

Selective Signal Extrapolation and its Application in Image and Video Communications

Der Technischen Fakultät der
Friedrich-Alexander-Universität Erlangen-Nürnberg
zur Erlangung des Grades

Doktor-Ingenieur

vorgelegt von

Katrin Meisinger

Erlangen, 2007

Als Dissertation genehmigt von
der Technischen Fakultät der
Friedrich-Alexander-Universität
Erlangen-Nürnberg

Tag der Einreichung:	10. April 2007
Tag der Promotion:	10. Juli 2007
Dekan:	Prof. Dr.-Ing. A. Leipertz
Berichterstatter:	Prof. Dr.-Ing A. Kaup Prof. Dr.-Ing T. Aach

Danksagung

Im Folgenden möchte ich einigen Personen für ihre Unterstützung während meiner Promotion meinen Dank aussprechen.

An erster Stelle bedanke ich mich bei Herrn Prof. Dr.-Ing. A. Kaup für die Möglichkeit, meine wissenschaftliche Arbeit unter seiner Leitung am Lehrstuhl für Multimediakommunikation und Signalverarbeitung durchführen zu können.

Weiterhin danke ich Herrn Prof. Dr.-Ing. T. Aach für sein besonderes Interesse an meiner Arbeit, das er durch die Begleitung meiner Dissertation und die Übernahme des Zweitgutachtens zum Ausdruck gebracht hat.

Die Zeit am Lehrstuhl war für mich eine angenehme Zeit, sowohl fachlich interessant als auch sehr unterhaltsam, weswegen ich allen ehemaligen Kollegen danke.

Ohne namentlich alle einzeln aufzuführen, möchte ich meinen ehemaligen Studien- und Diplomarbeitern für ihre hervorragende Arbeit danken.

Für die Korrektur meiner Ausarbeitung sowie für die vielen interessanten Diskussionen mit ganz unterschiedlichen Sichtweisen danke ich meinem Mann Fabian und Herrn Jürgen Seiler.

Ganz besonders danke ich meiner Familie. Vor allem danke ich meinem Mann Fabian für jeden gemeinsamen Tag.

Abstract

The estimation of unknown signal samples by extrapolation from known data arises in many signal processing applications. In this thesis, a very generic concept for the estimation of unknown signal samples is derived which is referred to as *selective extrapolation*. Several applications in image and video communications are discussed from the point of view of extrapolation. Unknown areas occurring in images and videos due to erroneous transmission over unreliable channels are concealed by the proposed selective extrapolation concept. The same technique is applied to the estimation of the video signal covered by TV logos in order to remove the undesired TV logo. Prediction in hybrid video coding is also interpreted as the extrapolation of the already coded samples in order to increase coding efficiency.

The derivation of the principle of selective extrapolation relies on arbitrary basis functions which allows for a straightforward adaptation to any given application. Since the generic concept is independent of the dimensions of the considered signal space, the proposed approach can be straightforwardly extended from the two-dimensional derivation to three-dimensions. The new concept of three-dimensional extrapolation allows for an interpretation of a video signal as a three-dimensional volume. Therefore, *spatial and temporal* correlations of the video signal can be *simultaneously* used for extrapolation purposes.

The interpretation of image and video signals as a linear combination of multi-dimensional frequencies explains why DFT like basis functions are especially suited in the considered extrapolation context. The application of DFT like basis functions leads to the concept of *frequency selective extrapolation*. The main advantage of frequency selective extrapolation is the ability to inherently adapt to the local signal characteristics. Image signals with different characteristics such as smooth areas, edges, patterns as well as noise-like areas can be extrapolated. With the novel three-dimensional approach, additionally motion and variations in luminance occurring from frame to frame can be inherently compensated.

Contents

1	Introduction	1
2	Signal Extrapolation	5
2.1	Applications for Extrapolation in Image and Video Communications	5
2.2	Spatial Signal Extrapolation	7
2.2.1	The Extrapolation Task	7
2.2.2	Extrapolation by Spectral Estimation	8
2.3	Related Techniques Used in Image Coding	11
2.4	3-D Signal Processing	11
3	Spatial Selective Extrapolation	13
3.1	Spatial Selective Extrapolation for 2-D Signals	13
3.1.1	Principle of Selective Extrapolation	14
3.1.2	Update of Selected Coefficient	15
3.1.3	Selection of Basis Function	17
3.2	Geometrical Interpretation of Selective Extrapolation	18
3.3	Suitable Basis Functions	25
3.4	Frequency Selective Extrapolation of Image Signals Using DFT Basis Functions	26
3.4.1	Spatial-Domain Solution	27
3.4.2	Frequency-Domain Solution	29
3.4.3	Summary of Frequency Selective Extrapolation	31
3.5	2-D Weighting Function	36
3.5.1	Extrapolation Result for Rectangular Binary Weighting Function .	39
3.5.2	Isotropic Weighting Function	44
3.5.3	Extrapolation Result for Isotropic Weighting Function	45
3.5.4	Discussion of Results	50
3.6	Frequency Selective Extrapolation Interpreted as Spectral Deconvolution .	54
4	Spatio-Temporal Selective Extrapolation	61
4.1	Spatio-Temporal Selective Extrapolation for 3-D Signals	61

4.2	Frequency Selective Extrapolation of Video Signals Using DFT Basis Functions	64
4.2.1	Spatio-Temporal-Domain Solution	66
4.2.2	Frequency-Domain Solution	67
4.2.3	3-D Isotropic Weighting Function	68
4.2.4	Discussion of Results	76
4.3	Alternative Approach: Spatio-Temporal Extrapolation Using Best Approximation	82
4.3.1	Basic Principle of Best Approximation	82
4.3.2	Geometrical Interpretation of Extrapolation by Best Approximation	84
4.3.3	2-D Extrapolation by Best Approximation Using DFT Basis Functions	88
4.3.4	3-D Extrapolation by Best Approximation Using DFT Basis Functions	91
4.4	Frequency Selective Extrapolation vs. Best Approximation	93
4.4.1	Extrapolation Result Using 2-D Best Approximation	93
4.4.2	Extrapolation Result Using 3-D Best Approximation	96
4.4.3	Discussion of Results	97
5	Error Concealment of Lost Block Coded Data	101
5.1	Related Work in Concealment of Lost Block Coded Data	102
5.1.1	Spatial Concealment of Block Losses	102
5.1.2	Spatio-Temporal Concealment of Block Losses	103
5.2	Concealment using Frequency Selective Extrapolation	105
5.2.1	Spatial Concealment Using 2-D Frequency Selective Extrapolation	105
5.2.2	Spatio-Temporal Concealment Using 3-D Frequency Selective Extrapolation	109
5.3	Comparison to Related Techniques	116
5.3.1	Results for Spatial Concealment	116
5.3.2	Results for Spatio-Temporal Concealment	119
5.4	Integration of Frequency Selective Extrapolation into H.264/AVC Decoder	124
5.4.1	Error Robustness in H.264/AVC	124
5.4.2	Error Concealment in Reference Software	125
5.4.3	Realization of Concealment Using Frequency Selective Extrapolation	126
5.4.4	Simulations and Results	128
5.4.5	Summary	141
6	Spatial Concealment of Lost Wavelet Coded Data	143
6.1	Fundamentals of JPEG2000	144
6.2	Tools for Error Control in JPEG2000	145
6.2.1	Effects of Transmission Errors	145
6.2.2	Error Robustness on Entropy Coding Level	146

6.2.3	Error Robustness on Packet Level	146
6.2.4	Concealment by Inserting Empty Packets	146
6.3	Concealment by Frequency Selective Extrapolation	147
6.4	Simulations and Results	148
6.4.1	Concealment of Lowpass Band Losses	149
6.4.2	Concealment of Highpass Band Losses	155
6.4.3	Concealment of Lower Layer Losses	155
6.4.4	Realization of Packet Error Robust JPEG2000 Decoder	156
6.5	Summary	157
7	Further Applications	159
7.1	TV Logo Removal	159
7.1.1	Logo Removal by Frequency Selective Extrapolation	160
7.1.2	Simulations and Results	161
7.1.3	Comparison to Related Techniques	168
7.2	Prediction	168
7.2.1	Prediction by Frequency Selective Extrapolation	169
7.2.2	Comparison to Related Techniques	171
8	Conclusions and Outlook	175
A	Notations	179
A.1	Conventions	179
A.2	Abbreviations	179
A.3	Mathematical Symbols	180
B	Titel, Inhaltsverzeichnis, Einleitung und Zusammenfassung	183
B.1	Titel	183
B.2	Inhaltsverzeichnis	183
B.3	Einleitung	186
B.4	Zusammenfassung und Schlussfolgerungen	189
	Bibliography	193

1 Introduction

Many signal processing applications imply the problem of estimating unknown signal samples from known data. This task can be interpreted as the extrapolation of known signal areas towards the unknown areas. The described problem occurs e.g. in medical image processing for the defect pixel interpolation of flat panel X-ray detectors. These detectors allow the immediate availability of the acquired images for display, but still may contain regions with inactive pixels amongst others due to manufacturing problems. Hence, the provided X-ray images include unknown areas caused by these inactive pixels site. The corresponding defects in the images can be replaced by extrapolating the signal from adjacent pixels [4], because the locations of the inactive pixels are known from calibration measurements.

Modeling the human auditory system including a high-resolution spectral analysis [50] represents an example for an audio signal processing application. Because only signal blocks of short time are available, the method for performing a high-resolution short-time spectral analysis is based on an extrapolation. The influence of the time-limited observation of the audio signal is removed in the spectral domain which corresponds to an extrapolation of the audio signal beyond the observed period of time.

A similar problem arises in the completely different field of astrophysics. For aperture synthesis, the limitations imposed by the recording radio telescopes on the measurements are eliminated [22]. The recorded so-called “dirty map” is “cleaned” from the so-called “dirty beam” formed by the recording telescopes. Since the observation space of the sky covered by the recording telescopes is limited, the method also corresponds to an extrapolation of the intensity values of the recorded objects beyond the observed part of the sky.

In nuclear physics, the prediction of nuclear masses in instable regions represents another extrapolation task [17]. In Fig. 1.1, the difference between two given models for nuclear mass prediction in dependence of the number of neutrons on the x-axis and the number of protons on the y-axis is shown [17]. Based on the extrapolation of the given pattern beyond its limits, the nuclear masses in instable regions can be predicted.

The problem of estimating unknown samples also arises in various image and video communication applications. Examples which are especially important in practice are interpreted in this thesis from the point of view of signal extrapolation. For instance, the problem of concealing corrupted video data caused by transmission errors in mobile video communications can be interpreted as an extrapolation of the surrounding available video

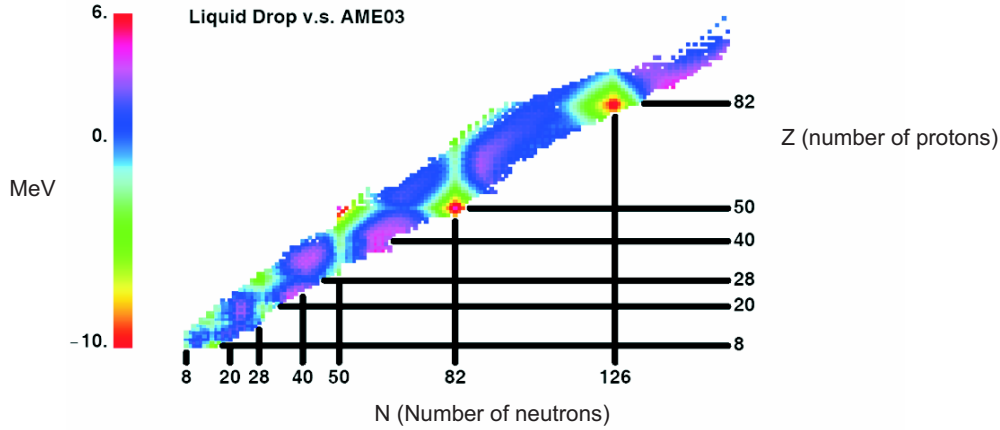


Figure 1.1: The difference between two models in nuclear mass prediction is shown according to [17]. For predicting nuclear masses in instable regions, the pattern is to be extrapolated beyond its limits.

signal into the missing area. The removal of undesired TV logos can be also considered in the extrapolation context, where the unknown pixels covered by the logo are to be estimated from the surrounding video signal. In hybrid video coding, prediction of the video signal is applied in order to increase coding efficiency. This signal processing step can also be interpreted as extrapolation of already coded pixels in order to predict following pixels.

Various methods have been proposed which solve the extrapolation task for two-dimensional signals by applying spectral estimation. Bandlimited extrapolation methods [44, 45, 53, 56] are known to cause specific artifacts in the signal domain and are therefore not suited for the extrapolation into larger areas. Spectral deconvolution methods [18, 50, 12, 4] achieve in general better results. However, they explicitly rely on Discrete Fourier Transform (DFT) basis functions and are therefore limited to the Fourier domain. Other basis functions which turn out to be useful for certain applications as, e.g., Discrete Cosine Transform (DCT) functions or polynomials can not be applied within this approach.

The general principles considered in this thesis are not only used for *extrapolation* tasks, but also for *coding* purposes. In [26], successive *approximation* relying on arbitrary basis functions is applied to object oriented image coding. Spectral estimation methods are usually not applied for the extrapolation of video signals. Conventionally, unknown samples in video signals are estimated by hybrid techniques using either spatial or temporal information as, e.g., in case of prediction in hybrid video coding. Most commonly, motion compensation is applied to explicitly exploit temporal information.

In this thesis, a generic estimation method is derived referred to as *selective extrapolation*.

tion. In contrast to previous work for extrapolation purposes [18, 50, 12, 4], this general approach relies on arbitrary basis functions which allows for an adaptation to the specific properties implied by the considered application. Based on the principle of successive approximation [26], the support area surrounding the area to be estimated is approximated by a linear combination of *selected* basis functions. The basis functions used for the approximation are defined on an area which also includes the unknown samples. Therefore, the approximation of the known area inherently provides an *extrapolation* into *unknown* areas. The error criterion minimized during the extrapolation uses a weighting function which allows for a further application specific adaptation of the general approach. For the applications considered in this thesis, a suitable weighting function is specified. The presented concept is independent of the dimensions of the signal space. Therefore, the approach of selective extrapolation as initially derived for two-dimensional (2-D) signals can be straightforwardly extended to 3-D signals. This is achieved by interpreting video signals as a 3-D volume. Hence, *spatial and temporal correlations* of the video signal can be used *at the same time* for the estimation of unknown areas. This interpretation of video signals leads to new insights compared to conventional hybrid approaches which apply either spatial or temporal prediction.

The work presented in this thesis is organized as follows. First, a review of spectral estimation methods used for signal extrapolation and related techniques is given in Chapter 2. In Chapter 3, the derivation of the generic concept of selective extrapolation is presented for the 2-D case. For the extrapolation of image signals, DFT like basis functions are especially suited, leading to the concept of *frequency selective extrapolation*. In Chapter 4, this concept is extended to the case of 3-D signals.

Following the theoretical derivations, frequency selective extrapolation is applied to solve several extrapolation problems typically occurring in image and video communications. In Chapter 5, the application of the developed approach to error concealment of lost block coded data for video communications in error-prone environments is described. After detailed investigations, frequency selective extrapolation is integrated into the reference software [54] of the H.264/AVC coder [23] as concealment feature.

Other coders such as the JPEG2000 coder [1] work with wavelet based coding principles. Since this coding principle differs from block based coders, a data loss has completely different effects. After investigations in combination with the error robustness tools provided by the standard, the 2-D frequency selective extrapolation is integrated into the JPEG2000 coder as concealment feature in Chapter 6.

In Chapter 7, two additional applications are investigated as interesting showcases of the derived frequency selective extrapolation technique: on the one hand the removal of undesired TV logos and on the other hand prediction in hybrid video coding. TV logos are commonly present for the entire sequence and located at the same time-invariant position. The concept of considering the video signal as a volume allows for extrapolating

the signal from spatial as well as temporal surrounding into the logo area as described in Section 7.1. In Section 7.2, prediction in hybrid video coding in order to increase the coding efficiency is also interpreted as extrapolation of already coded data. The 3-D extrapolation is integrated into the H.264/AVC coder as alternative coding mode.

The main results of this thesis are finally summarized in Chapter 8, where additionally an outlook to promising topics of future research work is presented.

2 Signal Extrapolation

Estimating unknown image areas from the surrounding image signal is an important topic of various applications in image and video communications. The surrounding consists of spatial information within the image. In case of video signals, temporal data from previous and/or following frames is additionally available. Examples for applications which are especially important in practice and considered in this thesis, such as concealment of lost data in erroneous image and video communications, removal of undesired TV logos, or prediction in hybrid video coding are discussed in Section 2.1 from the point of view of signal extrapolation.

A method of estimating unknown areas in image and video signals is derived in this thesis. The estimation problem is interpreted from the point of view of *signal extrapolation*: the signal is extended *beyond* an interval of known values. The image and video signals being processed are sampled, therefore the signal is extended beyond a limited number of known samples by discrete signal extrapolation. Here, we consider extrapolation as the general case, hence *interpolation* - the extension of values *within* an interval - is included, too. In short, we develop a method in order to estimate unknown samples in an image or video signal by extending the surrounding available signal as consistent as possible. The generic concept of selective extrapolation is specialized to the technique of *frequency selective* extrapolation for image and video signals by exploiting spatial and, wherever applicable, temporal correlations of the surrounding signal in the *spectral domain*.

There exists a variety of methods which can be used to extend signals towards unknown areas. One possibility is to draw conclusions about the area to be estimated from the spectrum of the given surrounding. The corresponding extrapolation task to be solved is introduced in Section 2.2.1. Extrapolation methods based on spectral estimation are reviewed in Section 2.2.2.

2.1 Applications for Extrapolation in Image and Video Communications

The task of extrapolating image signals in order to estimate unknown image contents arises in various image and video communication applications.

Transmission errors as they may occur in mobile video communications cause losses in the received video data. The effects of such losses are concealed by extrapolating the

surrounding available video signal into the missing area. Examples for the effects of such losses are presented in Fig. 2.1. For block based coders, the image is coded blockwise and several successive blocks constitute a packet which is sent to the receiver. In case of erroneous transmission, the blocks corresponding to the lost packets are substituted by black blocks as illustrated in Fig. 2.1 (a). Transmission errors cause different visual degradations for wavelet-based coded data as can be seen in Fig. 2.1 (b).

The same extrapolation principle can be exploited in order to remove undesired TV logos which are embedded in a video sequence by TV stations. Two examples of a logo are displayed in Fig. 2.2 (a), (b). The approach considered in this thesis replaces the logo by image data obtained from extrapolating the signal surrounding the logo with help of a detected logo mask as exemplified in Fig. 2.2 (c).



Figure 2.1: Concealment as an extrapolation problem: Lost areas caused by transmission errors are estimated by extrapolating the signal from the surrounding.

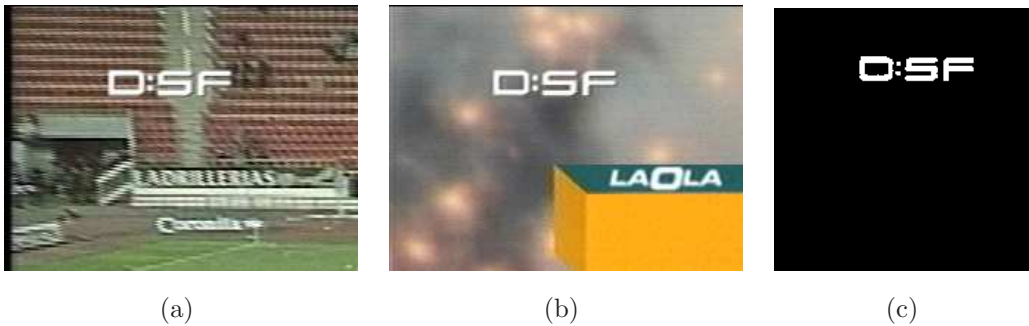


Figure 2.2: Logo inpainting as an extrapolation problem: The TV logo *DSF* in (a) and (b) should be removed with help of the detected logo mask in (c) by replacing them with the estimated signal extrapolated from the surrounding.

The coding efficiency in video coding is commonly increased by prediction which can be seen as extrapolation of the already coded signal in order to predict the samples to be coded. Then only the difference between the predicted and the actual signal has to be transmitted.

2.2 Spatial Signal Extrapolation

The extrapolation task to be solved is described in Section 2.2.1 for discrete 2-D signals. A review of extrapolation methods based on spectral estimation is given in Section 2.2.2.

2.2.1 The Extrapolation Task

Fig. 2.3 shows a part of the sampling grid of a 2-D discrete signal. The shaded area \mathcal{L} is composed of the area \mathcal{B} (dark gray) to be estimated from the given samples included in \mathcal{A} (light gray). Note that \mathcal{A} as well as \mathcal{B} can be arbitrarily shaped but that their union \mathcal{L} forms a circumscribing rectangle in our considerations. The extrapolation task can then be summarized: The support area \mathcal{A} is extrapolated to obtain the unknown area \mathcal{B} .

The intensities of the original samples in the area \mathcal{L} are denoted by $f[m, n]$, where m is the spatial dimension in row direction and n in column direction. The number of samples in the entire area \mathcal{L} equals $M \times N$. The observed signal can be interpreted as a windowed version of the original signal: The available signal is formed by the multiplication of the original signal with the binary window function $b[m, n]$ which is one in the support area and zero elsewhere

$$b[m, n] = \begin{cases} 1 & , \quad (m, n) \in \mathcal{A} \\ 0 & , \quad (m, n) \in \mathcal{B}. \end{cases} \quad (2.1)$$

The estimated signal is denoted by $g[m, n]$ which is also defined in the entire area \mathcal{L} . The signal in the support area \mathcal{A} is extrapolated towards the unknown area \mathcal{B} by approximating the observable samples in the original signal $f[m, n]$ by the estimated signal $g[m, n]$. Note that finally the observable samples of $f[m, n]$ in \mathcal{A} are kept and only the unknown samples belonging to \mathcal{B} are replaced by the corresponding samples of the estimate $g[m, n]$.

The error energy $E_{\mathcal{A}}$ between the available signal $f[m, n]$ and its approximation by the estimated signal $g[m, n]$ controls the quality of the approximation within the support area

$$E_{\mathcal{A}} = \sum_{(m,n) \in \mathcal{L}} b[m, n] (f[m, n] - g[m, n])^2. \quad (2.2)$$

It should be pointed out that in general the error criterion does not provide any conclusions about the quality of the signal in the area to be estimated.

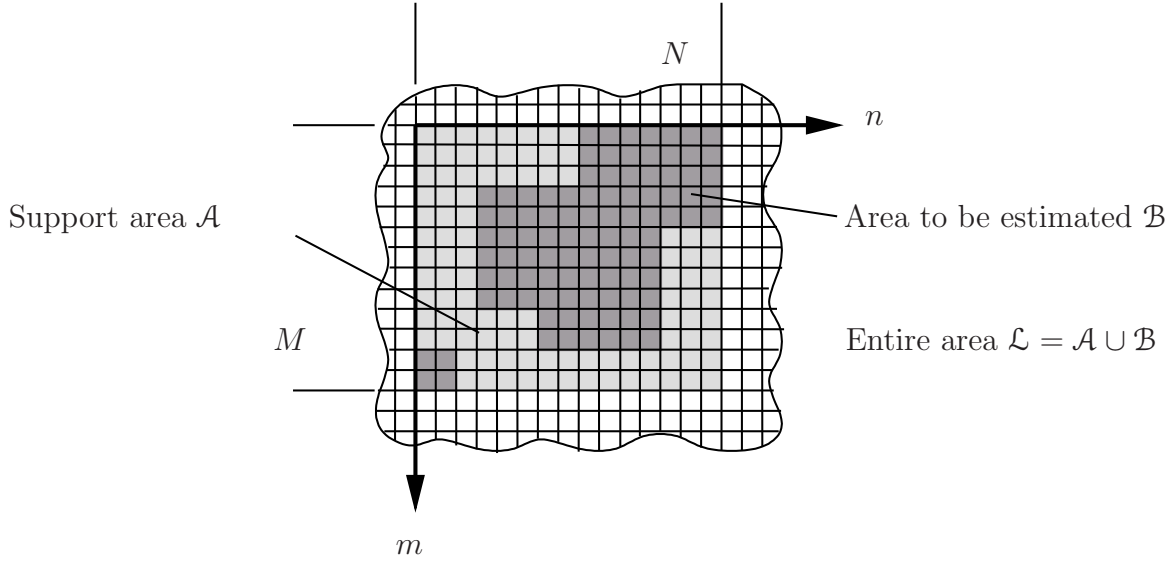


Figure 2.3: The principle of extrapolation: The unknown signal in the area \mathcal{B} (dark gray) is estimated by extrapolating the signal from the approximated support area \mathcal{A} (light gray).

2.2.2 Extrapolation by Spectral Estimation

For the 1-D and 2-D case, several methods have been presented in literature [44, 45, 18, 50] that apply a spectral analysis of discrete signals to solve the extrapolation task. The observed time-limited signal $f_b[m, n]$ can be modeled as a multiplication of the desired signal $f[m, n]$ by the time-limited binary window function $b[m, n]$. When expressing the extrapolation task in the frequency domain, the convolution of the unknown signal DFT $F[k, l]$ with the window DFT $B[k, l]$ leads to a blurred and spread version of the DFT of the observed signal $F_b[k, l]$ [42]

$$f_b[m, n] = f[m, n]b[m, n] \quad (2.3)$$

$$F_b[k, l] = \frac{1}{MN} F[k, l] \circledast B[k, l] = \frac{1}{MN} \sum_{\kappa=0}^{M-1} \sum_{\lambda=0}^{N-1} F[\kappa, \lambda] B[k - \kappa, l - \lambda] \quad (2.4)$$

where “ \circledast ” denotes cyclic convolution. The objective is to eliminate the influence of the known window DFT $B[k, l]$ from the DFT of the desired signal $F[k, l]$ by spectral analysis. Thereby, the estimate $G[k, l]$ is obtained and the corresponding spatial-domain signal $g[m, n] \circ \bullet G[k, l]$ is extrapolated beyond the known samples yielding the estimate for the unknown samples.

Band-limited Extrapolation

Several iterative approaches are known for Fourier-based spectral analysis. The techniques in [44] and [45] consider the extrapolation of time-windowed signals with known limited

bandwidth. Here, the concept derived in [44] and [45] for 1-D continuous signals is adopted for the case of discrete 2-D signals. It is assumed that the DFT of the original signal $F[k, l]$ is band-limited to the cut-off frequency bins k_c and l_c :

$$F[k, l] = 0 \quad \forall \quad k_c < k < M - k_c ; \quad l_c < l < N - l_c.$$

The iterative algorithm starts with the observed signal

$$f_b^{(0)}[m, n] = f_b[m, n] \circ \bullet F_b^{(0)}[k, l] = F_b[k, l].$$

In the ν th iteration step, the spectrum of the observed windowed signal $F_b^{(\nu)}$ is first limited to the known frequency-band by $P[k, l]$ with respect to the cut-off frequencies k_c and l_c

$$P[k, l] = \begin{cases} 1 & , \quad k \leq k_c \vee k \geq M - k_c ; l \leq l_c \vee l \geq N - l_c \\ 0 & , \quad \text{otherwise} \end{cases}. \quad (2.5)$$

Here, the 1-D ideal lowpass from [44] and [45] is generalized to a separable 2-D function. The band limitation yields the DFT of the estimated signal $G^{(\nu)}[k, l]$ in iteration ν

$$G^{(\nu)}[k, l] = F_b^{(\nu)}[k, l] P[k, l]. \quad (2.6)$$

The inverse transform to the spatial domain yields a signal $g^{(\nu)}[m, n]$ which is extrapolated beyond the known samples. Band-limitation, however, also alters the samples within the window. After replacing the altered samples by the known window-internal samples

$$f_b^{(\nu+1)}[m, n] = g^{(\nu)}[m, n] + (f_b[m, n] - g^{(\nu)}[m, n]) b[m, n],$$

the signal $f_b^{(\nu+1)}[m, n]$ is transformed again into the DFT domain yielding $F_b^{(\nu+1)}[k, l]$, where band-limitation by $P[k, l]$ is enforced. The extrapolation is obtained by iterating this procedure implying two transformations in each iteration. In [45], spectral components are additionally eliminated which are below an adaptive threshold. However, the spatial-domain signal obtained after band-limitation decays rapidly beyond the known samples in the extrapolated area, because the band-limitation by $P[k, l]$ according to (2.6) corresponds to a convolution with a sinc function in the spatial domain:

$$g^{(\nu)}[m, n] = f_b^{(\nu)}[m, n] \circledast \text{sinc} \left(\frac{2\pi k_c}{M} m, \frac{2\pi l_c}{N} n \right).$$

Therefore, the approach is not suitable for applications where the emphasis is placed on extrapolation into larger areas. This approach is extended in [16] for recovering a 1-D continuous signal from a non-uniformly sampled signal which is assumed to be band-limited. The extension to 2-D is presented in [53].

Other methods apply Projections Onto Convex Sets (POCS) for the extrapolation task. The signal in the entire area \mathcal{L} is transformed into the Discrete Fourier Transform

(DFT) domain. There, the spectrum is band-limited and then transformed back into the signal-domain. Samples in the known area are replaced by the original signal and the estimated samples in the unknown area are clipped to the valid range of amplitudes. This procedure is iteratively repeated and also requires two transformations in each iteration. It is applied in [56] to error concealment. The signal surrounding the missing block is classified in two categories: smooth and edge areas. Thus, the signal can be filtered correspondingly. In case of smooth areas, the lowpass filter $P[k, l]$ introduced in (2.5) is modified to:

$$P_{\text{smooth}}[k, l] = \begin{cases} 1 & , \quad \sqrt{k^2 + l^2} \leq R_{\text{th}} \vee \sqrt{(M-k)^2 + (N-l)^2} \leq R_{\text{th}} \\ 0 & , \quad \text{otherwise} \end{cases} \quad (2.7)$$

where R_{th} specifies the bandwidth radius. In case of an edge area, a bandpass filter

$$P_{\text{edge}}[k, l] = \begin{cases} 1 & , \quad |k - l \tan(\theta + 90^\circ)| \leq B_{\text{th}} \\ 0 & , \quad \text{otherwise} \end{cases} \quad (2.8)$$

is applied, where θ denotes the angle of the classified edge and B_{th} the bandwidth threshold. Hence, the filter characteristic is adopted to the signal content.

Extrapolation by Spectral Deconvolution

Other 2-D spectral approaches model the unknown area by the multiplication of a binary window function with the available signal. The fundamental idea to restore the missing samples is to remove the window function in the spectral domain by deconvolution. The approach aims at estimating $F[k, l]$ by $G[k, l]$ with only $D < MN$ spectral samples

$$G[k, l] = \sum_{(\kappa, \lambda) \in \mathcal{K}_D} G[\kappa, \lambda] \delta[k - \kappa, l - \lambda] \quad (2.9)$$

with the set \mathcal{K}_D consisting of D index pairs. The spectral components of $G[k, l]$ are determined iteratively by minimizing the mean squared error to the windowed original signal as defined in the error criterion (2.2). The inverse transform to the spatial domain yields a signal $g[m, n]$ which is extrapolated beyond the known samples. In [18], an iterative extrapolation approach for region oriented image analysis is described which selects the spectral component with the largest magnitude in each step. The undesirable signal decay in the extrapolated area is avoided due to signal expansion restricted to a small subset of dominant spectral coefficients as given by (2.9). As the previous approaches the technique alternates in each iteration between signal and frequency domain, and hence has a high computational complexity. This repetitive transformation is avoided by the methods proposed in [50, 51]. The desired spectral components are completely estimated in the frequency domain because the error energy in (2.2) can be expressed equivalently in the DFT-domain due to Parseval's theorem [50]. The deconvolution method and its

relation to the selective extrapolation method derived in this thesis is described in detail in Section 3.6. The application considered in [50] is modeling the human sense of hearing by a high resolution spectral analysis. In [51] some examples for a 2-D generalization of this spectrum estimation are given with application to image extrapolation.

The same principle has been used in [25] for the prediction of uncovered background in object-based video coding using spatial extrapolation. The surrounding known background signal is successively approximated yielding thereby a prediction for the uncovered area. Similar approaches are applied in [12] to restore missing non-rectangular textures and in [2] to conceal defects occurring in clinical X-ray imaging acquired by flat-panel detectors.

2.3 Related Techniques Used in Image Coding

Similar principles are used for image coding purposes. Extrapolation principles also occur as a by-product in object based transform coding. In [26], the texture of an arbitrarily shaped object is approximated successively by the estimated signal $g[m, n]$ which is defined as a parametric model given by a linear combination of a few weighted suitable basis functions $\varphi_{k,l}[m, n]$ defined over a circumscribing rectangle \mathcal{L} by

$$g[m, n] = \sum_{(k,l) \in \mathcal{K}} c_{k,l} \varphi_{k,l}[m, n] \quad (2.10)$$

with $c_{k,l}$ denoting the expansion coefficients. In (2.10), \mathcal{K} represents the set of basis functions used for the expansion of the parametric model $g[m, n]$. The expansion coefficients $c_{k,l}$ are determined by minimizing the error criterion (2.2) with respect to the support area. The texture is then cut to the shape of the object by discarding the extrapolated areas. This approach does not rely on the convolution theorem, and is thus applicable to transforms other than the DFT, such as the Discrete Cosine Transform (DCT) or the Discrete Hartley Transform (DHT).

A different approach is based on Matching Pursuits where the signal is decomposed by a linear expansion of waveforms [31] according to (2.10). Basis functions of limited extent such as Gabor functions are applied for coding the prediction error signal [41].

However, these approaches are designed for *transform coding* purposes whereas we concentrate on *signal extrapolation*.

2.4 3-D Signal Processing

For 3-D signals, the unknown signal areas are commonly predicted by hybrid approaches, i.e., either spatial or temporal information is used for the estimation but no combination. In case of spatial prediction, the block to be predicted is estimated from surrounding

data within the image, whereas temporal prediction exploits the similarity of subsequent frames.

Conventional video coding is also based on the hybrid coding approach. Temporal prediction exploits temporal information by motion compensation based on the motion vector principle. For block-based techniques, the displacement for each block of the image from one frame to the next one is described by a motion vector as for instance in the video coding standard H.264/AVC [23]. Similarly, error concealment methods such as the Boundary Matching Algorithm (BMA) [29, 61] take advantage of temporal information by restoring the motion vector.

A real 3-D video coder based on 3-D Matching Pursuits is presented in [47] by extending the 2-D Matching Pursuit approach to 3-D. This coder offers full scalability in terms of SNR, spatial and temporal scalability as opposed to a conventional video coder.

A completely different 3-D approach of predicting missing areas in order to restore severely degraded film sequences is based on stochastic techniques [28]. The missing areas are reconstructed by interpolation based on the motion and occlusion field estimation.

3 Spatial Selective Extrapolation

The basic idea of discrete signal extrapolation is to estimate unknown samples from the given surrounding signal content. Methods solving this problem by using spectral estimation have been reviewed in Section 2.2.2. In this chapter we derive our proposed approach: the algorithm of *selective extrapolation*. Based on the principle of successive approximation [24], the known signal area is approximated by a weighted linear combination of basis functions [26]. Since the basis functions are defined in the entire area including the region to be extrapolated, the approximation provides at the same time an estimation of the missing area. Approximating the *known* areas successively is extensively treated in [24, 26]. In this thesis, however, we are interested in estimating the *unknown* area by extrapolating the surrounding known signal.

First, the method of selective extrapolation is introduced generically for two-dimensional signals in Section 3.1. Next, in Section 3.2 the developed approach is interpreted from a geometrical point of view in order to obtain deeper theoretical insights. The extrapolation result significantly depends on the choice of basis functions which is addressed in Section 3.3. In this thesis, we choose multi-dimensional DFT functions as basis functions which are especially suited for image and video signals. Hence, the approach yields a *frequency selective extrapolation* which is derived in Section 3.4.1 for the spatial-domain. The algorithm can be implemented very efficiently in the frequency domain being subsequently described in Section 3.4.2. The error criterion minimized during the extrapolation procedure relies on a weighting function which allows for an application-specific adaptation of the error criterion. In Section 3.5, a suitable weighting function for the extrapolation context considered in this thesis as well as the effect on the extrapolation result compared to a non-adapted function is presented.

Finally, in Section 3.6 an alternative extrapolation approach is investigated. There exists a variety of methods in the literature based on deconvolution [22, 50, 12, 4]. We show that the algorithm of frequency selective extrapolation leads to a similar approach as deconvolution in the spectral domain.

3.1 Spatial Selective Extrapolation for 2-D Signals

In this section, we derive generically the algorithm of selective extrapolation for 2-D signals using general basis functions [27]. The unknown samples are estimated by minimizing

an error criterion between the available signal and its approximation as described in Section 3.1.1. A method for achieving the approximation and the resulting extrapolation is derived in the remainder of this section.

3.1.1 Principle of Selective Extrapolation

The generic extrapolation task to be solved has been already illustrated in Fig. 2.3. In order to extrapolate the observed signal beyond the given samples, the support area is approximated by the parametric model $g[m, n]$ as introduced in (2.10). This parametric model is defined for the entire area \mathcal{L} as linear combination of basis functions $\varphi_{k,l}[m, n]$ which are defined on the entire area \mathcal{L} , weighted by the expansion coefficients $c_{k,l}$. The number of available basis functions equals $M \times N$ and corresponds to the number of samples in the entire area \mathcal{L} .

The support area is approximated by minimizing an error criterion. As error criterion, the sum of the *weighted* squared error between the observed signal and the parametric model is defined. The weighting function $w[m, n]$ has only positive amplitudes $\rho[m, n]$ in the support area and is zero elsewhere

$$w[m, n] = \begin{cases} \rho[m, n] & , \quad (m, n) \in \mathcal{A} \\ 0 & , \quad (m, n) \in \mathcal{B}. \end{cases} \quad (3.1)$$

The weighted instantaneous error energy $E_{\mathcal{A}}$ between the available signal and its approximation by the parametric model controls the quality of the approximation within the support area

$$E_{\mathcal{A}} = \sum_{(m,n) \in \mathcal{L}} w[m, n] (f[m, n] - g[m, n])^2. \quad (3.2)$$

In [34], the weighting function with arbitrary positive amplitudes is introduced unlike previous works which are based on a binary weighting function [24, 26, 35] for the corresponding error criterion as defined in (2.2). The weighting function allows for emphasizing subareas in \mathcal{A} which are more important for the extrapolation and should therefore be involved stronger into the extrapolation procedure. The extrapolation result is significantly influenced by the weighting function depending on both, the shape of the area to be estimated and the surrounding signal. Therefore, $w[m, n]$ has to be specially designed for each application considered.

In order to determine the expansion coefficients $c_{k,l}$ of the parametric model (2.10), the error criterion is minimized by taking the partial derivative with respect to the unknown coefficients and equating it to zero

$$\frac{\partial E_{\mathcal{A}}}{\partial c_{k,l}} \stackrel{!}{=} 0 \quad (3.3)$$

yielding the following system of equations

$$\sum_{(k,l) \in \mathcal{K}} c_{k,l} \sum_{(m,n) \in \mathcal{L}} w[m,n] \varphi_{u,v}^2[m,n] = \sum_{(m,n) \in \mathcal{L}} w[m,n] f[m,n] \varphi_{u,v}[m,n]. \quad (3.4)$$

The system of equations resulting from minimizing (3.4) is underdetermined and does therefore not yield a unique solution. The minimization leads to a set of possible solutions because the number of basis functions equals the number of samples in \mathcal{L} which is larger than the number of given signal samples in \mathcal{A} . To overcome this problem caused by the underdetermined system of equations, we apply the technique of selective extrapolation. This iterative procedure approximates the signal within the support area successively subject to a specified error criterion in terms of one single weighted basis function per iteration: The expansion coefficient is optimally estimated as derived in Section 3.1.2 corresponding to the appropriately selected basis function as described in Section 3.1.3.

The parametric model is refined in each iteration step until an approximation with desired accuracy is reached. The successive approximation yields as result the parametric model $g[m,n]$ which approximates the data in the support area \mathcal{A} by a few dominant features in terms of *selected* weighted basis functions. At the same time a signal extrapolation to the unknown area \mathcal{B} is provided. Hence, we estimate the signal in the unknown area by *selective extrapolation*. Finally, the estimated area corresponding to \mathcal{B} is cut out of the parametric model $g[m,n]$ and inserted accordingly.

3.1.2 Update of Selected Coefficient

Accounting for the iterative extrapolation technique, the parametric model $g^{(\nu)}[m,n]$ describes the known signal content at iteration step ν by a linear combination of weighted selected basis functions

$$g^{(\nu)}[m,n] = \sum_{(k,l) \in \mathcal{K}^{(\nu)}} c_{k,l}^{(\nu)} \varphi_{k,l}[m,n] \quad (3.5)$$

with $\mathcal{K}^{(\nu)}$ denoting the set of basis functions used up to this step. Initially, $\mathcal{K}^{(\nu)}$ is empty for $\nu = 0$ and the parametric model $g^{(0)}[m,n]$ is zero. The residual error signal in the support area $(m,n) \in \mathcal{A}$ at iteration ν is then given with help of the window function $b[m,n]$ (2.1) by

$$r^{(\nu)}[m,n] = (f[m,n] - g^{(\nu)}[m,n]) b[m,n]. \quad (3.6)$$

In each iteration, the residual error signal in the support area is further reduced by $\Delta g[m,n]$ which represents the change of the parametric model from step ν to $\nu + 1$. $\Delta g[m,n]$ is updated in each iteration. Hence, the residual error signal in the next iteration can be expressed by

$$r^{(\nu+1)}[m,n] = r^{(\nu)}[m,n] - \Delta g[m,n] b[m,n]. \quad (3.7)$$

Analogously to (3.2), the *weighted instantaneous residual* error energy represents the error criterion and can be expressed by

$$E_{\mathcal{A}}^{(\nu+1)} = \sum_{(m,n) \in \mathcal{L}} w[m, n] \left(r^{(\nu)}[m, n] - \Delta g[m, n] \right)^2, \quad (3.8)$$

where $b[m, n]w[m, n] = w[m, n]$ has been used. The update of the parametric model $\Delta g[m, n]$ is determined by minimizing $E_{\mathcal{A}}^{(\nu+1)}$.

In the following of this section, we assume that an appropriate basis function $\varphi_{u,v}[m, n]$ with index (u, v) is already selected. Strategies how to select an appropriate basis function are treated in Section 3.1.3. In case of the selective extrapolation, the update of the parametric model $\Delta g[m, n]$ is chosen as the weighted actually selected basis function, i.e.,

$$\Delta g[m, n] = \Delta c \varphi_{u,v}[m, n] \quad (3.9)$$

where Δc is the optimal update of the currently available approximation to be estimated in the next iteration. Thus, Δc is updated in each iteration. This can be used to express the reduced error signal in (3.7) by

$$r^{(\nu+1)}[m, n] = r^{(\nu)}[m, n] - \Delta c \varphi_{u,v}[m, n] b[m, n]. \quad (3.10)$$

The error criterion generally defined in (3.8) is obtained by inserting (3.9)

$$E_{\mathcal{A}}^{(\nu+1)} = \sum_{(m,n) \in \mathcal{L}} w[m, n] \left(r^{(\nu)}[m, n] - \Delta c \varphi_{u,v}[m, n] \right)^2 \quad (3.11)$$

and minimized with respect to the unknown coefficient Δc

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c} \stackrel{!}{=} 0. \quad (3.12)$$

This yields a single equation in contrary to the underdetermined system of equations (3.4)

$$\Delta c \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}^2[m, n] = \sum_{(m,n) \in \mathcal{L}} w[m, n] r^{(\nu)}[m, n] \varphi_{u,v}[m, n] \quad (3.13)$$

which can be solved uniquely for Δc

$$\Delta c = \frac{\sum_{(m,n) \in \mathcal{L}} w[m, n] r^{(\nu)}[m, n] \varphi_{u,v}[m, n]}{\sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}^2[m, n]}. \quad (3.14)$$

The selected expansion coefficient $c_{u,v}^{(\nu+1)}$ is then updated by

$$c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c. \quad (3.15)$$

The index (u, v) is included in the set of basis functions used if it has not been selected in previous iterations

$$\mathcal{K}^{(\nu+1)} = \mathcal{K}^{(\nu)} \cup (u, v) \quad \text{if } (u, v) \notin \mathcal{K}^{(\nu)}. \quad (3.16)$$

Using the definition

$$r_w^{(\nu)}[m, n] = w[m, n] r^{(\nu)}[m, n], \quad (3.17)$$

we can simplify (3.13) to

$$\Delta c \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}^2[m, n] = \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{u,v}[m, n].$$

Analogously to (3.17), the residual error $r_w^{(\nu+1)}$ is obtained from (3.10) as

$$r_w^{(\nu+1)}[m, n] = r_w^{(\nu)}[m, n] - \Delta c \varphi_{u,v}[m, n] w[m, n], \quad (3.18)$$

where we recall that $b[m, n]w[m, n] = w[m, n]$.

