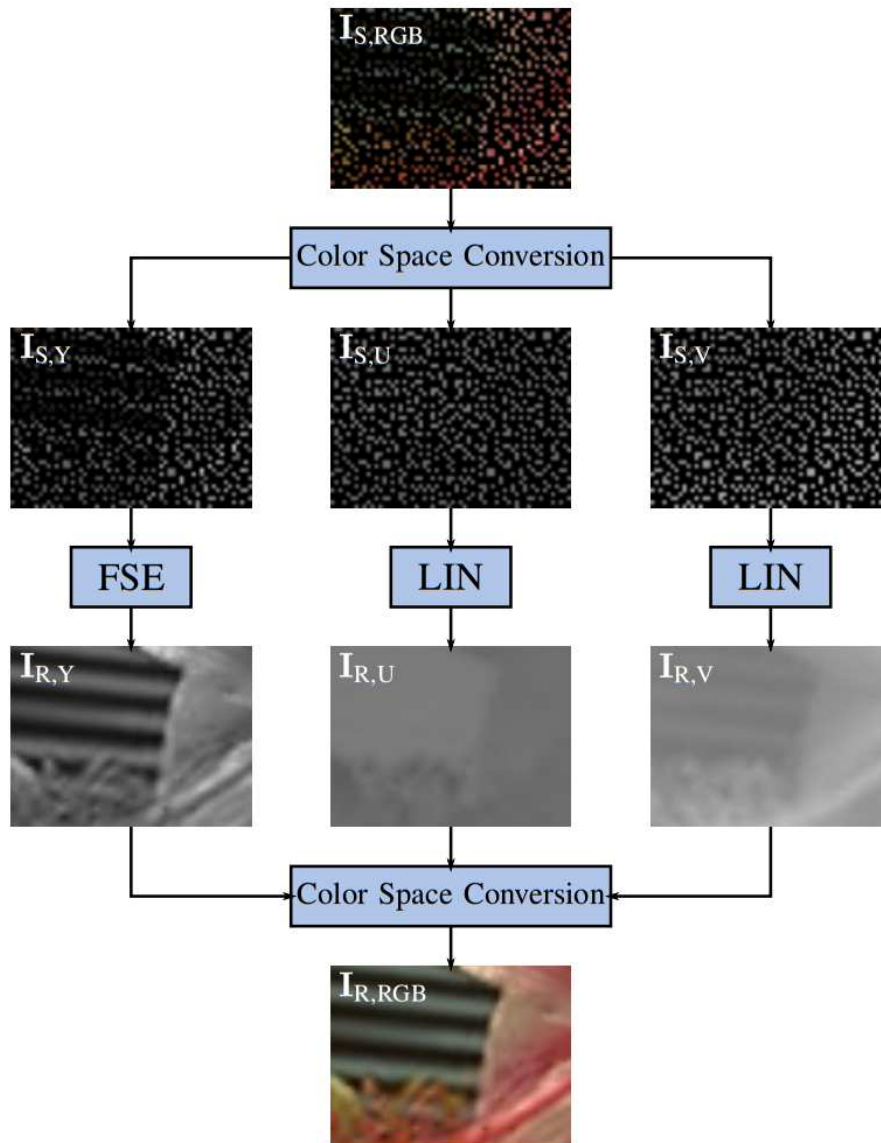


Abbildung 3.1: Hybrid-Verfahren für die FSE. [2]

Kapitel 4

Ergebnisse

Dieser Abschnitt stellt die beiden implementierten Modifikationen dem ursprünglichen Demonstrator gegenüber. Die Versionen werden hinsichtlich Laufzeit und PSNR bewertet und verglichen.

4.1 Testumgebung

Als Testsystem diente ein Intel Core i5-4690 @ 3.50GHz mit 16GB RAM.

Iterationen	100
FFT-Größe	32
Blockgröße	8
Randbreite	12
Abklingfaktor ρ	0.7
Orthogonalitätskorrekturfaktor γ	0.5
Gewichtungsfaktor für rekonstruierte Gebiete δ	0

Tabelle 4.1: Verwendete FSE-Parameter

Tabelle 4.1 zeigt die verwendeten Parameter für die FSE dar.

4.2 Laufzeit

Die Kernaufgabe war es die Bildrate des Demonstrator zu verbessern. Um die hierbei erzielten Ergebnisse zu untersuchen wurde die Zeit zur Verarbeitung und Anzeige eines Bildes gemessen und über mehrere Durchläufe gemittelt. Verwendet wurde hierzu die C-Funktion `gettimeofday()`, die auch zur Berechnung der im Demonstrator angezeigten Bildrate dient. Die erzielten Messergebnisse sind in Tabelle 4.2 dargestellt.