It should be noted that the above derivations imply that the *weighted instantaneous residual error energy* is minimized and not the energy of the weighted residual error. Thus, the error signal is weighted after squaring but before summation, i.e., the weighting function does not affect the residual error signal but the error criterion. This is pointed out by introducing the binary function in order to calculate the residual error first and then reformulating the result with help of $r_w^{(\nu)}[m, n]$.

3.1.3 Selection of Basis Function

For the calculation of the expansion coefficients $c_{u,v}^{(\nu+1)}$, we assumed that an appropriate basis function $\varphi_{u,v}[m, n]$ was already selected for the expansion of the parametric model from iteration ν to $\nu + 1$. The question not addressed so far is *how* to select this function. Generally, we seek to select that basis function $\varphi_{u,v}[m, n]$ which maximises the reduction of the error criterion from step ν to $\nu + 1$. Consequently, we calculate the weighted residual error energy from iteration ν to $\nu + 1$, taking into account that the residuum $r_w^{(\nu+1)}[m, n]$ is orthogonal to the weighted selected basis function

$$\begin{aligned} E_{\mathcal{A}}^{(\nu+1)} + \Delta E_{\mathcal{A}}^{(\nu+1)} &= E_{\mathcal{A}}^{(\nu)} \\ \sum_{(m,n) \in \mathcal{L}} w[m, n] (r^{(\nu+1)}[m, n])^2 + \sum_{(m,n) \in \mathcal{L}} w[m, n] (\Delta g[m, n])^2 &= \sum_{(m,n) \in \mathcal{L}} w[m, n] (r^{(\nu)}[m, n])^2. \end{aligned} \quad (3.19)$$

The error criterion $E_{\mathcal{A}}^{(\nu+1)}$ becomes minimum if the reduction of $E_{\mathcal{A}}^{(\nu)}$ by $\Delta E_{\mathcal{A}}^{(\nu+1)}$ becomes maximum. Therefore, we select that basis function with index (u, v) which results in

a maximal decrease of the error criterion. Combining (3.9) and (3.14), $\Delta E_{\mathcal{A}}^{(\nu+1)}$ can be written as

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \Delta c^2 \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k,l}^2[m, n] \\ &= \frac{\left(\sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{k,l}[m, n] \right)^2}{\sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k,l}^2[m, n]}. \end{aligned} \quad (3.20)$$

The basis function $\varphi_{u,v}[m, n]$ is then selected according to

$$(u, v) = \underset{(k,l)}{\operatorname{argmax}} \Delta E_{\mathcal{A}}^{(\nu+1)}. \quad (3.21)$$

As can be seen from (3.19), the convergence of the algorithm is assured since the error energy in the approximated area is reduced in each step.

The algorithm is initialized by

$$g^{(0)}[m, n] = 0 \quad (3.22)$$

and terminates if the reduction of the residual error energy $\Delta E_{\mathcal{A}}$ normalized per pixel with respect to its contribution according to the weighting functions drops below a pre-specified threshold Δ_{\min}

$$\frac{\Delta E_{\mathcal{A}}^{(\nu+1)}}{\sum_{(m,n) \in \mathcal{L}} w[m, n]} < \Delta_{\min}.$$

The termination criterion is then given with respect to the threshold E_{\min}

$$\Delta E_{\mathcal{A}}^{(\nu+1)} < E_{\min} = \Delta_{\min} \sum_{(m,n) \in \mathcal{L}} w[m, n]. \quad (3.23)$$

The successive approximation procedure results in a parametric model $g[m, n]$ given in the entire area \mathcal{L} . The desired signal extrapolation to the missing area \mathcal{B} is then obtained as the corresponding area of the parametric model $g[m, n]$.

3.2 Geometrical Interpretation of Selective Extrapolation

In the following, we interpret the principle of selective extrapolation from a geometrical point of view. The area \mathcal{L} is composed of the support area \mathcal{A} and the area to be extrapolated \mathcal{B} . Each function in \mathcal{L} can be interpreted as a $L \times 1$ vector where $L = M N$. Hence,

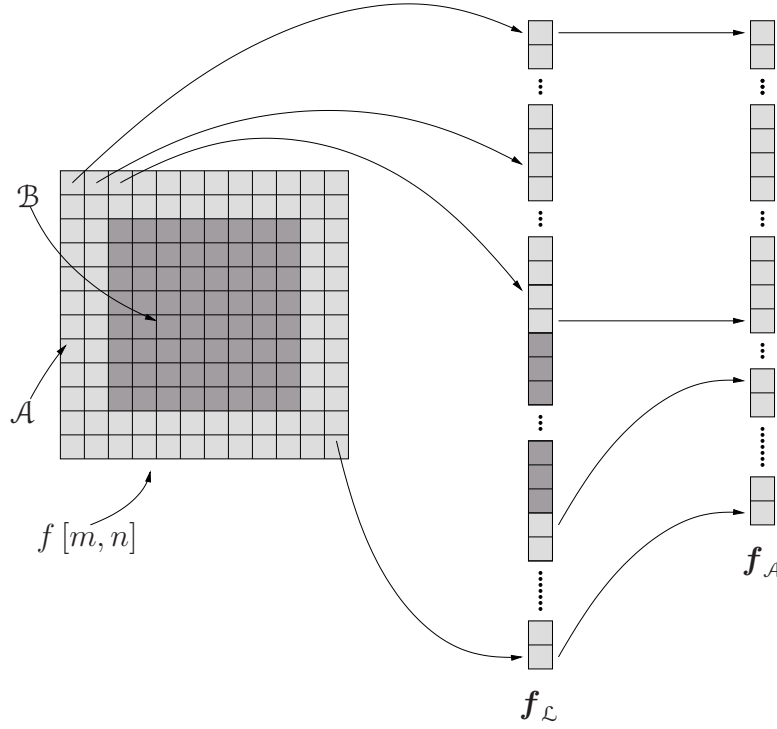


Figure 3.1: Geometrical representation of a two-dimensional function by stacking the columns of the function below each other.

the image signal $f[m, n]$ can be interpreted as a vector $\mathbf{f}_{\mathcal{L}}$ with L elements. Therefore, the columns of $f[m, n]$ are subsequently stacked columnwise below each other. On the other hand it is also possible to compose a vector $\mathbf{f}_{\mathcal{A}}$ which covers only the support area \mathcal{A} . The vector $\mathbf{f}_{\mathcal{A}}$ is derived from the vector $\mathbf{f}_{\mathcal{L}}$ by discarding the elements originally located in \mathcal{B} . Let A and B denote the number of elements included in \mathcal{A} and \mathcal{B} , respectively. The vector $\mathbf{f}_{\mathcal{L}}$ is thus shortened by B elements yielding $\mathbf{f}_{\mathcal{A}}$ with A elements. This relation is depicted graphically in Fig. 3.1.

The basis functions $\varphi_{k,l}[m, n], \forall (m, n) \in \mathcal{L}$ compose the basis vectors $\varphi_{\mathcal{L}}[\lambda]$. In this thesis, we restrict ourselves to sets of basis functions which are linearly independent to each other with respect to \mathcal{L} and additionally mutually orthogonal

$$\varphi_{\mathcal{L}}[i]^T \varphi_{\mathcal{L}}[j] = 0, \text{ for } i \neq j$$

and span a L dimensional vector space corresponding to \mathcal{L} .

In contrast, the observable components of the basis vectors $\varphi_{\mathcal{L}}[\lambda]$ that are summarized in the corresponding shortened vector $\varphi_{\mathcal{A}}[\lambda]$ are in general not mutually orthogonal anymore

$$\varphi_{\mathcal{A}}[i]^T \varphi_{\mathcal{A}}[j] \neq 0, \text{ for } i \neq j.$$

However, the number of available basis vectors has not changed. There exist L vectors $\varphi_{\mathcal{A}}[\lambda]$ spanning a vector space with only maximally A dimensions.

The aim of selective extrapolation can be interpreted therefore in the following way: The approximation of the vector $\mathbf{f}_{\mathcal{L}}$ of the original signal by the parametric model takes place in the known subspace \mathcal{A} . The vector $\mathbf{f}_{\mathcal{A}}$ is approximated by the vector of the parametric model $\mathbf{g}_{\mathcal{A}}$ within \mathcal{A} . The aim is to extend the vector $\mathbf{g}_{\mathcal{A}}$ beyond \mathcal{A} into the unknown subspace \mathcal{B} due to the fact that the basis vectors are defined in \mathcal{L} . The optimization criterion has to be evaluated with respect to the observable part of $\mathbf{f}_{\mathcal{L}}$ in \mathcal{A} , i.e. the vector $\mathbf{f}_{\mathcal{A}}$. The subspace \mathcal{A} is composed of L different basis vectors but has only the dimension A . As derived in Section 3.1, we select one basis vector per iteration for the approximation. For presentational convenience we consider the special case that the weighting function $w[m, n] = b[m, n]$, implying $\rho[m, n] = 1$. Therefore, the weighting function can be discarded in the following discussion.

The residual error criterion is maximally reduced for the vector with the largest length $|\Delta c \varphi_{\mathcal{A}}[\lambda]|$. Hence, the basis vector $\varphi_{\mathcal{A}}[\lambda]$ which maximizes

$$\Delta E_{\mathcal{A}}^{(\nu+1)} = |\Delta c \varphi_{\mathcal{A}}[\lambda_{u,v}]|^2 = \Delta c^2 \varphi_{\mathcal{A}}[\lambda_{u,v}]^T \varphi_{\mathcal{A}}[\lambda_{u,v}] \quad (3.24)$$

is selected. In order to determine the expansion coefficient Δc , the vector $\mathbf{r}_{\mathcal{A}}^{(\nu)}$ is projected onto the selected basis vector

$$\Delta c = \frac{\mathbf{r}_{\mathcal{A}}^{(\nu)T} \varphi_{\mathcal{A}}[\lambda]}{\varphi_{\mathcal{A}}[\lambda]^T \varphi_{\mathcal{A}}[\lambda]}. \quad (3.25)$$

From (3.25) it follows that the length of the vector $\mathbf{r}_{\mathcal{A}}^{(\nu+1)}$ becomes minimum because $\mathbf{r}_{\mathcal{A}}^{(\nu+1)}$ and $\Delta c \varphi_{\mathcal{A}}[\lambda]$ are mutually orthogonal

$$\begin{aligned} \left(\mathbf{r}_{\mathcal{A}}^{(\nu)} - \Delta c \varphi_{\mathcal{A}}[\lambda] \right)^T \varphi_{\mathcal{A}}[\lambda] &= 0 \\ \Rightarrow \mathbf{r}_{\mathcal{A}}^{(\nu+1)T} \Delta c \varphi_{\mathcal{A}}[\lambda] &= 0. \end{aligned}$$

The combination of basis function and expansion coefficient $\Delta c \varphi_{\mathcal{A}}[\lambda]$ is added to the vector of the parametric model $\mathbf{g}_{\mathcal{A}}^{(\nu)}$ in \mathcal{A} so far calculated. This procedure is repeated iteratively by approximating $\mathbf{f}_{\mathcal{A}}$ successively by $\mathbf{g}_{\mathcal{A}}^{(\nu)}$ yielding

$$\mathbf{g}_{\mathcal{A}}^{(\nu)} = \sum_{\lambda \in \mathcal{K}^{(\nu)}} c_{\lambda}^{(\nu)} \varphi_{\mathcal{A}}[\lambda]. \quad (3.26)$$

After termination, the vector $\mathbf{g}_{\mathcal{L}}^{(\nu)}$ of the parametric model is composed with respect to the complete space by \mathcal{L}

$$\mathbf{g}_{\mathcal{L}}^{(\nu)} = \sum_{\lambda \in \mathcal{K}^{(\nu)}} c_{\lambda}^{(\nu)} \varphi_{\mathcal{L}}[\lambda], \quad (3.27)$$

with the same expansion coefficients $c_\lambda^{(\nu)}$ as in (3.26). Thereby, the extension of the original signal $\mathbf{f}_\mathcal{A}$ beyond the support area \mathcal{A} into the unknown area \mathcal{B} is achieved.

The principle of selective extrapolation is exemplified step-wise by Fig. 3.2. The area \mathcal{L} considered in this example consists of three pixels, the missing area \mathcal{B} of one pixel, and the support area \mathcal{A} of two pixels. Consequentially there exist three basis functions and their associated basis vectors $\varphi_\mathcal{L}[1]$, $\varphi_\mathcal{L}[2]$, and $\varphi_\mathcal{L}[3]$ which form an orthogonal basis in \mathcal{L} . This orthogonal basis is shown in Fig. 3.2 (a). The vector $\mathbf{f}_\mathcal{L}$ of the original signal which is located in \mathcal{L} is also depicted. For illustration purposes, the location of $\mathbf{f}_\mathcal{L}$ within the vector space corresponding to \mathcal{L} is sketched in. However, the observable part of the vector of the original signal $\mathbf{f}_\mathcal{L}$ composes the vector $\mathbf{f}_\mathcal{A}$ in the subspace \mathcal{A} which is only two-dimensional. The three basis vectors $\varphi_\mathcal{A}[1]$, $\varphi_\mathcal{A}[2]$, and $\varphi_\mathcal{A}[3]$ span this subspace \mathcal{A} of two dimensions, i.e. the plane \mathcal{A} , but they are not mutually orthogonal anymore. The reduction of the space \mathcal{L} to the subspace \mathcal{A} with the respective vectors is clarified by Fig. 3.2 (b). For illustration purposes, the location where the plane \mathcal{A} spanned by the vectors $\varphi_\mathcal{A}[1]$, $\varphi_\mathcal{A}[2]$, $\varphi_\mathcal{A}[3]$ intersects the vector space \mathcal{L} is indicated.

The approximation of the vector $\mathbf{f}_\mathcal{A}$ of the observable part of the original signal takes place in the subspace \mathcal{A} . The first iteration of the approximation procedure within \mathcal{A} is shown in Fig. 3.2 (c). Since for the approximation in \mathcal{A} the location of $\mathbf{f}_\mathcal{L}$ is unknown, the relation of the plane \mathcal{A} to the vector space \mathcal{L} is not sketched in anymore. In fact, the true approximation situation is shown where $\mathbf{f}_\mathcal{A}$ is approximated in \mathcal{A} without any knowledge of $\mathbf{f}_\mathcal{L}$. The parametric model is initialized by the zero vector, i.e. the residual $\mathbf{r}_\mathcal{A}^{(0)}$ equals the vector $\mathbf{f}_\mathcal{A}$. The basis vector $\varphi_\mathcal{A}[2]$ is selected in the first iteration because the vector $\Delta c \varphi_\mathcal{A}[2]$ reduces the residual energy maximally. The expansion coefficient $c_2^{(1)} = \Delta c$ is determined by the projection of $\mathbf{f}_\mathcal{A}$ onto $\varphi_\mathcal{A}[2]$. The residual error $\mathbf{r}_\mathcal{A}^{(1)}$ becomes minimum because the vectors $\mathbf{r}_\mathcal{A}^{(1)}$ and $\Delta c \varphi_\mathcal{A}[2]$ are orthogonal with respect to each other. The vector of the parametric model is then given after the first iteration by $\mathbf{g}_\mathcal{A}^{(1)} = c_2^{(1)} \varphi_\mathcal{A}[2]$. The second iteration of the approximation procedure in the plane \mathcal{A} is shown in Fig. 3.2 (d), where $\varphi_\mathcal{A}[3]$ is selected. Next, $\mathbf{r}_\mathcal{A}^{(1)}$ is projected onto $\varphi_\mathcal{A}[3]$. In other words, the expansion coefficient $c_3^{(2)} = \Delta c$ is obtained when the approximation error $\mathbf{r}_\mathcal{A}^{(2)}$ is orthogonal to $c_3^{(2)} \varphi_\mathcal{A}[3]$. The vector of the parametric model is then given by

$$\mathbf{g}_\mathcal{A}^{(2)} = \mathbf{g}_\mathcal{A}^{(1)} + c_3^{(2)} \varphi_\mathcal{A}[3] = c_2^{(1)} \varphi_\mathcal{A}[2] + c_3^{(2)} \varphi_\mathcal{A}[3].$$

The vector $\mathbf{f}_\mathcal{A}$ is approximated further by $\mathbf{g}_\mathcal{A}$ according to the same procedure. After termination, the parametric model $\mathbf{g}_\mathcal{L}$ can be composed within the complete space \mathcal{L} . The selected basis vectors $\varphi_\mathcal{L}[\lambda]$ in \mathcal{L} are multiplied by the respective calculated expansion coefficients $c_\lambda^{(\nu)}$. This is shown in Fig. 3.2 (e) for $\mathbf{g}_\mathcal{L}^{(2)}$ which is then finally given by

$$\mathbf{g}_\mathcal{L}^{(2)} = \mathbf{g}_\mathcal{L}^{(1)} + c_3^{(2)} \varphi_\mathcal{L}[3] = c_2^{(1)} \varphi_\mathcal{L}[2] + c_3^{(2)} \varphi_\mathcal{L}[3]. \quad (3.28)$$

Note that the vector $\mathbf{g}_\mathcal{L}^{(2)}$ points into a completely different direction than $\mathbf{g}_\mathcal{A}^{(2)}$. For presentational clarity and to avoid confusions, the dashed auxiliary lines to indicate the locations

of the vectors within the respective space as in Fig. 3.2 (b) are discarded. However, it is more important that the approximation by the vector of the parametric model $\mathbf{g}_{\mathcal{A}}$ takes place in a *different* space than the composition of the vector of the extrapolation by $\mathbf{g}_{\mathcal{L}}$. Because the vector $\mathbf{f}_{\mathcal{L}}$ is unknown, it is not possible to draw any conclusions how good the approximation by $\mathbf{g}_{\mathcal{L}}$ actually is. Therefore, a better or even perfect approximation of $\mathbf{f}_{\mathcal{A}}$ by $\mathbf{g}_{\mathcal{A}}$ in \mathcal{A} does not necessarily mean a closer extrapolation of $\mathbf{g}_{\mathcal{L}}$ to $\mathbf{f}_{\mathcal{L}}$ in \mathcal{L} . In the applications considered, we rely on the fact that the unknown part of the original signal is an extension of the signal contents beyond \mathcal{A} into \mathcal{B} .

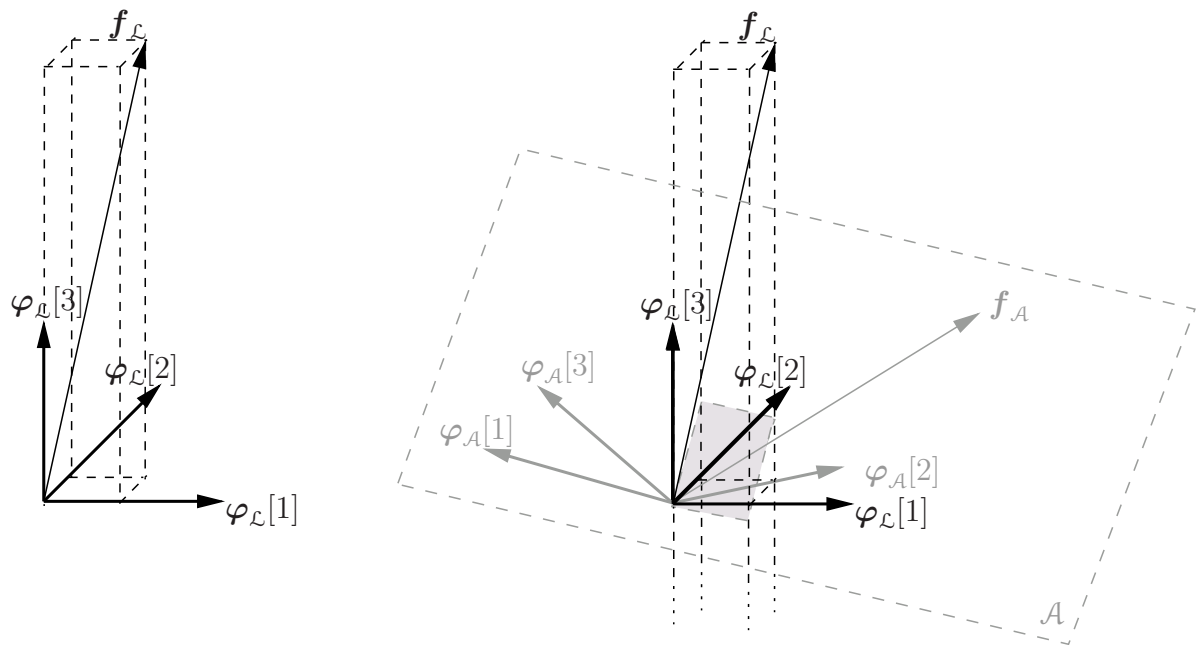
The weighting function $w[m, n]$ included in (3.14) should be also incorporated in the vector representation of the update Δc (3.25). Therefore, the vector $\mathbf{w}_{\mathcal{A}}$ with the elements corresponding to the weighting function $w[m, n]$ is introduced. The diagonal matrix $\text{diag}\{\mathbf{w}_{\mathcal{A}}\}$ has the elements of $\mathbf{w}_{\mathcal{A}}$ on its main diagonal. Hence, the elementwise multiplication of the weighting function with the residual error is included by rewriting (3.25) as

$$\Delta c = \frac{\left(\text{diag}\{\mathbf{w}_{\mathcal{A}}\} \mathbf{r}_{\mathcal{A}}^{(\nu)}\right)^T \boldsymbol{\varphi}_{\mathcal{A}}[\lambda]}{\left(\text{diag}\{\mathbf{w}_{\mathcal{A}}\} \boldsymbol{\varphi}_{\mathcal{A}}[\lambda]\right)^T \boldsymbol{\varphi}_{\mathcal{A}}[\lambda]}$$

(3.24) has to be rewritten accordingly

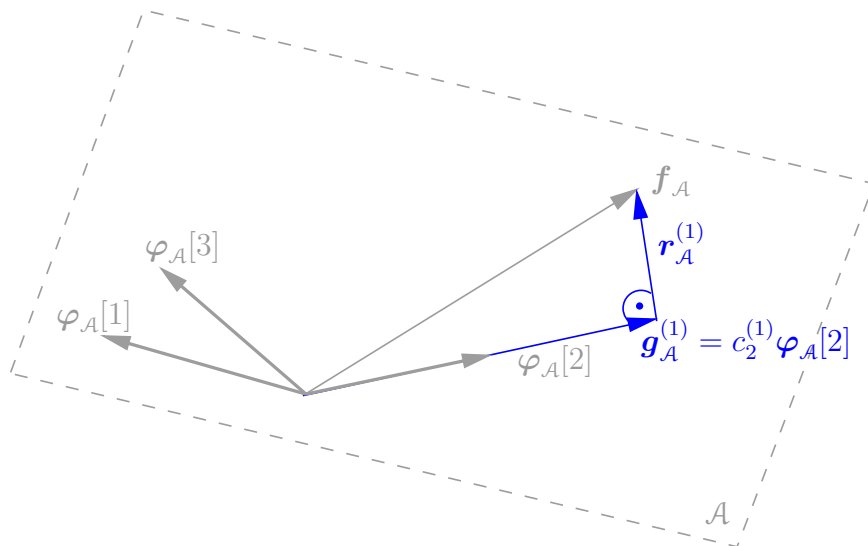
$$\Delta E_{\mathcal{A}}^{(\nu+1)} = \Delta c^2 \text{diag}\{\mathbf{w}_{\mathcal{A}}\} \boldsymbol{\varphi}_{\mathcal{A}}[\lambda_{u,v}]^T \boldsymbol{\varphi}_{\mathcal{A}}[\lambda_{u,v}].$$

It should be pointed out that the weighted scalar product does not affect the derivations and interpretations of the algorithm.

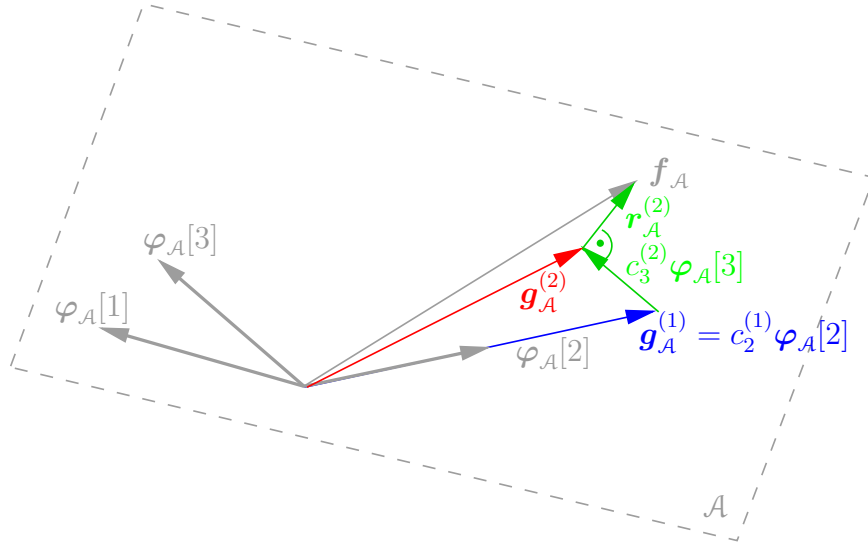


(a)

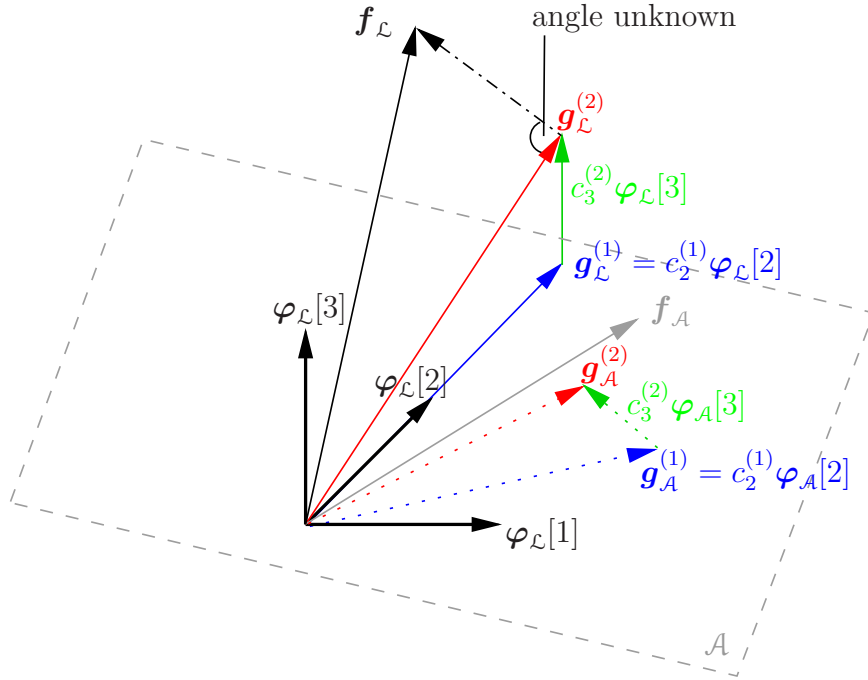
(b)



(c)



(d)



(e)

Figure 3.2: Geometrical interpretation of selective extrapolation: (a) Original vector $\mathbf{f}_{\mathcal{L}}$ within space \mathcal{L} spanned by orthogonal basis $\varphi_{\mathcal{L}}[\lambda]$. (b) Reduction of vector $\mathbf{f}_{\mathcal{L}}$ in \mathcal{L} to $\mathbf{f}_{\mathcal{A}}$ within observable support area \mathcal{A} spanned by $\varphi_{\mathcal{A}}[\lambda]$. (c) 1st iteration: approximation of $\mathbf{f}_{\mathcal{A}}$ by $\mathbf{g}_{\mathcal{A}}^{(1)}$ within \mathcal{A} . (d) 2nd iteration: approximation of $\mathbf{f}_{\mathcal{A}}$ by $\mathbf{g}_{\mathcal{A}}^{(2)}$ within \mathcal{A} . (e) Aim of selective extrapolation beyond \mathcal{A} : Composition of $\mathbf{g}_{\mathcal{L}}^{(2)}$ within \mathcal{L} .

3.3 Suitable Basis Functions

In this thesis we treat extrapolation problems in image and video signal processing. The aim is that the signal extrapolated into the unknown area is a consistent extension of the signal in the support area and, most important, that the extrapolation result looks visually pleasant. For a consistent extension of image or video signals, respectively, suitable basis function are needed which is treated in the following.

So far, we derived generically an algorithm in order to extrapolate any kind of real-valued discrete signal beyond its borders by a weighted linear combination of basis functions. The set of basis functions used for the extrapolation should be able to extend the signal as consistently as possible. The success of the extrapolation depends significantly on the choice of basis functions which is determined by the underlying signal. We are aiming at extrapolating image and video signals. The textures in images vary strongly, they can be smooth, consisting of edges or be detailed, as noise like areas or regular patterns. All these different textures should be extended beyond its borders. In video signals, additionally to the different textures motion trajectories should be extended. Our objective is to search for a set of basis functions which fullfils all these requirements at the same time.

Thus, we take advantage of the extrapolating properties of basis functions. In general periodic functions are suited because they are able to extend the signal periodically. In contrary, functions of limited extent such as polynomials or wavelets are not suited because they lack the extrapolation ability if the missing area becomes larger. Therefore, periodic functions like the DCT or DFT are suited for the extrapolation of image or video signals.

Different periodic basis functions such as the 2-D DCT and 2-D DFT are discussed with respect to their extrapolation properties in the following for two-dimensional signals. The DCT contains vertical and horizontal basis images as shown in Fig. 3.3(a). As opposed to the DFT basis images which provide diagonal orientations and a phase relation. The real part is displayed in Fig. 3.3(b) and the imaginary part in Fig. 3.3(c), both showing the same orientation but different phasing.

Using the DFT allows to extend also diagonal edges and structures due to the orientations in contrast to the DCT [35]. The DCT is not capable of resolving single diagonal orientations [5] because only mirrored diagonal orientations can be resolved. Hence, the DFT is better suited for *extrapolation purposes*. In [26], the DCT is applied in region based image coding. However, coding aims at concentrating as much energy as possible in as few transform coefficients as possible. As well-known, the DCT meets these requirements and is widely used in image coding. Therefore, the DCT is the better choice in terms of *coding*.

For reasons mentioned above, the DFT basis functions are applied to the extrapolation of image and video signals derived in Section 3.4 and Section 4.2, respectively.

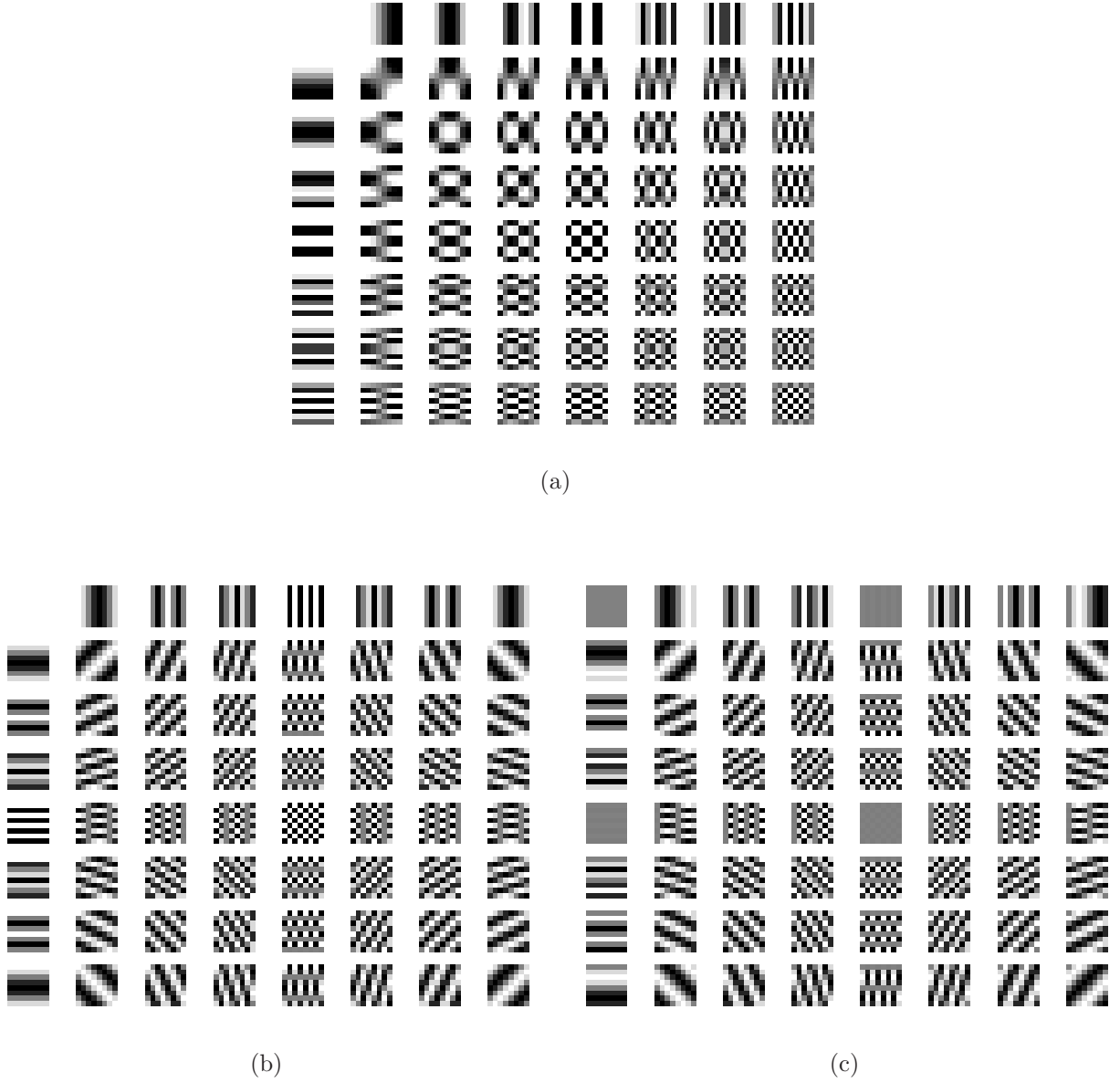


Figure 3.3: 8×8 basis images. (a) DCT. (b) Real and (c) imaginary part of DFT.

3.4 Frequency Selective Extrapolation of Image Signals Using DFT Basis Functions

Based on the two-dimensional DFT functions, we use

$$\varphi_{k,l}[m,n] = e^{j\frac{2\pi}{M}mk} e^{j\frac{2\pi}{N}nl} \quad (3.29)$$

as basis functions in order to extrapolate an image signal beyond its available area [35] according to Section 3.3. M denotes the number of rows and N the number of columns.

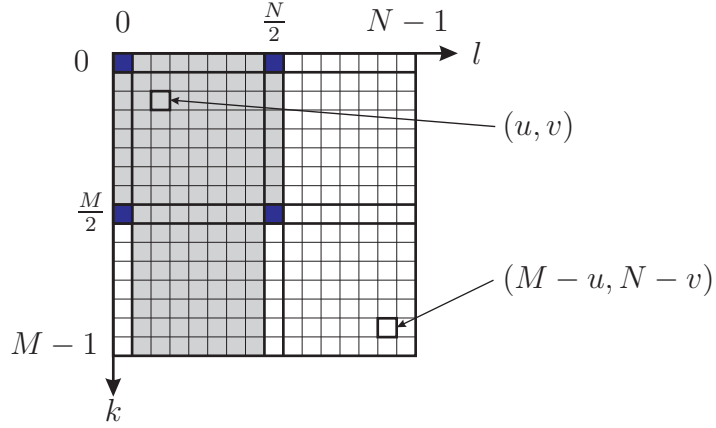


Figure 3.4: The conjugate complex symmetry for a DFT spectrum of a 2-D real-valued signal is shown. Blue: Real-valued DFT coefficients. Colored area: Coefficients which have to be calculated.

$M \times N$ equals both the number of pixels in \mathcal{L} and the number of basis functions.

The basis functions fulfill the following conjugate complex symmetry

$$\varphi_{M-k, N-l}[m, n] = \varphi_{k, l}^*[m, n]. \quad (3.30)$$

The location in the frequency plane of a pair of conjugate complex basis functions is illustrated in Fig. 3.4 for M, N being even. In case of real-valued signals such as image signals, also the DFT-coefficients exhibit this conjugate complex symmetry. The discrete frequencies belonging to a real-valued spectrum due to a real-valued input signal are highlighted in blue. They compose the set \mathcal{M} for even M, N

$$\mathcal{M} = \left\{ (0, 0), \left(0, \frac{N}{2} \right), \left(\frac{M}{2}, 0 \right), \left(\frac{M}{2}, \frac{N}{2} \right) \right\}. \quad (3.31)$$

Due to the symmetry properties, the DFT spectrum of a real-valued signal is completely described by the shaded region shown in Fig. 3.4.

3.4.1 Spatial-Domain Solution

To ensure that the approximation $g^{(\nu)}[m, n]$ in each iteration ν yields a real-valued signal, we modify (3.5) for the parametric model to

$$g^{(\nu)}[m, n] = \frac{1}{2MN} \sum_{(k, l) \in \mathcal{K}^{(\nu)}} (c_{k, l}^{(\nu)} \varphi_{k, l}[m, n] + c_{M-k, N-l}^{(\nu)} \varphi_{M-k, N-l}[m, n]), \quad (3.32)$$

where the conjugate complex symmetry of the expansion coefficients is taken into account as already illustrated in Fig. 3.4 according to

$$c_{M-k, N-l}^{(\nu)} = c_{k, l}^{(\nu)*}. \quad (3.33)$$

The principle of the approximation procedure remains unchanged and proceeds for the coefficient update as described in Section 3.1.2 and for the selection of an appropriate basis function according to Section 3.1.3.

The residual error signal in iteration $\nu + 1$ is gained by subtracting $\Delta g[m, n]$ from the already obtained residual error signal in step ν . Taking the symmetry properties into account, the update of the parametric model

$$\Delta g[m, n] = \frac{1}{2MN} (\Delta c \varphi_{u,v}[m, n] + \Delta c^* \varphi_{M-u, N-v}[m, n]) \quad (3.34)$$

further approximates the residual error signal in the next iteration

$$r_w^{(\nu+1)}[m, n] = r_w^{(\nu)}[m, n] - \frac{1}{2MN} (\Delta c \varphi_{u,v}[m, n] + \Delta c^* \varphi_{M-u, N-v}[m, n]) w[m, n]. \quad (3.35)$$

The error criterion (3.8) is adapted accordingly

$$E_{\mathcal{A}}^{(\nu+1)} = \sum_{(m,n) \in \mathcal{L}} w[m, n] \left(r^{(\nu)}[m, n] - \frac{1}{2MN} (\Delta c \varphi_{u,v}[m, n] + \Delta c^* \varphi_{M-u, N-v}[m, n]) \right)^2$$

and minimized by partially derivating with respect to Δc and Δc^* and equating the result to zero

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c} \stackrel{!}{=} 0 \quad \wedge \quad (3.36)$$

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c^*} \stackrel{!}{=} 0. \quad (3.37)$$

Evaluating $\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c^*} \stackrel{!}{=} 0$ leads to

$$\begin{aligned} & \Delta c \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}[m, n] \varphi_{M-u, N-v}[m, n] \\ & + \Delta c^* \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{M-u, N-v}[m, n] \varphi_{M-u, N-v}[m, n] \\ & = 2MN \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{M-u, N-v}[m, n]. \end{aligned} \quad (3.38)$$

A conjugate complex equation to (3.38) is obtained for (3.36). Solving the two equations for Δc yields a solution for Δc and a conjugate complex equation for Δc^* which is derived in detail in the next section for a frequency-domain solution.

Here, also the conjugate complex coefficient has to be updated due to the symmetry requirements

$$\begin{aligned} c_{u,v}^{(\nu+1)} &= c_{u,v}^{(\nu)} + \Delta c \\ c_{M-u, N-v}^{(\nu+1)} &= c_{M-u, N-v}^{(\nu)} + \Delta c^*. \end{aligned} \quad (3.39)$$

In each step, the weighted basis function with index (u, v) is selected which maximally decreases the residual error criterion $\Delta E_{\mathcal{A}}^{(\nu+1)}$ with respect to Δc

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \frac{1}{2(MN)^2} \left(|\Delta c|^2 \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}[m, n] \varphi_{u,v}^*[m, n] \right. \\ &\quad \left. + \operatorname{Re}\{\Delta c^2 \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}^2[m, n]\} \right) \\ \Rightarrow (u, v) &= \operatorname{argmax}_{(k,l)} \Delta E_{\mathcal{A}}^{(\nu+1)}. \end{aligned} \quad (3.40)$$

Due to the symmetry properties of 2-D real-valued signals, the search area of the respective optimum basis function is limited to the shaded region in Fig. 3.4.

3.4.2 Frequency-Domain Solution

The derivation of 2-D signal extrapolation is carried out in the spatial-domain so far. Using DFT basis functions allows us to express all equations in the frequency domain. The evaluation of the sums in (3.38), (3.40) is computationally very expensive but with help of the DFT an efficient implementation of the extrapolation algorithm can be obtained. As derived in the following, only one DFT transform in the beginning and one inverse DFT transform at the end is necessary while all intermediate computations can be expressed in the frequency domain.

Using DFT functions as basis functions, the summation of the product of a function $x[m, n]$ and the basis function $\varphi_{k,l}[m, n]$ over \mathcal{L} yields the DFT $X[k, l]$ of $x[m, n]$ at frequency bin (k, l)

$$X[k, l] = \sum_{(m,n) \in \mathcal{L}} x[m, n] \varphi_{k,l}^*[m, n]. \quad (3.41)$$

Obviously, the expansion coefficient in the approximation becomes the DFT coefficient according to (3.41) and (3.32) except for a constant

$$G^{(\nu)}[k, l] = \begin{cases} c_{k,l}^{(\nu)}, & (k, l) \in \mathcal{M} \\ \frac{1}{2}c_{k,l}^{(\nu)}, & \text{otherwise} \end{cases}. \quad (3.42)$$

In the frequency domain, the multiplication of the weighting function with the complex exponential $\varphi_{u,v}[m, n]$ is equivalent to a shift of its conjugate complex DFT by u and v

$$\sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{u,v}[m, n] \varphi_{k,l}[m, n] = W^*[k + u, l + v].$$

Hence, (3.38) can be expressed in the frequency domain according to

$$\Delta c W[0, 0] + \Delta c^* W[2u, 2v] = 2MN R_w^{(\nu)}[u, v]. \quad (3.43)$$

Solving (3.43) and its conjugate complex equation for Δc yields

$$\Delta c = 2MN \frac{R_w^{(\nu)}[u, v] W[0, 0] - R_w^{*(\nu)}[u, v] W[2u, 2v]}{W[0, 0]^2 - |W[2u, 2v]|^2}. \quad (3.44)$$

Considering the discrete frequencies with a real-valued spectrum, for coefficients and basis functions with $(k, l) \in \mathcal{M}$ the following holds

$$\begin{aligned} \Delta c &= \Delta c^* \\ \varphi_{k,l}[m, n] &= \varphi_{k,l}^*[m, n] \end{aligned} \quad (3.45)$$

which can be used to simplify (3.43). Hence, a combination of the results leads to the following expression for the unknown Δc

$$\Delta c = \begin{cases} MN \frac{R_w^{(\nu)}[u, v]}{W[0, 0]}, & (u, v) \in \mathcal{M} \\ 2MN \frac{R_w^{(\nu)}[u, v] W[0, 0] - R_w^{*(\nu)}[u, v] W[2u, 2v]}{W[0, 0]^2 - |W[2u, 2v]|^2}, & \text{otherwise} \end{cases} \quad (3.46)$$

with \mathcal{M} defined in (3.31). A conjugate complex solution is obtained for Δc^* . The expansion coefficient and its conjugate complex counterpart have to be updated by Δc and Δc^* , respectively, according to (3.39).

In order to select the optimum basis function, the residual error energy (3.40) is expressed in the frequency domain as

$$\Delta E_{\mathcal{A}}^{(\nu+1)} = \frac{1}{2(MN)^2} (|\Delta c|^2 W[0, 0] + \text{Re}\{\Delta c^2 W^*[2u, 2v]\}).$$

Taking the simplification properties for real-valued coefficients (3.45) into account, the insertion of Δc according to (3.46) leads to the selection of the basis function with index (u, v) which maximizes

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \begin{cases} 2 \frac{R_w^{(\nu)}[k, l]^2}{W[0, 0]}, & (k, l) \in \mathcal{M} \\ 2 \frac{|R_w^{(\nu)}[k, l]|^2 W[0, 0] - \text{Re}\{R_w^{(\nu)}[k, l]^2 W^*[2k, 2l]\}}{W[0, 0]^2 - |W[2k, 2l]|^2}, & \text{otherwise} \end{cases} \\ \Rightarrow (u, v) &= \underset{(k, l)}{\text{argmax}} \Delta E_{\mathcal{A}}^{(\nu+1)}. \end{aligned} \quad (3.47)$$

Since the DFT coefficients are real-valued for $(k, l) \in \mathcal{M}$, the magnitude is not necessary for the energy calculation. In case of $(k, l) \notin \mathcal{M}$, the energy decrease is calculated for the discrete frequency (k, l) and its conjugate complex frequency. However, for $(k, l) \in \mathcal{M}$ the energy decrease would be only calculated for the considered frequency. Thus, the energy decrease is doubled in (3.47) for $(k, l) \in \mathcal{M}$. In fact, we do not consider the total maximum energy decrease but the energy decrease for the discrete frequencies (k, l) and $(M - k, N - l)$.

Due to the symmetry properties of the coefficients, the search area is limited as already mentioned to the shaded half-plane in Fig. 3.4.

If the termination criterion is not reached yet, the DFT spectrum of the weighted residual error (3.18) is calculated by

$$R_w^{(\nu+1)}[k, l] = R_w^{(\nu)}[k, l] - \frac{1}{2MN} (\Delta c W[k - u, l - v] + \Delta c^* W^*[k - u, l - v]). \quad (3.48)$$

Due to the symmetry properties only the DFT spectrum of the residual corresponding to the shaded area in Fig. 3.4 has to be updated in each step.

3.4.3 Summary of Frequency Selective Extrapolation

In this section, we clarify the principle of frequency selective extrapolation and demonstrate it by means of a representative example.

The extrapolation algorithm is summarized by the flow graph in Fig. 3.5. In the beginning, the algorithm is initialized with $g^{(0)}[m, n] = 0$, therefore the residual signal equals the weighted original signal in the first iteration

$$r_w^{(0)}[m, n] = w[m, n]f[m, n].$$

Since all equations are expressed in the frequency domain, there is only one DFT transform of $r_w^{(0)}[m, n]$ and $w[m, n]$ required in the beginning. The algorithm works in the DFT domain until the approximation accuracy is sufficient. The signal in the support area is approximated successively by computing one expansion coefficient per iteration and its conjugate complex coefficient if applicable. First, that basis function $\varphi_{u,v}[m, n]$ is selected which maximizes the decrease of the residual error energy. Then the respective coefficient $c_{u,v}^{(\nu)}$ is computed by minimizing the residual error energy. Subsequently, the residual error signal in the support area is computed and further approximated by the next, selected coefficient. If $\Delta E_A^{(\nu+1)}$ drops below a predefined threshold E_{\min} (3.23), the algorithm terminates and the parametric model is given in the entire area by the inverse DFT of the DFT-domain parametric model $G^{(\nu)}[k, l]$

$$g^{(\nu)}[m, n] = \text{IDFT}_{M,N}\{G^{(\nu)}[k, l]\}. \quad (3.49)$$

The unknown samples are replaced by the corresponding area of the parametric model $g[m, n]$. Fast Fourier Transform (FFT) algorithms realize efficiently the DFT which are especially efficient if the transformation length is a power of two [13]. This can be achieved by zero padding of the signal to the desired transformation length.

In the following, the principle of frequency selective extrapolation is illustrated by means of an example. We do not go into detail as parameters or implementational issues but rather demonstrate the extrapolation procedure. As typical application example we

choose concealment of isolated block losses. In Fig. 3.6 top the image Baboon with losses of size 16×16 pixels is shown. The missing block and the adjacent 13 pixels in each direction are transformed by the next larger FFT size of 64×64 to the frequency domain. As weighting function $w[m, n]$ we apply a radial-symmetric function in the support area, the amplitudes of which decay with distance

$$w[m, n] = \begin{cases} 0.74 \sqrt{(m - \frac{M-1}{2})^2 + (n - \frac{N-1}{2})^2} & , \quad (m, n) \in \mathcal{A} \\ 0 & , \quad (m, n) \in \mathcal{B} \end{cases}$$

The explanation why this model is especially suited for this application as well as the parameter selection is discussed in detail in Section 3.5. The result with the concealed losses is shown in the corresponding figure on the bottom.

In each iteration, one DFT coefficient is updated. Fig. 3.7 (a) shows a magnified part of the original image which should be restored by frequency selective extrapolation. The progress of the parametric model with increasing number of iterations is demonstrated by Fig. 3.7 (b) to (l). In the first iteration, the DC component is chosen. Then, subsequently AC frequencies are selected with respect to the image content. The average number of iterations per block depends on the image content. Smooth areas require only a few iterations in contrast to detailed areas such as the fur of the Baboon. For example, in Fig. 3.7 (c) the direction of the structure of the fur is already noticeable but has subsequently to be refined by more iterations in order to achieve a natural appearance. The algorithm terminates if either the residual error energy drops below the threshold $\Delta_{\min} = 15$ or 11 iterations are reached because a further improvement for the extrapolated area can not be expected. 8 iterations are done on average per block, i.e. the image content is sufficiently described by 8 DFT coefficients on average.

The demonstration shows the strength of the algorithm applied to extrapolation problems in image processing which is the ability to extrapolate all kinds of different textures: smooth areas, edges as well as detailed areas as noise like structures or patterns.

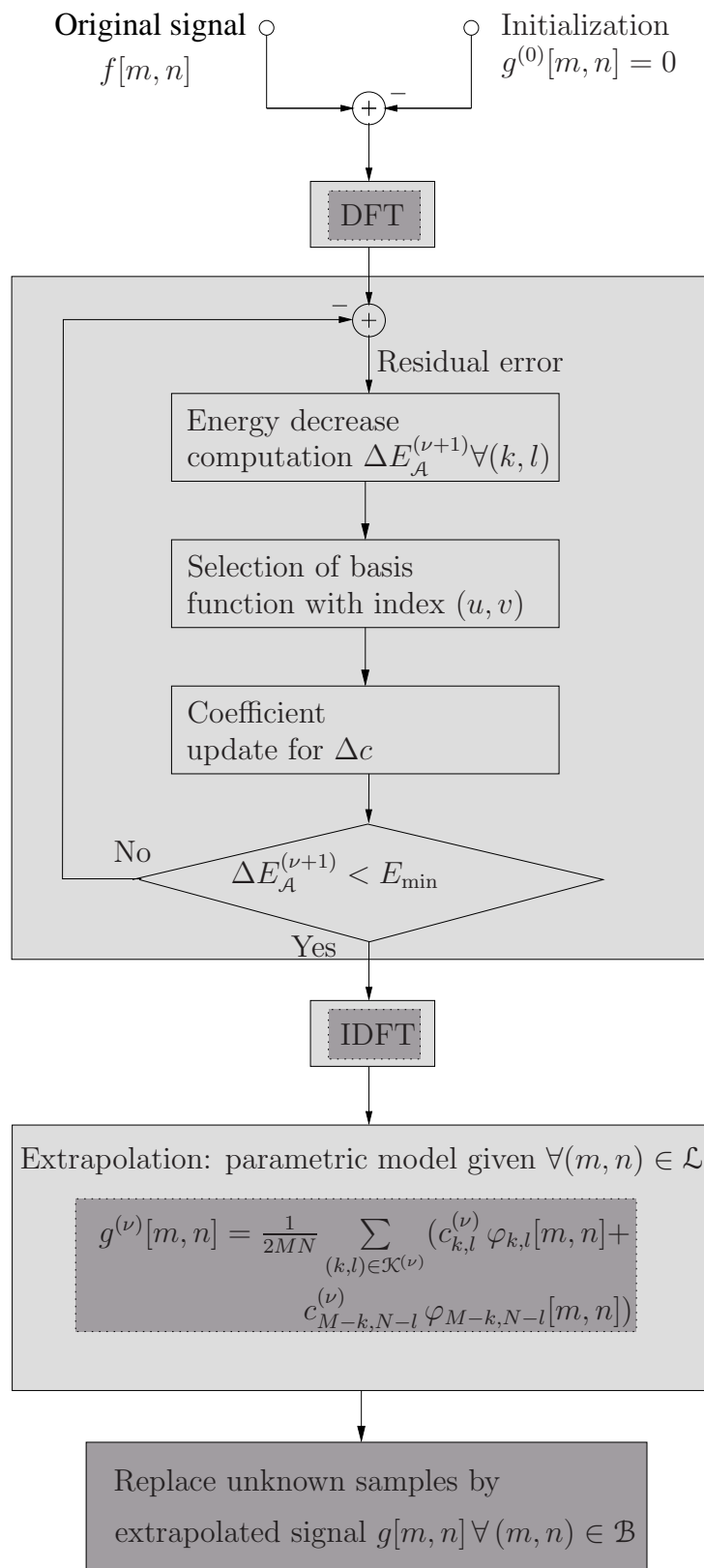


Figure 3.5: Flow graph describing the principle of frequency selective extrapolation.

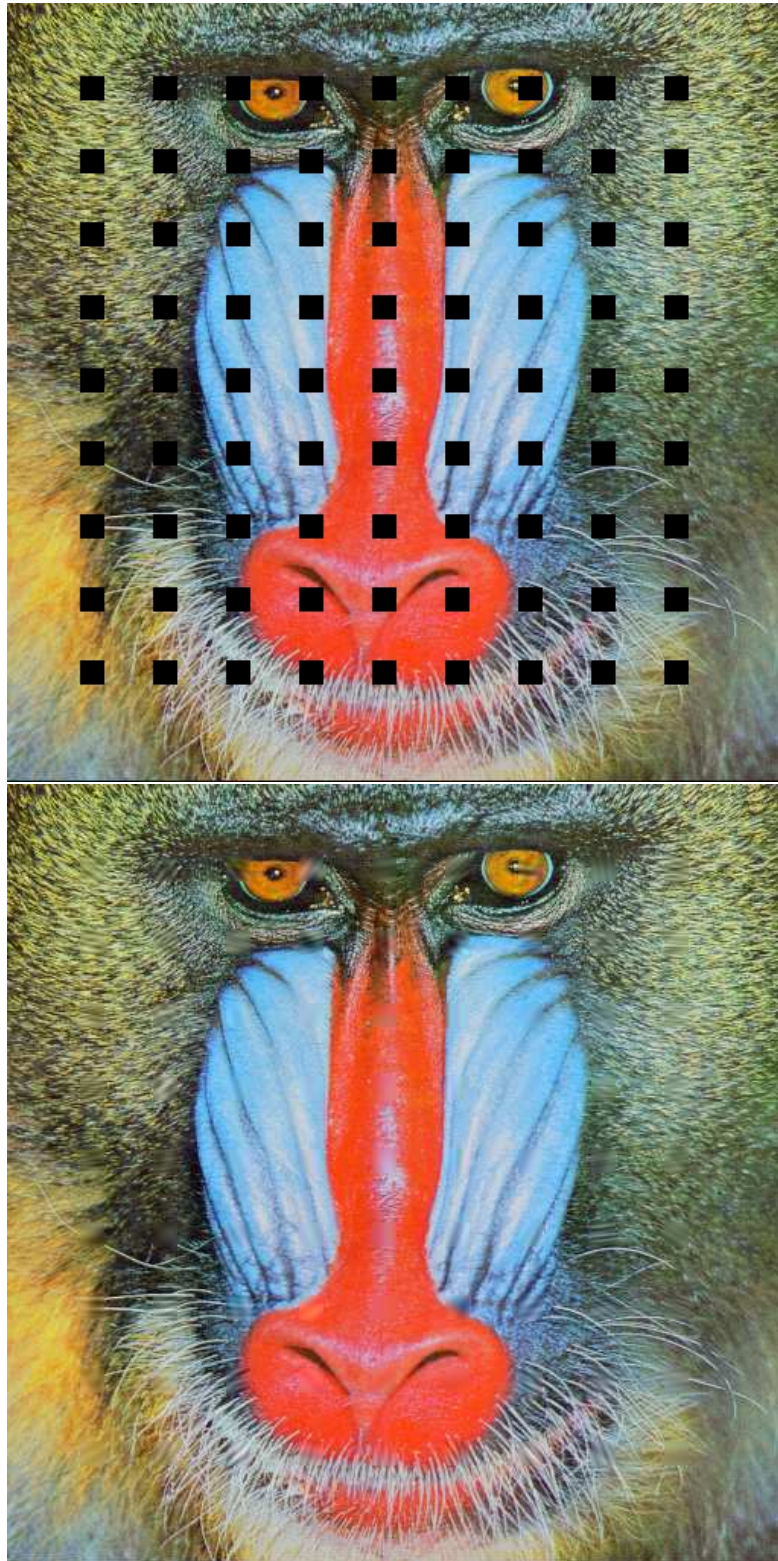


Figure 3.6: Top: Baboon with losses of 16×16 pixels. Bottom: Unknown pixels are replaced by frequency selective extrapolation with 8 iterations on average per block.

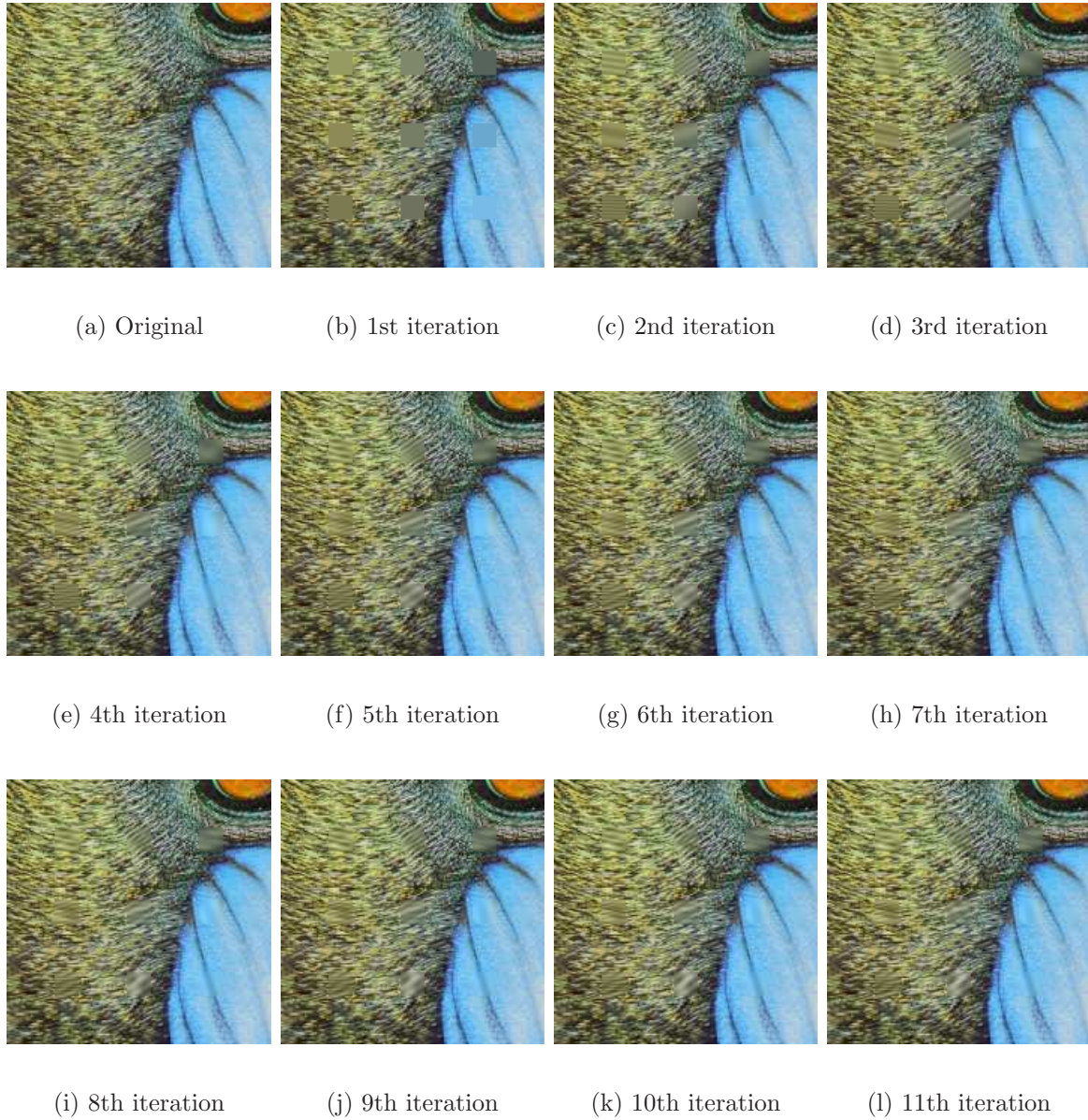


Figure 3.7: Progress of parametric model with increasing number of iterations. (a) Original. (b)-(l) Iteration 1 to 11.

3.5 2-D Weighting Function

Introducing the non-binary weighting function [34] allows for an application-specific and signal-dependent adaptation of the error criterion (3.2). Hence, the right choice of an appropriate weighting function $w[m, n]$ as described in Section 3.1.1 is of great importance. In the following, the effect of such an application-specific weighting function on the extrapolation result is discussed by means of a representative example. The applications considered in this thesis are characterized by a connected unknown area which is centrally located within the considered block and surrounded by known data. Therefore, the extrapolation of the surrounding signal into a quadratic unknown area serves as representative example. As size of the unknown block we choose 16×16 pixels which corresponds to isolated block losses typically appearing in erroneous image and video transmission. However, this scenario of error concealment in image and video communication is discussed in detail in Chapter 5.

First, results are shown for the non-adaptable binary weighting function in Section 3.5.1 as proposed in [35]. Investigations have been done for different adaptable

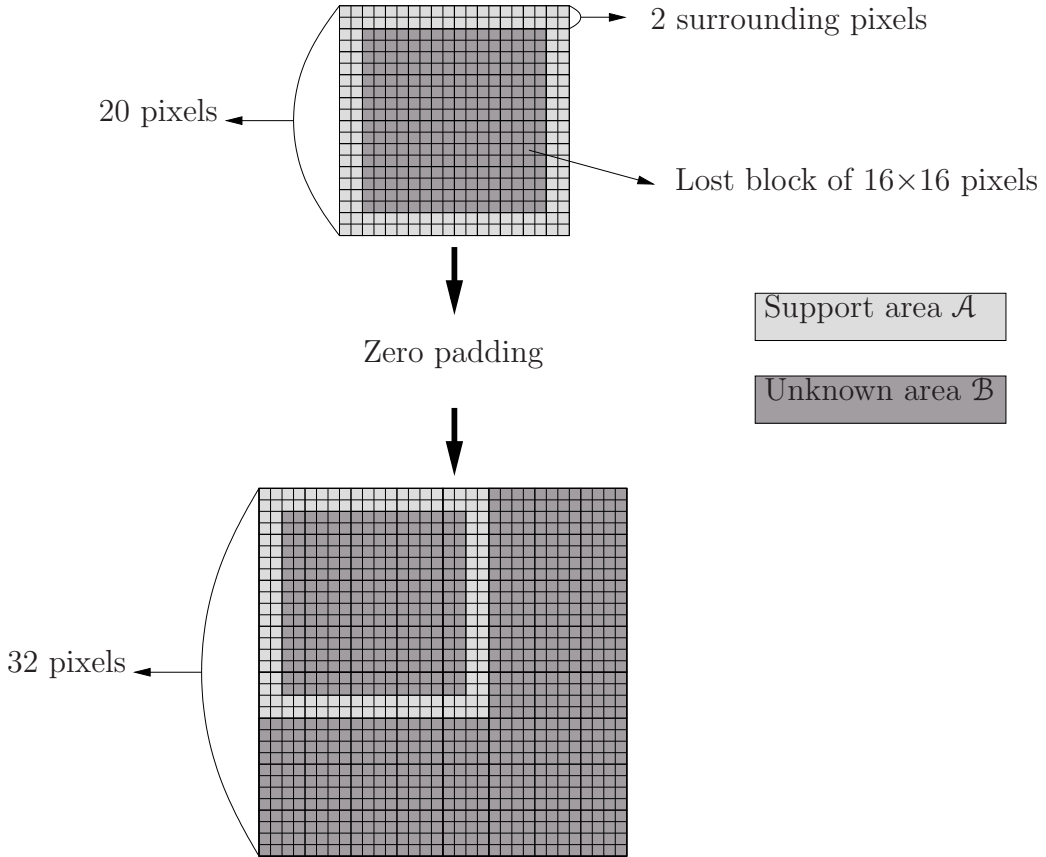


Figure 3.8: The missing block of 16×16 pixels and a number of known surrounding pixels form a block which is zero-padded to the transformation length of the FFT.

weighting functions. Only marginal improvements compared to the rectangular binary function could be achieved for rectangular non-binary functions where the amplitudes decay along the m -axis and along the n -axis with distance to the unknown block and the loci of constant amplitudes are therefore rectangular frames surrounding the unknown block. In contrast, improvements could be achieved for functions which do not consist of rectangular frames surrounding the unknown block anymore. Different Auto Regressive (AR) models for the weighting function as the isotropic, the separable, and a generalized model were investigated. For these models, the loci of constant amplitudes are ellipses or circles, i.e., the diagonals are not overrated anymore. Since the isotropic model according to [34] has shown the best performance, it is introduced in Section 3.5.2 as suitable model to be used for the weighting function in the considered extrapolation context. The effect on the extrapolation result and the improvement compared to the binary function is presented in Section 3.5.3.

Block losses of 16×16 pixels are inserted into different images for the simulations. For the realization of frequency selective extrapolation as derived in Section 3.4.2, the missing block of 16×16 pixels and a surrounding of known pixels form a block which is zero-padded to the transformation length of the DFT. The DFT can be accomplished by a suitable FFT implementation. Especially efficient algorithms are available if the transformation length is a power of two. An example is given in Fig. 3.8 where the missing block of 16×16 is surrounded by two known pixels forming a block of 20×20 pixels. This block is zero-padded and a 2-D DFT of size 32×32 is performed. Note that also the zero-padded area belongs to the unknown area where the weighting function equals zero, but that only the missing 16×16 pixels are replaced.

The simulated losses are applied to the three different test images *Lena*, *Peppers*, and *Baboon* which are depicted in Fig. 3.9 with the corresponding original images. The images have a size of 512×512 pixels and are given in RGB format. Since the frequency ranges of the distinct R, G, and B channels overlap [42], color artifacts appear in the concealed image when the algorithm of frequency selective extrapolation is applied to each channel separately. Thus, the images are first converted to the YUV format and then the concealment is performed with respect to each of the YUV channels separately.

The performance of concealment is defined in terms of Peak Signal to Noise Ratio (PSNR) in dB [42] which is given for one missing block of an 8-bit image by

$$\text{PSNR} = 10 \log_{10} \frac{255^2 B}{\sum_{(m,n) \in \mathcal{B}} (f[m,n] - g[m,n])^2} \quad (3.50)$$

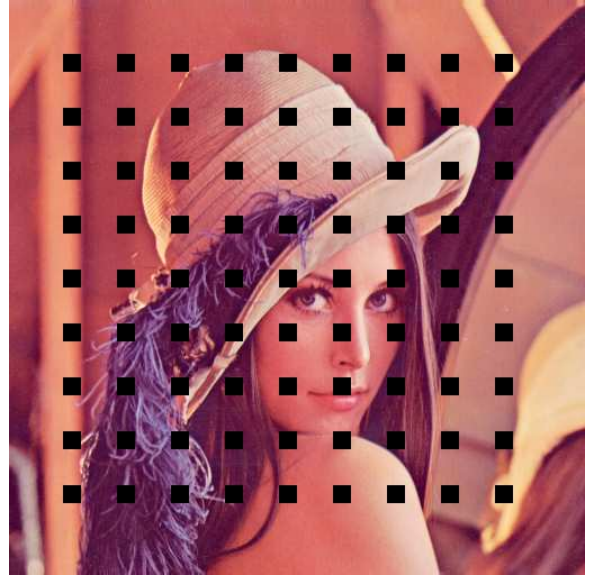
with B denoting the number of samples in the area to be replaced \mathcal{B} . For the evaluations in this section uncoded images are used. Thus, the PSNR is measured only at the concealed parts. The PSNR given in the evaluations is the PSNR of the luminance component.

Different parameters influence the quality of the frequency selective extrapolation and

have to be adjusted with respect to the considered application. To give an example, the extent of the support area is an important parameter, because artifacts are produced if it is chosen too large and the signal in the unknown area has no strong correlation to the signal in the support area any more. On the other hand, the support area should not be chosen too small because then it is not possible to restore details. Besides signal dependent factors, spectral factors of the weighting function play an important role. Furthermore, we investigate the influence of the DFT size on the extrapolation performance.



(a) *Lena*, original



(b) *Lena*, block losses of 16×16 pixels



(c) *Peppers*, original



(d) *Peppers*, block losses of 16×16 pixels

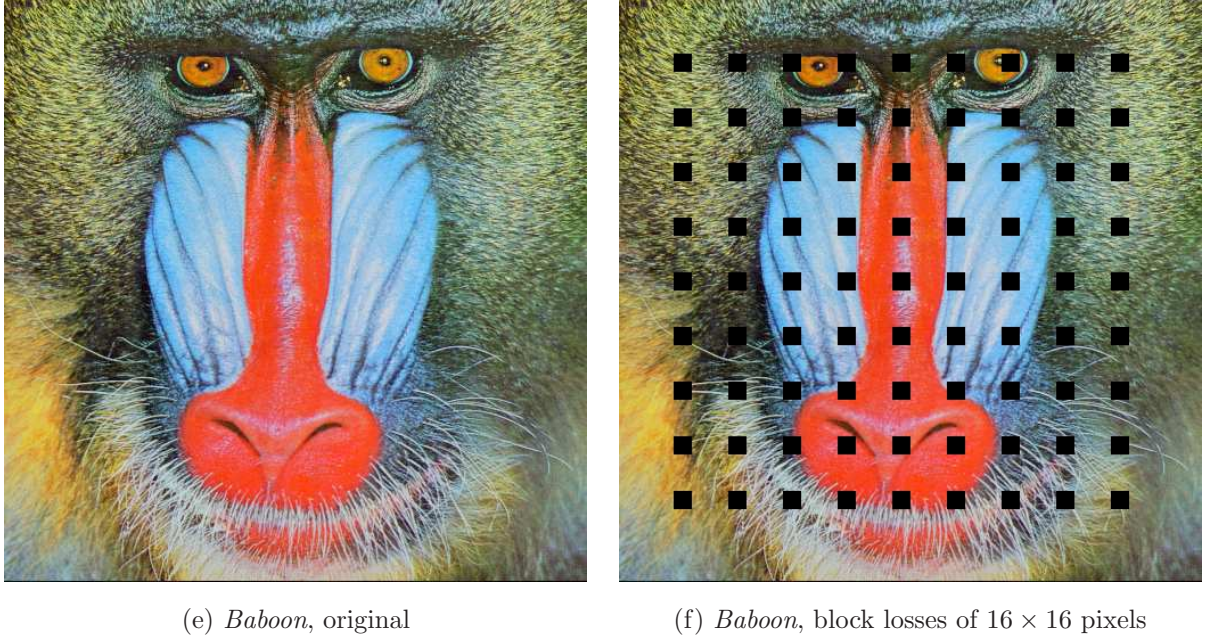


Figure 3.9: The original test images together with their corresponding corrupted images for isolated 16×16 block losses.

3.5.1 Extrapolation Result for Rectangular Binary Weighting Function

In the following of this section, we consider the case of a rectangular binary weighting function given by:

$$w[m, n] = b[m, n] = \begin{cases} 1 & , \quad (m, n) \in \mathcal{A} \\ 0 & , \quad (m, n) \in \mathcal{B}. \end{cases} \quad (3.51)$$

The window function $b[m, n]$ is introduced in (2.1).

In Fig. 3.10 PSNR results obtained for estimating isolated 16×16 losses using the rectangular binary weighting function are presented for the three different test images Fig. 3.9 (b), (d), (f). The performance is shown with respect to the number of iterations for different sizes of the support area expressed by the number of known surrounding pixels. In all simulations, the DFT size is chosen to 128×128 .

According to Fig. 3.10, in the first iteration the best solution is achieved if the number of known pixels is one, i.e., only the adjacent pixels are taken into account. For the luminance component, the DC component is commonly chosen which can be extrapolated best possibly from the direct surrounding. However, in order to restore details, both, a larger support area and more iterations are necessary.

First, we evaluate the impact of the size of the support area. For the image *Lena*, the best results are obtained for a small number of known pixels because the image content

often changes abruptly. If there are details in the support area which do not belong to the content to be estimated, the performance decreases. Therefore, using only one known pixel surrounding achieves the best result. However, a larger number of known pixels achieves better results for the image *Peppers* due to the clear structures and monotonous areas. The best result is provided by 6 known surrounding pixels. Usually, the image content of the missing block becomes more uncorrelated to the image content of the surrounding as the distance from the missing block increases. Therefore, the performance decreases again for a larger number of known surrounding pixels. The fur of the *Baboon* shows a noise-like frequency behavior. In this case, a larger number of 12 known pixels is necessary for the reconstruction of distinct details.

The more accurate image details should be restored, the more iterations are required. On the other hand, even if the approximation in the support area becomes better with more iterations, the performance of the extrapolation in the missing area can decrease. If too many iterations are done, the DFT basis images might become visible depending on the image content. This is indicated by Fig. 3.10 in case of the image *Lena* where only 2 iterations at 1 pixel surrounding yield the best result. For such a small number of surrounding pixels containing only a small part of the signal, hardly any details can be restored. The performance increases with an increasing number of iterations for the image *Peppers* until a saturation point of approximately 7 iterations is reached. The performance decreases again for more iterations for a small number of surrounding pixels. In case of the image *Baboon* the performance also increases first and decreases again if a certain number of iterations is exceeded. Depending on the number of surrounding pixels, this point is reached sooner for a small and later for a large number.

The results shown in Fig. 3.10 indicate that iterating beyond a required maximum number of iterations does not improve the visual impression anymore. Therefore, the maximum number of iterations is limited. Smooth areas require less iterations than detailed areas as can be observed from the illustrating example in Fig. 3.7, where the progress of the parametric model is shown with respect to the number of iterations. For the result of e.g. the second iteration depicted in Fig. 3.7 (c), it can be observed that the smooth areas within the cheek are restored but the fur of the Baboon still needs improvement for a good visual quality. Therefore, the error energy decrease per iteration represents a suitable termination criterion when dropping below the threshold Δ_{\min} (3.23). The actual value of Δ_{\min} is chosen as a compromise between concealment quality and the number of required iterations, i.e., the computational complexity.

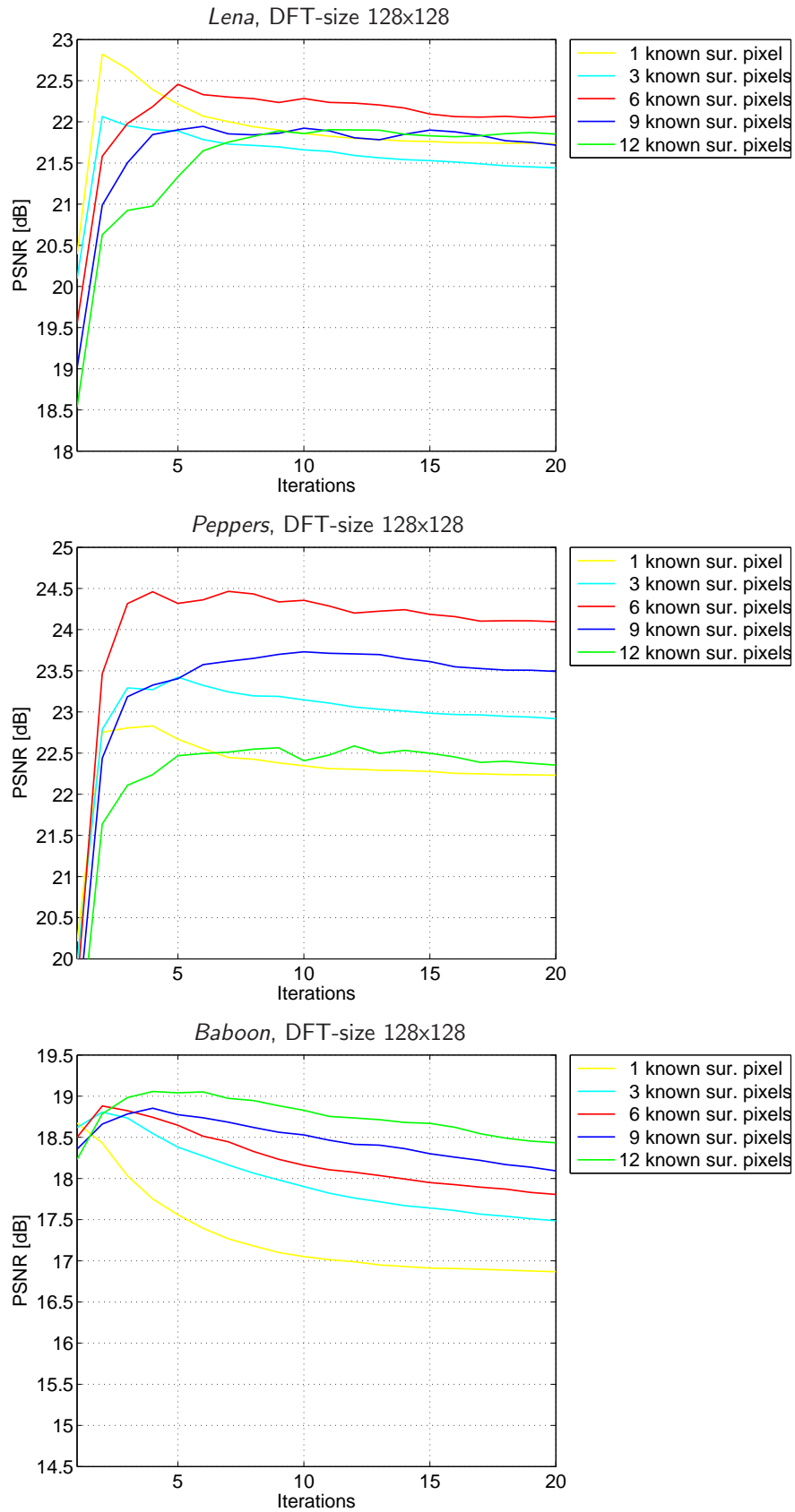


Figure 3.10: PSNR results for **binary weighting function** and 16×16 losses at a DFT size of 128×128 .



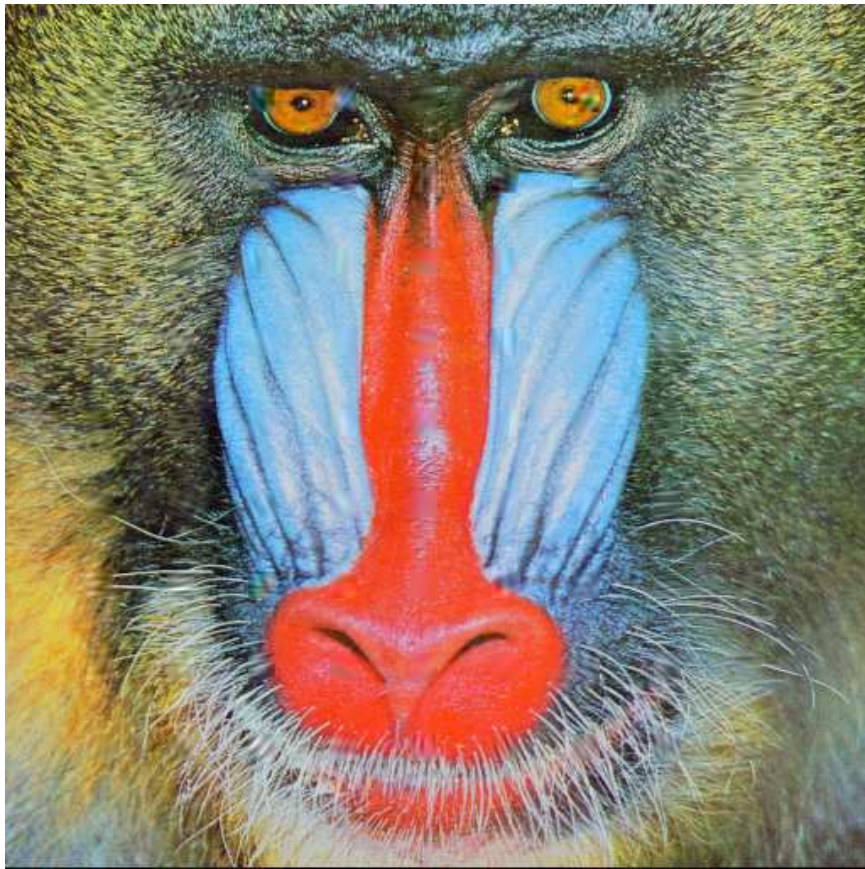


Figure 3.11: Concealed images using **binary weighting function** at DFT size 128×128 . *Lena*: 1 surrounding pixel, maximally 3 iterations; 22.6 dB at 3 iterations on average per block. *Peppers*: 6 surrounding pixel, maximally 7 iterations; 24.5 dB at 5.7 iterations on average per block. *Baboon*: 12 surrounding pixel, maximally 7 iterations; 19.0 dB at 6.9 iterations on average per block.

The visual results of concealing the losses shown in Fig. 3.9 are displayed in Fig. 3.11. For all images, a DFT of 128×128 is applied. The termination criterion is chosen such that the concealment quality is not affected but the computational cost reduced with $\Delta_{\min} = 6$. For the image *Lena*, only one known pixel at maximally three iterations is chosen yielding 22.7 dB. One more iteration to the optimum parameter is allowed in order to be able to restore a few more details. However, the DFT basis image effect is already visible, e.g., it can be noticed for the restored blocks at the hat. The concealed image *Peppers* is generated using 6 known surrounding pixels at maximally 7 iterations achieving 24.5 dB. Also maximally 7 iterations are done for the image *Baboon* with 19.0 dB. The combination with a large number of 12 surrounding pixels allows to even imitate the structure of the fur. However, artifacts occur at the transition of the nose to the cheek and color artifacts occur at the eyes.

3.5.2 Isotropic Weighting Function

Introducing the non-binary weighting function as proposed in [34] allows for an application-specific adaptation of the error criterion. As already mentioned in Section 3.1.1, the weighting function (3.1) should emphasize areas which are more important for the extrapolation over less important ones. In the considered extrapolation context characterized by a centrally located, connected unknown area surrounded by known data, it is reasonable to assume that the closer a known pixel is to the unknown area, the more important it is for the extrapolation. Therefore, the 2-D isotropic function [34]

$$\rho_{\text{isotrop}}[m, n] = \hat{\rho} \sqrt{\left(m - \frac{M-1}{2}\right)^2 + \left(n - \frac{N-1}{2}\right)^2}, \quad \hat{\rho} \leq 1 \quad (3.52)$$

represents a suitable model, where $\hat{\rho}$ is a predefined constant which determines the radial-symmetric decay of $\rho_{\text{isotrop}}[m, n]$ with increasing distance from the center at $(\frac{M-1}{2}, \frac{N-1}{2})$. The loci of constant amplitudes are circles as depicted in Fig. 3.12. This means for the weighting function

$$w[m, n] = \rho[m, n]b[m, n]$$

that the importance of a pixel decays with its distance and the influence of a pixel is stronger, the closer it is to the missing area.

However, the weighting function depends not only on $\hat{\rho}$ but also on the shape of the missing area given by the considered application. Fig. 3.13 depicts the resulting weighting function for the representative example of an unknown block of 16×16 pixels. It corresponds to a single block loss in error concealment where all surrounding blocks are available.

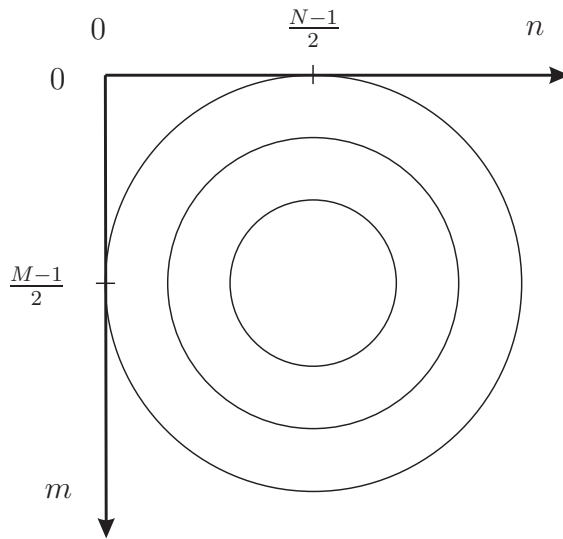


Figure 3.12: Isotropic model with circles as loci of constant amplitudes.

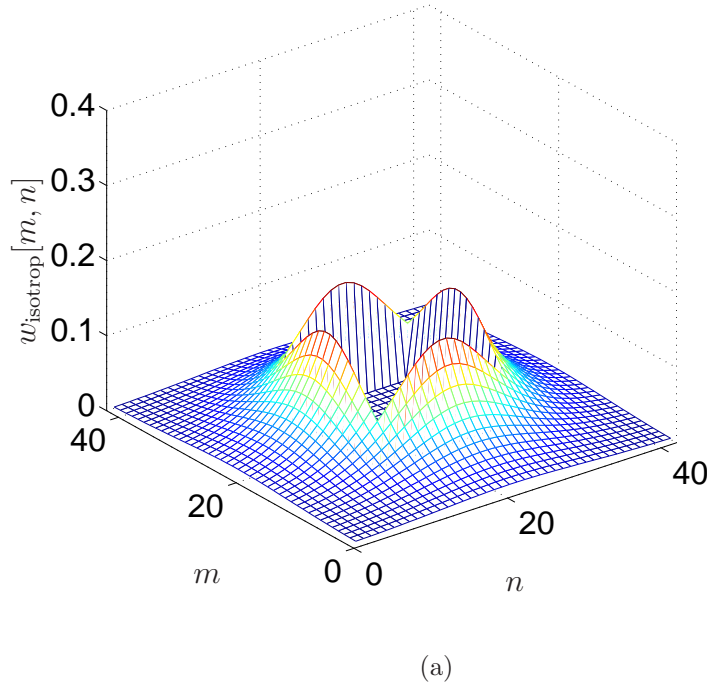


Figure 3.13: Weighting function using the isotropic model with $\hat{\rho}=0.8$ for an unknown block of size 16×16 (area \mathcal{B}). 13 surrounding pixels (support area \mathcal{A}) are used for reconstructing the unknown area.

3.5.3 Extrapolation Result for Isotropic Weighting Function

In the following, the effect on the extrapolation result of using the application-specific weighting function introduced in the last section instead of using the binary weighting function is shown. The performance applying the isotropic weighting function is influenced by the following parameters:

- The value of the decay parameter $\hat{\rho}$ as introduced in (3.52).
- The size of the support area expressed by the number of known surrounding pixels.
- The termination criteria: number of iterations and Δ_{\min} .
- The DFT size.

Decay Parameter $\hat{\rho}$

First, the performance is investigated with respect to $\hat{\rho}$ for the three test images *Lena*, *Peppers*, and *Baboon*. Furthermore, the improvement which can be achieved compared to the rectangular binary weighting function is evaluated. The corresponding results are depicted in Fig. 3.14.

PSNR is measured with respect to the number of iterations for varying $\hat{\rho}$ at 13 known surrounding pixels which is presented by the graphs in the left column of Fig. 3.14 for each test image. Further, the results as a function of the number of known surrounding pixels at 11 iterations are presented correspondingly in the right column. The size of the applied DFT-transform is 64×64 . For comparison, for each image the best results using the rectangular binary weighting function are displayed which are achieved by the parameters selected individually according to Fig. 3.10. Note that the DFT size for the binary weighting function is doubled in each dimension to 128×128 , because better results are achieved compared with a DFT size of 64×64 .

As can be noticed from the results for the image *Lena*, i.e. the two figures in the top row of Fig. 3.14, the best results can be obtained for $0.66 \leq \hat{\rho} \leq 0.74$. For the image *Peppers* according to the 2 graphs in the middle row of Fig. 3.14, slightly larger values, i.e. $0.74 \leq \hat{\rho} \leq 0.86$, achieve the best results. This is in accordance to the results for the binary weighting function where also a larger support area for *Peppers* compared to *Lena* yields the better results. In case of the image *Baboon*, the results are different as shown by the graphs in the bottom row of Fig. 3.14. Here, $0.7 \leq \hat{\rho} \leq 0.9$ leads to the best results but the range of the achieved PSNR values is very narrow. However, PSNR is not suited to evaluate the visual quality of noise-like textures as the fur of the *Baboon*. Only slight variations in phase result in large PSNR losses even if they are not visible. Therefore, the visual impressions are also discussed in Section 3.5.4, following the objective evaluations presented here.