	Zeit pro Bild	Beschleunigungsfaktor
Ursprüngliches Verfahren	6846,59ms	1
Umstrukturierte Iteration	7224,17ms	0,95
Hybridverfahren	2342,36ms	2,92

Tabelle 4.2: Vergleich der Laufzeiten für die Verarbeitung eines Bildes und der Beschleunigung gegenüber dem ursprünglichen Demonstrator für die beiden implementierten Änderungen

Die umstrukturierte Iteration führt um eine Verlangsamung der Verarbeitung um fast 400ms. Dies entspricht einer Reduktion der Bildrate um 5%. Dieser Verlust ist auf die zusätzlichen Variablenzugriffe zurückzuführen, die nötig sind, um die hinzugefügten Fourierbasisfunktionen für die nächste Iteration zwischenspeichern.

Die Laufzeit für das implementierte Hybrid-Verfahren hingegen wurde um einen Faktor von 2,92 beschleunigt. Dieser Gewinn von knapp drei ist zu erklären, da statt drei Farbkämen nur noch ein Luminanzkanal mit der FSR rekonstruiert wird, die den Großteil der Rechenleistung beansprucht. Die Vorgänge wie das Maskieren oder Wiedergeben des rekonstruierten Bildes sowie die Farbtransformation und die lineare Extrapolation der Chrominanzkanäle fallen dagegen kaum ins Gewicht. Dieses Ergebnis ist auch ähnlich zu den in [?] erzielten, wo ein das Verfahren ebenfalls um einen Faktor von etwa 2,9 beschleunigt werden konnte.

4.3 Bildqualität

Ein zweiter wichtiger Aspekt ist der Einfluss der vorgenommenen Änderungen auf die Qualität der rekonstruierten Bilder. Zum Vergleich wurde hier das PSNR der Bilder, sowie der einzelnen Kanäle, herangezogen. Tabelle 4.3 zeigt die Ergebnisse.

	ursprüngliches Verfahren	umstrukturierte Iteration	Hybridverfahren
RGB	32,15dB	32,18dB	30,98dB
R	31,92dB	31,96dB	30,54dB
G	32,23dB	32,26dB	31,81dB
B	32,31dB	32,34dB	30,70dB

Tabelle 4.3: Vergleich der PSNR-Werte für die drei Verfahren in den drei Farbkanälen und der Kombination

Die umstrukturierte Iteration zeigt hier nur minimale Änderungen von weniger als 0,5dB im Vergleich zur ursprünglichen Implementierung. Dieses Ergebnis ist nicht verwunderlich, da an den Berechnungen, die für die Rekonstruktion durchgeführt werden, keine Änderung vorgenommen wurden, ausschließlich die Reihenfolge wurde abgewandelt.

Für das implementierte Hybrid-Verfahren zeigen sich hingegen starke Verluste, von 1,17dB im Gesamtbild und bis zu 1,6dB in den einzelnen Kanälen. Hierbei ist zu bemerken, dass der grüne Farbkanal deutlich weniger von den Verlusten betroffen ist als rot und blau. Dies ist zu erklären, da der grüne Farbkanal die stärksten Relation mit dem Luminanzkanal hat, während die anderen beiden Kanäle stark mit den Chrominanzkanälen korrelieren. Die Verluste sind auf die deutlich leistungsschwächere lineare Extrapolation der beiden Chrominanzkanäle zurückzuführen.

Dieses Ergebnis steht in Gegensatz zu den Resultaten aus [2], wo nur ein PSNR-Verlust von weniger als 0,1dB zu verzeichnen war. Dieser Unterschied kann auf viele Arten zu-

stande kommen. In dieser Arbeit wurde ein anderer Testdatensatz verwendet, als im oben genannten Paper. Zudem kann die lineare Interpolation des Bildes auf eine andere Weise implementiert sein.

Kapitel 5

Schluss

Dieser Bericht stellt die Laufzeitoptimierung des Extrapolations-Demonstrators zur Vorführung der FSE dar. Die beiden vorgestellten Verfahren wurden kurz vorgestellt, erläutert und hinsichtlich Geschwindigkeit sowie Bildqualität evaluiert.

Das erste vorgestellte Verfahren, das mit einer Umstrukturierung der iterativen Modellbildung in der FSR die Laufzeit verbessern sollte, zeigte hierbei keinen Erfolg. Die Bildrate des Demonstrators verschlechterte sich um 5% und die Bildqualität blieb unbeeinflusst. Somit kann dieses Verfahren nicht zu einer Beschleunigung des Demonstrators beitragen.