In summary, the evaluations based on Fig. 3.14 show that the best results for each individual test image can be obtained for a narrow range of the value $\hat{\rho}$. Therefore, we choose $\hat{\rho} = 0.74$ for the further evaluations because it leads for all test images to very satisfying results with only marginal losses in PSNR compared to the optimum parameters.

Size of Support Area

In the right column of Fig. 3.14 the performance is evaluated with respect to the support area expressed by the number of known surrounding pixels. For all test images the results show the same behavior: the performance increases with an increasing number of known pixels. 12 to 16 pixels are a good choice if $\hat{\rho}$ is chosen appropriately. Therefore, 13 known surrounding pixels are chosen as parameter in the following.

The amplitude of the weighting function is attenuated almost to zero at 13 surrounding pixels as Fig. 3.13 shows. Note that the parameter $\hat{\rho}$ can be used to control the effective number of known surrounding pixels.

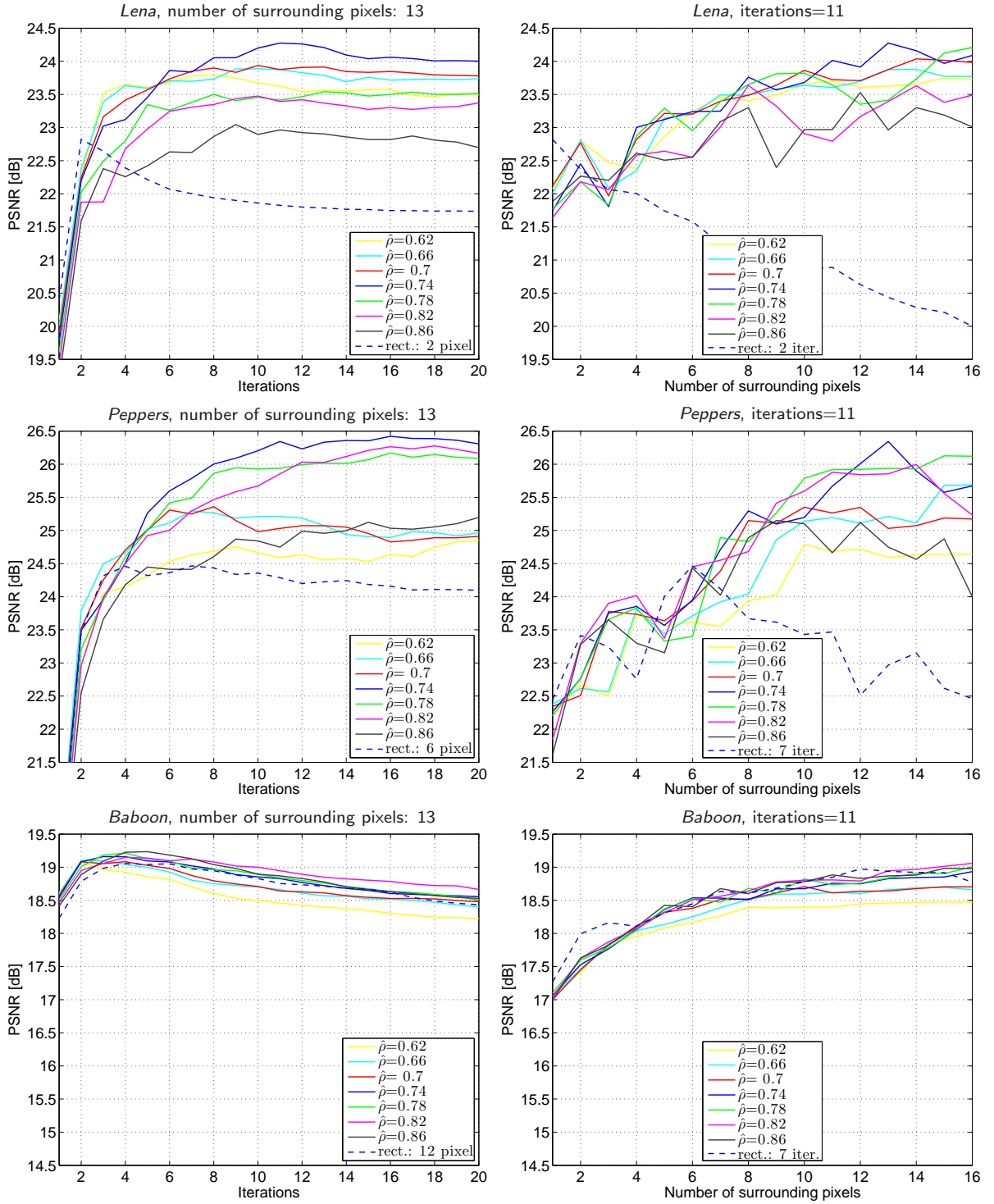


Figure 3.14: PSNR results for **isotropic weighting function** for isolated 16×16 block losses. For comparison, the results for the *rectangular binary* weighting function are also given. Isotropic weighting function: DFT size 64×64 . Rectangular binary weighting function: DFT size 128×128 .

Termination Criteria

First, the number of iterations is evaluated as termination criterion. Taking a look at the graphs in the left column of Fig. 3.14 shows that the performance generally increases with an increasing number of iterations until a saturation point is reached. Since we imitate the signal in the support area successively by a linear combination of DFT-like basis functions, detailed areas require more iterations. This has been already illustrated by the demonstration in Fig. 3.7 for part of the image *Baboon*. In the first iteration the DC component and then subsequently higher frequencies are selected. Hence, restoring the fur of the *Baboon* requires more iterations. However, for all images we can observe a saturation point where further iterations do not lead to an improvement anymore or even decrease the quality. This phenomenon has been already observed for the binary weighting function as shown in Fig. 3.10. For a small support area corresponding to only a small part of the signal, not many details can be restored, i.e. only a few iterations are appropriate. Hence, 11 iterations are chosen as allowed maximum number of iterations. For the image *Baboon* this means a slight decrease in terms of PSNR which, however, does not reflect the visual impression due to the reasons mentioned above.

The second termination criterion Δ_{\min} limits also the computational cost. For smooth areas it is not necessary to run 11 iterations because one or two might be already sufficient. Therefore, Δ_{\min} is set to 15 with no loss in quality in the considered cases.

DFT Size

Further, the influence of the DFT size on the performance is evaluated. The results for the DFT sizes 64×64 and 128×128 comparing the isotropic to the rectangular binary weighting function with respect to the number of iterations are presented in Fig. 3.15. In case of the isotropic weighting function, the parameters chosen ($\hat{\rho} = 0.74$, 13 surrounding pixel) are applied. For the binary weighting function the results are presented for 1, 6, and 12 surrounding pixels.

In case of the image *Lena*, the DFT size does not influence the result for the isotropic as well as the binary weighting function. For the binary weighting function only smooth areas and hardly any details can be restored because the image content is changing abruptly. Thus, the number of iterations and the number of surrounding pixels has to be kept low. In the first iteration the DC component is calculated which is not affected by the DFT size at all. Hence, for this image a larger DFT size does not improve the performance.

For the image *Peppers*, more details can be restored in general. Hence, more iterations and surrounding pixels are necessary. For the isotropic weighting, the differences between the results obtained with the different DFT sizes are again only marginal. Although the performance for the larger DFT size increases faster, the maximum achievable PSNR differs only in 0.1 dB. However, differences can be observed for the binary weighting

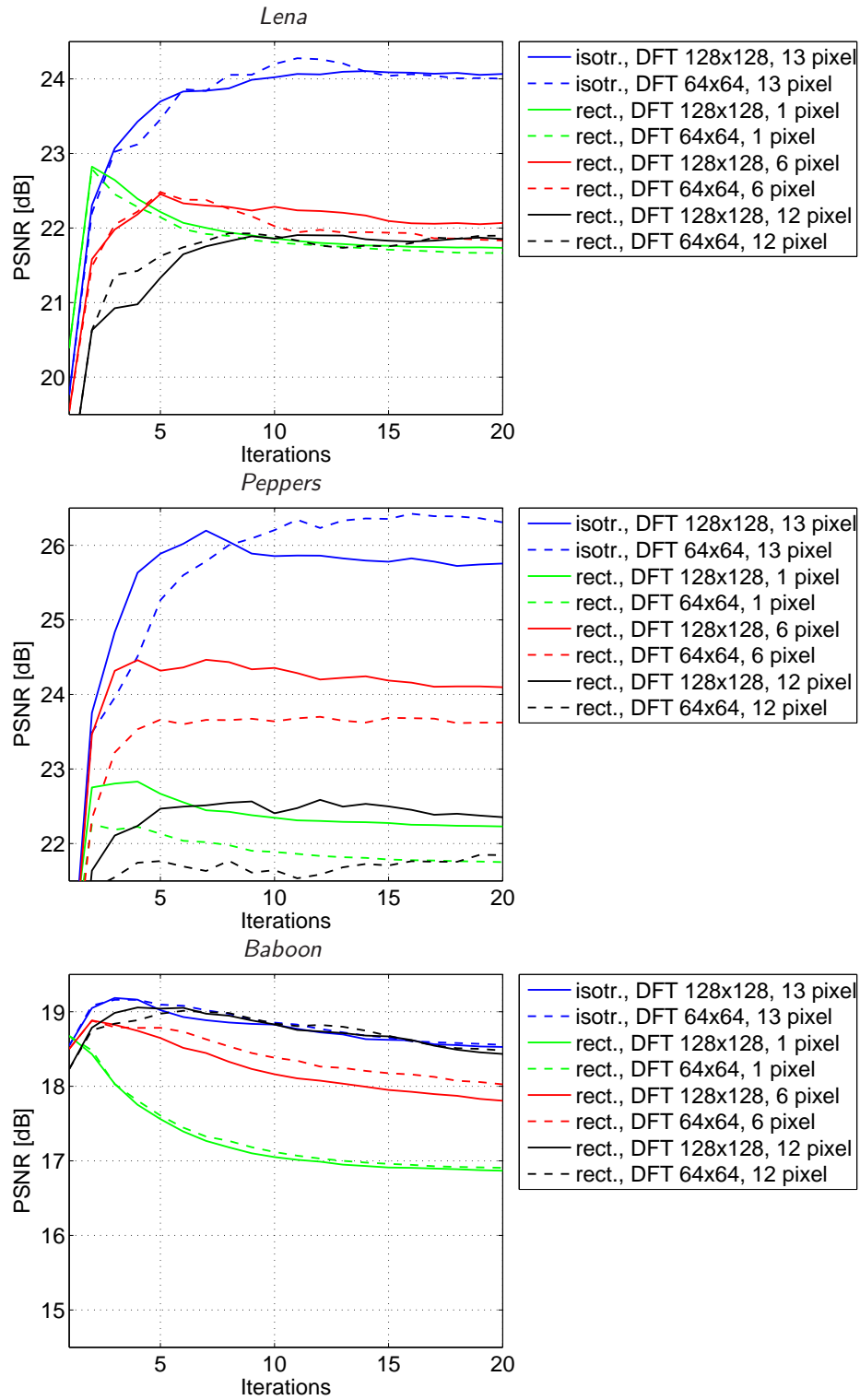


Figure 3.15: Evaluation of the DFT size for 16×16 losses comparing the isotropic weighting function (isotr.) to the rectangular binary weighting function (rect.).

function. For 6 known pixels and 7 iterations as chosen in Section 3.5.1, the difference in PSNR is even 0.8 dB when using the larger DFT size. The same behavior can be noticed for the other numbers of surrounding pixels.

In case of the image *Baboon*, the results for the different DFT sizes differ only marginally. However, as already mentioned, it is difficult to evaluate the performance without subjective results. They are provided in the next section.

In summary, in case of the isotropic weighting function no major differences between the DFT sizes 64×64 and 128×128 can be observed. Hence, the DFT size of 64×64 is applied. For the binary weighting function the larger size of 128×128 is chosen due to the gains for the image *Peppers*. Hence, the quality of the extrapolation result using the isotropic weighting function could therefore be improved significantly at smaller DFT sizes [34] compared to using the binary weighting function [35]. Using the isotropic function, only the next larger power of two to the size of the support area is necessary as DFT size and further extensions do not show improvements.

3.5.4 Discussion of Results

In the previous section, the effect of an application-specific weighting function on the extrapolation was investigated. Based on these evaluations, the achieved results are summarized in this section. Furthermore, subjective results are given.

The evaluations have shown that the 2-D isotropic model as specified in (3.52) represents a suitable model when used for the estimation of unknown blocks surrounded by known data. A fixed set of parameters can be specified for applying the isotropic weighting function which is important in practice. For all test images including various textures, this set achieved very satisfying results. For the estimation of blocks of size 16×16 pixels, the chosen parameters are summarized in Table 3.1.

In Table 3.2 the extrapolation results for concealing 16×16 unknown pixels are summarized. The comparison of the results for the isotropic weighting function with the optimum parameters to the fixed set of parameters shows that the loss in PSNR from 0.1 dB to 0.5 dB is indeed only marginal.

Table 3.2 shows further the objective gains achieved by using the isotropic instead of the binary weighting function. Comparing the corresponding results for the optimum parameter selection, gains up to 2.2 dB are obtained. For the image *Lena* 1.5 dB and for *Peppers* even 2.2 dB are achieved. In case of the image *Baboon*, 0.3 dB are reached but subjective results presented later confirm the significant improvement in perceived visual quality. Compared to using the binary weighting function, results achieved with the isotropic weighting function applying the fixed parameter set according to Table 3.1 yields still 1.4 dB improvement for *Lena* and 1.8 dB for *Peppers*.

It should be pointed out that in case of the binary weighting function it has not

Table 3.1: Parameters for estimating blocks of size 16×16 pixels using frequency selective extrapolation with isotropic weighting function (Section 3.5.3).

$\hat{\rho}$		0.74
no. of known pixel		13
Termination	max. no. of iterations	11
criterion	Δ_{\min}	15
DFT size		64×64

Table 3.2: PSNR results for estimating blocks of size 16×16 pixels measured in dB.

Weighting function	Isotropic	Isotropic	Rectangular binary
Parameter selection	Optimum	Set (Table 3.1)	Optimum
Lena	24.3 dB	24.2 dB	22.8 dB
Peppers	26.7 dB	26.3 dB	24.5 dB
Baboon	19.3 dB	18.8 dB	19.0 dB

been possible to find a single set of parameters that yields satisfying results for all test images. This implies that for each image the parameters have to be adjusted individually in advance. Furthermore, the size of the DFT is doubled in each dimension compared to using the isotropic function implying a higher computational complexity. The images with the concealed blocks obtained by using the isotropic weighting function are presented in Fig. 3.16. Note that all images are generated with the same set of parameters according to Table 3.1 at a DFT size of 64×64 . The results are very convincing and it is hard to observe any artifacts. The corresponding images obtained for the binary weighting function are shown in Fig. 3.11. Comparing 3.11 with Fig. 3.16, it can be noticed that for *Lena* the artifacts in the background and the blurring within the feathers as observed for the binary window are avoided by the isotropic weighting function. In Fig. 3.11 showing *Peppers*, the green artifact between the two green peppers in the foreground is also not present in Fig. 3.16. Also the color artifacts for the image *Baboon* and the artifacts of the block at the transition from the nose to the cheek are reduced. Hence, the objective results can be confirmed by subjective evaluations.



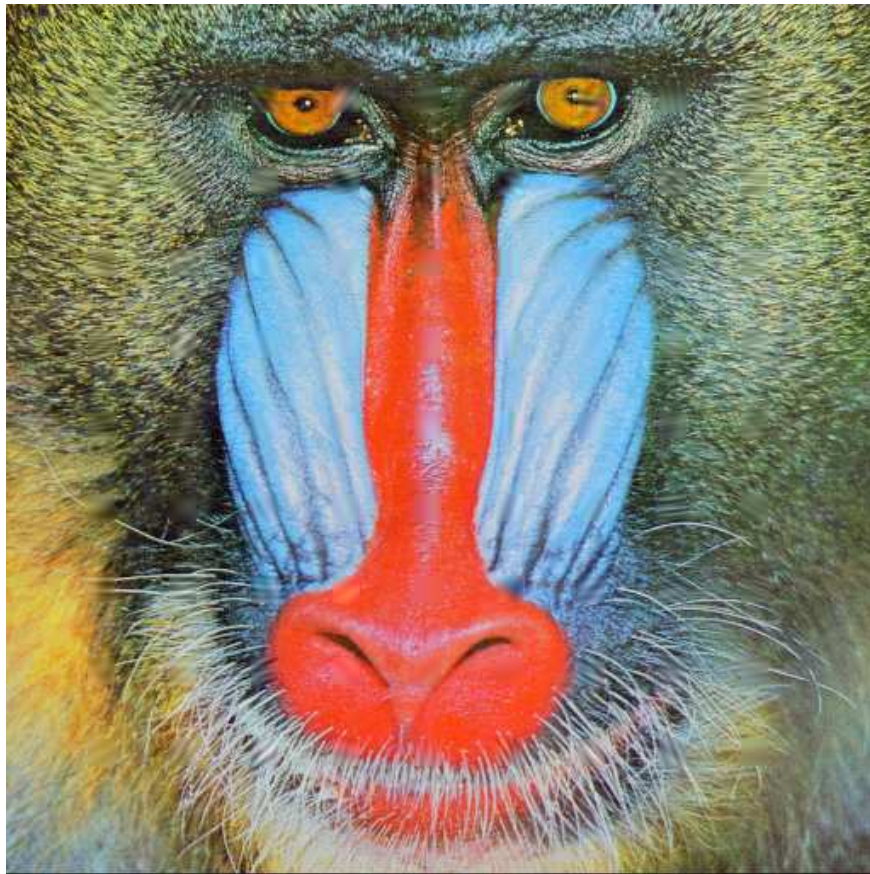


Figure 3.16: Concealed images using the **isotropic weighting function** at DFT size 64×64 with parameters according Table 3.1. *Lena*: 24.2 dB at 7.2 iterations on average per block. *Peppers*: 26.3 dB at 6.7 iterations on average per block. *Baboon*: 18.8 dB at 8.9 iterations on average per block.

In summary, the quality of the extrapolation result using frequency selective extrapolation could be improved significantly by applying an application-specific weighting function instead of a simple binary weighting function. The isotropic weighting function provides a suitable function when estimating unknown areas from the surrounding image data. Very convincing results are achieved for both, subjective and objective evaluations. A fixed set of parameters can be specified which allows to sufficiently restore different textures such as smooth areas, edges, and detailed areas. The computational complexity is determined by the image content, smooth areas require only a few, detailed areas more iterations. Only the next larger power of two as DFT size has to be applied in order to use a FFT.

3.6 Frequency Selective Extrapolation Interpreted as Spectral Deconvolution

In the following, we show that methods known under the name of spectral deconvolution [50, 12, 4] as already reviewed in Section 2.2.2 lead to a similar approach as selective extrapolation using DFT basis functions implemented in the frequency domain.

In this chapter we introduced the method of selective extrapolation for two dimensions. The derivation of the algorithm is carried out generically which implies that the method can be applied with any kind of basis functions. The choice of suitable basis functions depends on the underlying signal. For image and video signals, we proposed multi-dimensional DFT functions as basis functions and, thus, we referred to the approach as *frequency selective extrapolation*. It turned out that the entire algorithm can be implemented efficiently in the frequency domain. Using arbitrary basis functions does in general not lead to such a compact representation.

In the literature, there are methods known under the name of *spectral deconvolution* which is a special case of selective extrapolation if DFT functions are used as basis functions. The partitioning of the area \mathcal{L} in \mathcal{A} and \mathcal{B} can be interpreted as a windowing of the area \mathcal{L} . The multiplication of the window function with the original signal leads to a convolution of the respective spectra in the frequency domain. The fundamental idea is to remove the impact of this window function by spectral deconvolution. However, this method is limited to the DFT-domain only and does not allow for a more generic approach.

Deconvolution algorithms are in general applied to most different topics. Removing successively the impact of a window function by deconvolution is for example derived by Högbom [22] in radio astronomy. There, the imposed limitations of the recording radio telescopes on the measurements are eliminated for the aperture synthesis by deconvolution. The method is called CLEAN because the recorded “dirty map” is “cleaned” from the “dirty beam” formed by the recording telescopes. Here, the deconvolution takes place in the signal-domain. A completely different area where spectral deconvolution is applied represents the modeling of the human sense of hearing with high-resolution analysis by Sottek [50]. In image processing, spectral deconvolution allows a Fourier-based analysis of textures of non-rectangular image patches as described in [12] by Clark *et al.* A similar principle is applied by Aach in medical image processing for interpolating defect pixels of flat panel detectors with spectral deconvolution [4].

In the following, the approach of 2-D spectral deconvolution is derived based on the one-dimensional case presented in [50]. It is shown that the deconvolution algorithm is a special case of selective extrapolation. In case DFT basis functions are inserted, this results in the frequency selective extrapolation implemented in the frequency domain according to Section 3.4.2. The extension to 3-D is straightforward and therefore not

further discussed here.

The multiplication of the desired signal $f[m, n]$ by the window function $w[m, n]$ is denoted by $f_w[m, n]$ and corresponds to a convolution of the respective DFTs $F[k, l]$ and $W[k, l]$ resulting in $F_w[k, l]$ [42]

$$f_w[m, n] = f[m, n]w[m, n] \quad (3.53)$$

$$F_w[k, l] = \frac{1}{MN} F[k, l] * W[k, l] = \frac{1}{MN} \sum_{\kappa=0}^{M-1} \sum_{\lambda=0}^{N-1} F[\kappa, \lambda] W[k - \kappa, l - \lambda] \quad (3.54)$$

where “ $*$ ” denotes convolution.

The approach aims at estimating $F[k, l]$ by $G[k, l]$ with only D spectral samples

$$G[k, l] = \sum_{(\kappa, \lambda) \in \mathcal{K}_D} G[\kappa, \lambda] \delta[k - \kappa, l - \lambda] \quad (3.55)$$

with the set \mathcal{K}_D consisting of D index pairs. Thus, the DFT spectrum of the original signal $F[k, l] \bullet \circ f[m, n]$ for $0 \leq (m, k) \leq M - 1; 0 \leq (n, l) \leq N - 1$ is estimated by $G[k, l]$ which corresponds to the parametric model introduced in frequency selective extrapolation.

For our considerations, the signals $f[m, n]$ and $g[m, n]$ are real-valued. The estimated values $g[m, n]$ are determined by maximally reducing the mean squared error to the original signal weighted by $w[m, n]$

$$\begin{aligned} E_{\mathcal{A}} &= \sum_{(m, n) \in \mathcal{L}} w[m, n] (f[m, n] - g[m, n])^2 \\ &= \sum_{(m, n) \in \mathcal{L}} (f_w[m, n]f[m, n] - 2f_w[m, n]g[m, n] + w[m, n]g^2[m, n]). \end{aligned} \quad (3.56)$$

According to Parseval’s theorem [42]

$$\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n]y^*[m, n] = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X[k, l]Y^*[k, l]$$

the error energy can be expressed equivalently in the DFT-domain

$$\begin{aligned} E_{\mathcal{A}} &= \frac{1}{MN} \sum_{(k, l) \in \mathcal{K}_D} \left(F_w[k, l]F^*[k, l] - 2F_w[k, l]G^*[k, l] \right. \\ &\quad \left. + \left(\frac{1}{MN} W[k, l] * G[k, l] \right) G^*[k, l] \right). \end{aligned} \quad (3.57)$$

Differentiating the error criterion with respect to the DFT coefficients $G[k, l]$ and equating it to zero

$$\frac{\partial E_{\mathcal{A}}}{\partial G[k, l]} = \frac{\partial E_{\mathcal{A}}}{\partial G^*[k, l]} \stackrel{!}{=} 0; \quad \forall (k, l) \in \mathcal{K}_D \quad (3.58)$$

leads to the following system of equations

$$\frac{1}{MN}G[k, l] * W[k, l] = F_w[k, l]; \quad \forall (k, l) \in \mathcal{K}_D. \quad (3.59)$$

In summary, $G[k, l]$ has to be determined as follows: For L DFT-components $\in \mathcal{K}_L$, the convolution of $W[k, l]$ and the estimated values $G[k, l]$ has to equal the DFT of the weighted original signal $F_w[k, l]$ in order to yield the minimum approximation error.

The question considered next is which DFT-components should be selected in order to obtain a minimum error with respect to the signal. As error criterion we consider the deviation of the approximation error which depends on the estimation $G[k, l]$ having contributions only for all $(k, l) \in \mathcal{K}_D$

$$\Delta E_{\mathcal{A}} = \frac{1}{MN} \sum_{(k, l) \in \mathcal{K}_D} \left(2F_w[k, l]G^*[k, l] - \left(\frac{1}{MN}W[k, l] * G[k, l] \right) G^*[k, l] \right)$$

Substituting the convolution of $W[k, l]$ with $G[k, l]$ by (3.59), the maximum reduction of the error criterion can be expressed as

$$\Delta E_{\mathcal{A}} = \frac{1}{MN} \sum_{(k, l) \in \mathcal{K}_D} F_w^*[k, l]G[k, l]. \quad (3.60)$$

For an optimum selection of D DFT-components we have $\binom{L}{D}$ possibilities to choose from. Further, a unique solution is not assured or less than D spectral samples might be sufficient. To overcome these problems, an iterative selection strategy is applied by selecting one DFT-component with index (u, v) per iteration.

Iterative Spectral Deconvolution

If the DFT of the original signal consists only of one DFT-component corresponding to a real-valued harmonic described by index (u, v) , the estimate $G[k, l]$ according to (3.59) yields the true spectrum $F[k, l]$. The estimated signal $g[m, n]$ which is the inverse transformed DFT of $G[k, l]$ has also values in the unknown area, hence, the signal is extrapolated beyond the known area. Weighting the extrapolated signal $g[m, n]$ by $w[m, n]$ results then in $f_w[m, n]$.

However, if the original signal consists of more spectral components, the subtraction of the DFT of the weighted estimated signal $w[m, n]g[m, n]$ from the DFT of the weighted original signal $w[m, n]f[m, n]$ does not vanish, but yields the DFT of the weighted residual signal $R_w[k, l] \bullet \text{---} \circ r_w[m, n] = w[m, n]r[m, n]$. The spectral deconvolution is accomplished by an iterative approach as noted, hence the DFT of the weighted residual signal in iteration ν is composed of

$$R_w^{(\nu)}[k, l] = F_w[k, l] - \frac{1}{MN}G^{(\nu)}[k, l] * W[k, l]. \quad (3.61)$$

Thereby, in each iteration the impact of the estimated DFT component with index (u, v) on less dominant components is eliminated with $R_w^{(\nu)}[u, v] = 0$. The DFT of the residual is further approximated by eliminating successively the impact of dominant spectral components.

The algorithm is initialized with $G^{(0)}[k, l] = 0$ resulting in $R_w^{(0)}[k, l] = F_w[k, l]$. The update of the estimated DFT spectrum is described by $\Delta G[u, v]$

$$G^{(\nu+1)}[u, v] = G^{(\nu)}[u, v] + \Delta G[u, v]. \quad (3.62)$$

$\Delta G[u, v]$ equals the update of the DFT-domain parametric model in frequency selective extrapolation if $\Delta G[u, v] = \Delta c$. Hence, the DFT of the residual error $R_w^{(\nu+1)}[k, l]$ in the next iteration $\nu + 1$ can be expressed based on (3.61) as

$$R_w^{(\nu+1)}[k, l] = R_w^{(\nu)}[k, l] - \frac{1}{MN} \Delta G[u, v] * W[k, l]. \quad (3.63)$$

In order to determine $\Delta G[u, v]$, (3.59) has to be revised for the selected index pair (u, v) to

$$\frac{1}{MN} \Delta G[u, v] * W[u, v] = R_w^{(\nu)}[u, v]. \quad (3.64)$$

Applying the notation of (3.55), the convolution of the unit impulse with the DFT of the window function results in a shift by u and v

$$\Delta G[k, l] \delta[k - u, l - v] * W[k, l] = \Delta G[u, v] W[k - u, l - v]. \quad (3.65)$$

Hence, inserting (3.65) in (3.64) and evaluating it for (u, v) yields the unknown $\Delta G[u, v]$

$$\Delta G[u, v] = MN \frac{R_w^{(\nu)}[u, v]}{W[0, 0]}. \quad (3.66)$$

With $\Delta G[u, v]$, the reduction of the error criterion in each iteration depends only on (u, v) and, regarding (3.60), equals

$$\begin{aligned} \Delta \tilde{E}_A^{(\nu+1)} &= \frac{1}{MN} R_w^{*(\nu)}[u, v] \Delta G[u, v] \\ &= \frac{|R_w^{(\nu)}[u, v]|^2}{W[0, 0]}. \end{aligned} \quad (3.67)$$

So far, the estimation assumed the selection of a real-valued DFT-component for the approximation. But if the chosen index (u, v) belongs to a complex-valued DFT-component, i.e. if $(u, v) \notin \mathcal{M}$, a conjugate complex index pair is chosen as already described for the frequency selective extrapolation in order to yield a real-valued signal in the spatial domain. Hence, the following equation has to be considered

$$\begin{aligned} \frac{1}{MN} (\Delta G[k, l] \delta[k - u, l - v] * W[k, l] \\ + \Delta G[k, l] \delta[k - M + u, l - N + v] * W[k, l]) = R_w^{(\nu)}[k, l] \end{aligned}$$

which yields for the selected index (u, v)

$$\frac{1}{MN} (\Delta G[u, v] W[0, 0] + \Delta G^*[u, v] W[2u, 2v]) = R_w^{(\nu)}[u, v]$$

and a conjugate complex equation. They can be solved for the unknown coefficient $\Delta G[u, v]$

$$\Delta G[u, v] = \gamma MN \frac{R_w^{(\nu)}[u, v] W[0, 0] - R_w^{*(\nu)}[u, v] W[2u, 2v]}{W[0, 0]^2 - |W[2u, 2v]|^2}. \quad (3.68)$$

The clean gain factor γ lies in the range $0.1 < \gamma < 1.0$ and is introduced in [22, 12] based on the assumption that multiple components have contributed to the amplitude of the considered frequency bin due to the convolution. Therefore, the true value $\Delta G[u, v]$ is lower. Low values of γ improve the stability of the algorithm at the expense of requiring more iterations in order to fully extract the clean component. For a complex-valued component, the relation $\Delta G[u, v] = \frac{1}{2}\Delta c$ is already summarized in (3.42) for frequency selective extrapolation.

The DFT of the residual error in the next iteration is then given by

$$R_w^{(\nu+1)}[k, l] = R_w^{(\nu)}[k, l] - \frac{1}{MN} (\Delta G[u, v] W[k - u, l - v] + \Delta G^*[u, v] W[k + u, l + v]). \quad (3.69)$$

Based on (3.60), the error criterion is reduced in each iteration by

$$\begin{aligned} \Delta \tilde{E}_{\mathcal{A}}^{(\nu+1)} &= \frac{1}{MN} (R_w^{*(\nu)}[u, v] \Delta G[u, v] + R_w^{(\nu)}[u, v] \Delta G^*[u, v]) \\ &= 2 \frac{|R_w^{(\nu)}[u, v]|^2 W[0, 0] - \text{Re}\{R_w^{(\nu)}[u, v]^2 W^*[2u, 2v]\}}{W[0, 0]^2 - |W[2u, 2v]|^2}. \end{aligned} \quad (3.70)$$

After initializing with $G^{(0)} = 0$, the approach of selective deconvolution can be summarized by the following steps:

- Select index (u, v) at which error criterion is maximally reduced [50, 4]

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \begin{cases} 2 \Delta \tilde{E}_{\mathcal{A}}^{(\nu+1)}, & (k, l) \in \mathcal{M} \\ \Delta \tilde{E}_{\mathcal{A}}^{(\nu+1)}, & \text{otherwise} \end{cases} \\ \Rightarrow (u, v) &= \underset{(k, l)}{\text{argmax}} \Delta E_{\mathcal{A}}^{(\nu+1)} \end{aligned} \quad (3.71)$$

$\Delta \tilde{E}_{\mathcal{A}}^{(\nu+1)} \forall (k, l) \in \mathcal{M}$ has to be doubled for the same reasons as for frequency selective extrapolation, i.e., the energy decrease has to be calculated *per* frequency bin. Next, (u, v) and $(M - u, N - v)$ are included into $\mathcal{K}^{(\nu+1)}$ if they have not been selected yet. In [22, 12] a simplified selection criterion according to (3.67) is applied.

- Compute $\Delta G[u, v]$ according to (3.66) for a real-valued component and according to (3.68) for a complex-valued component. Then, $G^{(\nu+1)}[u, v] = G^{(\nu)}[u, v] + \Delta G[u, v]$ can be computed, where $G^{(\nu+1)}[M - u, N - v] = G^{(\nu+1)*}[u, v]$.

- Compose the estimated DFT-spectrum $G^{(\nu+1)}[k, l]$ by

$$G^{(\nu+1)}[k, l] = \sum_{(\kappa, \lambda) \in \mathcal{K}^{(\nu+1)}} G^{(\nu+1)}[\kappa, \lambda] \delta[k - \kappa, l - \lambda] \quad (3.72)$$

- Evaluate termination criterion according to (3.23):
 - $\Delta E_{\mathcal{A}}^{(\nu+1)} > E_{\min}$: Calculate the DFT of the residual error according to (3.61) using (3.65) for a real-valued component and (3.69) for a complex-valued component, respectively. Then, restart from the beginning.
 - $\Delta E_{\mathcal{A}}^{(\nu+1)} < E_{\min}$: The DFT-domain estimate is given after termination according to (3.72), yielding the estimated signal by an inverse DFT $g[m, n] \leftarrow \bullet G[k, l]$. Finally, the unknown samples are replaced by the extrapolated pixels cutted out of $g[m, n]$.

The fundamental idea and the extrapolation concept of frequency selective extrapolation and spectral deconvolution are summarized in Table 3.3. For basis functions based on the DFT, the derivations of frequency selective extrapolation result in corresponding equations as for spectral deconvolution, although the concepts behind the approaches differ.

The fundamental idea behind **spectral deconvolution** is that $G[k, l]$ is the estimate of the original spectrum $F[k, l]$ in terms of dominant frequencies successively obtained by spectral deconvolution of the DFT of the window function. The inverse DFT yields the estimated signal $g[m, n]$ with components in the entire signal area, accomplishing thereby the inherent signal extrapolation.

In contrast, the main concept behind **selective extrapolation** is the estimation of the original signal $f[m, n]$ by a parametric model $g[m, n]$ as a linear combination of basis functions which are defined in the entire signal area. The successive approximation of the support area in terms of selected weighted basis functions provides the extrapolation in the unknown area.

Table 3.3 shows that the equation for the update of the expansion coefficient Δc (3.46) in frequency selective extrapolation corresponds to the update of the estimated DFT component $\Delta G[u, v]$ expressed in (3.66) and (3.68). The selection criterion of the basis function in the next iteration (3.47) and the selection criterion of the DFT component (3.71) are equal for both approaches.

Table 3.3: Comparison of frequency selective extrapolation to spectral deconvolution.

Frequency Selective Extrapolation	Spectral Deconvolution
Fundamental Idea	
$g[m, n]$ estimates original $f[m, n]$ by a linear combination of basis functions defined in entire signal area	$G[k, l]$ estimates original $F[k, l]$ in terms of dominant frequencies
Extrapolation Concept	
Linear combination of selected basis functions provides extrapolation	Inverse DFT of estimate yields $g[m, n]$ which provides extrapolation

$g[m, n]$	
Parametric model (DFT fcts. inserted) $g^{(\nu)}[m, n] = \frac{1}{2MN} \sum_{(k,l) \in \mathcal{K}^{(\nu)}} (c_{k,l}^{(\nu)} \varphi_{k,l}[m, n] + c_{k,N}^{(\nu)*} \varphi_{k,N}^*[m, n])$	IDFT of estimate $G[k, l]$ $g^{(\nu)}[m, n] = \text{IDFT}_{M,N}\{G^{(\nu)}[k, l]\}$

$G[k, l]$	
DFT of parametric model $\text{DFT}_{M,N}\{g^{(\nu)}[m, n]\} = G^{(\nu)}[k, l] = \frac{1}{2} \sum_{(k,l) \in \mathcal{K}^{(\nu)}} (c_{k,l}^{(\nu)} + c_{M-k,N-l}^{(\nu)})$	The estimate $G^{(\nu)}[k, l] = \sum_{(\kappa,\lambda) \in \mathcal{K}^{(\nu)}} G^{(\nu)}[\kappa, \lambda] \delta[k - \kappa, l - \lambda]$

Update equation	
$\Delta c = \begin{cases} MN \frac{R_w^{(\nu)}[u,v]}{W[0,0]}, & (u, v) \in \mathcal{M} \\ 2MN \frac{R_w^{(\nu)}[u,v] W[0,0] - R_w^{*(\nu)}[u,v] W[2u,2v]}{W[0,0]^2 - W[2u,2v] ^2}, & \text{otherwise} \end{cases}$	$\Delta G[u, v] = \gamma \times \begin{cases} MN \frac{R_w^{(\nu)}[u,v]}{W[0,0]}, & (u, v) \in \mathcal{M} \\ MN \frac{R_w^{(\nu)}[u,v] W[0,0] - R_w^{*(\nu)}[u,v] W[2u,2v]}{W[0,0]^2 - W[2u,2v] ^2}, & \text{otherwise} \end{cases}$

Selection of basis function	
$\Delta E_{\mathcal{A}}^{(\nu+1)} = \begin{cases} 2 \frac{R_w^{(\nu)}[k,l]^2}{W[0,0]}, & (k, l) \in \mathcal{M} \\ 2 \frac{ R_w^{(\nu)}[k,l] ^2 W[0,0] - \text{Re}\{R_w^{(\nu)}[k,l]^2 W^*[2k,2l]\}}{W[0,0]^2 - W[2k,2l] ^2}, & \text{otherwise} \end{cases}$ $\Rightarrow (u, v) = \underset{(k,l)}{\text{argmax}} \Delta E_{\mathcal{A}}^{(\nu+1)}$	

4 Spatio-Temporal Selective Extrapolation

The concept derived for two-dimensions in the last chapter is extended to spatio-temporal selective extrapolation for three-dimensional signals in Section 4.1. In case of video signals, information about the area to be extrapolated can additionally be gained from previous and/or subsequent frames. Using DFT basis functions, this approach leads also to the frequency selective extrapolation which is presented in Section 4.2.4 for the efficient implementation in the frequency domain.

Finally, in Section 4.3 an alternative extrapolation approach is investigated. First, on basis of [26] extrapolation using best approximation is described. The difference to the selective approach derived in Section 3.1 is due to a best possible approximation of the known area in each iteration. However, a better approximation of the known area does not necessarily lead to a better extrapolation to the unknown area. Therefore, the extrapolation performance of best approximation is subsequently evaluated and compared to the performance of frequency selective extrapolation in Section 4.4.

4.1 Spatio-Temporal Selective Extrapolation for 3-D Signals

In the last chapter, the concept of selective extrapolation is derived for two-dimensional signals. Since the generic concept is independent of the dimensions of the signal space, it can be straightforwardly extended to three-dimensional signals which is presented in the following. In case of video signals, additional information from previous and/or following frames is obtained about the signal content to be estimated [37, 36]. We introduce a mathematical description of the video signal in *spatial and temporal* direction at the same time. By doing so, we provide a spatio-temporal extrapolation technique which allows for estimating image areas by exploiting simultaneously spatial and temporal correlations of the video signal.

Our approach approximates the known signal by a weighted linear combination of 3-D basis functions from spatial as well as temporal directions and extrapolates it into the missing area. In contrast to the two-dimensional approach derived in Section 3.1, we do not only exploit the surrounding image content within the image of the signal to be

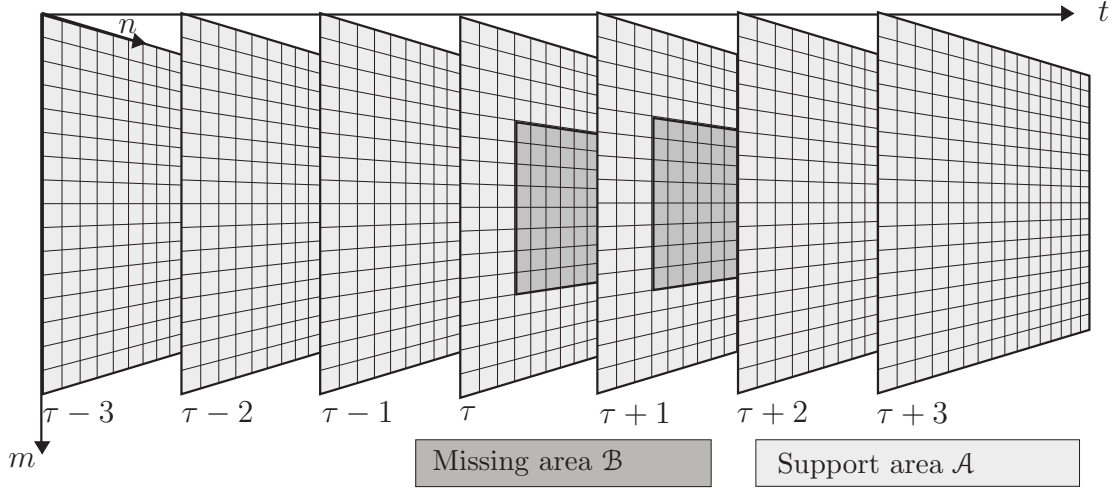


Figure 4.1: Image areas used for 3-D extrapolation consisting of the area to be estimated and its known surrounding.

predicted but also the image content of preceding and/or proceeding frames.

Fig. 4.1 shows a possible sequence of seven frames where the spatial dimensions are denoted by m, n and the temporal dimension by t . The area \mathcal{B} shaded dark gray reaches over two frames from τ to $\tau + 1$ and is to be extrapolated from the area \mathcal{A} . Note that the area to be extrapolated can be of spatio-temporal extension and is not limited to one frame. The support area \mathcal{A} ranges from three previous frames over the surrounding of the area to be estimated in the frame τ and $\tau + 1$ to two subsequent frames. Only the areas which are used for the reconstruction of the missing area are shown in Fig. 4.1. The entire region \mathcal{L} - consisting of the region to be estimated \mathcal{B} and the support area \mathcal{A} - is described by a volume.

Analogously to the 2-D extrapolation described in Section 3.1, the known pixels $f[m, n, t]$ are approximated iteratively by the parametric model $g^{(\nu)}[m, n, t]$. The spatio-temporal description by $g^{(\nu)}[m, n, t]$ approximates the support area by a linear combination of basis functions $\varphi_{k,l,p}[m, n, t]$ weighted by expansion coefficients $c_{k,l,p}^{(\nu)}$ according to

$$g^{(\nu)}[m, n, t] = \sum_{(k,l,p) \in \mathcal{K}^{(\nu)}} c_{k,l,p}^{(\nu)} \varphi_{k,l,p}[m, n, t]. \quad (4.1)$$

The set $\mathcal{K}^{(\nu)}$ consists of all basis functions used so far for the approximation. The basis functions are defined in the entire area \mathcal{L} and their number $M \times N \times T$ equals the number of pixels in \mathcal{L} . Here, the principle is described for real-valued basis functions and expansion coefficients.

With help of the 3-D window function $b[m, n, t]$

$$b[m, n, t] = \begin{cases} 1, & (m, n, t) \in \mathcal{A} \\ 0, & (m, n, t) \in \mathcal{B} \end{cases} \quad (4.2)$$

the residual error signal in the support area is calculated in this step

$$r^{(\nu)}[m, n, t] = b[m, n, t] (f[m, n, t] - g^{(\nu)}[m, n, t]) . \quad (4.3)$$

Assuming an appropriate basis function $\varphi_{u,v,q}[m, n, t]$ is already selected, the residual error signal is further reduced by the update of the parametric model $\Delta g[m, n, t]$

$$r_w^{(\nu+1)}[m, n, t] = r_w^{(\nu)}[m, n, t] - w[m, n, t] \Delta g[m, n, t] \quad (4.4)$$

which is expressed with help of the introduced variable

$$r_w^{(\nu)}[m, n, t] = w[m, n, t] r^{(\nu)}[m, n, t] \quad (4.5)$$

as described for the 2-D case in (3.35).

In case of taking only the selected coefficient into account for the update step, $\Delta g[m, n, t]$ is determined for three-dimensional signals analogously to the two-dimensional case (3.9) by

$$\Delta g[m, n, t] = \Delta c \varphi_{u,v,q}[m, n, t]. \quad (4.6)$$

In order to determine the expansion coefficients, the weighted residual error energy between the original signal and its approximation by the parametric model is evaluated with respect to the support area

$$E_{\mathcal{A}}^{(\nu+1)} = \sum_{(m,n,t) \in \mathcal{A}} w[m, n, t] (r^{(\nu)}[m, n, t] - \Delta g[m, n, t])^2, \quad (4.7)$$

where the 3-D weighting function $w[m, n, t]$

$$w[m, n, t] = \begin{cases} \rho[m, n, t], & (m, n, t) \in \mathcal{A} \\ 0, & (m, n, t) \in \mathcal{B} \end{cases} \quad (4.8)$$

has only positive amplitudes in the support area and is zero elsewhere. Generally, $w[m, n, t]$ allows to emphasize pixels which are more important for the extrapolation and has to be specified for the actual application. The weighting function is a straightforward extension to 3-D from the 2-D function (3.1).

Assuming a basis function with index (u, v, q) is already selected, the coefficient update by Δc is computed by minimizing the error criterion (4.7) by taking the derivative with respect to the unknown coefficient and setting it to zero

$$\Delta c = \frac{\sum_{(m,n,t) \in \mathcal{L}} r_w^{(\nu)}[m, n, t] \varphi_{u,v,q}[m, n, t]}{\sum_{(m,n,t) \in \mathcal{L}} w[m, n, t] \varphi_{u,v,q}^2[m, n, t]} \quad (4.9)$$

The coefficient $c_{u,v,q}^{(\nu+1)}$ is subsequently updated

$$c_{u,v,q}^{(\nu+1)} = c_{u,v,q}^{(\nu)} + \Delta c. \quad (4.10)$$

Per iteration, we choose the basis function $\varphi_{u,v,q}[m, n, t]$ which leads to a maximum reduction of the residual error criterion obtained with help of (4.9) by

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \sum_{(m,n,t) \in \mathcal{A}} w[m, n, t] (\Delta c \varphi_{k,l,p}[m, n, t])^2 \\ &= \frac{\left(\sum_{(m,n,t) \in \mathcal{L}} r_w^{(\nu)}[m, n, t] \varphi_{k,l,p}[m, n, t] \right)^2}{\sum_{(m,n,t) \in \mathcal{L}} w[m, n, t] \varphi_{k,l,p}^2[m, n, t]} \\ &\Rightarrow (u, v, q) = \arg \max_{(k,l,p)} \Delta E_{\mathcal{A}}^{(\nu+1)}. \end{aligned} \quad (4.11)$$

The index of the selected basis function is included in the set of used basis functions in case it has not been selected yet

$$\mathcal{K}^{(\nu+1)} = \mathcal{K}^{(\nu)} \cup (u, v, q) \quad \text{if} \quad (u, v, q) \notin \mathcal{K}^{(\nu)}. \quad (4.12)$$

The algorithm terminates if the reduction of the residual error energy drops below a pre-specified threshold.

The derivations in this section provide the extension of the concept of selective extrapolation from 2-D as developed in Section 3.1 to 3-D. Summarizing the main points of the algorithm, the image content in the spatio-temporal volume is described *simultaneously* in *spatio-temporal* direction by dominant features in terms of weighted basis functions.

The basis functions are defined in the entire volume, therefore the approximation provides at the same time an estimation of the missing samples which in turn can be of spatio-temporal extend. Finally, the extrapolated area is cut out of the parametric model.

4.2 Frequency Selective Extrapolation of Video Signals Using DFT Basis Functions

2-D DFT basis functions are used in order to extrapolate image signals. For video signals, additionally to textures in 2-D motion trajectories in temporal direction have to be extended. In the following, we apply basis functions based on the 3-D DFT for the extrapolation of video signals according to

$$\varphi_{k,l,p}[m, n, t] = e^{j \frac{2\pi}{M} mk} e^{j \frac{2\pi}{N} nl} e^{j \frac{2\pi}{T} tp} \quad (4.13)$$

where the described volume $M \times N \times T$ consists of the spatial size $M \times N$ extended by T in the temporal dimension.

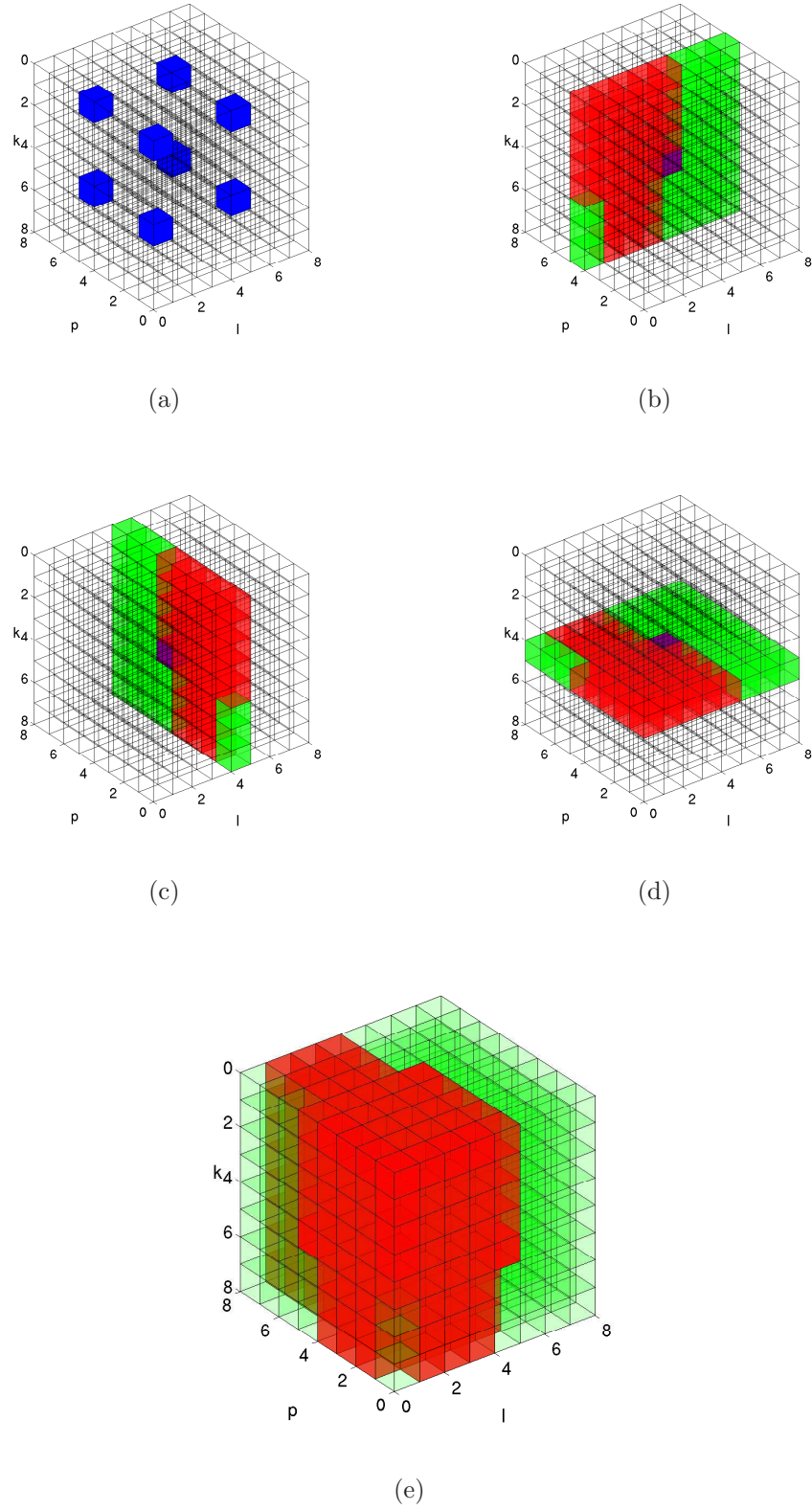


Figure 4.2: The DFT symmetry properties for 3-D real-valued signals are shown for $M = N = T = 8$. (a) Real-valued coefficients are marked in blue. (b)-(e) Due to the symmetry properties, the coefficients in red have to be computed and the conjugate complex coefficients in green can be obtained. Purple marks the point of conjugate complex symmetry.

As for 2-D, the basis functions exhibit the conjugate complex symmetry

$$\varphi_{M-k,N-l,T-p}[m,n,t] = \varphi_{k,l,p}^*[m,n,t]. \quad (4.14)$$

Due to symmetry properties of the DFT for real-valued signals, the set \mathcal{M} consists of the indices of frequency bins corresponding to a real-valued spectrum for M, N, T being even

$$\mathcal{M} = \left\{ (0,0,0), \left(\frac{M}{2}, 0, 0\right), \left(0, \frac{N}{2}, 0\right), \left(0, 0, \frac{T}{2}\right), \left(\frac{M}{2}, \frac{N}{2}, 0\right), \left(\frac{M}{2}, 0, \frac{T}{2}\right), \right. \\ \left. \left(0, \frac{N}{2}, \frac{T}{2}\right), \left(\frac{M}{2}, \frac{N}{2}, \frac{T}{2}\right) \right\}. \quad (4.15)$$

They are marked blue in the 3-D frequency volume shown in Fig. 4.2(a). As for 2-D, the DFT-spectrum of a real-valued signal such as a video signal shows conjugate complex symmetries which is illustrated in the following. In the three following figures (b) to (d), the purple colour indicates the point of symmetry for conjugate complex symmetry within a 2-D plane, red indicates the area of values which have to be calculated and green indicates the DFT values which can be obtained due to conjugate complex symmetry properties. In Fig. 4.2(b) in the section plane at $p = \frac{T}{2}$, in (c) at $l = \frac{N}{2}$, and in (d) at $k = \frac{M}{2}$, the 2-D symmetry properties of Fig. 3.4 can be recognized. Fig. 4.2(e) shows finally in red the values which have to be computed in order to describe completely the 3-D DFT-spectrum of a real-valued signal and in green the values which can be obtained due to conjugate complex symmetry properties within the volume.

4.2.1 Spatio-Temporal-Domain Solution

The video signal in the spatio-temporal volume is approximated by the corresponding parametric model in iteration ν , i.e. by

$$g^{(\nu)}[m,n,t] = \frac{1}{2MNT} \times \sum_{(k,l,p) \in \mathcal{K}^{(\nu)}} (c_{k,l,p}^{(\nu)} \varphi_{k,l,p}[m,n,t] + c_{M-k,N-l,T-p}^{(\nu)} \varphi_{M-k,N-l,T-p}[m,n,t]) \quad (4.16)$$

where the expansion coefficients exhibit the conjugate complex symmetry

$$c_{M-k,N-l,T-p}^{(\nu)} = c_{k,l,p}^{(\nu)*}. \quad (4.17)$$

The residual error in iteration ν is altered by the update of the parametric model

$$\Delta g[m,n,t] = \frac{1}{2MNT} (\Delta c \varphi_{u,v,q}[m,n,t] + \Delta c^* \varphi_{M-u,N-v,T-q}[m,n,t]) \quad (4.18)$$

yielding the residual error in the next iteration $\nu + 1$

$$\begin{aligned} r_w^{(\nu+1)}[m, n, t] &= r_w^{(\nu)}[m, n, t] \\ &- \frac{1}{2MNT} (\Delta c \varphi_{u,v,q}[m, n, t] + \Delta c^* \varphi_{M-u, N-v, T-q}[m, n, t]) w[m, n, t]. \end{aligned} \quad (4.19)$$

The error criterion for the 3-D signals (4.7) is minimized analogously to the 2-D case in (3.36). The resulting 3-D frequency-domain solutions for Δc and Δc^* are discussed in Section 4.2.2. With the solution Δc , the unknown expansion coefficient and its conjugate complex counterpart are updated

$$\begin{aligned} c_{u,v,q}^{(\nu+1)} &= c_{u,v,q}^{(\nu)} + \Delta c \\ c_{M-u, N-v, T-q}^{(\nu+1)} &= c_{M-u, N-v, T-q}^{(\nu)} + \Delta c^*. \end{aligned} \quad (4.20)$$

The maximum energy decrease (4.11) for the frequency (u, v, q) leads to the selection of the optimum basis function depending on Δc

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \frac{1}{2(MNT)^2} \left(|\Delta c|^2 \sum_{(m,n,t) \in \mathcal{L}} w[m, n, t] \varphi_{k,l,p}[m, n, t] \varphi_{k,l,p}^*[m, n, t] \right. \\ &\quad \left. + \operatorname{Re}\{\Delta c^2 \sum_{(m,n,t) \in \mathcal{L}} w[m, n, t] \varphi_{k,l,p}^2[m, n, t]\} \right) \\ \Rightarrow (u, v, q) &= \operatorname{argmax}_{(k,l,p)} \Delta E_{\mathcal{A}}^{(\nu+1)}. \end{aligned} \quad (4.21)$$

4.2.2 Frequency-Domain Solution

As in the 2-D case, we express all equations in the frequency domain in order to solve the extrapolation problem efficiently. Minimizing the 3-D error criterion (4.7) leads to a solution for Δc and Δc^* , respectively. Recalling the simplifications for real-valued DFT coefficients in the 2-D case (3.45), we obtain for Δc

$$\Delta c = \begin{cases} MNT \frac{R_w^{(\nu)}[u,v,q]}{W[0,0,0]}, & (u, v, q) \in \mathcal{M} \\ 2MNT \frac{R_w^{(\nu)}[u,v,q] W[0,0,0] - R_w^{(\nu)*}[u,v,q] W[2u, 2v, 2q]}{W[0,0,0]^2 - |W[2u, 2v, 2q]|^2}, & \text{otherwise} \end{cases} \quad (4.22)$$

and a conjugate complex equation for Δc^* .

The energy decrease per discrete frequency (4.21) is rewritten in the frequency domain

$$\Delta E_{\mathcal{A}}^{(\nu+1)} = \frac{1}{2M^2 N^2 T^2} (|\Delta c|^2 W[0, 0, 0] + \operatorname{Re}\{\Delta c^2 W^*[2u, 2v, 2q]\}) \quad (4.23)$$

Inserting Δc according to (4.22) leads to the selection of the optimum basis function according to the maximum energy decrease *per* discrete frequency

$$\begin{aligned} \Delta E_{\mathcal{A}}^{(\nu+1)} &= \begin{cases} 2 \frac{R_w^{(\nu)}[u,v,q]^2}{W[0,0,0]} & , \quad (k,l,p) \in \mathcal{M} \\ 2 \frac{|R_w^{(\nu)}[u,v,q]|^2 W[0,0,0] - \text{Re}\{R_w^{(\nu)}[u,v,q]^2 W^*[2u,2v,2q]\}}{(W[0,0,0]^2 - |W[2u,2v,2q]|^2)} & , \quad \text{otherwise} \end{cases} \quad (4.24) \\ \Rightarrow (u,v,q) &= \arg \max_{(k,l,p)} \Delta E_{\mathcal{A}}^{(\nu+1)}. \end{aligned}$$

If the approximation of the support area is not sufficient yet, the residuum in the DFT-domain is further reduced by

$$\begin{aligned} R_w^{(\nu+1)}[k,l,p] &= R_w^{(\nu)}[k,l,p] \\ &- \frac{1}{2MNT} (\Delta c W[k-u, l-v, p-q] + \Delta c^* W[k+u, l+v, p+q]). \end{aligned} \quad (4.25)$$

Otherwise the algorithm terminates if $\Delta E_{\mathcal{A}}^{(\nu+1)}$ drops below ΔE_{\min} and the spatio-temporal parametric model is given in the entire volume by an inverse DFT of $G^{(\nu)}[k,l,p]$

$$g^{(\nu)}[m,n,t] = \text{IDFT}_{M,N,T}\{G^{(\nu)}[k,l,p]\}. \quad (4.26)$$

4.2.3 3-D Isotropic Weighting Function

The right choice of an appropriate weighting function $w[m,n,t]$ is of great importance and depends on the application. In Section 3.5, it is already shown for 2-D that the different weighting functions affect the extrapolation result significantly. By choosing an signal and application adapted weighting function instead of a simple binary weighting function, the quality of the extrapolated area could be improved remarkably. The adapted weighting function takes pixels closer to the unknown area more into account as more distant pixels and is represented by an 2-D isotropic model. Based on these results, a suitable 3-D weighting function is specified and investigated in the following.

Again, as representative example a centrally located block of 16×16 unknown pixels is extrapolated from its known surrounding. Since the video signal is interpreted as a 3-D volume, data from previous and subsequent frames about the area to be estimated is also available which has to be incorporated into the weighting function. The application example corresponds to 3-D frequency selective extrapolation applied to error concealment of block losses which is described in detail in Chapter 5.

The principle of the successful 2-D weighting function relying on the isotropic model (3.52) where pixels closer to the missing area are taken more into account as more distant pixels is adopted for the 3-D case. Therefore, the 2-D isotropic function (3.52) is extended to 3-D according to

$$\rho_{\text{isotrop}}[m,n,t] = \hat{\rho} \sqrt{\left(m - \frac{M-1}{2}\right)^2 + \left(n - \frac{N-1}{2}\right)^2 + \left(t - \frac{T-1}{2}\right)^2}, \quad \hat{\rho} \leq 1 \quad (4.27)$$

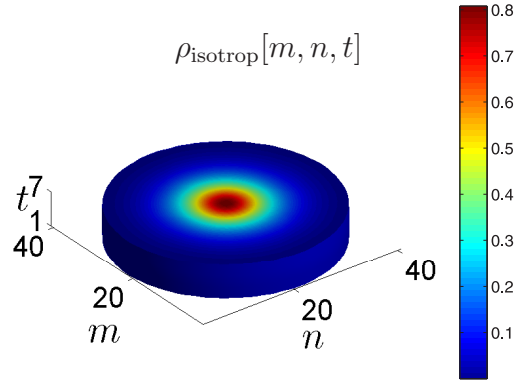


Figure 4.3: 3-D isotropic function for a volume of 7 frames ($\hat{\rho}=0.8$).

where the loci of constant amplitudes are spheres as depicted for a cut-out of 7 frames in Fig. 4.3. Pixels with the same distance to the center $(\frac{M-1}{2}, \frac{N-1}{2}, \frac{T-1}{2})$ in spatial as well as temporal direction have the same weight.

Applying this model does not allow to weight the spatial and temporal direction separately. However, intensive investigations in [32] comparing different models where space and time are treated separately have shown that the 3-D isotropic model provides the best performance.

Evaluations for 3-D Isotropic Weighting Function

The investigations are done for uncoded YUV sequences. Therefore, the quality of the extrapolation result is measured in terms of PSNR (3.50) as previously in Section 3.5 only for the extrapolated parts of the luminance component.

The error patterns of isolated block losses with 16×16 pixels are inserted into a frame of different video sequences. For evaluation purposes, four different YUV sequences in common intermediate format (CIF) with 352×288 pixels are tested. The original frame and the corresponding frame with inserted block losses of the test sequences *Flowergarden*, *Foreman*, *Table Tennis*, and *Crew* are depicted in Fig. 4.4.

Investigations are done using the 3-D isotropic weighting function defined in (4.27). The following parameters influence the performance and are evaluated in the following:

- The value of the decay parameter $\hat{\rho}$ as introduced in (4.27).
- The size of the support area along the time axis in terms of previous and subsequent frames and along the space axis in terms of number of surrounding pixels.
- The termination criteria: number of iterations and Δ_{\min} .
- The DFT size.



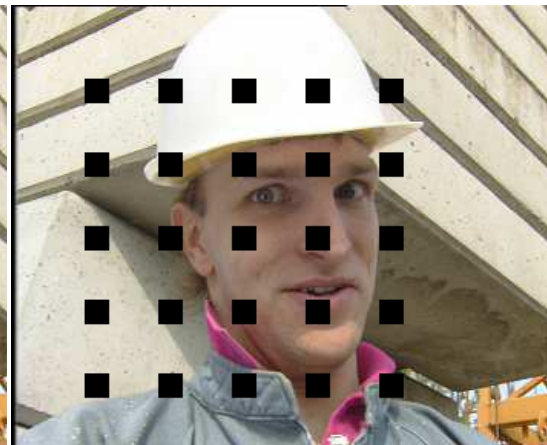
(a)



(b)



(c)



(d)



(e)



(f)



Figure 4.4: Sequences used for concealment with corresponding corrupted frames. 4.4(a): 9th frame of sequence *Flowergarden*. 4.4(c): 205th frame of sequence *Foreman*. 4.4(e): 82th frame of sequence *Table Tennis*. 4.4(g): 316th frame of sequence *Crew*.

Decay Parameter $\hat{\rho}$

The performance is investigated in terms of PSNR with respect to $\hat{\rho}$ using the 3-D weighting function as defined in (4.27). The results for concealing the isolated block losses which are inserted for each of the four different test sequences into a single frame as shown in Fig. 4.4 are presented in Fig. 4.5. PSNR is measured for varying $\hat{\rho}$ with respect to the number of surrounding pixels. The size of the DFT transform is $64 \times 64 \times 32$. As termination criteria maximally 200 iterations and $\Delta_{\min} = 0.1$ are applied. Two previous and two subsequent frames are taken into account for the extrapolation. The support area in spatial direction consists of 13 surrounding pixels according to the evaluation results for 2-D in Section 3.5.3.

Fig. 4.5 shows that the best results are obtained for $0.75 \leq \hat{\rho} \leq 0.85$. As already mentioned, space and time can not be treated separately with the 3-D isotropic model. In [32], investigations have been done for applying the 2-D model to each time index separately, i.e., the parameter $\hat{\rho}$ takes different values along the time axis, in ascending and in descending order. However, the investigations have shown that the isotropic 3-D model provides the best performance. Therefore, the 3-D model with the fixed value $\hat{\rho} = 0.8$ is chosen in the following.

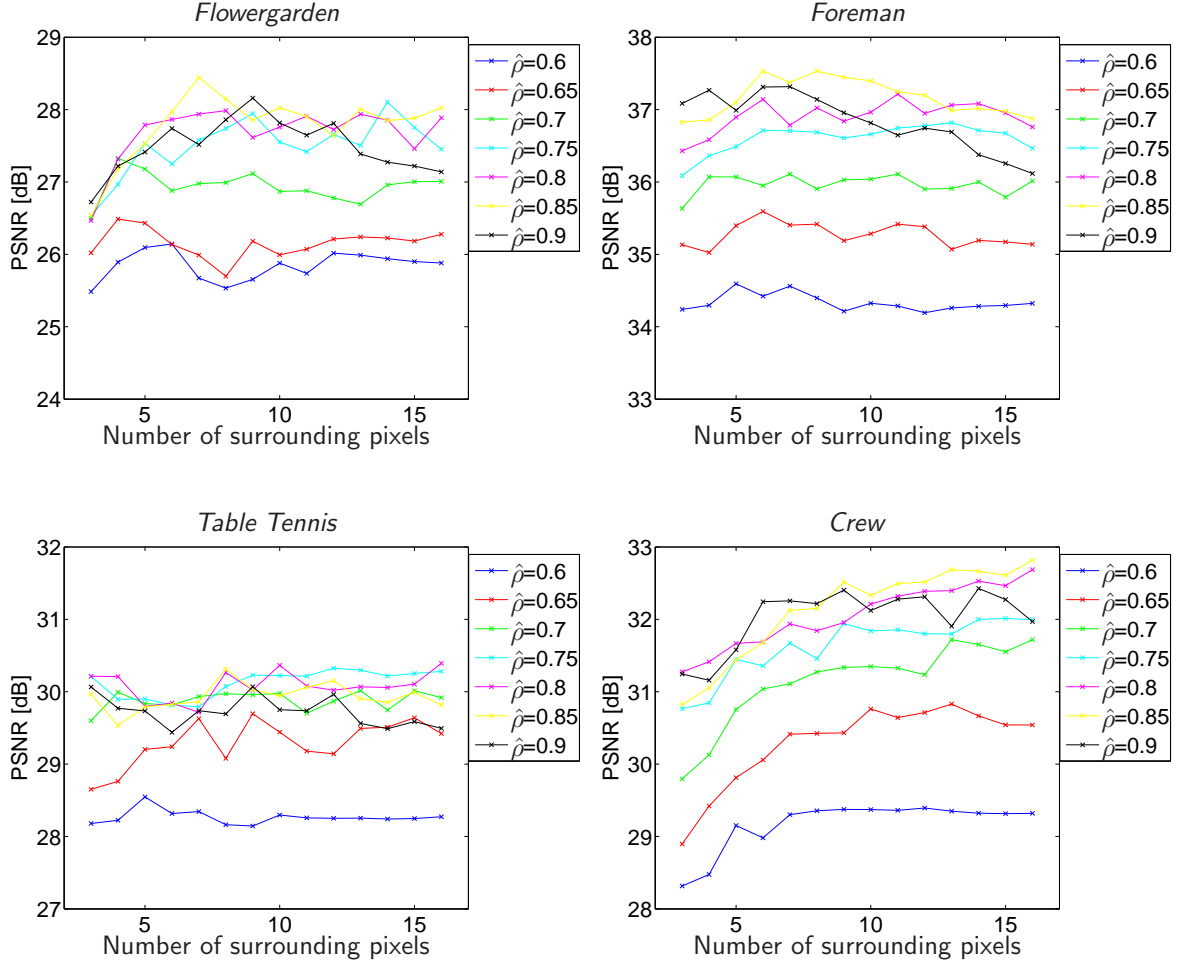


Figure 4.5: PSNR measurements for the decay parameter $\hat{\rho}$ (4.27) for isolated block losses. Set of parameters used: maximum number of iterations: 200, $\Delta_{\min}=0.1$, $N_V=2$, $N_N=2$, 13 surrounding pixels in spatial direction, DFT size $64 \times 64 \times 32$.

Size of Support Area

Let us now take a closer look at the impact of the number of previous N_V and/or subsequent N_N frames involved for the calculation of the parametric model. Fig. 4.6 depicts the results for the four test sequences. The application of both, previous and subsequent frames yields always better results than using only previous frames, whereas the optimum number of frames depends on the motion of the sequence. For slow motion sequences as *Foreman* and *Crew*, the results are very similar for one, two, or three previous and subsequent frames. In case of *Flowergarden* and *Table Tennis* including stronger motion, the results improve using two or three previous and subsequent frames. In case only previous frames are available, one previous frame yields the worst result for all sequences.

In summary, two or three previous frames lead to a satisfying result for all sequences.

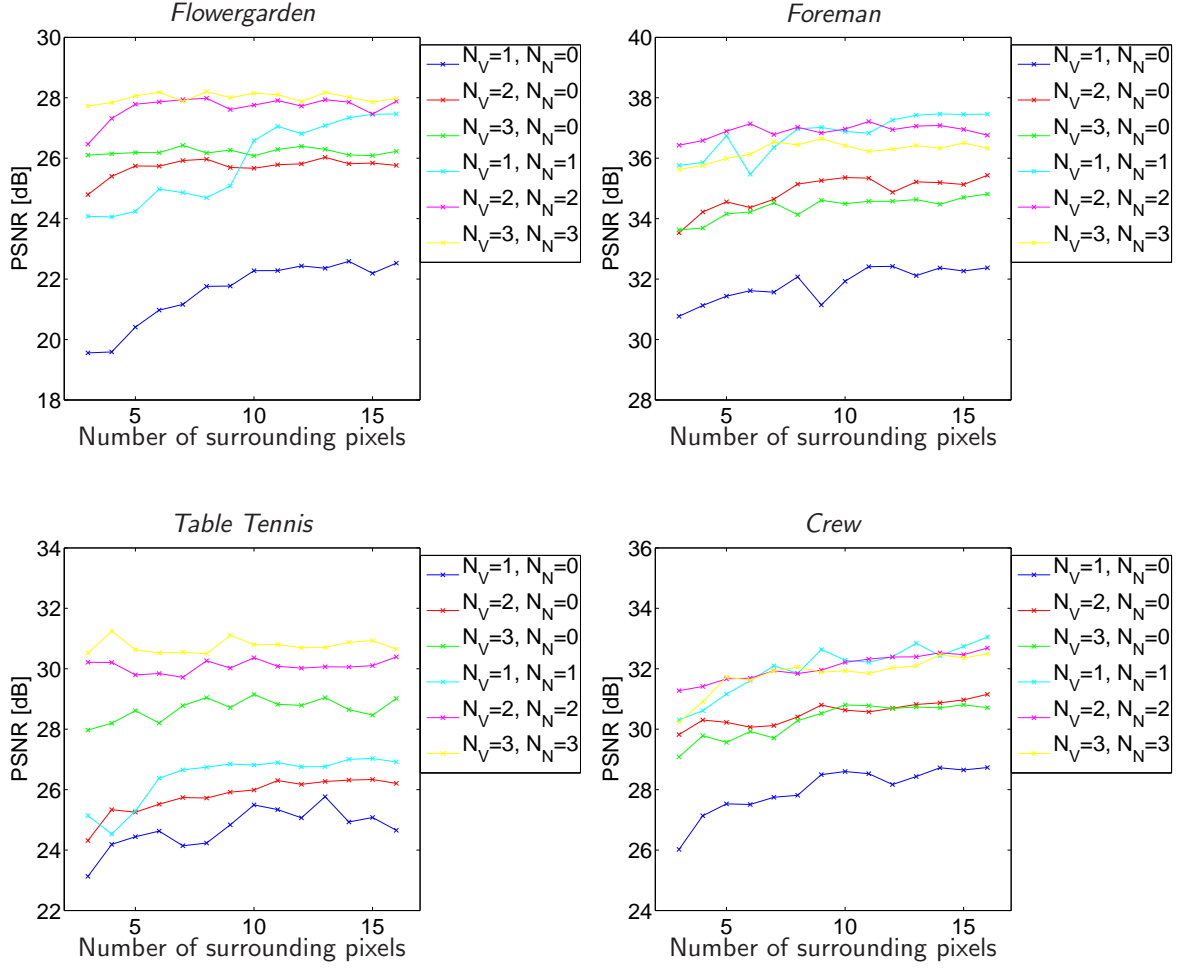


Figure 4.6: PSNR measurements with respect to the number of involved previous N_V and subsequent N_N frames. Set of parameters used: maximum number of iterations: 200, $\Delta_{\min}=0.1$, $\hat{\rho}=0.8$, DFT size $64 \times 64 \times 32$.

If no subsequent frames are available, the best result is also obtained for two or three previous frames. In the following, we choose $N_V = N_N = 2$.

In order to finally determine the size of the support area, the extension in spatial direction in terms of number of surrounding pixels is investigated. The evaluation according to Fig. 4.6 shows that for 10 to 16 surrounding pixels the best results can be achieved. The image content of several succeeding frames is considered in a volume. The spatial extension has to assure that the corresponding volume covers the motion. Here, 13 surrounding pixels are chosen as extension of the support area in spatial direction.

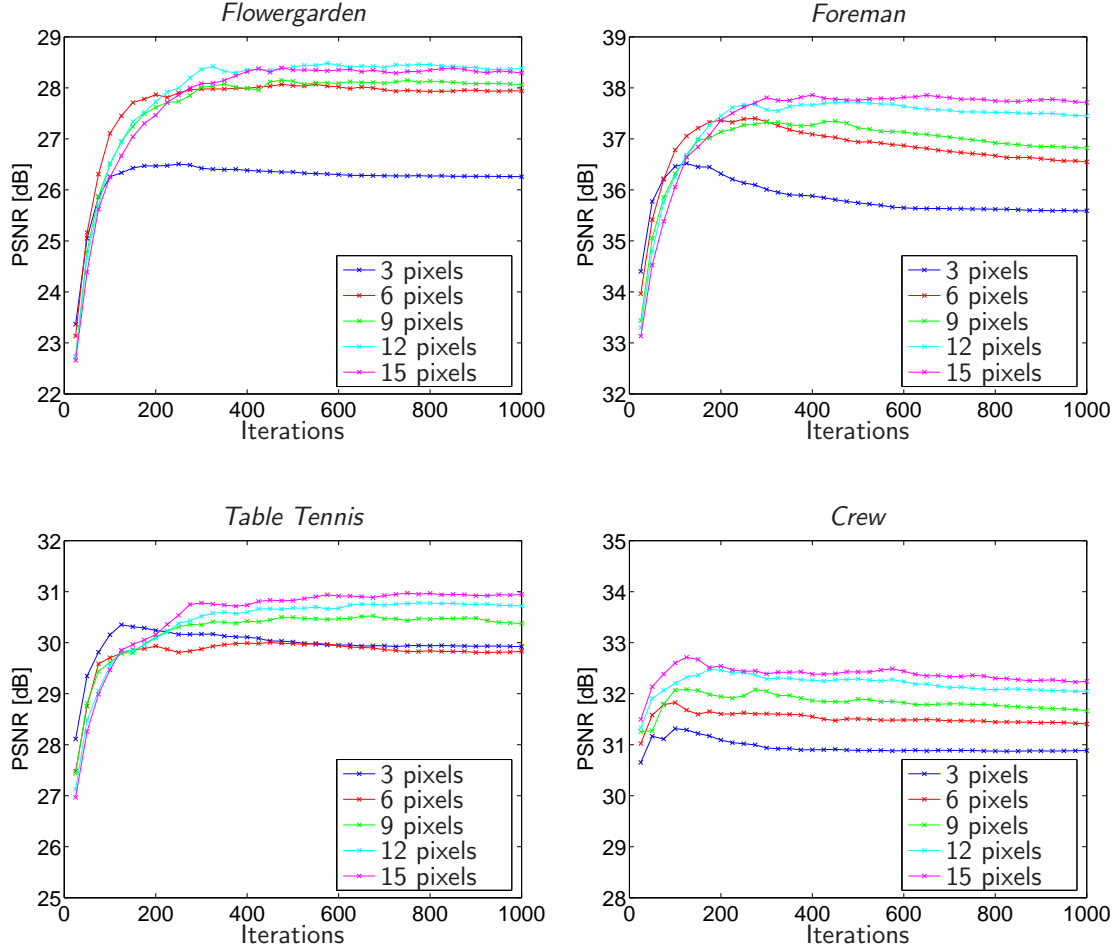


Figure 4.7: PSNR measurements with respect to number of iterations. Set of parameters used: $\Delta_{\min}=0.0$, $\hat{\rho}=0.8$, $N_V=2$, $N_N=2$, DFT size $64 \times 64 \times 32$.

Termination Criteria

Next, the termination criteria are investigated. We start with the maximum number of iterations. As the demonstration of the principle for 3-D in Section 4.2.4 has shown, in general more iterations are needed for the approximation of the 3-D signal in the volume compared to 2-D. Fig. 4.7 depicts the PSNR results for the number of iterations with respect to the number of surrounding pixels. We can observe a saturation point where further iterations do not yield a significant improvement anymore. This holds if the support area is chosen appropriately. If it is chosen too small, the performance even decreases. Trading off a marginal improvement versus computational load, 200 iterations are maximally allowed. The investigations in [32] show that $\Delta_{\min} = 0.1$ is a good choice because the corresponding PSNR value is close to the optimum.

If the computational load has to be limited, the maximum number of iterations can be decreased and Δ_{\min} increased.

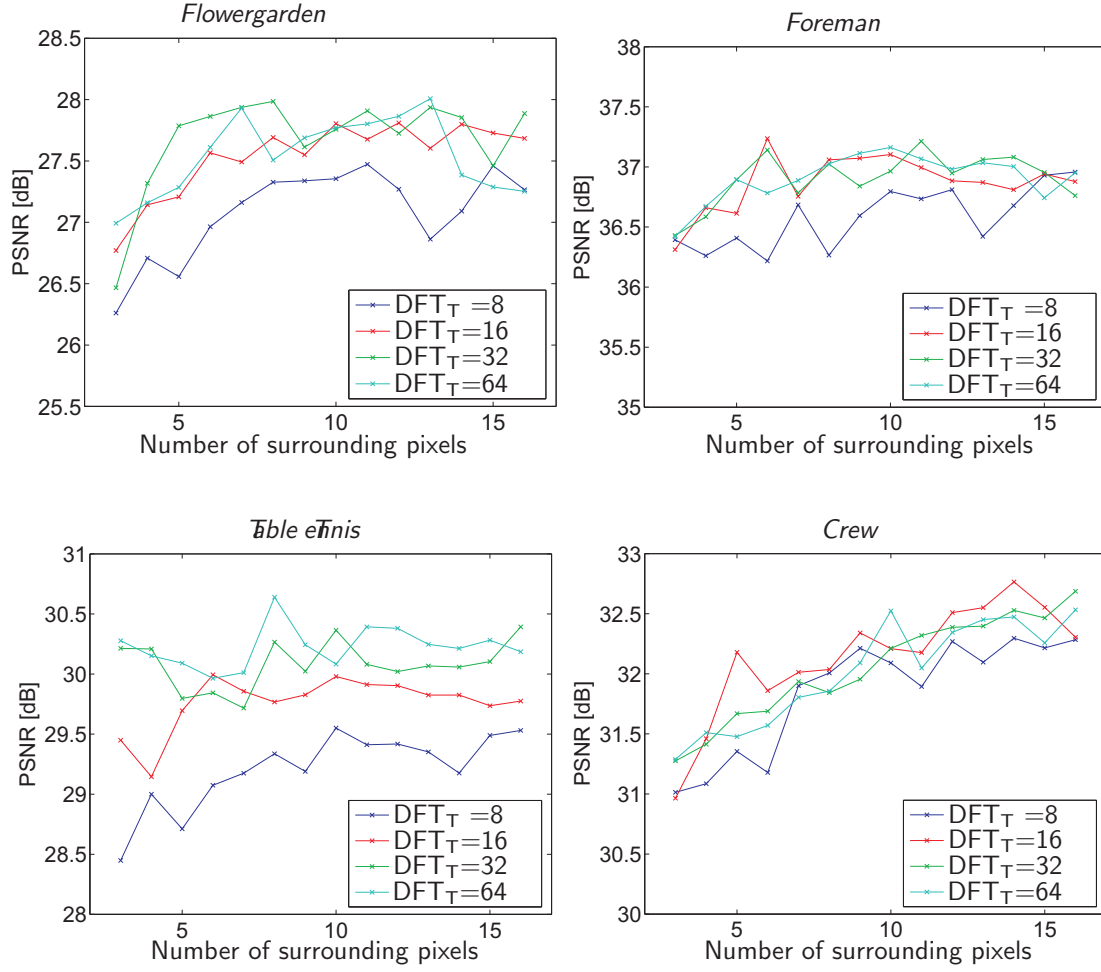


Figure 4.8: PSNR measurements with respect to DFT size in temporal direction DFT_T . Set of parameters used: maximum number of iterations: 200, $\Delta_{\min}=0.1$, $\hat{\rho}=0.8$, $N_V=2$, $N_N=2$, DFT size in spatial direction 64×64 .

DFT Size

The DFT size for the spatial dimensions is chosen to 64×64 according to the evaluations for 2-D in Section 3.5.3. The size of the DFT in temporal direction is considered next. In Fig. 4.8 the performance with respect to the DFT length in temporal direction DFT_T is depicted. The results for $DFT_T = 8$ are always worse than for the larger DFT lengths. Since the isotropic function is already truncated after a few frames in temporal direction, interpolated spectra due to zero padding have to be used. The results for $DFT_T = 16, 32, 64$ are similar. In the following, $DFT_T = 32$ is chosen.

Table 4.1: Parameters for spatio-temporal extrapolation of 16×16 pixels with 3-D isotropic weighting function (Section 4.2.3).

$\hat{\rho}$		0.8
Support area	no. of known pixel	13
	no. of frames involved $N_V = N_N$	2
Termination criterion	max. no. of iterations	200
	Δ_{\min}	0.1
DFT size		$64 \times 64 \times 32$

4.2.4 Discussion of Results

In the following, the results of applying 3-D frequency selective extrapolation to the estimation of unknown areas in video signals are discussed. Furthermore, objective results demonstrating the principle of the 3-D concept are presented. The improvements of using 3-D extrapolation by considering the video signal in a volume compared to 2-D extrapolation are also shown.

As in case of 2-D frequency selective extrapolation, an application-specific weighting function for 3-D could be specified in the considered extrapolation context. The area to estimated is centrally located and surrounded by original data. The evaluations in the last section have shown that the 3-D isotropic model provides as for 2-D a suitable application adapted model for the 3-D weighting function. A fixed set of parameters can be specified as for 2-D also for 3-D which is very important in practice. For the extrapolation into blocks of size 16×16 pixels, the set of parameters is given in Table 4.1.

The principle of frequency selective extrapolation was illustrated in Section 3.4.3 for 2-D signals for the representative example. In the following, we want to provide also subjective results and demonstrate the principle for 3-D. The procedure such as the realization in the frequency domain remains unchanged. Here, the results for the 3-D case and the effects of extending the support area from 2-D to 3-D are presented. The parameters according to Table 4.1 are applied.

In Fig. 4.9(a) the frame #9 of the sequence *Flowergarden* containing block losses of size 16×16 is shown. The missing pixels are to be concealed by 3-D frequency selective extrapolation, with the result depicted in Fig. 4.9(d). By involving two previous and subsequent frames, the extrapolation result can be significantly improved compared to 2-D which is displayed in Fig. 4.9(c).

Using temporal data implies that the image contents usually move. However, the motion between the frames is inherently compensated by the algorithm which can be seen in Fig. 4.9(b), where for comparison the block spatially located at the same position in

the previous frame is copied by temporal block replacement [59]. Motion artifacts occur at the edge of the tree for example.

To further illustrate the inherent motion compensation, we take a look at Fig. 4.10. In the first row, the input data to the algorithm for the reconstruction of a single block is displayed in figures (a) to (e). We conceal the loss occurring in frame #9 depicted in Fig. 4.10 (c) by taking advantage of the spatial surrounding as well as the respective blocks in the two previous frames, Fig. 4.10 (a), (b), and the two subsequent frames (d), (f). Obviously the tree is moving across the 5 figures. The algorithm works in the frequency domain until the approximation accuracy is reached. The evolution of the parametric model with respect to the number of iterations is displayed step-wise in figures (f) to (y), demonstrating the inherent motion compensation. After termination the parametric model is transformed to the spatio-temporal domain and the loss is replaced by the corresponding samples of the parametric model.

Fig. 4.11 depicts a further example. The losses occurring in Fig. 4.11(a) are concealed applying the 3-D approach which is displayed in Fig. 4.11(d). We observe again that using the spatio-temporal surrounding as basis for extrapolation improves the result significantly compared to using only the spatial surrounding for the 2-D approach shown in Fig. 4.11(c). The 3-D result appears sharper and more details can be restored. Comparing the result to the temporal block replacement [59] depicted in Fig. 4.11(b) shows another advantage. The deviations in luminance occurring in the previous frame due to a flash light can be compensated, too. This point is further illustrated by Fig. 4.12.

The five figures in the top row (a) to (e) show the original data which is input to the algorithm. The luminance varies from frame to frame, especially in frame #315 Fig. 4.12 (b) where the flash of a camera illuminates the scene. In Fig. 4.12 (f) to (j) the progress of the parametric model with respect to the number of iterations is depicted illustrating the ability to compensate the variations in luminance.

The examples presented in this section clearly show the improvements of the 3-D spatio-temporal approach compared to the 2-D case. With the new concept of 3-D extrapolation, a video signal can be interpreted as a 3-D volume. Hence, spatial and temporal correlations of the video signal can be used for the extrapolation at the same time. The 3-D algorithm is not only able to extrapolate different 2-D textures, but is additionally able to extend motion trajectories and to compensate for changes in luminance.



(a)



(b)



(c)



(d)

Figure 4.9: *Flowergarden* sequence. (a) Unknown blocks. (b) Temporal block replacement. (c) 2-D frequency selective extrapolation. (d) 3-D frequency selective extrapolation.