Das implementierte Hybrid-Verfahren, welches mithilfe einer Farbtransformation den Rechenaufwand minimieren sollte, konnte eine Beschleunigung um einen Faktor von 2,92 erzielen. Allerdings führte das Verfahren auch zu einem Verlust im PSNR von annähernd 1,2dB. Da in einer vorgehenden Arbeit mit diesem Hybrid-Verfahren bereits mit Verlusten unter 0,1dB vergleichbare Geschwindigkeitsgewinne erzielt werden konnten, besteht an diesem Verfahren noch Potential für Verbesserung in folgenden Arbeiten am Demonstrator.

Anhang A

Anhang Kapitel



Abbildung A.1: Originalbild



Abbildung A.2: Mit ursprünglichem Verfahren rekonstruiertes Bild



Abbildung A.3: mit Hybridverfahren rekonstruiertes Bild

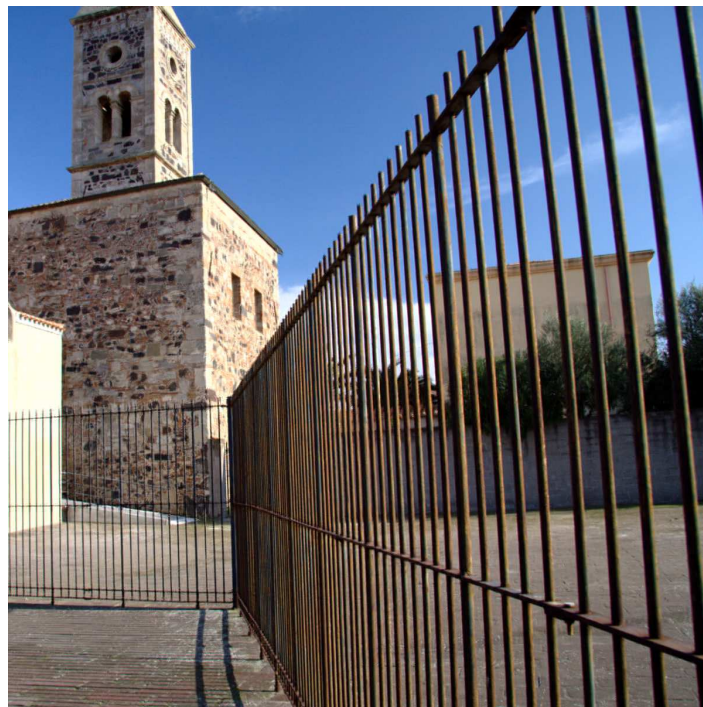


Abbildung A.4: Mit umgestellter Iteration rekonstruiertes Bild

Abbildungsverzeichnis

2.1	Ein zu rekonstruierender Block \mathcal{L} bestehend aus dem bekannten Unterstützungsgebiet \mathcal{A} , bereits rekonstruiertem Gebiet \mathcal{R} und unbekanntem Pixeln \mathcal{A} . Der rot umrahmte Bereich ist zu rekonstruieren. [4] . . .	4
3.1	Hybrid-Verfahren für die FSE. [2]	8
A.1	Originalbild	15
A.2	Mit ursprünglichem Verfahren rekonstruiertes Bild	15
A.3	mit Hybridverfahren rekonstruiertes Bild	16
A.4	Mit umgestellter Iteration rekonstruiertes Bild	16

Tabellenverzeichnis

4.1	Verwendete FSE-Parameter	9
4.2	Vergleich der Laufzeiten für die Verarbeitung eines Bildes und der Beschleunigung gegenüber dem ursprünglichen Demonstrator für die beiden implementierten Änderungen	10
4.3	Vergleich der PSNR-Werte für die drei Verfahren in den drei Farbkanälen und der Kombination	11

Literaturverzeichnis

- [1] M. Schöberl, J. Seiler, S. Foessel, and A. Kaup, “Increasing imaging resolution by covering your sensor,” in *IEEE International Conference on Image Processing (ICIP)*, pp. 1937–1940, Sep 2011.
- [2] M. Bätz, A. Eichenseer, M. Jonscher, J. Seiler, and A. Kaup, “Accelerated hybrid image reconstruction for non-regular sampling color sensors,” in *IEEE Int. Conf. on Visual Communications and Image Processing (VCIP)*, pp. 217–220, Dec 2014.
- [3] J. Seiler and A. Kaup, “Complex-valued frequency selective extrapolation for fast image and video signal extrapolation,” *IEEE Signal Processing Letters*, vol. 17, pp. 949 – 952, Nov 2010.
- [4] J. Seiler, M. Jonscher, M. Schöberl, and A. Kaup, “Resampling images to a regular grid from a non-regular subset of pixel positions using frequency selective reconstruction,” *IEEE Transactions on Image Processing (IEEE TIP)*, vol. 24, pp. 4540–4555, Nov 2015.