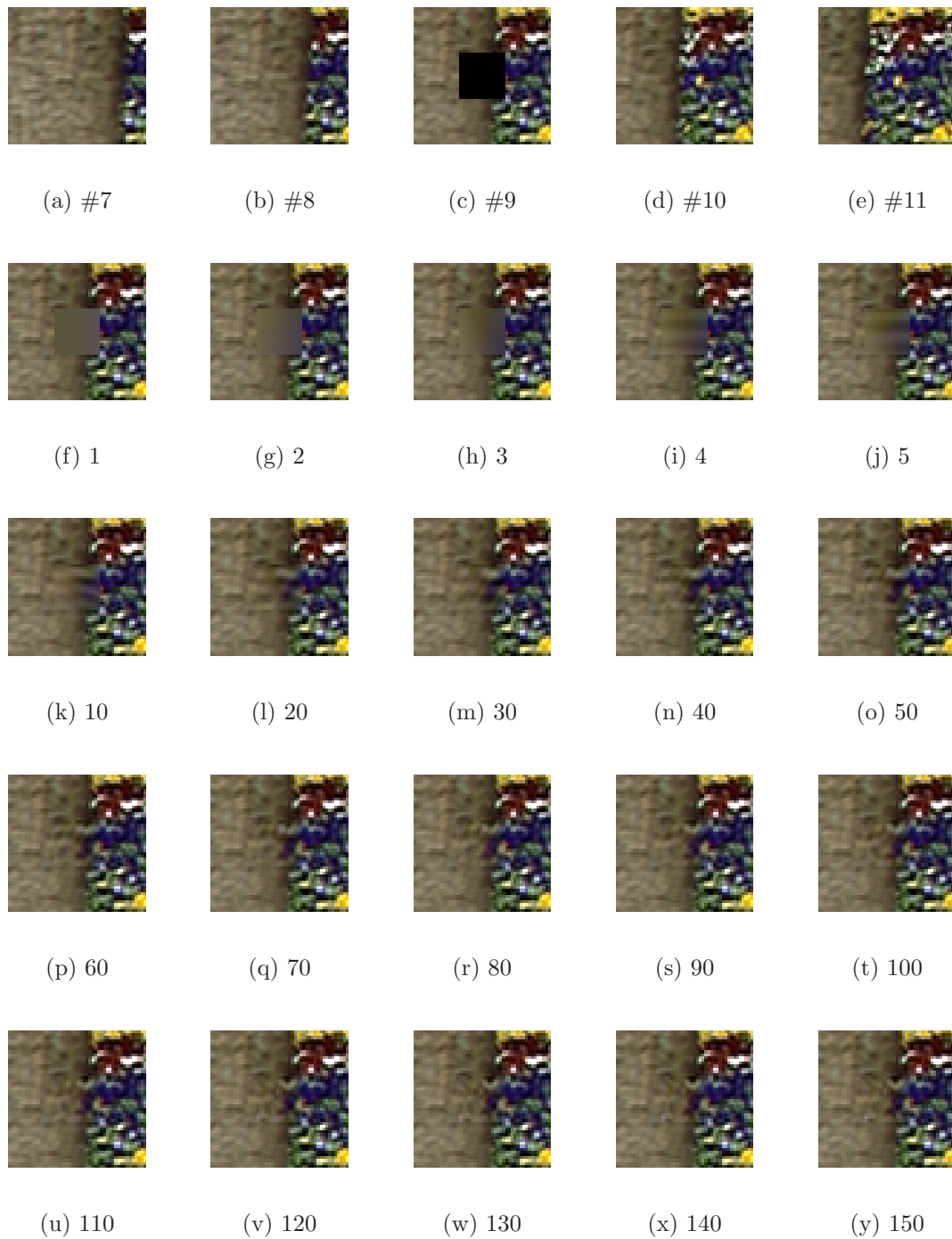
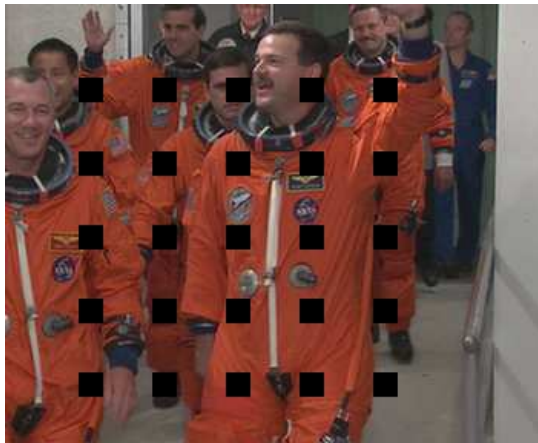


Figure 4.10: Original input data of loss occurring in frame #9 including the respective areas in the two previous and subsequent frame. (a) Frame #7 to (e) frame #11. Progress of parametric model with increasing number of iterations. (f)-(y) Iteration 1 to 150.



(a)



(b)



(c)



(d)

Figure 4.11: *Crew* sequence. (a) Unknown blocks. (b): Temporal block replacement. (c): 2-D frequency selective extrapolation. (d): 3-D frequency selective extrapolation.



Figure 4.12: Top: Original input data of loss occurring in frame #316 including the respective areas in the two previous and subsequent frame. (a) Frame #314 to (e) frame #318. Bottom: Progress of parametric model with increasing number of iterations. (f)-(j) Iteration 1 to 100.

4.3 Alternative Approach: Spatio-Temporal Extrapolation Using Best Approximation

An alternative extrapolation approach is investigated in this section. Extrapolation using best approximation of the support area in the update step is described based on [26].

In order to estimate unknown samples, the selective extrapolation strategy is derived in Section 3.1. There, the residual error signal is further approximated in each iteration by the update of the parametric model $\Delta g[m, n]$ (3.7). In Section 3.1.2, the approximation by $\Delta g[m, n]$ consisted only of the currently selected, weighted basis function (3.9). However, in each update step the expansion coefficients of the previously selected basis functions are actually altered. We derive in the following the approximation of the known area by taking *all* basis functions selected so far into account and refer to it as *best approximation*. We investigate whether a better extrapolation in the area to be estimated can be achieved for the case of approximating the known area best possibly. By this procedure, the residual error criterion with respect to the known area is maximally reduced in each step. On the other hand, a better approximation of the known area does not necessarily have to yield a better extrapolation.

The idea of best approximation is introduced in [26] for object oriented transform coding using DCT basis functions. First, we introduce the principle of best approximation in general according to [26]. Then we derive the best approximation for the special case of applying DFT basis functions [48] analogously to Section 3.4, i.e. we introduce the spatial-domain solution and then the frequency-domain solution for 2-D. Since the derivations for 3-D are analogously to 2-D, we give finally in brief the results for a spatio-temporal extrapolation using the best approximation.

4.3.1 Basic Principle of Best Approximation

We assume as previously that an appropriate basis function $\varphi_{u,v}[m, n]$ is already selected and that its index is included in the set of used basis functions $\mathcal{K}^{(\nu+1)}$

$$\mathcal{K}^{(\nu+1)} = \mathcal{K}^{(\nu)} \cup \{u, v\} \quad \text{if } (u, v) \notin \mathcal{K}^{(\nu)}. \quad (4.28)$$

The strategy how to select the basis function follows Section 3.1.2.

In each iteration, the residual is approximated by an updated version of the linear combination of all basis functions applied so far

$$\Delta g[m, n] = \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} \Delta c_{u,v} \varphi_{u,v}[m, n]. \quad (4.29)$$

Hence, the residual error in iteration $\nu+1$ is determined by adjusting all basis functions

used best possibly

$$r_w^{(\nu+1)}[m, n] = r_w^{(\nu)}[m, n] - \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} \Delta c_{u,v} \varphi_{u,v}[m, n] w[m, n]. \quad (4.30)$$

The revised error criterion taking all linear combinations of used basis functions into account is expressed by

$$E_{\mathcal{A}}^{(\nu+1)} = \sum_{(m,n) \in \mathcal{L}} w[m, n] \left(r^{(\nu)}[m, n] - \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} \Delta c_{u,v} \varphi_{u,v}[m, n] \right)^2. \quad (4.31)$$

The updates of all selected coefficients are obtained by the partial derivative of $E_{\mathcal{A}}^{(\nu+1)}$ with respect to all $\Delta c_{u,v}$

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c_{u,v}} = 0; \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)}. \quad (4.32)$$

The error criterion is maximally reduced in each step due to the consideration of all basis functions and yields a system of equations for each coefficient $\Delta c_{u,v}$; $\forall (u, v) \in \mathcal{K}^{(\nu+1)}$

$$\sum_{(k,l) \in \mathcal{K}^{(\nu+1)}} \Delta c_{k,l} \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k,l}[m, n] \varphi_{u,v}[m, n] = \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{u,v}[m, n] \quad (4.33)$$

$\forall (u, v) \in \mathcal{K}^{(\nu+1)}.$

The resulting system of equations can be solved uniquely for all $\Delta c_{u,v}$; $\forall (u, v) \in \mathcal{K}^{(\nu+1)}$ as long as the number of selected basis functions does not exceed the dimension of \mathcal{A} .

In order to solve (4.33) for the unknown updates $\Delta c_{k,l}$, we express this equation in matrix notation. Therefore, the indices of the basis functions which belong to $\mathcal{K}^{(\nu+1)}$ are numbered from 0 to ν

$$\mathcal{K}^{(\nu+1)} = \{(u_0, v_0); (u_1, v_1); \dots (u_\nu, v_\nu)\} = \{(k_0, l_0); (k_1, l_1); \dots (k_\nu, l_\nu)\}. \quad (4.34)$$

Further, all update variables and residuals are summarized in two $(\nu + 1) \times 1$ vectors $\Delta \mathbf{c}$ and $\mathbf{r}_w^{(\nu)}$, respectively,

$$\Delta \mathbf{c} = \begin{pmatrix} \Delta c_{k_0, l_0} \\ \Delta c_{k_1, l_1} \\ \vdots \\ \Delta c_{k_\nu, l_\nu} \end{pmatrix} \quad (4.35)$$

$$\mathbf{r}_w^{(\nu)} = \begin{pmatrix} \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{u_0, v_0}[m, n] \\ \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{u_1, v_1}[m, n] \\ \vdots \\ \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{u_\nu, v_\nu}[m, n] \end{pmatrix} \quad (4.36)$$

The scalar products of the weighted basis functions are summarized in the $(\nu + 1) \times (\nu + 1)$ matrix \mathbf{w}

$$\mathbf{w} = \begin{pmatrix} \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k_0, l_0}[m, n] \varphi_{u_0, v_0}[m, n] & \dots & \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k_\nu, l_\nu}[m, n] \varphi_{u_0, v_0}[m, n] \\ \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k_0, l_0}[m, n] \varphi_{u_1, v_1}[m, n] & \dots & \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k_\nu, l_\nu}[m, n] \varphi_{u_1, v_1}[m, n] \\ \vdots & \ddots & \vdots \\ \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k_0, l_0}[m, n] \varphi_{u_\nu, v_\nu}[m, n] & \dots & \sum_{(m,n) \in \mathcal{L}} w[m, n] \varphi_{k_\nu, l_\nu}[m, n] \varphi_{u_\nu, v_\nu}[m, n] \end{pmatrix}$$

Hence, (4.33) can be expressed in matrix notation

$$\mathbf{w} \Delta \mathbf{c} = \mathbf{r}_w^{(\nu)}$$

and solved for the unknown $\Delta \mathbf{c}$

$$\Delta \mathbf{c} = \mathbf{w}^{-1} \mathbf{r}_w^{(\nu)}. \quad (4.37)$$

The expansion coefficients are obtained by updating *all* available coefficients $c_{u,v}^{(\nu)}$ for $(u, v) \in \mathcal{K}^{(\nu+1)}$

$$c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c_{u,v} \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)}. \quad (4.38)$$

4.3.2 Geometrical Interpretation of Extrapolation by Best Approximation

After deriving the principle of best approximation, we interpret the best approximation also from a geometrical point of view and emphasize major differences to the selective extrapolation derived in Section 3.2. The task of extrapolating the original signal beyond its observable part is the same. The basis vectors in the different spaces \mathcal{A} and \mathcal{L} remain also unchanged. The only difference consists in *how* the approximation in subspace \mathcal{A} takes place.

Applying the best approximation, the vector of the parametric model $\mathbf{g}_\mathcal{A}^{(\nu)}$ is completely recalculated in each iteration step which is the most important difference compared to the selective approach. $\mathbf{g}_\mathcal{A}^{(\nu)}$ is obtained by projecting $\mathbf{f}_\mathcal{A}$ onto the subspace spanned by all basis functions $\varphi_\mathcal{A}[\lambda]$ selected so far. Hence, the vector of the approximation error is orthogonal to the subspace spanned by all basis functions selected so far after each iteration. Therefore, it is not possible to choose an already selected basis vector again. With selective extrapolation, the vector $\mathbf{r}_\mathcal{A}$ is projected onto the just selected basis function in each iteration and the result of the projection is added to the vector of the parametric model $\mathbf{g}_\mathcal{A}^{(\nu)}$. Hence, in selective extrapolation a basis vector can be selected

multiple times. The criterion for selecting a basis function remains unchanged. The final step is again the same, in both cases the parametric model $\mathbf{g}_{\mathcal{L}}$ is constructed in \mathcal{L} by the linear combination of the expansion coefficients $c_{\lambda}^{(\nu)}$ and basis functions $\varphi_{\mathcal{L}}[\lambda]$.

For exemplifying the best approximation we choose the same example as for the selective extrapolation. The area \mathcal{L} consists of three pixels, the missing area \mathcal{B} of one pixel and the support area \mathcal{A} of two pixels. The orthogonal basis functions in \mathcal{L} and the vector of the original signal $\mathbf{f}_{\mathcal{L}}$ are depicted in Fig. 4.13 (a). The location of $\mathbf{f}_{\mathcal{L}}$ within \mathcal{L} is sketched in for illustration purposes. In Fig. 4.13 (b) the reduction of the space \mathcal{L} to \mathcal{A} including all respective vectors is shown. Furthermore, the location where the plane \mathcal{A} spanned by the vectors $\varphi_{\mathcal{A}}[1]$, $\varphi_{\mathcal{A}}[2]$, $\varphi_{\mathcal{A}}[3]$ intersects the vector space \mathcal{L} is illustrated. In the first iteration the basis vector $\varphi_{\mathcal{A}}[2]$ is selected. Since the approximation of $\mathbf{f}_{\mathcal{A}}$ of the observable part of the original signal takes place in the subspace \mathcal{A} without any knowledge of $\mathbf{f}_{\mathcal{L}}$, the relation of the plane \mathcal{A} to the space \mathcal{L} is not shown anymore. The vector $c_2^{(1)}\varphi_{\mathcal{A}}[2]$ is obtained by projection of $\mathbf{f}_{\mathcal{A}}$ onto $\varphi_{\mathcal{A}}[2]$. This equals the parametric model in the first iteration $\mathbf{g}_{\mathcal{A}}^{(1)}$ which is displayed in Fig. 4.13 (c). The set of all basis functions selected so far consists of a single basis function in the first iteration, leading to the same parametric model $\mathbf{g}_{\mathcal{A}}^{(1)}$ for both approaches which is also shown by the consistency of Fig. 4.13 (c) with Fig. 3.2 (c). The criterion for selecting a basis function is the same for both approaches which leads to the choice of $\varphi_{\mathcal{A}}[3]$ in the second iteration. The vector of the parametric model is obtained by projecting $\mathbf{f}_{\mathcal{A}}$ onto the vector space spanned by $\varphi_{\mathcal{A}}[2]$ and $\varphi_{\mathcal{A}}[3]$. The two-dimensional vector $\mathbf{f}_{\mathcal{A}}$ is projected onto a two-dimensional vector space which allows to approximate $\mathbf{f}_{\mathcal{A}}$ perfectly by linear combination of $\varphi_{\mathcal{A}}[2]$ and $\varphi_{\mathcal{A}}[3]$

$$\mathbf{g}_{\mathcal{A}}^{(2)} = c_2^{(2)}\varphi_{\mathcal{A}}[2] + c_3^{(2)}\varphi_{\mathcal{A}}[3]$$

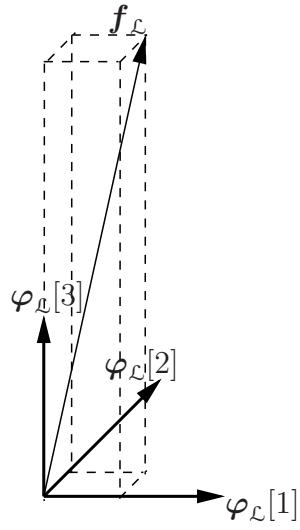
as also shown in Fig. 4.13 (d).

Finally, the vector of the parametric model $\mathbf{g}_{\mathcal{L}}^{(2)}$ within \mathcal{L} is composed by

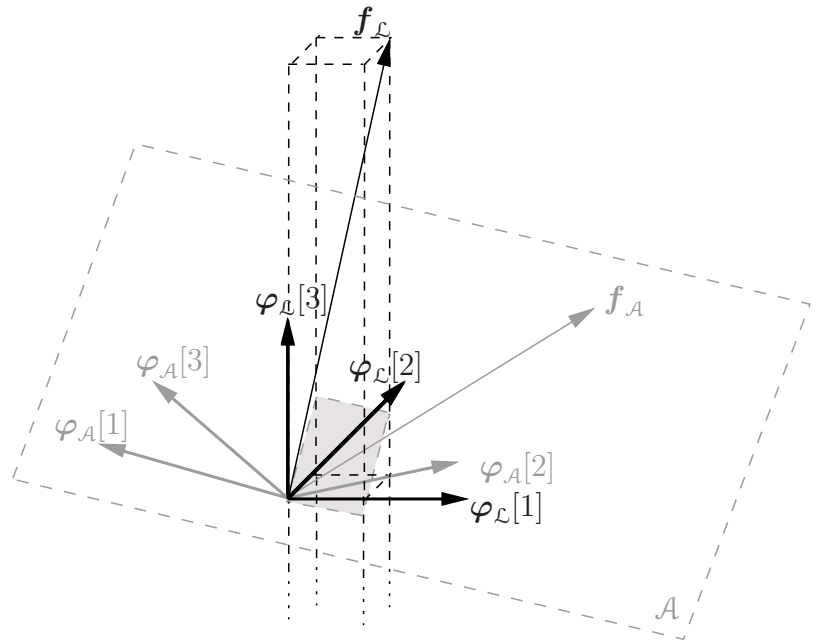
$$\mathbf{g}_{\mathcal{L}}^{(2)} = c_2^{(2)}\varphi_{\mathcal{L}}[2] + c_3^{(2)}\varphi_{\mathcal{L}}[3] \quad (4.39)$$

as illustrated in Fig. 4.13 (e). As for selective extrapolation, it should be emphasized that the vectors $\mathbf{g}_{\mathcal{L}}^{(2)}$ and $\mathbf{g}_{\mathcal{A}}^{(2)}$ point into different directions due to the different spaces of approximation and extrapolation. The dashed auxiliary lines to indicate the locations of the vectors within the respective space as in Fig. 4.13 (b) are discarded in order to avoid confusions.

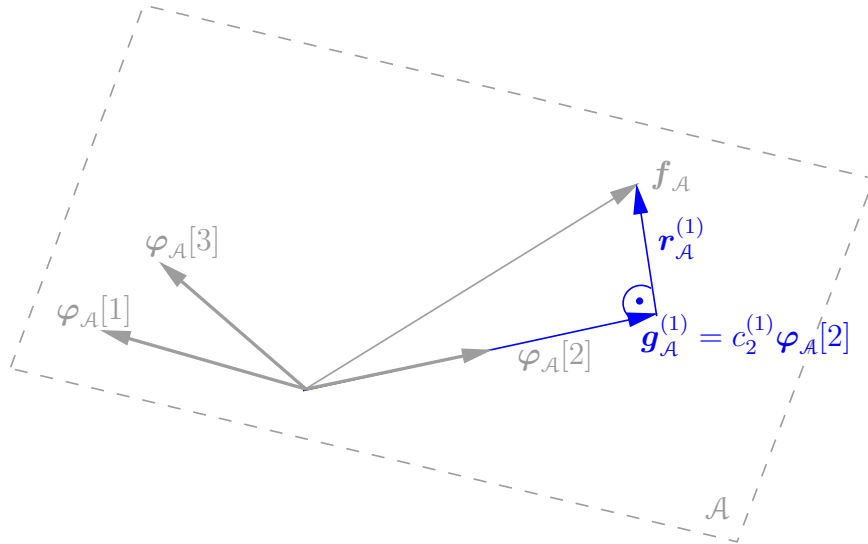
The expansion coefficient of the first selected basis function $\varphi_{\mathcal{A}}[2]$ has changed significantly from iteration 1 to 2 which is depicted in Fig. 4.13 (c) and (d), respectively, by the different expansion coefficients $c_2^{(1)}$ and $c_2^{(2)}$. This deviation occurs due to the non-orthogonal basis functions with respect to \mathcal{A} . The vector of the approximation error $\mathbf{r}_{\mathcal{A}}^{(\nu)}$ is orthogonal to the subspace spanned by the basis functions selected so far. The basis



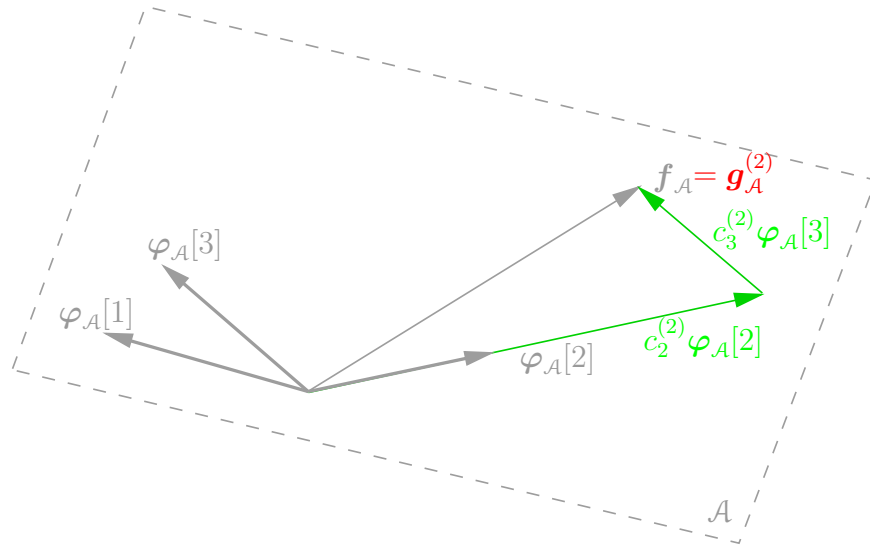
(a)



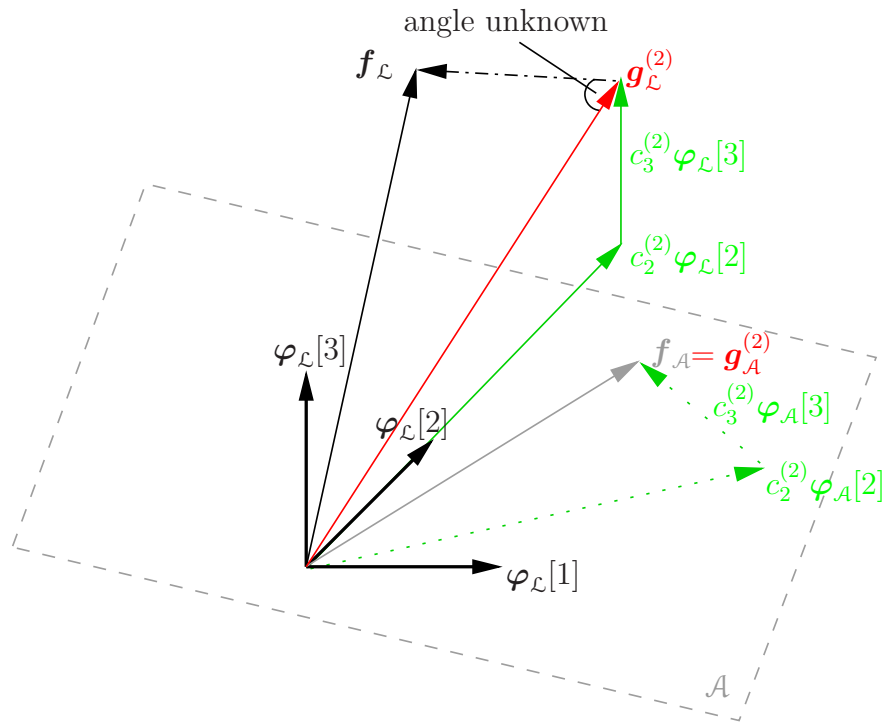
(b)



(c)



(d)



(e)

Figure 4.13: Geometrical interpretation of extrapolation by best approximation: (a) Original vector $f_{\mathcal{L}}$. (b) Reduction of vector $f_{\mathcal{L}}$ within space \mathcal{L} to $f_{\mathcal{A}}$ within observable support area \mathcal{A} . (c) 1st iteration: approximation of $f_{\mathcal{A}}$ by $g_{\mathcal{A}}^{(1)}$ within \mathcal{A} . (d) 2nd iteration: approximation of $f_{\mathcal{A}}$ by $g_{\mathcal{A}}^{(2)}$ within \mathcal{A} . (e) Aim of extrapolation by best approximation beyond \mathcal{A} : Composition of $g_{\mathcal{L}}^{(2)}$ within \mathcal{L} .

function which will be selected in the next iteration does due to the non-orthogonality of the basis functions in \mathcal{A} not only point into the direction of $\mathbf{r}_{\mathcal{A}}^{(\nu)}$, but has also components in the subspace spanned by the previously selected basis functions. The component into the direction of $\mathbf{r}_{\mathcal{A}}^{(\nu)}$ is required in order to decrease the approximation error. Since at the same time components are included which are located in the already considered subspace, the previously computed projection of $\mathbf{f}_{\mathcal{A}}$ on that subspace is not valid anymore. Consequently, all expansion coefficients of the parametric model have to be recalculated.

4.3.3 2-D Extrapolation by Best Approximation Using DFT Basis Functions

In order to perform the extrapolation for image signals by best approximation, we insert basis functions based on the 2-D DFT according to (3.29) into the equations derived in the last section. The definition of the parametric model $g[m, n]$ according to (3.32) remains unchanged. Due to the symmetry requirements for real-valued signals (3.30), the approximation of the parametric model in the next iteration by the update $\Delta g[m, n]$ taking all selected basis functions into account has to be revised to

$$\Delta g[m, n] = \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{u,v} \varphi_{u,v}[m, n] + \Delta c_{u,v}^* \varphi_{u,v}^*[m, n]) \quad (4.40)$$

Consequently, the residual error signal in the next iteration is computed by

$$r_{\mathbf{w}}^{(\nu+1)}[m, n] = r_{\mathbf{w}}^{(\nu)}[m, n] - \frac{1}{2MN} \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{u,v} \varphi_{u,v}[m, n] + \Delta c_{u,v}^* \varphi_{u,v}^*[m, n]) w[m, n] \quad (4.41)$$

For the resulting approximation error energy we obtain

$$\begin{aligned} E_{\mathcal{A}}^{(\nu+1)} &= \sum_{(k,l) \in \mathcal{L}} w[m, n] \left(r^{(\nu)}[m, n] \right. \\ &\quad \left. - \frac{1}{2MN} \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{u,v} \varphi_{u,v}[m, n] + \Delta c_{u,v}^* \varphi_{u,v}^*[m, n]) b[m, n] \right)^2. \end{aligned} \quad (4.42)$$

In order to minimize the residual error criterion $E_{\mathcal{A}}^{(\nu+1)}$ in iteration $\nu + 1$, the partial derivatives of the approximation error energy with respect to each $\Delta c_{u,v}$ and each $\Delta c_{u,v}^*$ have to become zero

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c_{u,v}} \stackrel{!}{=} 0 \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)} \quad \wedge \quad (4.43)$$

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c_{u,v}^*} \stackrel{!}{=} 0 \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)}. \quad (4.44)$$

This minimization procedure leads to the following system of equations for each $\Delta c_{u,v}^* \forall (u, v) \in \mathcal{K}^{(\nu+1)}$

$$\begin{aligned} & \sum_{(k,l) \in \mathcal{K}^{(\nu+1)}} \left(\Delta c_{k,l} \sum_{(m,n) \in \mathcal{L}} (\varphi_{k,l}[m, n] \varphi_{u,v}^*[m, n] w[m, n]) \right. \\ & \left. + \Delta c_{k,l}^* \sum_{(m,n) \in \mathcal{L}} (\varphi_{k,l}^*[m, n] \varphi_{u,v}[m, n] w[m, n]) \right) \\ & = 2MN \sum_{(m,n) \in \mathcal{L}} r_w^{(\nu)}[m, n] \varphi_{u,v}^*[m, n] \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)} \end{aligned} \quad (4.45)$$

and to a conjugate complex system of equations for $\Delta c_{u,v}$ for each $(u, v) \in \mathcal{K}^{(\nu+1)}$.

It is now possible, to express this equation in the frequency domain

$$\begin{aligned} \sum_{(k,l) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{k,l} W^*[k - u, l - v] + \Delta c_{k,l}^* W[k + u, l + v]) &= 2MNR_w^{(\nu)}[u, v] \\ &\quad \forall (u, v) \in \mathcal{K}^{(\nu+1)} \end{aligned} \quad (4.46)$$

Analogously, we obtain a system of equations for (4.43) being conjugate complex to (4.46).

The set $\mathcal{K}^{(\nu+1)}$ of the indices of the basis functions selected so far can be split into two disjoint subsets. The subset $\mathcal{K}_{\mathcal{M}}^{(\nu+1)}$ consists of the indices of the basis functions which belong to the set of real-valued DFT components \mathcal{M} (3.31). The second subset $\mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)}$ contains accordingly all indices of basis functions which are not part of \mathcal{M}

$$\mathcal{K}_{\mathcal{M}}^{(\nu+1)} = \{(u, v) \mid (u, v) \in \mathcal{K}^{(\nu+1)} \wedge (u, v) \in \mathcal{M}\} \quad (4.47)$$

$$\mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)} = \{(u, v) \mid (u, v) \in \mathcal{K}^{(\nu+1)} \wedge (u, v) \notin \mathcal{M}\} \quad (4.48)$$

The sum in (4.46) can be split into two sums using the two subsets, for the real-valued and complex-valued components. The sum over $(k, l) \in \mathcal{K}_{\mathcal{M}}^{(\nu+1)}$ can be simplified due to symmetry properties

$$\begin{aligned} & \sum_{(k,l) \in \mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)}} (\Delta c_{k,l} W^*[k - u, l - v] + \Delta c_{k,l}^* W[k + u, l + v]) \\ & + \sum_{(k,l) \in \mathcal{K}_{\mathcal{M}}^{(\nu+1)}} 2\Delta c_{k,l} W[k + u, l + v] = 2MNR_w^{(\nu)}[u, v] \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)} \end{aligned} \quad (4.49)$$

As previously described, we solve (4.49) and its conjugate complex equation using matrix notation. Therefore, we number again the indices of the basis functions which belong to $\mathcal{K}_{\mathcal{M}}^{(\nu+1)}$ from 0 to i and the indices of $\mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)}$ from $i + 1$ to ν

$$\mathcal{K}_{\mathcal{M}}^{(\nu+1)} = \{(u_0, v_0); \dots; (u_i, v_i)\} = \{(k_0, l_0); \dots; (k_i, l_i)\} \quad (4.50)$$

$$\mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)} = \{(u_{i+1}, v_{i+1}); \dots; (u_\nu, v_\nu)\} = \{(k_{i+1}, l_{i+1}); \dots; (k_\nu, l_\nu)\} \quad (4.51)$$

We compose the two $(2\nu - i + 1) \times 1$ vectors $\Delta \mathbf{c}$ and $\mathbf{R}_w^{(\nu)}$ and the matrix \mathbf{W} with dimension $(2\nu - i + 1) \times (2\nu - i + 1)$

$$\Delta \mathbf{c} = \begin{pmatrix} \Delta c_{k_0, l_0} \\ \vdots \\ \Delta c_{k_i, l_i} \\ \Delta c_{k_{i+1}, l_{i+1}} \\ \vdots \\ \Delta c_{k_\nu, l_\nu} \\ \Delta c_{k_{i+1}, l_{i+1}}^* \\ \vdots \\ \Delta c_{k_\nu, l_\nu}^* \end{pmatrix} \quad (4.52)$$

$$\mathbf{R}_w^{(\nu)} = \begin{pmatrix} R_w^{(\nu)} [u_0, v_0] \\ \vdots \\ R_w^{(\nu)} [u_i, v_i] \\ R_w^{(\nu)} [u_{i+1}, v_{i+1}] \\ \vdots \\ R_w^{(\nu)} [u_\nu, v_\nu] \\ R_w^{(\nu)*} [u_{i+1}, v_{i+1}] \\ \vdots \\ R_w^{(\nu)*} [u_\nu, v_\nu] \end{pmatrix} \quad (4.53)$$

$$\mathbf{W} = \begin{pmatrix} 2W[k_0 + u_0, l_0 + v_0] & \cdots & 2W[k_i + u_0, l_i + v_0] & W^*[k_{i+1} - u_0, l_{i+1} - v_0] & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots \\ 2W[k_0 + u_i, l_0 + v_i] & \cdots & 2W[k_i + u_i, l_i + v_i] & W^*[k_{i+1} - u_i, l_{i+1} - v_i] & \cdots \\ \\ 2W[k_0 + u_{i+1}, l_0 + v_{i+1}] & \cdots & 2W[k_i + u_{i+1}, l_i + v_{i+1}] & W^*[k_{i+1} - u_{i+1}, l_{i+1} - v_{i+1}] & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots \\ 2W[k_0 + u_\nu, l_0 + v_\nu] & \cdots & 2W[k_i + u_\nu, l_i + v_\nu] & W^*[k_{i+1} - u_\nu, l_{i+1} - v_\nu] & \cdots \\ \\ 2W[k_0 - u_{i+1}, l_0 - v_{i+1}] & \cdots & 2W[k_i - u_{i+1}, l_i - v_{i+1}] & W^*[k_{i+1} + u_{i+1}, l_{i+1} + v_{i+1}] & \cdots \\ \\ 2W[k_0 - u_\nu, l_0 - v_\nu] & \cdots & 2W[k_i - u_\nu, l_i - v_\nu] & W^*[k_{i+1} + u_\nu, l_{i+1} + v_\nu] & \cdots \\ \\ \cdots & W^*[k_\nu - u_0, l_\nu - v_0] & W[k_{i+1} + u_0, l_{i+1} + v_0] & \cdots & W[k_\nu + u_0, l_\nu + v_0] \\ \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & W^*[k_\nu - u_i, l_\nu - v_i] & W[k_{i+1} + u_i, l_{i+1} + v_i] & \cdots & W[k_\nu + u_i, l_\nu + v_i] \\ \\ \cdots & W^*[k_\nu - u_{i+1}, l_\nu - v_{i+1}] & W[k_{i+1} + u_{i+1}, l_{i+1} + v_{i+1}] & \cdots & W[k_\nu + u_{i+1}, l_\nu + v_{i+1}] \\ \cdots & \vdots & \vdots & \ddots & \vdots \\ \cdots & W^*[k_\nu - u_\nu, l_\nu - v_\nu] & W[k_{i+1} + u_\nu, l_{i+1} + v_\nu] & \cdots & W[k_\nu + u_\nu, l_\nu + v_\nu] \\ \\ \cdots & W^*[k_\nu + u_{i+1}, l_\nu + v_{i+1}] & W[k_{i+1} - u_{i+1}, l_{i+1} - v_{i+1}] & \cdots & W[k_\nu - u_{i+1}, l_\nu - v_{i+1}] \\ \vdots & \ddots & \vdots & \vdots & \ddots \\ \cdots & W^*[k_\nu + u_\nu, l_\nu + v_\nu] & W[k_{i+1} - u_\nu, l_{i+1} - v_\nu] & \cdots & W[k_\nu - u_\nu, l_\nu - v_\nu] \end{pmatrix} \quad (4.54)$$

Hence, we express (4.49) and its conjugate complex counterpart in matrix notation

$$\mathbf{W} \Delta \mathbf{c} = 2MN \mathbf{R}_w^{(\nu)} \quad (4.55)$$

and solve for the adaptation vector of all coefficients selected so far

$$\Delta \mathbf{c} = 2MN \mathbf{W}^{-1} \mathbf{R}_w^{(\nu)}. \quad (4.56)$$

If the matrix \mathbf{W} is ill-conditioned which might occur for many iterations, the inverse matrix \mathbf{W}^{-1} can be substituted by the pseudo-inverse matrix \mathbf{W}^+ referred to as Moore-Penrose pseudo-inverse [46]:

$$\mathbf{W}^+ = (\mathbf{W}^H \mathbf{W})^{-1} \mathbf{W}^H.$$

The expansion coefficients $c_{u,v}^{(\nu+1)}$, $\forall (u, v) \in \mathcal{K}^{(\nu+1)}$ in step $\nu+1$ are obtained by adding the adaptation update $\Delta c_{u,v}$ to the expansion coefficients $c_{u,v}^{(\nu)}$ in step ν

$$c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c_{u,v} \quad \forall (u, v) \in \mathcal{K}^{(\nu+1)} \quad (4.57)$$

For further approximations, the weighted residual error in the frequency domain is required. The transform of (4.41) yields

$$R_w^{(\nu+1)}[k, l] = R_w^{(\nu)}[k, l] - \frac{1}{2MN} \sum_{(u,v) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{u,v} W^*[k-u, l-v] + \Delta c_{u,v}^* W[k+u, l+v]) \quad (4.58)$$

With help of the derivations it is possible to implement the best approximation completely in the frequency domain. As in the selective case, the parametric model is initialized by $G^{(0)}[k, l] = 0$ yielding a residual error signal in the first iteration of

$$R_w^{(0)}[k, l] = \text{DFT}_{M,N}\{r_w^{(0)}[m, n]\} = \text{DFT}_{M,N}\{w[m, n]f[m, n]\}.$$

Further, only the transform of the weighting function $W[k, l] = \text{DFT}_{M,N}\{w[m, n]\}$ and an inverse transform of the parametric model $g[m, n] = \text{IDFT}_{M,N}\{G[k, l]\}$ after termination are required. As termination criterion also the drop of the decrease of the residual error criterion below the threshold E_{\min} (3.23) is applied.

4.3.4 3-D Extrapolation by Best Approximation Using DFT Basis Functions

Three-dimensional extrapolation using best approximation works according to the same principle as for two-dimensions. Therefore, we skip the derivations and give the most important equations for applying basis functions based on the DFT. They are given mainly in the frequency domain as required for a corresponding frequency domain implementation.

The selection of the basis function to be used for further approximating the residual is analogously to the selective case based on (4.24). Accordingly, the residual error is further approximated by

$$\Delta g[m, n, t] = \sum_{(u,v,q) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{u,v,q} \varphi_{u,v,q}[m, n, t] + \Delta c_{u,v,q}^* \varphi_{u,v,q}^*[m, n, t]) \quad (4.59)$$

The updates are obtained by equating the partial derivatives of the approximation error energy $E_{\mathcal{A}}^{(\nu+1)}$ with respect to the updates of all basis functions selected so far to zero, i.e.,

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c_{u,v,q}} \stackrel{!}{=} 0 \quad \forall (u, v, q) \in \mathcal{K}^{(\nu+1)} \quad \wedge \quad (4.60)$$

$$\frac{\partial E_{\mathcal{A}}^{(\nu+1)}}{\partial \Delta c_{u,v,q}^*} \stackrel{!}{=} 0 \quad \forall (u, v, q) \in \mathcal{K}^{(\nu+1)}. \quad (4.61)$$

(4.61) leads to the following equation

$$\begin{aligned} & \sum_{(k,l,p) \in \mathcal{K}_{\mathcal{M}}^{(\nu+1)}} (\Delta c_{k,l,p} W^*[k - u, l - v, p - q] + \Delta c_{k,l,p}^* W[k + u, l + v, p + q]) \\ & + \sum_{(k,l,p) \in \mathcal{K}_{\mathcal{M}}^{(\nu+1)}} (2\Delta c_{k,l,p} W[k + u, l + v, p + q]) = 2MNT R_{\mathbf{w}}^{(\nu)}[u, v, q] \quad \forall (u, v, q) \in \mathcal{K}^{(\nu+1)} \end{aligned} \quad (4.62)$$

where the set $\mathcal{K}^{(\nu+1)}$ is again split into two disjoint subsets $\mathcal{K}_{\mathcal{M}}^{(\nu+1)}$ and $\mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)}$

$$\mathcal{K}_{\mathcal{M}}^{(\nu+1)} = \{(u, v, q) \mid (u, v, q) \in \mathcal{K}^{(\nu+1)} \wedge (u, v, q) \in \mathcal{M}\} \quad (4.63)$$

$$\mathcal{K}_{\overline{\mathcal{M}}}^{(\nu+1)} = \{(u, v, q) \mid (u, v, q) \in \mathcal{K}^{(\nu+1)} \wedge (u, v, q) \notin \mathcal{M}\}. \quad (4.64)$$

A conjugate complex equation to (4.62) is obtained for (4.60).

In order to solve the system of equations for the 3-D case, the equations have to be expressed in matrix notation according to the 2-D procedure (4.52-4.56).

The expansion coefficients are updated by

$$c_{u,v,q}^{(\nu+1)} = c_{u,v,q}^{(\nu)} + \Delta c_{u,v,q} \quad \forall (u, v, q) \in \mathcal{K}^{(\nu+1)} \quad (4.65)$$

Finally, the residual error in the frequency domain $R_{\mathbf{w}}^{(\nu)}$ can be expressed in the next iteration by

$$\begin{aligned} R_{\mathbf{w}}^{(\nu+1)}[k, l, p] &= R_{\mathbf{w}}^{(\nu)}[k, l, p] - \frac{1}{2MNT} \\ &\times \sum_{(u,v,q) \in \mathcal{K}^{(\nu+1)}} (\Delta c_{u,v,q} W[k - u, l - v, p - q] + \Delta c_{u,v,q}^* W[k + u, l + v, p + q]) \end{aligned} \quad (4.66)$$

As it has been shown, the algorithm derived for 2-D can be extended straightforwardly to the 3-D case including its efficient implementation in the frequency domain.

4.4 Frequency Selective Extrapolation vs. Best Approximation

In the following, the extrapolation performance of frequency selective extrapolation is compared to the performance of best approximation as introduced in Section 4.3. Therefore, best approximation is applied to the already familiar representative example of estimating an unknown block of 16×16 pixels from its known surrounding. In practice, this corresponds to concealment of isolated 16×16 block losses.

First, the results obtained for 2-D extrapolation are presented in Section 4.4.1 and the corresponding 3-D results in Section 4.4.2. The interpretation of the results is given in Section 4.4.3.

4.4.1 Extrapolation Result Using 2-D Best Approximation

The 2-D isotropic weighting function as introduced in Section 3.5.2 is also a suitable weighting function for best approximation in our extrapolation context. Based on the investigations in [48], the following parameters are chosen for 16×16 block losses: the decay parameter $\hat{\rho} = 0.8$, the support area includes 16 known surrounding pixels, the DFT size is 64×64 , and the termination criteria are chosen to maximally 20 iterations and $\Delta_{\min} = 1$.

First, the energy of the error between the approximation and the original signal in

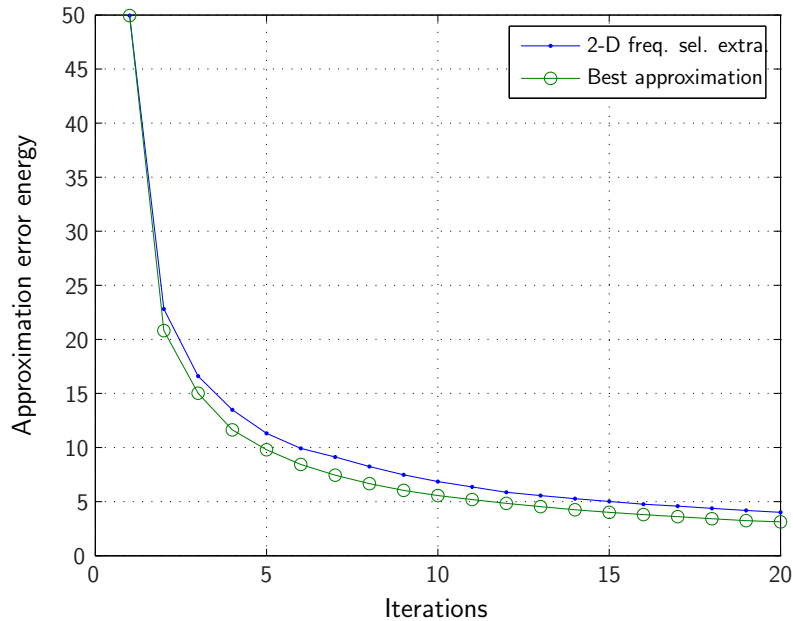


Figure 4.14: Energy of approximation error for *Lena* for isolated 16×16 block losses. Set of parameters used: $\hat{\rho} = 0.8$, $\Delta_{\min} = 0$, 16 surrounding pixels, DFT size 64×64 .

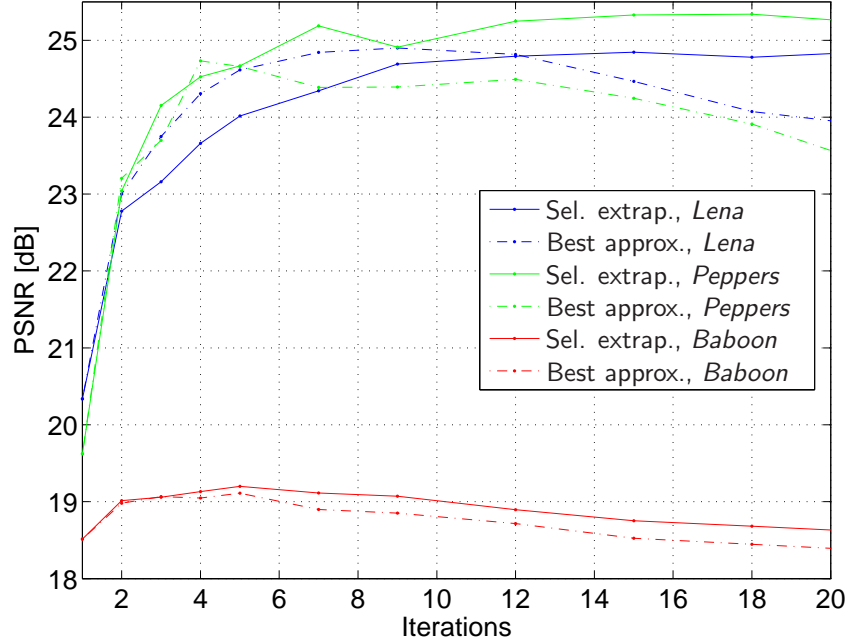


Figure 4.15: PSNR results comparing **best approximation** to **selective extrapolation** for isolated 16×16 block losses. Set of parameters used: $\hat{\rho} = 0.8$, $\Delta_{\min} = 0$, 16 surrounding pixels, DFT size 64×64 .

the support area is evaluated. Therefore, the energy of the approximation error per pixel obtained by best approximation is compared to frequency selective extrapolation with respect to the number of iterations for *Lena* in Fig. 4.14. For both methods, the parameters mentioned above are applied. The algorithms work identical in the first iteration and produce therefore the same result. In the second iteration, the best approximation yields a stronger decrease in approximation error energy. With an increasing number of iterations, the curves run nearly parallel at the interval of the gain achieved in the second iteration.

So far, we have evaluated the energy of the approximation error in the known area. Next, we evaluate the performance of the extrapolation in the unknown area objectively as well as subjectively.

The extrapolation performance for best approximation is compared to frequency selective extrapolation in terms of PSNR in Fig. 4.15. The results are presented for concealing 16×16 losses in *Lena*, *Peppers*, and *Baboon*. For *Lena*, the best approximation yields the better result in the first few iterations. Then, the performance of the best approximation decreases, whereas the performance of frequency selective extrapolation remains nearly constant. For *Peppers*, the performance of both algorithms is similar until 5 iterations are reached. As for *Lena*, the performance of the best approximation decays rapidly afterwards. In contrast, the performance increases further for frequency selective extrapolation with an increasing number of iterations. For *Baboon*, the performance of the



Figure 4.16: Extrapolation by **2-D best approximation** for 16×16 block losses, 24.11 dB.



Figure 4.17: Result by **2-D frequency selective extrapolation** for 16×16 losses, 24.83 dB.

two algorithms shows a similar behavior, although the performance of frequency selective extrapolation is slightly better than best approximation.

In summary, although the best approximation yields the best possible *approximation of the support area* in each iteration, in general it does not yield the better *extrapolation of the unknown area*. Only for a few iterations, the best approximation may achieve better results, but with an increasing number of iterations the frequency selective extrapolation performs better. The performance of best approximation even decreases rapidly for an increasing number of iterations. Besides, the complexity of the best approximation is computational very expensive due to the matrix inversion required in each iteration, especially for an increasing number of iterations.

Visual results are presented in the following in order to confirm the objective results. Fig. 4.16 illustrates the result obtained with best approximation for *Lena* and isolated 16×16 block losses. For comparison, the corresponding result for frequency selective extrapolation is shown in Fig. 4.17. Although single blocks are excellent concealed with best approximation, other blocks suffer from strong artifacts as for instance above the hat or at the shoulder. Similar artifacts, but by far not as strong, occur also for frequency selective extrapolation. Besides, the DFT basis images are more pronounced for the best approximation. Furthermore, the artifacts occur also in the chrominance components and lead to color artifacts. The same visual effects can be observed for *Peppers* and *Baboon* according to [48].

4.4.2 Extrapolation Result Using 3-D Best Approximation

The 3-D isotropic model as introduced in (4.27) is also applied as weighting function for extrapolation by 3-D best approximation. According to the investigations in [48], the following parameters are selected for the estimation of 16×16 block losses in this section: the decay parameter $\hat{\rho} = 0.8$, the support area includes in spatial direction 16 surrounding pixels and in temporal direction 2 previous and 2 subsequent frames, and the DFT size equals $64 \times 64 \times 16$.

The results comparing the extrapolation performance in terms of PSNR with respect to the number of iterations are depicted in Fig. 4.18 for the sequences *Flowergarden*, *Crew*, and *Vimto* in CIF format. The comparison of the 3-D extrapolation approaches shows a similar behavior as already observed for 2-D. Better results may be achieved using best approximation only for a few iterations. An increasing number of iterations does not improve the result anymore, but the performance may even decrease rapidly. In contrast, the performance for frequency selective extrapolation yields the better result with an increasing number of iterations compared to best approximation and it does not decrease anymore. In [48] it was further observed that the performance of best approximation generally increases with an increasing size of the support area, i.e. in the 3-D context

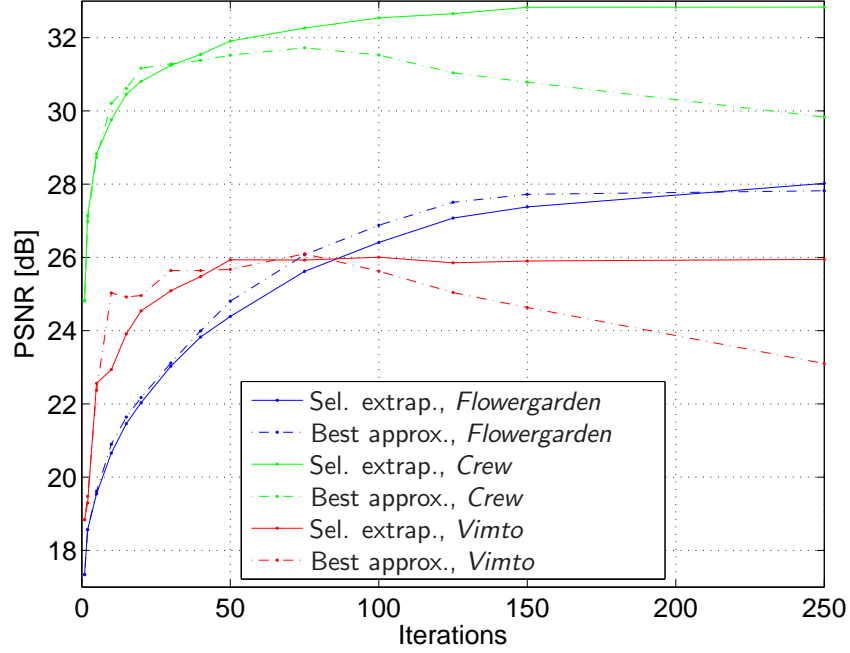


Figure 4.18: PSNR results comparing **3-D best approximation** to **3-D selective extrapolation** for isolated 16×16 block losses. Set of parameters used: $\hat{\rho} = 0.8$, $\Delta_{\min} = 0$, 16 surrounding pixels, $N_V = N_N = 2$ frames, DFT size $64 \times 64 \times 16$.

with more frames being involved. Visual results are not presented here, but as it has been discussed in [48] similar effects as for 2-D can be observed.

4.4.3 Discussion of Results

The geometrical interpretations as described in Section 4.3.2 for best approximation and in Section 3.2 for selective extrapolation are the basis for the following discussions. For best approximation, in each iteration the vector of the approximation error is projected onto the selected basis vectors. The basis vectors in the subspace of the support area are not mutually orthogonal. The considered image signals commonly consist only of a few dominant frequencies. Using DFT like functions as basis functions, the basis functions can be interpreted as the frequency components of the signal and the expansion coefficient as the corresponding frequency amplitude. If the number of iterations corresponds approximately to the number of dominant frequencies, these frequencies are selected and the projection onto the corresponding subspace yields a good parametric model. If the number of iterations exceeds the number of dominant frequencies, the performance of the extrapolation result decreases, because frequencies are selected which are only weakly present in the original signal. The inclusion of these frequencies into the parametric model leads to a strong change of the dominant frequencies resulting from the non-orthogonality of the

basis functions in the approximation subspace. The components of the weak frequencies in direction of the dominant frequencies have to be compensated. Thus, the amplitudes of the dominant frequencies of the signal do not correspond to the corresponding amplitudes in the parametric model anymore. The parametric model consists of a different spectral composition than the original signal, although it yields the best possible approximation of the known area. This also explains why the performance of the best approximation decreases rapidly with an increasing number of iterations.

In order to confirm these statements, simulations with artificially generated signals are evaluated. A white noise signal is generated and subsequently a varying number of dominant frequencies are inserted. Single frequencies are excited with an amplitude five times as high as the mean of the amplitudes of all frequencies. For a 16×16 block with a 8×8 block loss, the PSNR is measured for concealing with best approximation and frequency selective extrapolation. The PSNR difference for one to ten dominant frequencies with respect to the number of iterations is depicted in Fig. 4.19. If less iterations are considered than the signal has dominant frequencies, the performance of best approximation and frequency selective extrapolation are comparable. In case the number of dominant frequencies equals approximately the number of iterations, the best approximation performs better than frequency selective extrapolation which is visible at the valley of the PSNR value along the main diagonal. The performance of best approximation becomes worse than frequency selective extrapolation if the number of iterations exceeds the number of dominant frequencies.

Further investigations in [48] have shown that the number of iterations required for a sufficient visual quality of the extrapolated areas exceeds the number of dominant frequencies for 2-D as well as 3-D. Hence, for a desired quality of the extrapolation result, the best approximation performs worse than frequency selective extrapolation. However, the performance of best approximation generally increases if the support area becomes larger in relation to the area to be extrapolated according to [48]. The reason is that the lack of mutual orthogonality of the basis functions in the subspace of the support area becomes less significant.

In summary, although the best approximation achieves the best possible approximation of the support area, it leads to an inferior extrapolation result in the unknown area compared to frequency selective extrapolation. Furthermore, the best approximation may lead to severe visual artifacts.

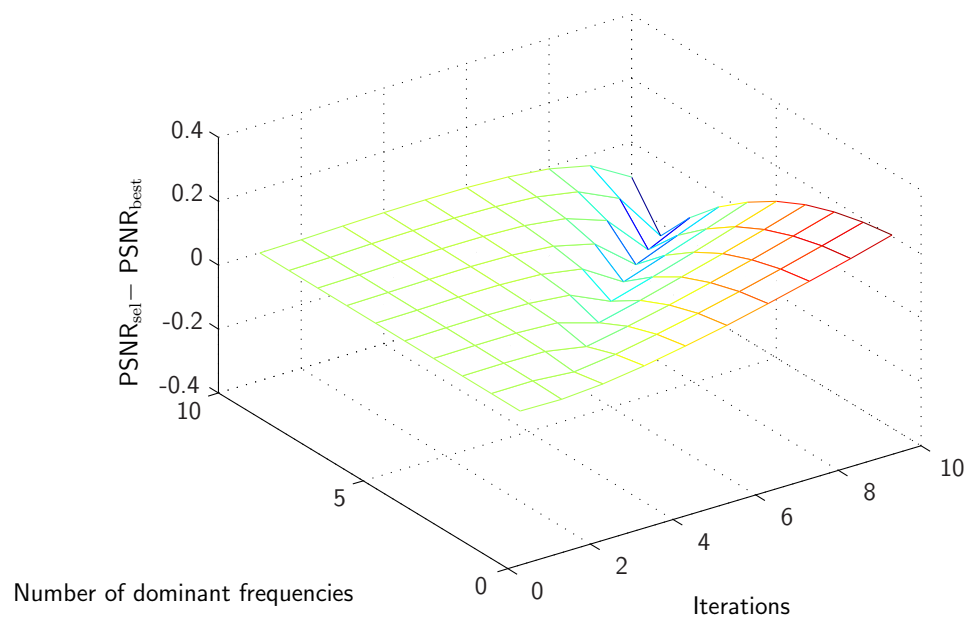


Figure 4.19: PSNR difference comparing **selective extrapolation** to **best approximation** for a 16×16 block with 8×8 losses, DFT size 16×16 .

5 Error Concealment of Lost Block Coded Data

Multimedia services require error-free transmission of digital image and video data over distinct communication channels. The data is vulnerable to transmission errors due to compression. In case of transmission over unreliable channels as a transmission channel in mobile communications or the Internet, transmission errors may lead to data losses. Hence, a robust data stream is required using robustness tools like packetization and synchronization markers in order to localize the data losses [59]. Furthermore, the application of robustness techniques allows to decode the images or video frames except for the lost regions. At the decoder side error concealment is applied in order to conceal the effects of data losses. In case of video transmission, concealment is especially important in order to limit error propagation caused by predictive coding.

In this chapter, the frequency selective extrapolation method derived in the last chapters is applied to various concealment tasks in image and video communications. Depending on the coding scheme, transmission errors cause different type of losses:

- For block-based coders like JPEG or MPEG transmission errors cause block losses. In predictive coding commonly applied, the data loss may even lead to consecutive block losses.
- In wavelet-based coding like JPEG2000, the visual effects of errors are different because transmission errors cause losses of wavelet coefficients in the subband.

First, a review of related methods developed for error concealment in block based coding is given in Section 5.1. Then, the concept of frequency selective extrapolation is applied to error concealment of block losses in Section 5.2. Investigations analyze the performance and the impact of parameters in simulations. A comparison of the concealment performance of frequency selective extrapolation to related techniques follows in Section 5.3. Finally, frequency selective extrapolation is realized in a video decoder as concealment method. Based on the results of Section 5.2, the developed extrapolation methods are integrated into the state-of-the-art video decoder H.264/AVC (Advanced Video Coding) [23] as concealment feature in Section 5.4. The application of frequency selective extrapolation to concealment of lost wavelet coded data and its integration into the JPEG2000 decoder is discussed in Chapter 6.

5.1 Related Work in Concealment of Lost Block Coded Data

Spatial and temporal correlations of the image and video signal, respectively, are exploited in order to reconstruct the missing areas by concealment. Depending on the information used for concealment, three categories can be distinguished: spatial, temporal, and spatio-temporal methods. Spatial concealment methods make use of the spatial correlations in image and video signals. The image content of damaged blocks is estimated from the correctly received neighboring blocks. Temporal concealment methods take advantage of temporal correlations in video signals by estimating the missing data from previously transmitted frames. Both, spatial as well as temporal approaches fail in certain concealment scenarios: Spatial methods often have problems to reconstruct high-frequency and detailed image contents, whereas temporal approaches may fail in case of complex and fast object motions, scene cuts, or luminance variations. Positive synergy effects are obtained by combining the spatial and temporal approaches to spatio-temporal methods which simultaneously exploit spatial *and* temporal correlations of the video signal.

Several spatial concealment methods are reviewed in Section 5.1.1. In Section 5.1.2, temporal error concealment methods are briefly described. The performance of frequency selective extrapolation is compared to the performance achieved by related techniques in Section 5.3.

5.1.1 Spatial Concealment of Block Losses

Spatial concealment methods are applied to losses occurring in image transmission or video transmission of Intra coded frames. There exists a variety of different techniques how to estimate the unknown area from the correctly received surrounding. One possibility is to apply spectral estimation methods and use techniques as described in Section 2.2.2. Other algorithms work in the spatial-domain.

A standard approach of Wang et. al [60] allows the reconstruction of smooth areas. Based on the assumption that the image content is changing smoothly, the algorithm tries to restore the transition along the block boundary as smooth as possible. Hence, edges can not be restored and the image appears blurry.

Another approach based on Projection onto Convex Sets (POCS) [56] belongs to the category of band-limited extrapolation by spectral estimation reviewed in Section 2.2.2. The surrounding blocks are first classified as either monotonous areas or areas containing edges. Subsequently, the missing block including the surrounding blocks are transformed by a DFT. According to its classification, the block is filtered. In case of a smooth area, a lowpass filter according to (2.7) is applied and otherwise a bandpass in direction of the edge (2.8). After the inverse DFT the surrounding blocks are replaced by the correct samples

and the amplitudes are clipped to the valid range. The procedure is repeated iteratively and requires two transforms per iteration implying a high computational complexity. A single occurring edge can be restored but multiple orientations can not be resolved. The reconstruction of smooth areas does not always succeed due to the overshoot effects caused by band-limited extrapolation methods. Further, block effects occur at the block boundaries.

The approach [8] places emphasis on a very efficient implementation. The missing block and the adjacent correct pixels are transformed to the DCT-domain. Based on the assumption that higher frequencies are less important than lower frequencies, the DCT-coefficients corresponding to higher frequencies are set at zero. Solving the system of equations for the unknown pixels yields the missing block. Diagonal edges can not be restored as well as details due to the cancellation of the high frequencies.

The method [49] is performed in the spatial-domain. The missing block is expressed by a weighted linear combination of the surrounding blocks. The weighting coefficients are obtained with a least squares solution minimizing the boundary error between the known and missing blocks. Partitioning the surrounding blocks can yield an improved reconstruction of diagonal edges.

The pixel-based method of [63] predicts each pixel from the available next neighbors. The missing block is pixel-based reconstructed from eight directions and given by a weighted linear combination of these reconstructed blocks. Hence, the method is computationally very complex. In principle the method achieves good results, edges can be restored but the reconstruction of details is not always satisfying.

5.1.2 Spatio-Temporal Concealment of Block Losses

Temporal concealment methods are commonly applied to conceal losses of temporally predicted macroblocks in Inter coding mode as they occur for erroneous video transmission. These methods try to exploit similarities along the time axis of subsequent frames. As already mentioned in Section 2.4, we are not aware of any existing methods using spectral estimation for temporal concealment. Common approaches are based on the motion vector recovery principle.

The simplest temporal error concealment technique places the block spatially located at the same position in the previous frame instead of the lost block. This approach is called Temporal Block Replacement (TR) [59]. The replacement with the block spatially located at the same position in the previous frame corresponds to a motion vector of zero length in block based motion compensation. For non-moving image contents this algorithm is well suited for concealment. In case of moving contents, however, this algorithm leads to severe motion artifacts. Therefore, other methods try to restore the motion vector of the missing block in order to compensate the motion.

The Boundary Matching Algorithm (BMA) [29] takes advantage of temporal information in order to conceal lost blocks. The match of the block compensated with different motion vectors and its neighboring correctly received blocks is evaluated by the difference of adjacent pixel values at the block boundaries. Thereby, the zero motion vector, the vector of the block in the previous frame, the vectors of the neighboring blocks, the median, and the average of the neighboring vectors are tested. The vector which results in a minimum boundary error is selected. The BMA assumes that only the motion vectors are lost. The Extended Boundary Matching Algorithm (EBMA) is applied if also the prediction error signal is lost. Additionally to the motion vectors the prediction error signals of the neighboring blocks and an assumed zero prediction error are used. The combination of prediction error and motion vector is selected which is minimizing the boundary error.

The BMA method exploits the spatial correlation of pixels for the recovery of the motion vector. In contrast, the Decoder Motion Vector Algorithm (DMVE) [67] uses the temporal correlation of succeeding frames for the estimation of the motion vector. The surrounding of the lost block consisting of a frame of two to eight lines of pixels is used to find the block in the previous frame which matches best. Also already concealed pixels in the surrounding are taken into account. Full search in the previous frame is performed in order to find the best match. Several methods are suggested in order to speed up the algorithm. For example, the possible set of motion vectors can be limited to a set of test vectors which is referred to as optional candidate search. In [19], additional post processing by a deblocking filter is proposed.

A motion compensated frame interpolation method [66] can also be used for error concealment. First, the motion vectors embedded in the bit stream of a coded video are examined if they reflect the true motion. For the non-reliable motion vectors, overlapped block bidirectional motion estimation is used. The obtained motion vectors are post processed in order to smooth the motion vector field. Bidirectional Overlapped Block Motion Compensation (OBMC) is recovering the lost macroblock.

In [20], a fading scheme for block loss recovery is suggested. A boundary error criterion based on BMA determines if either spatial, temporal, or fading of both methods is used for recovering the lost image samples in a macroblock. The boundary error is propagated into the lost MB by interpolation and determines pixelwise which method is to be applied for concealment.

The previously described BMA, DMVE, and OBMC work in two steps: First, the motion vector is restored and, second, the motion is compensated. Other methods base on the principle of motion vector field interpolation computing one motion vector per pixel [6, 7]. They aim at reproducing not only translational motion but also rotations and scalings. The motion field is interpolated and compensated for each pixel from the motion vectors of the neighboring blocks. However, this approach strongly depends on the reception of the motion vectors of the neighboring blocks. Therefore, the motion field

is often obtained by a combination of the motion interpolation and the BMA which is referred to as Motion Field Interpolation Boundary Matching [6]. The extension to using multiple frames for motion compensation is described in [7].

5.2 Concealment using Frequency Selective Extrapolation

In the following, frequency selective extrapolation is applied to concealment of transmission losses. Depending on the surrounding available data, two cases are addressed: In Section 5.2.1, 2-D frequency selective extrapolation is used for spatial concealment. 3-D frequency selective extrapolation is applied to spatio-temporal concealment in 5.2.2.

We investigate both, isolated and consecutive macroblock (MB) losses of 16×16 pixels that typically appear in erroneous image and video transmission.

5.2.1 Spatial Concealment Using 2-D Frequency Selective Extrapolation

In this section, 2-D frequency selective extrapolation as derived in Section 3.4 is applied to spatial concealment. For specifying a suitable problem-specific weighting function, the principle of applying frequency selective extrapolation to the estimation of centrally located unknown blocks surrounded by known data was already explained in Section 3.5. An isotropic model which decays radial symmetrically and involves pixels closer to the missing area stronger into the extrapolation than distant pixel achieved large gains up to 2.2 dB compared to a simple binary weighting function. The representative example chosen for these general investigations in Section 3.5 corresponds to spatial concealment of isolated block losses. Therefore, using frequency selective extrapolation together with the

Table 5.1: Parameters for concealment of 16×16 losses using frequency selective extrapolation with isotropic weighting function (Section 3.5.3).

$\hat{\rho}$		0.74
no. of known pixel		13
Termination	max. no. of iterations	11
criterion	Δ_{\min}	15
DFT size		64×64
Consecutive losses: attenuation of concealed blocks		0.1

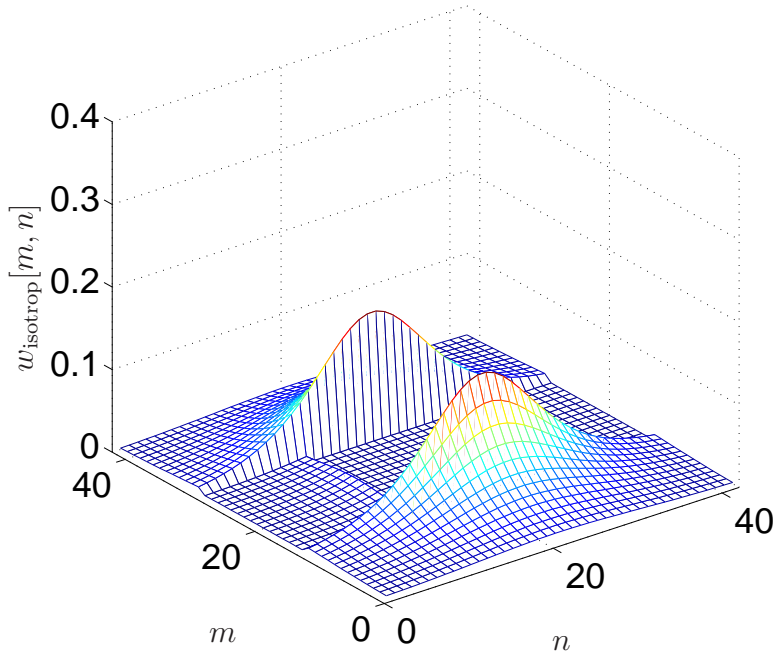


Figure 5.1: Weighting function using the isotropic model with $\hat{\rho}=0.8$ for consecutive lost blocks of size 16×16 (area \mathcal{B}). 13 surrounding pixels (support area \mathcal{A}) are used for reconstructing the lost area.

isotropic weighting function (3.52) as concealment method provides very convincing results for both, subjective and objective evaluations done in Section 3.5.3 and Section 3.5.4. A fixed set of parameters can be specified which allows to sufficiently restore different textures such as smooth areas, edges, and detailed areas. For convenience, the parameter selection is reviewed in Table 5.1. The visual results for concealment of isolated block losses achieved with 2-D frequency selective extrapolation correspond to the results presented in Section 3.5.4 and are therefore already given by Fig. 3.16.

In the following, concealment using 2-D frequency selective extrapolation for consecutive block losses is discussed. The block next to the missing one in raster scan order is not available in case of consecutive block losses. Furthermore, the previous block is an already extrapolated block weighted by 0.1 in order to include it in the concealment procedure but with limiting influence to avoid error propagation. The resulting weighting function is shown in Fig. 5.1. For comparison, the resulting weighting function in case of isolated block losses is already given in Fig. 3.13.

Finally, visual results are presented for consecutive macroblock losses of 16×16 pixels. The losses are concealed macroblock-wise in raster scan order. Blocks which are already concealed are taken into account with only limited influence, applying the weighting function depicted in Fig. 5.1. Again, the parameters according to Table 5.1 are used. The test images with the losses and the corresponding concealed images are displayed in

Fig. 5.2. For the image *Lena* artifacts appear in Fig. 5.2 (b) at horizontal structures such as the shoulder. There, the loss covers the transition between shoulder and background which can not be restored. Artifacts also arise at diagonal structures like the hat. Also details are lost which are entirely covered by the loss like the pupils of the *Baboon* in Fig. 5.2 (f). However, the visual impression of the images *Peppers* in Fig. 5.2 (d) and *Baboon* in Fig. 5.2 (f) with mainly vertical structures is very convincing. For the image *Baboon* also the structure of the fur and the beard can be restored. In consideration of the extent of the losses the visual quality of the extrapolated areas is remarkable.

The conclusions from these evaluations allow to integrate the method as concealment feature into the reference software of the H.264/AVC decoder in Section 5.4.3.

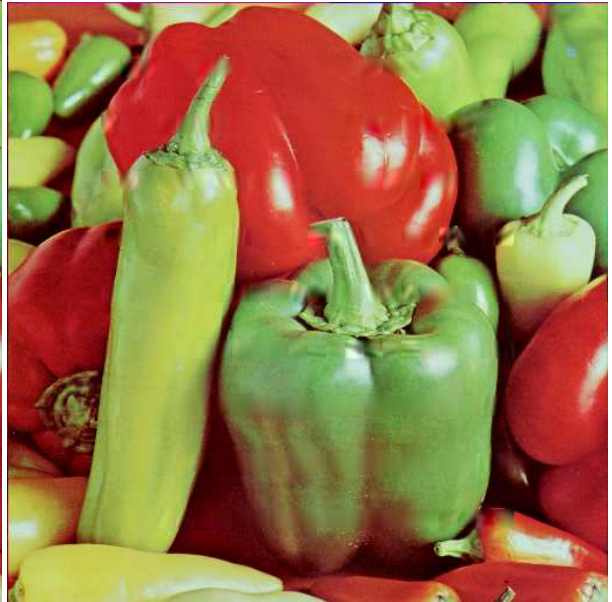


(a)

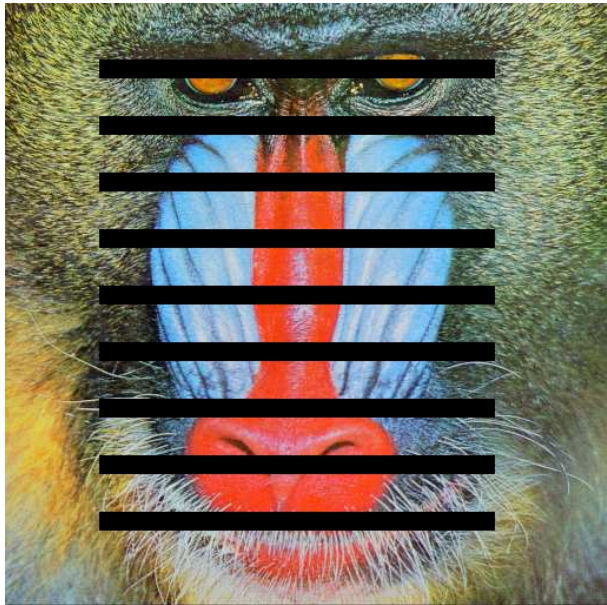
(b)



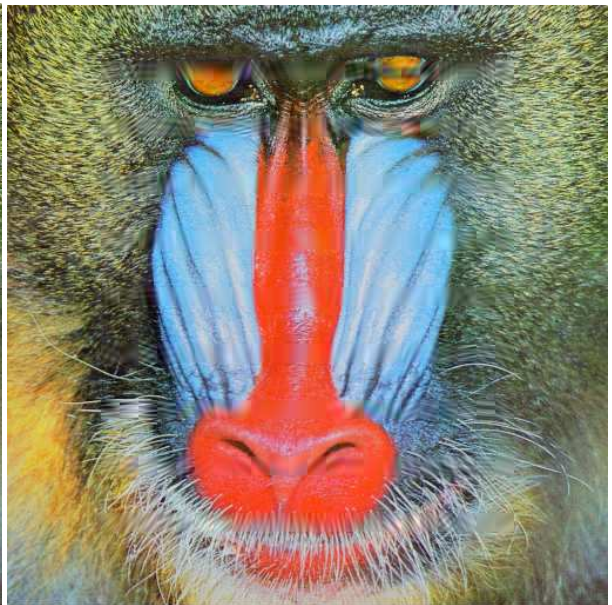
(c)



(d)



(e)



(f)

Figure 5.2: Test images with consecutive block losses and corresponding concealed images using the **isotropic weighting function** at DFT size 64×64 with parameters according Table 5.1.

5.2.2 Spatio-Temporal Concealment Using 3-D Frequency Selective Extrapolation

In this section, 3-D frequency selective extrapolation as derived in Section 4.2 is applied to spatio-temporal concealment of isolated and consecutive MB losses. The principle of spatio-temporal extrapolation from a 3-D volume into a centrally located block is already described in Section 4.2.3. For isolated MB losses occurring in video communications, this example corresponds to spatio-temporal concealment of a MB of 16×16 pixels. In Section 4.2.3, the problem-specific weighting function $w[m, n, t]$ relying on the 3-D isotropic model (4.27) is specified where the importance of pixels involved in the extrapolation decreases with distance to the lost block in spatial as well as temporal direction. Subsequent evaluations have shown the suitability of this weighting function. The principle shape of the 3-D isotropic model is illustrated in Fig. 4.3. The resulting weighting function $w[m, n, \tau]$ for the frame τ containing the missing block corresponds to the 2-D weighting function shown in Fig. 3.13.

In case consecutive losses are occurring, the already concealed block in raster scan order is taken into account with limited influence. Here, the concealed block is weighted with 0.1 as for 2-D, see Fig. 5.1. Subjective results are also presented. The conclusions from these evaluations are the basis for integrating this method as spatio-temporal concealment feature into the reference software of the H.264/AVC decoder in Section 5.4. The impact on subsequent frames caused by error propagation due to predictive coding is addressed in Section 5.4.3.

As for 2-D, a fixed set of parameters can be specified using the 3-D isotropic weighting function according to Section 4.2.4 which is given again in Table 5.2 for convenience. PSNR results achieved with this set of parameters are presented in Table 5.3 for isolated 16×16 block losses. Further, the average number of iterations per block is given. For

Table 5.2: Parameters for spatio-temporal concealment of 16×16 losses using 3-D frequency selective extrapolation with 3-D isotropic weighting function (Section 4.2.3).

$\hat{\rho}$		0.8
Support area	no. of known pixel	13
	no. of frames involved $N_V = N_N$	2
Termination criterion	max. no. of iterations	200
	Δ_{\min}	0.1
DFT size		$64 \times 64 \times 32$
Consecutive losses: attenuation of concealed blocks		0.1

Table 5.3: PSNR results and average number of iterations per block for block losses for parameters according to Table 5.2

Sequence	Flowergarden $N_V = N_N = 2$	Flowergarden $N_V = 2; N_N = 0$	Crew $N_V = N_N = 2$	Crew $N_V = 2; N_N = 0$
$\Delta_{\min} = 0.1$	27.94 dB 177.80 It.	26.03 dB 177.24 It.	32.40 dB 147.40 It.	30.82 dB 148.72 It.
$\Delta_{\min} = 1.0$	27.35 dB 116.68 It.	25.48 dB 103.96 It.	31.50 dB 30.0 It.	30.60 dB 30.76 It.
$\Delta_{\min} = 2.0$	26.57 dB 73.8 It.	24.84 dB 66.48 It.	30.98 dB 17.44 It.	30.19 dB 17.40 It.
Sequence	Foreman $N_V = N_N = 2$	Foreman $N_V = 2; N_N = 0$	Table Tennis $N_V = N_N = 2$	Table Tennis $N_V = 2; N_N = 0$
$\Delta_{\min} = 0.1$	37.06 dB 114.84 It.	35.22 dB 114.40 It.	30.07 dB 166.52 It.	26.27 dB 164.36 It.
$\Delta_{\min} = 1.0$	33.98 dB 26.96 It.	33.14 dB 27.68 It.	29.47 dB 54.72 It.	26.03 dB 53.00 It.
$\Delta_{\min} = 2.0$	33.20 dB 18.44 It.	32.33 dB 18.48 It.	28.94 dB 34.32 It.	25.69 dB 33.48 It.

each sequence two cases are presented with a support area using the actual and only previous frames simulating unidirectional coding and a support area using previous and subsequent frames simulating bidirectional coding. In general, the results are better if previous and subsequent frames can be used for the extrapolation. However, using only previous frames yields still very satisfying results.

The number of required iterations depends on the image content within the considered volume. If the approximation quality is decreased by increasing the threshold Δ_{\min} , the algorithm terminates after less iterations. However, with less iterations still very satisfying results can be achieved as Table 5.3 shows. The difference in terms of PSNR is 1.1 dB to 1.4 dB for $N_V = N_N = 2$ except for the *Foreman* sequence. There, 3.92 dB are lost but the quality of the reconstructed areas with 33.14 dB is still sufficiently high. For using only previous frames, the loss is less in general. In fact, the computational complexity of the 3-D extrapolation could be reduced by increasing Δ_{\min} while providing visually pleasing concealment results.

Visual results for the sequences *Flowergarden* and *Crew* are presented in Fig. 5.3 and Fig. 5.4, respectively. In the top row, the error patterns and in the middle row the

Table 5.4: Comparison of 3-D extrapolation for isolated and consecutive block losses. Parameters used according to Table 5.2 with $N_V = N_N = 2$.

	Flower	Foreman	Table Tennis	Crew
Isolated losses	27.94 dB	37.06 dB	30.07 dB	32.40 dB
Consecutive losses	28.60 dB	36.11 dB	30.56 dB	32.09 dB

corresponding concealed frames are shown using the parameters according to Table 5.2. The bottom graphs in Fig. 5.3 depict the number of iterations used per block for each color component separately where the blocks are numbered in raster scan order. For the luminance component Y more iterations are required as for the color components U and V in order to yield the desired quality, because the chrominance components contain less signal energy than the luminance component. Further, for smooth areas only a few iterations are done in contrary to high frequent areas as the flowers in the meadow. In case of the sequence *Flowergarden*, 177.8 iterations on average per block yielding 27.94 dB are done for isolated losses and 154.73 iterations yielding 28.60 dB for consecutive losses. For concealment of losses in the sequence *Crew*, less iterations are required. In case of isolated block losses 147.4 iterations on average with 32.4 dB PSNR and 130.17 iterations with 32.09 dB PSNR for consecutive losses are used.

The previous evaluations show that the 3-D frequency selective extrapolation is able to reconstruct smooth and complex image contents as well as edges. The approximation of the edge of the tree in the sequence *Flowergarden* in Fig. 5.3 proves that also motion can be estimated. Fig. 5.4 shows for the sequence *Crew* that also abrupt luminance changes can be compensated.

Further investigations presented in [32] emphasize the inherent motion compensation. There, the four test sequences are motion compensated prior to 3-D extrapolation. Better results could not be achieved by using prior motion compensation. This means that the 3-D extrapolation accomplishes inherent motion compensation. However, block based motion compensation in video coding does not necessarily imitate the true motion and only translational motion can be compensated. In contrast, 3-D extrapolation is more flexible and can compensate deformations better.

These results already indicate that the performance for concealing consecutive block losses does not decrease compared to isolated block losses. This is demonstrated in more detail in Table 5.4 for all test sequences. For consecutive losses the previous block in raster scan order is weighted by 0.1, as for 2-D.

The visual results corresponding to Table 5.4 in case of *Foreman* and *Table Tennis* are presented in the following. In Fig. 5.5 the concealed frames for isolated block losses are depicted. The corresponding error patterns are shown in Fig. 4.4(d) and in Fig. 4.4(f),

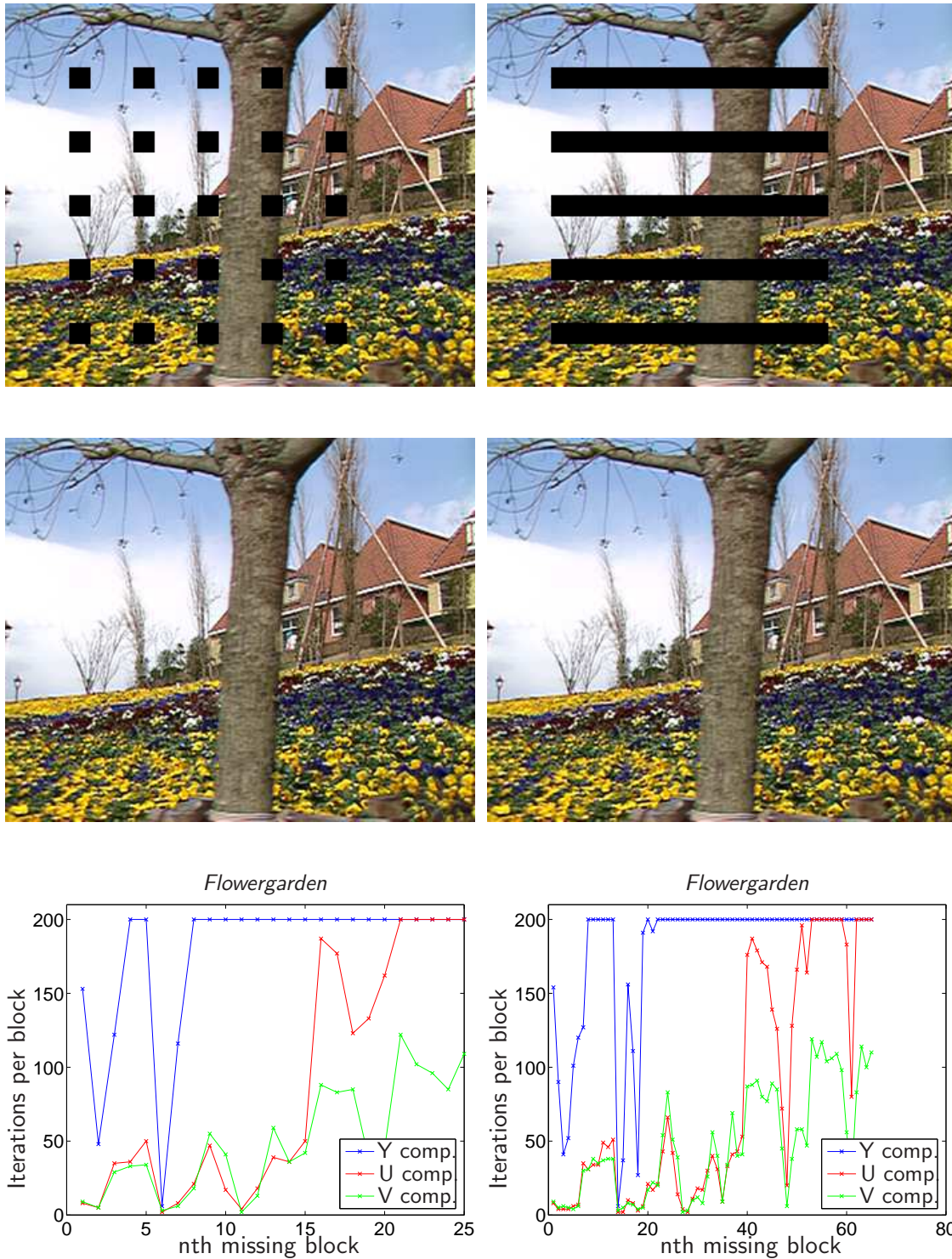


Figure 5.3: Results for sequence *Flowergarden* using parameters according to Table 5.2 with $N_V = N_N = 2$ for isolated and consecutive block losses. Top row: Frames with losses. Middle row: Concealed frames. Bottom row: Iterations per block numbered in raster scan order.

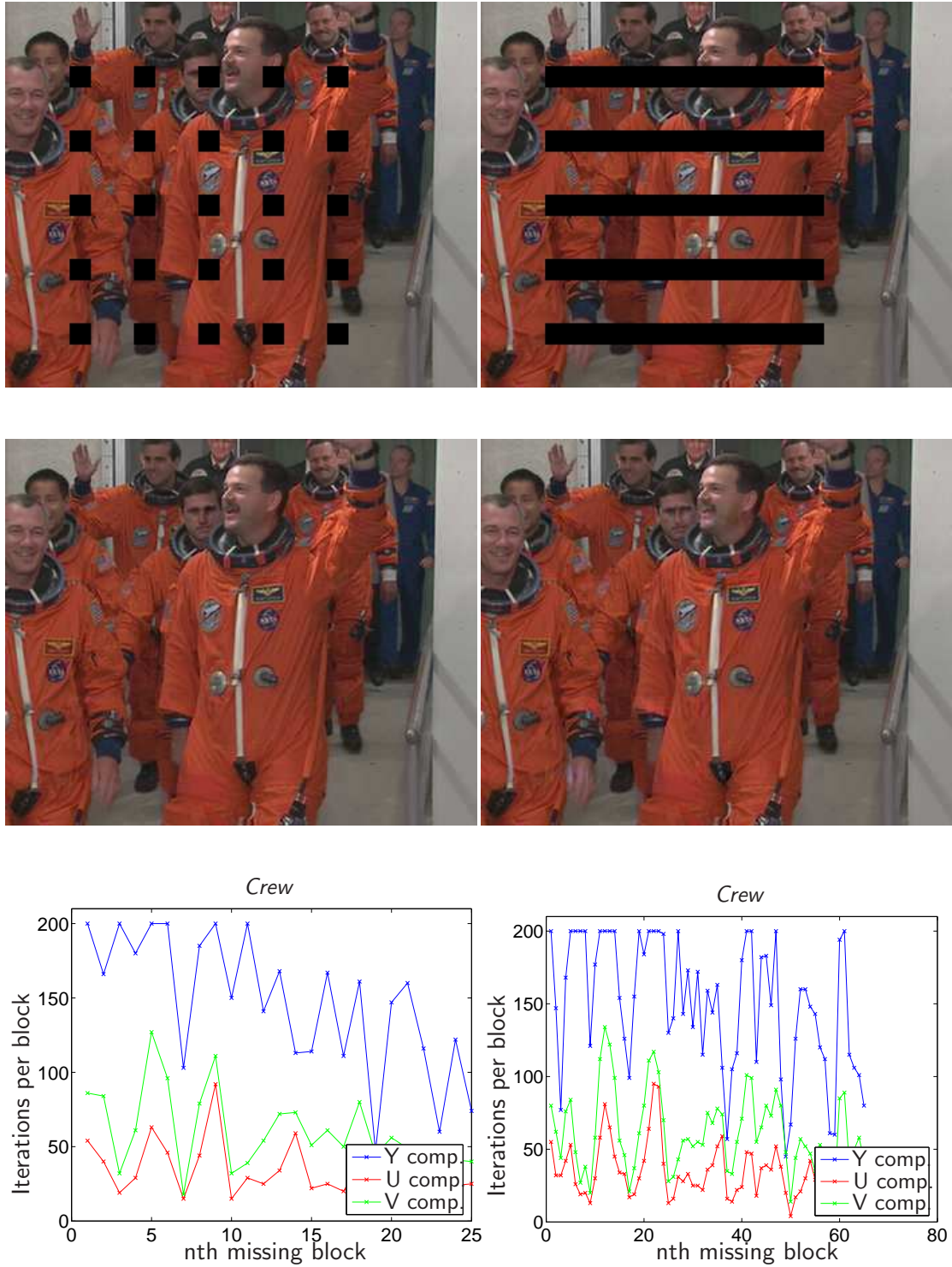


Figure 5.4: Results for sequence *Crew* using parameters according to Table 5.2 with $N_V = N_N = 2$ for isolated and consecutive block losses. Top row: Frames with losses. Middle row: Concealed frames. Bottom row: Iterations per block numbered in raster scan order.



(a)



(b)

Figure 5.5: Concealed isolated losses using 3-D frequency selective extrapolation with parameters according to Table 5.2 and $N_V = N_N = 2$. (a) *Foreman*. (b) *Table Tennis*.

respectively. The visual quality of the missing areas is excellent and it is almost impossible to identify artifacts. Also the results for concealment of consecutive block losses, as shown in Fig. 5.6 for the sequences *Foreman* and *Table Tennis* with the corresponding loss pattern, confirm the excellent concealment properties of the 3-D algorithm.



(a)



(b)



(c)



(d)

Figure 5.6: Frame with consecutive losses and corresponding concealed frame using 3-D frequency selective extrapolation with parameters according to Table 5.2 and $N_V = N_N = 2$.

5.3 Comparison to Related Techniques

The performance of the developed concealment strategies using frequency selective extrapolation as described in Section 5.2 is compared to the performance of related techniques reviewed in Section 5.1.

5.3.1 Results for Spatial Concealment

In Section 5.2.1, 2-D frequency selective extrapolation is applied to spatial concealment. The achieved concealment performance is compared in this section to the performance of related spatial concealment techniques reviewed in Section 5.1.1. The maximally smooth recovery [60], POCS [56], the spatial-domain interpolation according to [8], and the sequential error concealment [63] are considered for the comparison. The corresponding results are shown in Table 5.5. The results using frequency selective extrapolation are obtained with the fixed parameter set according to Table 5.1.

For the images *Lena* and *Peppers*, the frequency selective extrapolation performs as good as the sequential error concealment method. The difference to the next best method maximally smooth recovery is 0.5 dB for *Lena* and for *Peppers* even 2.1 dB. In Fig. 5.7 the corresponding visual results are presented for *Peppers*. The error pattern is shown in Fig. 5.7 (a) and the result of maximally smooth recovery in Fig. 5.7 (b). Due to this maximally smooth recovery edges can not be restored and appear blurry, in contrast to Fig. 5.7 (c) with restored edges concealed by the sequential method. Edges and even diagonal edges can also be restored with frequency selective extrapolation as can be noticed from Fig. 5.7 (d).

For *Baboon*, maximally smooth recovery provides the best PSNR, although the visual evaluation shows a better perceived quality for sequential error concealment and frequency selective extrapolation. In Fig. 5.8, the corresponding visual results are shown. Even if the best PSNR is obtained using maximally smooth recovery, the concealed parts as the

Table 5.5: The performance of spatial error concealment techniques in comparison.

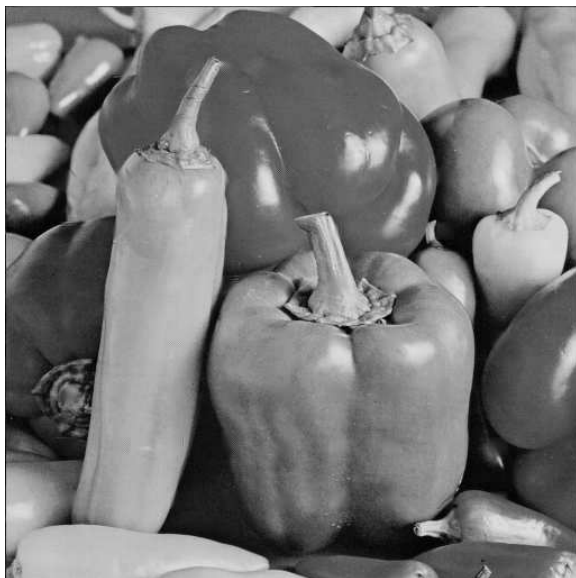
	16 × 16 Block loss		
	<i>Lena</i>	<i>Peppers</i>	<i>Baboon</i>
Maximally smooth recovery [60]	23.7 dB	24.2 dB	19.5 dB
POCS [56]	22.3 dB	22.1 dB	18.9 dB
Spatial-domain interpolation [8]	21.2 dB	23.3 dB	16.4 dB
Sequential error concealment [63]	24.2 dB	26.9 dB	18.8 dB
2-D frequency selective extrapolation	24.2 dB	26.3 dB	18.8 dB



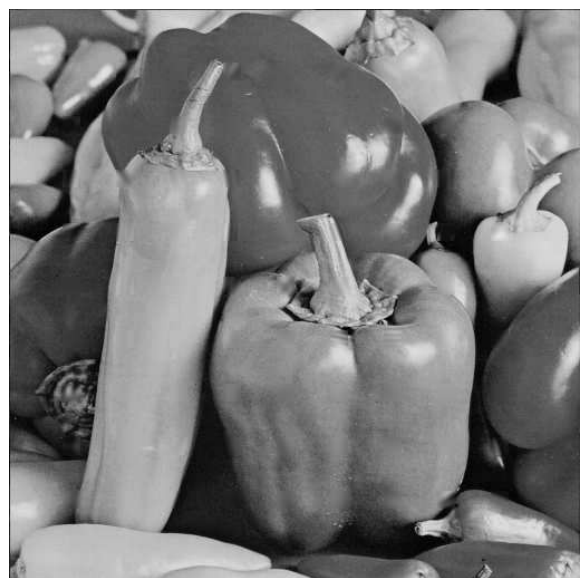
(a)



(b) Maximally smooth recovery [60], 24.2 dB

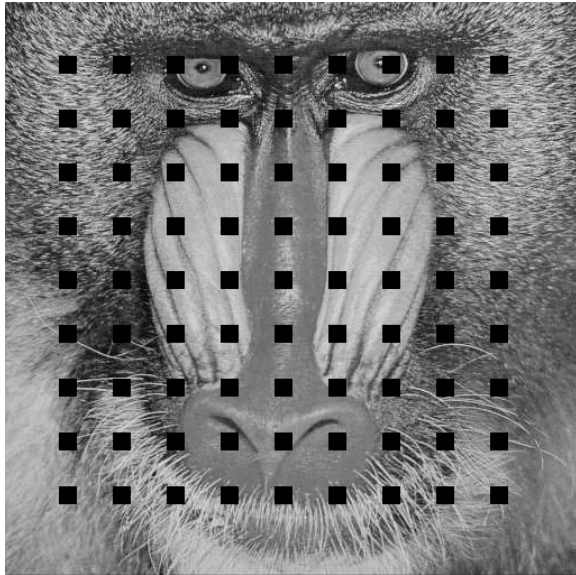


(c) Sequential error concealment [63], 26.9 dB

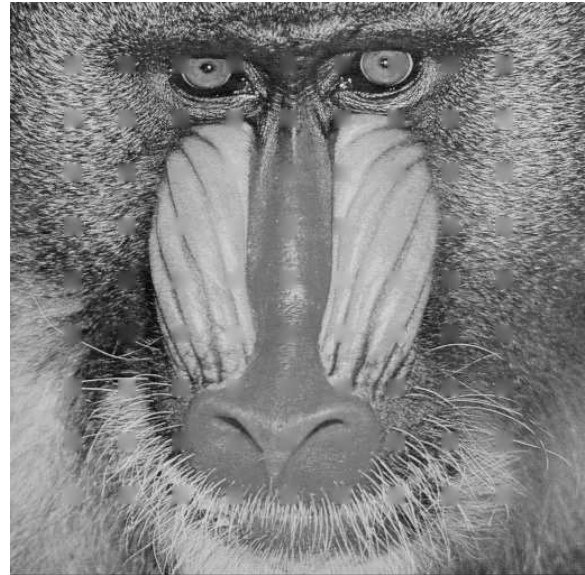


(d) Frequ. selective extrapolation, 26.3 dB

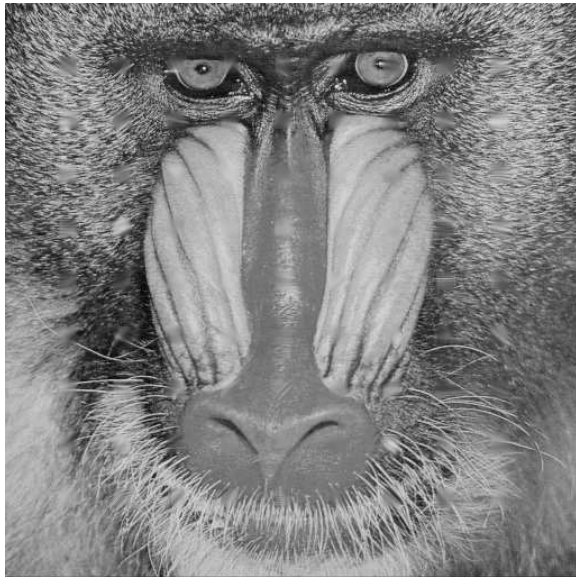
Figure 5.7: Comparison of different spatial concealment techniques for the image *Peppers*.



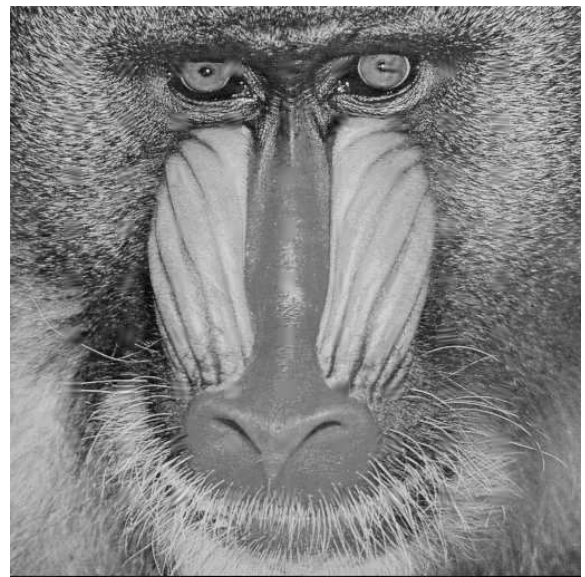
(a)



(b) Maximally smooth recovery [60], 19.5 dB



(c) Sequential error concealment [63], 18.8 dB



(d) Frequ. selective extrapolation, 18.8 dB

Figure 5.8: Comparison of different spatial concealment techniques for the image *Baboon*.

fur appear blurry. The PSNR is better especially in the detailed areas due to this lowpass effect as for algorithms which try to restore details. Although the visual impression is better, small but not visible phase shifts cause the low PSNR. The sequential concealment method exploits the correlation between pixels in order to predict the missing pixel. If this assumption is not fulfilled in areas with low correlation like the fur, the algorithm fails. This area cannot be predicted which results in white spots in Fig. 5.8 (c). For comparison, Fig. 5.8 (d) shows the concealed image with the proposed method. As can be noticed, the subjective performance is better than it is indicated by the PSNR values.

In conclusion, the comparison of frequency selective extrapolation to related techniques shows that its performance is comparable to the sequential concealment method [63] and superior to the other techniques [60, 56, 8] investigated. However, [63] has problems in concealing detailed areas and, moreover, is computational very complex due to the pixel based recovery. The simulation results have clearly shown that, in contrast to the other methods, frequency selective extrapolation is not only able to restore smooth areas, but it is also capable to deal with edges and detailed areas.

5.3.2 Results for Spatio-Temporal Concealment

In the following, the performance of 3-D frequency selective extrapolation applied to error concealment according to Section 5.2.2 is compared to the temporal concealment methods as described in Section 5.1.2. The parameters according to Table 5.2 are used for 3-D extrapolation based concealment. The result of the Temporal Blockreplacement (TR) algorithm copying simply the block from the previous frame illustrates the motion of the sequences. As temporal methods based on motion vector recovery the Decoder Motion Vector Estimation (DMVE) [67], the Boundary Matching Algorithm (BMA) and the Extended Boundary Matching Algorithm (EBMA) described in [29] are considered for the comparison. Furthermore, the results of Overlapped Block Motion Compensation (OBMC) [66] with bidirectional motion compensation taking the previous and the subsequent frame into account are compared. Motion compensated prediction with full search at pixel accuracy is performed in order to obtain the motion vectors and the prediction error. The motion vectors of the blocks to be concealed and additionally the prediction error signal in case of EBMA, DMVE, and OBMC are discarded. For comparison, the results of the spatial 2-D extrapolation are also presented.

Table 5.6 summarizes the results for isolated and Table 5.7 for consecutive losses. The OBMC with bidirectional motion compensation has a comparable performance to 3-D extrapolation, except for the *Crew* sequence where 3-D extrapolation is better. For isolated losses, OBMC performs better whereas for consecutive losses the performance of 3-D extrapolation is superior in most cases. For the remaining simulations, the concealment performance of 3-D extrapolation outperforms the other algorithms up to several dB in

Table 5.6: Comparison of different temporal concealment techniques to frequency selective extrapolation for isolated block losses.

	Flower	Foreman	Table Tennis	Crew
TR [59]	17.79 dB	28.77 dB	19.75 dB	19.55 dB
BMA [29]	28.99 dB	36.40 dB	25.19 dB	26.46 dB
EBMA [29]	22.95 dB	32.13 dB	22.25 dB	24.44 dB
DMVE [67]	25.43 dB	36.12 dB	28.28 dB	21.45 dB
OBMC [66]	28.30 dB	38.17 dB	32.50 dB	25.80 dB
2-D	17.04 dB	27.43 dB	19.94 dB	27.62 dB
3-D	27.94 dB	37.06 dB	30.07 dB	32.40 dB

Table 5.7: Comparison of different temporal concealment techniques to frequency selective extrapolation for consecutive block losses.

	Flower	Foreman	Table Tennis	Crew
TR [59]	17.99 dB	28.64 dB	20.40 dB	19.25 dB
BMA [29]	26.16 dB	33.22 dB	25.16 dB	25.89 dB
EBMA [29]	25.22 dB	33.46 dB	24.06 dB	24.02 dB
DMVE [67]	25.44 dB	35.32 dB	22.78 dB	20.52 dB
OBMC [66]	27.59 dB	38.57 dB	27.76 dB	24.71 dB
2-D	16.12 dB	22.47 dB	19.25 dB	26.15 dB
3-D	28.60 dB	36.11 dB	30.56 dB	32.09 dB

PSNR. The only exception occurs for isolated block losses for *Flowergarden*, because there the BMA algorithm is 1 dB better than the 3-D extrapolation and 0.7 dB better than OBMC. The vector of the homogenous motion can be reconstructed very well from the vectors of *all* surrounding blocks. To receive all surrounding vectors is unlikely in a real-world scenario because several blocks in a row are commonly coded in one packet. Additionally, the prediction error signal is commonly lost and not reconstructed. In contrast to the other techniques, the performance of 3-D extrapolation does not decrease for consecutive losses even if less data is available for the estimation.

Next, we want to confirm the gained insights subjectively. The results for concealing the consecutive block losses for the sequence *Flowergarden* in Fig. 5.3 are shown in Fig. 5.9. The results for the BMA algorithm are not presented, instead, the results for the more

realistic EBMA are given. Fig. 5.9(a) shows that there is significant motion present in the sequence and that simple block replacement with 17.99 dB yields strong motion artifacts. In Fig. 5.9(b) the results using the EBMA with 25.22 dB and in Fig. 5.9(c) using the DMVE with 25.44 dB are presented. Obviously, the algorithms are not able to compensate the motion which is apparent at the edges of the tree. The performance of the OBMC using bidirectional motion compensation yields a better result with 27.56 dB. However, motion artifacts occur also for instance at the bush or the roof of the house on the right hand side next to the tree. The worst result is obtained using the 2-D extrapolation displayed in Fig. 5.9(e). Using additionally the third dimension time, the 3-D extrapolation significantly improves the performance to 28.60 dB. 3-D extrapolation is able to inherently compensate the motion with an improvement of 1.04 dB compared to the next best temporal method OBMC. Also details like the branches of the tree, the bushes, or the flowers in the meadow can be extrapolated.

Fig. 5.10 shows the results for concealing consecutive losses in the sequence *Crew* depicted in Fig. 5.4. Using TR visualizes in Fig. 5.10(a) that significant luminance changes occur from frame to frame at slow motion. The results of the temporal concealment methods relying on the motion vector recovery principle such as EBMA, DMVE, and OBMC are shown in Fig. 5.10(b), Fig. 5.10(c), and Fig. 5.10(d), respectively. The algorithms are not able to compensate the change in luminance caused by a flash of a camera. Therefore, wrong motion vectors are determined leading to very annoying artifacts. In this case, the spatial method 2-D extrapolation shown in Fig. 5.10(e) is advantageous and the lost areas can be reconstructed fairly good. The spatio-temporal extrapolation improves the result by reconstructing detailed areas like badges on the uniforms proven by Fig. 5.10(f) compared to 2-D. Additionally, the variations in luminance can be compensated which leads to an improvement from 7.28 dB to 11.47 dB compared to the temporal methods.



(a) TR 17.99 dB



(b) EBMA 25.22 dB



(c) DMVE 25.44 dB



(d) OBMC 27.59 dB



(e) 2-D extrapolation 16.12 dB



(f) 3-D extrapolation 28.60 dB

Figure 5.9: Comparison of techniques for concealing consecutive losses in *Flowergarden*.



(a) TR 19.25 dB



(b) EBMA 24.02 dB



(c) DMVE 20.52 dB



(d) OBMC 24.71 dB



(e) 2-D extrapolation 26.15 dB



(f) 3-D extrapolation 32.09 dB

Figure 5.10: Comparison of techniques for concealing consecutive losses in Crew.

5.4 Integration of Frequency Selective Extrapolation into H.264/AVC Decoder

The state-of-the-art video coding standard H.264/MPEG-4 Advanced Video Coding (AVC) [23], in short H.264/AVC, was released in 2003 as a joint project of the Joint Video Team (JVT) of the two standardization organizations Motion Picture Experts Group (MPEG) and International Telecommunication Union Telecommunication Standardization Sector (ITU-T). As common for video standards, the H.264/AVC defines only the syntax and the semantic of the bitstream. The implementation details including coder optimizations are left unspecified. The main concepts of the video coding standard are summarized in [43, 55].

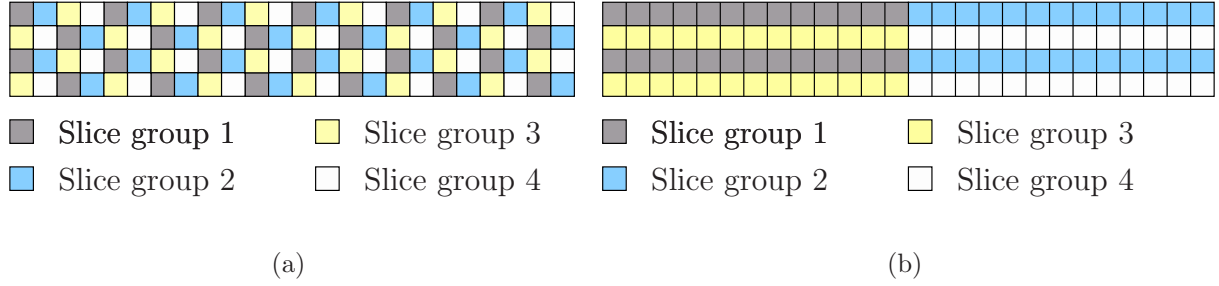
The codec H.264/AVC was not designed for a special purpose but for a variety of applications. Besides the increased data compression and the use for different applications, the standard provides several novel tools for improving the error robustness. Error robustness tools enable the transmission of coded video data in error prone environments. The main concepts are described in Section 5.4.1.

In Section 5.4.2 the error concealment methods used in the reference software are reviewed. The realization of the frequency selective extrapolation in the reference software decoder as concealment feature is given in Section 5.4.3. Both methods are evaluated by simulations in Section 5.4.4.

5.4.1 Error Robustness in H.264/AVC

Several error robustness tools are provided by the H.264/AVC standard in order to enable the transmission of coded video sequences in error prone environments [52]. A Macroblock (MB) consists of 16×16 luminance values and 8×8 chrominance values for a YUV signal in the most common format. Several MBs are organized in slices. The beginning of a slice thus provides a possible synchronization point. Due to the concept of *slice structured coding*, the slices can be coded and decoded independently. This increases the error robustness and allows the use of concealment techniques. Furthermore, a parallel processing of the slices is possible. The size of a slice can be adjusted to the maximum transfer unit (MTU) of the underlying network. In case of IP packet based transmission over the Internet this would be 1400 bytes [62]. Hence, in case of a packet loss, a complete slice gets lost. However, H.264/AVC differentiates between the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL). The VCL is responsible for coding the video data as efficient as possible. The adaptation of the VCL data to the network is done by the NAL.

Error robustness can additionally be increased by Flexible Macroblock Ordering (FMO) which modifies the assignment of MBs to a slice [52]. With help of FMO a frame

Figure 5.11: FMO patterns. (a) *Dispersed*. (b) *Interleaved*.

can be divided into different MB patterns. Two FMO patterns are shown in Fig. 5.11: the mode *dispersed* in Fig. 5.11 (a) and *interleaved* in Fig. 5.11 (b). In both cases, the MBs are assigned to four slice groups visualized by four different colours. In case of a loss, the MBs with one colour get lost. Thus, the two loss patterns used in the simulations for uncoded data are produced. For the mode *dispersed* in Fig. 5.11 (a), a loss results in isolated losses causing a checkerboard like loss pattern and for the mode *interleaved* in Fig. 5.11 (b) in consecutive MB losses.

Error robustness is, however, increased at the expense of coding efficiency which has to be traded off for the respective application.

5.4.2 Error Concealment in Reference Software

As already mentioned, error concealment is non-normative in common video coding standards. In the following the concealment procedure for the reference software of the H.264/AVC standard is reviewed as described in [61]. In the H.264/AVC reference software, spatial concealment is applied for Intra coded MBs only, losses of Inter coded MBs are concealed by exploiting temporal information.

First, the frame is decoded by H.264/AVC except for the lost areas. The status of each MB is stored as “correctly received” or “lost” in a status map. The concealment processing starts at the frame boundaries and proceeds inwards columnwise. The status of a MB is updated after concealment and marked by “concealed” in the status map as illustrated in Fig. 5.12.

The concealment algorithm used for Intra coded MBs in the H.264/AVC reference decoder implementation [61] is based on weighted pixel averaging from surrounding image data. The inverse distance from the adjacent correctly received pixel serves as weight. Only correctly received pixels are taken into account for the interpolation if available, otherwise also already concealed ones. Thus, in case of MB losses in a row, only the top and bottom MB are used for concealment.

For Inter coded MBs, either TR or motion vector recovery based on EBMA [61] is

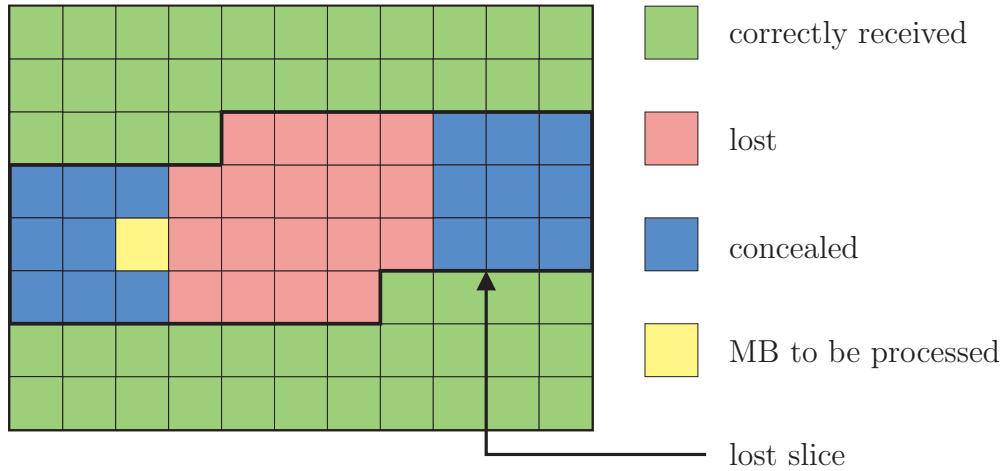


Figure 5.12: The condition of each MB is stored in the status map.

applied. If the average motion vector length of all correctly received motion vectors of the entire frames is below the threshold $\frac{1}{4}$, the lost MBs are replaced by copying the MBs spatially located at the same position from the previous frame. If the average motion vector length exceeds this threshold, the temporal concealment method is based on the EBMA algorithm. The prediction error signal is lost in case of transmission errors and no prediction error signal recovery is done. As test vectors, the zero motion vector and all motion vectors of the correctly received neighboring blocks are used. For motion compensation in H.264/AVC video coding, a MB can be split into several blocks down to a size of 4×4 pixels which is rate-distortion optimized. Hence, multiple vectors per MB can be transmitted. Furthermore, five reference frames are available for prediction. For concealment, the motion vector of the MB to be concealed is estimated from the surrounding 16×16 or 8×8 blocks pointing to the corresponding reference frame. Thus, 4×4 blocks are merged to 8×8 blocks. The test vector which results in a minimum boundary error between the MB to be concealed and the correctly received neighboring blocks is chosen as motion vector.

5.4.3 Realization of Concealment Using Frequency Selective Extrapolation

The algorithm of the frequency selective extrapolation applied to concealment is integrated into the H.264/AVC reference software version JM9.3 [54]. The concealment processing order of the MBs according to the status map illustrated by Fig. 5.12 is left unchanged. 2-D as well as 3-D extrapolation is used for concealment.

2-D extrapolation is used for concealment of Intra coded MBs. If several MBs in a row are lost, for uncoded images the already concealed MB is taken into account for the extrapolation with limited influence as shown in Fig. 5.1. In case of coded data

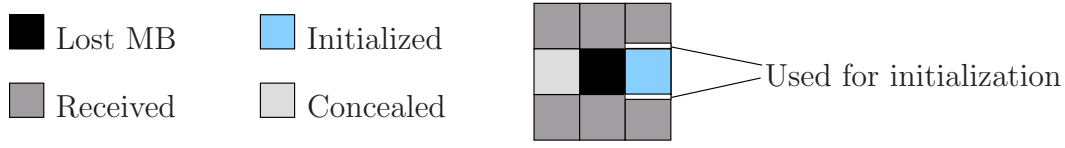


Figure 5.13: Status of MBs used for concealment in case of consecutive losses.

as considered here, we extend this idea so that the MB to be concealed is completely surrounded by image data which can be used for extrapolation. Therefore, a missing surrounding MB is initialized by weighted interpolation from adjacent top and bottom pixels of the correctly received MB where the inverse distance from the respective pixel serves as weight. Thereby, the effect of visible basis images can be reduced. The status of MBs used for concealment providing the input signal $f[m, n]$ is shown in Fig. 5.13. An attenuation of the concealed block is not necessary anymore in contrast to uncoded images, so that all surrounding blocks contribute equally to the extrapolation.

Based on the investigations for uncoded images summarized in Section 5.2.1, the following parameters are chosen for 2-D extrapolation. The coefficient $\hat{\rho}$ is set to 0.75. The 8 MBs surrounding the lost MB are taken as support area. The 9 MBs according to Fig. 5.13 are padded with zeros to a block of size 64×64 which is then transformed by a suitable FFT implementation. Further, the algorithm terminates if the residual error decrease falls below the threshold $\Delta_{\min} = 15$, or if the maximum number of 20 iterations is reached.

3-D extrapolation is used for concealing Intra as well as Inter coded MBs. Since the 3-D extrapolation does not use any information based on motion vectors, the method can be easily applied to Intra coded frames. As already shown by the investigations for uncoded frames in Section 5.3.2, the concealment result can be improved significantly using three dimensions instead of two dimensions. Because no motion information is available for the first frame of a sequence, it is concealed by the 2-D extrapolation method. For Inter coded blocks in case of slow motion, i.e., if the average motion vector length of all correctly received motion vectors is below the threshold $\frac{1}{4}$, the lost MBs are replaced by copying the MBs spatially located at the same position from the previous frame. If the average motion vector length exceeds this threshold, the image content of the lost blocks is estimated instead by using the 3-D frequency selective extrapolation. In contrast to the reference software, both, the correctly received and the already concealed blocks are used for the calculation of the parametric model.

In Section 5.2.2 we investigated the performance for uncoded frames with respect to different parameters. Based on the results, the following parameters are chosen for the 3-D approach. For the simulations, we used the *Baseline* profile of the H.264/AVC coder in order to code the test sequences. The Baseline profile supports only P and no B frames, hence, only previous frames can be taken into account for the extrapolation.

We used 2 previous frames in the evaluations. The lost MB and its adjacent MBs form a block of 48×48 pixels which is zero-padded to the next larger DFT size of 64×64 . The same configuration is applied to the 9 MBs spatially located at the same position in the 2 previous frames. The total DFT size of $64 \times 64 \times 16$ per processed block is obtained by further zero padding in temporal direction. Hence, also the DFT size is halved in temporal direction compared to the uncoded case summarized in Table 5.2 in order to reduce the computational complexity. Further, the termination thresholds are both slightly altered to the threshold $\Delta_{\min} = 0.5$ per pixel or a maximum number of 150 iterations. In Section 5.2.2 it was shown that these changes are only at the expense of a marginal loss in PSNR. The coefficient $\hat{\rho}$ is set to 0.8.

5.4.4 Simulations and Results

Spatial Concealment of Intra coded MBs

The performance of *2-D extrapolation* is compared to the *spatial concealment method* used in the reference software [61] for concealing lost *Intra* coded MBs.

In order to evaluate the performance, 150 frames of two VQEG (video quality experts group) sequences (VQEG5 *Canoe*, VQEG7 *Fast Food*) and 200 frames of the well-known test sequences *City*, *Harbour*, and *Crew* in PAL format (720×576 pixels) are encoded by the H.264/AVC software using the Baseline profile. The Intra frame period is 10, implying that each I frame is followed by 9 P frames. Here, we want to compare the performance of spatial concealment methods and its impact on future frames. Therefore, the sequences are encoded with fixed slice sizes using the FMO patterns *dispersed* and *interleaved* (see Fig. 5.11). Packet losses are introduced in I frames according to a regular pattern.

The PSNR of the entire sequence depends on the number of lost MBs, i.e. the more blocks are lost, the worse the PSNR is. Therefore, three sets of results are analyzed:

- The absolute number of lost MBs is eliminated by calculating the PSNR for the *lost areas* only.
- The PSNR is calculated for *I frames* only in order to evaluate the overall impression of the frame including coding artifacts.
- The PSNR for *all frames* is evaluated taking the effects into account which are caused by error propagation due to predictive coding.

The objective results measured in terms of PSNR are presented in Table 5.8. First, we consider the performance of the two algorithms concerning PSNR evaluated for the concealed areas only. In case of the mode *dispersed*, the performance for the frequency selective extrapolation method is better than H.264/AVC from comparable performance in terms of PSNR up to considerable 1.69 dB for the sequence *Fast Food*. The visual

Table 5.8: Objective performance comparison of spatial methods for concealment of *Intra* coded MBs measured in PSNR (dB) for H.264/AVC coded sequences.

FMO Pattern	Dispersed		Interleaved	
Concealment	Spatial ref. [61]	2-D extrap.	Spatial ref. [61]	2-D extrap.
Coded sequ.	Fast Food 37.07		Fast Food 37.08	
lost areas	23.81	25.52	21.77	22.15
I-frames	29.61	31.11	28.32	28.71
all frames	31.39	32.62	30.09	30.42
Coded sequ.	Canoe 34.58		Canoe 34.61	
lost areas	23.20	23.46	20.28	20.32
I-frames	28.72	28.98	26.57	26.62
all frames	30.87	31.06	29.30	29.32
Coded sequ.	Harbour 35.82		Harbour 35.83	
lost areas	20.81	21.01	20.69	20.70
I-frames	26.52	26.70	28.39	28.40
all frames	27.39	27.50	29.14	29.11
Coded sequ.	City 35.64		City 35.65	
lost areas	23.42	23.25	21.91	22.00
I-frames	28.85	28.69	29.42	29.48
all frames	29.41	29.26	30.00	30.04
Coded sequ.	Crew 37.64		Crew 37.65	
lost areas	27.41	28.21	27.21	27.53
I-frames	32.54	33.19	34.00	34.24
all frames	33.33	33.86	34.44	34.65

impression shows a clearer result. It should, however, be pointed out again that the average difference to the original signal measured by PSNR does not necessarily reflect the subjective visual quality of the image. This has already been observed for concealing losses in the image *Baboon*. As discussed in the following, the extrapolation approach provides a significantly better perceived image quality compared to the reference software H.264/AVC.

Fig. 5.14 shows the concealed results for the sequence *City* for the checkerboard like loss pattern of the mode *dispersed*. In Fig. 5.14 (a), the frame concealed by H.264/AVC



(a) Spatial method H.264/AVC [61], 28.84 dB



(b) 2-D extrapolation, 28.59 dB

Figure 5.14: Concealed frame 101 of sequence *City*, mode *dispersed*, Intra MBs lost.



(a) Spatial method H.264/AVC [61], 28.66 dB



(b) 2-D extrapolation, 28.80 dB

Figure 5.15: Concealed frame 81 of sequence *Canoe*, mode *interleaved* Intra MBs lost.

is displayed and in Fig. 5.14 (b) by 2-D extrapolation. The PSNR of the top frame is 28.84 dB and of the bottom frame 28.59 dB. Although the PSNR is better for the spatial method of the reference software, the visual impression is clearly different. The 2-D extrapolation method is able to reconstruct the vertical pattern of the building on the left hand side as well as the cross-like pattern of the windows of the building on the right hand side. The interpolation method in contrast blurs the patterns and causes also errors in the luminance. This results in very annoying visual artifacts which appear also in the following frames due to predictive coding.

As shown in Table 5.8, the PSNR gains of 2-D extrapolation over the spatial method of H.264/AVC are lower for the *interleaved* mode. Taking a look at the results for the lost areas, they reach from equal quality in terms of PSNR to a gain of 0.38 dB. Nevertheless, the visual result obtained for 2-D extrapolation is by far more convincing than H.264/AVC as demonstrated by Fig. 5.15 as in case of the mode *dispersed*. Fig. 5.15 (a) shows the frame concealed by H.264/AVC and Fig. 5.15 (b) by 2-D extrapolation. H.264/AVC uses only the top and bottom block for concealment which leads to blurring in vertical direction. This is visible in the *Canoe* sequence Fig. 5.15 (a) where diagonal structures like the paddel or the back of the man cannot be restored. These artifacts do not occur for frequency selective extrapolation as can be seen from Fig. 5.15 (b). Additionally, the concealment results of the 2-D extrapolation method for smooth areas like the water or the background appear very naturally.

As already mentioned, calculating the PSNR only for the lost areas eliminates the dependence of the number of lost MBs per frame. Thus, the concealment performance is explicitly reflected and the performance for the mode *dispersed* can be directly compared to the mode *interleaved*. For the *dispersed* pattern, concealment is more successful up to 3.37 dB compared to *interleaved*. In case of the mode *dispersed*, the lost block is surrounded by correct blocks which improves the concealment result significantly. However, the coding efficiency is reduced because prediction is not allowed over slice boundaries. Therefore, error robustness has to be traded off against efficiency for the respective application.

Taking a look at Table 5.8, a gain in PSNR for the lost areas apparently results in a gain in PSNR for all frames. A better concealment of I frames allows a better prediction for the following P frames and therefore reduces error propagation caused by predictive coding, too.

Spatio-Temporal Concealment of Intra and Inter coded MBs

In the following, the concealment performance of *3-D extrapolation* is compared against the non-normative methods used in the reference software [61].

In order to evaluate the performance, 65 frames of the test sequences *Canoe*, *Fast Food*, *Crew*, and *Discovery* in PAL format (720×576 pixels) are encoded by the H.264/AVC

software using the *Baseline* profile. The Intra frame period is 20. In order to simulate a DVB-T like transmission, we choose the slice size to 188 byte which is the corresponding size of a video data packet. Hence, in case of a loss, a slice simulating one packet gets lost. We introduce packet losses randomly according to a mobile scenario [58]. In order to obtain more reliable results, 10 realizations of a channel are simulated for each condition. Both, weak and strong distortions are introduced. The sequences are encoded using the FMO patterns *dispersed* and *interleaved* (see Fig. 5.11). Again, the increased error robustness reduces the coding efficiency. However, we are mainly interested in evaluating the concealment performance of the proposed algorithm.

Three different sets of experiments are considered in the following:

- The proposed 3-D extrapolation is applied to concealing *Inter* coded blocks. The concealment results are compared against the temporal method used in the reference software.
- Since the 3-D extrapolation does not use any information based on motion vectors, the method is also applied to *Intra* coded MBs. Note that the first frame is concealed by the 2-D extrapolation method because no motion information is yet available. The performance is compared against the spatial concealment method used in the reference software for a sequence of only Intra coded frames.
- The 3-D extrapolation is used as concealment method for losses occurring in both, *Intra* and *Inter* coded frames. The results are compared against the corresponding methods used in the reference software.

Both, objective results measured in terms of PSNR and subjective results visualized by images are presented.

The first set of experiments uses 3-D extrapolation for concealment of lost Inter coded blocks. The objective results comparing the performance of 3-D extrapolation to the temporal method of the reference software in terms of PSNR are shown in Table 5.9. In case of the mode *interleaved*, the two algorithms show comparable performance. While the extrapolation shows better results for the sequence *Canoe*, the motion vector recovery works better for the sequence *Fast Food*, especially for severe losses. Clear conclusions, however, can be drawn from the performance results for the mode *dispersed*. There, the 3-D extrapolation outperforms the motion vector recovery method in each case. The performance for *Canoe* is still slightly better, but for *Fast Food* the performance has significantly improved even up to 1.08 dB compared to the reference in case of strong distortions.

Visual results are presented in Fig. 5.16. Results for checkerboard like losses are depicted, comparing H.264/AVC concealment in Fig. 5.16 (a) to 3-D extrapolation in Fig. 5.16 (b). Strong motion is involved in this scene. The motion vectors can not follow

Table 5.9: Concealment results for PAL sequences comparing 3-D extrapolation to the temporal method of the reference software. 3-D extrapolation is used for *Inter* MBs.

	Distortions	Method	Canoe	Fast Food
Interleaved	-	-	34.65 dB	36.86 dB
	Strong	3-D extrapolation H.264/AVC [61]	29.72 dB	29.37 dB
			29.54 dB	30.48 dB
	Weak	3-D extrapolation H.264/AVC [61]	32.81 dB	35.01 dB
			32.68 dB	35.64 dB
Dispersed	-	-	34.58 dB	36.81 dB
	Strong	3-D extrapolation H.264/AVC [61]	32.49 dB	34.60 dB
			32.33 dB	33.52 dB
	Weak	3-D extrapolation H.264/AVC [61]	33.91 dB	36.04 dB
			33.76 dB	35.65 dB

the true motion, hence, the recovery leads to wrong vectors causing very annoying artifacts. The 3-D extrapolation method in contrast can compensate the motion inherently. Successful concealment is very important because it also improves the quality of the subsequent frames due to predictive coding. On the top left border of Fig. 5.16 (a) concealed by H.264/AVC, the effects of error propagation are visible. No error propagation artifacts occur in case of 3-D extrapolation based concealment on the top left border of the frame in Fig. 5.16 (b).

As already mentioned, the 3-D extrapolation can be easily applied to concealment of Intra coded blocks. The performance of the spatial method used in the reference software is to be compared to 3-D extrapolation. For this experiment, 10 frames of the sequences *Canoe* and *Fast Food* are Intra coded. Again, 10 runs causing weak and strong distortions are done. The results are displayed in Table 5.10. Obviously, the 3-D extrapolation outperforms the spatial concealment method of the reference software in any case by using both, spatial and temporal information. For the sequence *Fast Food* improvements up to more than 2 dB can be gained. Fig. 5.17 confirms this result also visually. The blocks concealed by the spatial method of the reference software in Fig. 5.17 (a) disturb the visual impression at the transition from the towel to the background, whereas the blocks concealed by 3-D extrapolation in Fig. 5.17 (b) are hardly visible.

Therefore, 3-D extrapolation is used to conceal both, Intra and Inter coded MBs. The overall performance and the comparison to the reference software is shown in Table 5.11. For the overall performance, two more sequences *Crew* and 'Documentation' are simulated. The performance for the sequences *Canoe* and *Fast Food* has slightly increased by



(a) Temporal method H.264/AVC [61], 36.58 dB



(b) 3-D extrapolation, 36.73 dB

Figure 5.16: Concealed frame 43 of sequence *Fast Food*, mode *dispersed*.



(a) Spatial method H.264/AVC



(b) 3-D extrapolation

Figure 5.17: Concealed frame 8, sequence *Fast Food*, mode *interleaved*, Intra coded.

Table 5.10: Concealment results for *Intra* coded PAL sequences. 3-D extrapolation used for *Intra* coded frames.

	Distortions	Method	Canoe	Fast Food
Interleaved	-	-	38.32 dB	37.35 dB
	Strong	3-D Extrapolation	35.98 dB	35.70 dB
		Spatial H.264/AVC	35.69 dB	33.58 dB
	Weak	3-D Extrapolation	37.62 dB	36.93 dB
		Spatial H.264/AVC	37.46 dB	35.94 dB
Dispersed	-	-	38.32 dB	37.34 dB
	Strong	3-D Extrapolation	37.32 dB	36.31 dB
		Spatial H.264/AVC	36.72 dB	35.34 dB
	Weak	3-D Extrapolation	38.02 dB	37.00 dB
		Spatial H.264/AVC	37.80 dB	36.71 dB

incorporating the 3-D extrapolation also for I frames. However, only 3 frames are coded Intra in each sequence of 65 frames at an I frame period of 20. Considering the mode *interleaved*, the performance of 3-D extrapolation for the sequence *Fast Food* could be improved considerably compared to Table 5.9, but is slightly worse than for the reference software. In contrast, the results for *Canoe* and *Crew* are slightly better. The results for the sequence *Discovery* obtained by extrapolation are significantly better than by H.264/AVC. Concealment by recovering motion vectors fails because the sequence contains many scene cuts. Visual results are given later for this case. In case of the mode *dispersed*, the 3-D extrapolation used for concealment performs better for each sequence and loss case. For the sequence *Discovery* and severe degradations gains even up to 1.72 dB are achieved.

In Fig. 5.18, the results for the sequence *Crew* obtained for the temporal method of the reference software in Fig. 5.18 (a) and for 3-D extrapolation in Fig. 5.18 (b) are compared for the mode *interleaved*. Based on the motion vector recovery principle, the motion compensated image content is copied into the frame in Fig. 5.18 (a). Hence, the changes in luminance to the previous frames caused by the flash of a camera can not be compensated. However, the concealment works better compared to the motion vector recovery based methods as shown in Fig. 5.10 for uncoded frames. The advanced motion compensation of H.264/AVC allows for a better concealment result. The annoying artifacts are avoided by the spatio-temporal extrapolation method which is able to compensate the variations in luminance.

The visual impression is even more distorted in case of a scene cut, as Fig. 5.19 shows.

Table 5.11: Concealment results in dB for PAL sequences. 3-D extrapolation used for *Intra* as well as *Inter* coded frames (except 1st frame, concealed by 2-D extrapolation).

	Dist.	Method	Canoe	Fast Food	Crew	Discovery
Interleaved	-	-	34.65	36.86	36.92	37.18
	Strong	3-D Extrapolation H.264/AVC	29.79	30.00	36.42	34.47
			29.54	30.48	36.08	33.66
	Weak	3-D Extrapolation H.264/AVC	32.84	35.04	36.65	36.16
			32.68	35.64	36.60	35.62
	-	-	34.58	36.81	36.92	37.20
Dispersed	Strong	3-D Extrapolation H.264/AVC	32.59	34.85	36.42	36.20
			32.33	33.52	36.08	34.48
	Weak	3-D Extrapolation H.264/AVC	33.95	36.12	36.72	36.91
			33.76	35.65	36.59	36.47
	-	-	34.58	36.81	36.92	37.20
	-	-	34.58	36.81	36.92	37.20

The principle is the same, the motion compensated blocks are copied from the previous frame even if they do not fit at all. The result is depicted in Fig. 5.19 (a) for checkerboard like losses. Incorporating both spatial *and* temporal information by 3-D extrapolation leads to a visually more appealing result. Obviously, the spatially surrounding information dominates the colour information for the extrapolated area, even though the patterns from the previous frames causing slight artifacts in colour are visible. As can be noticed from Fig. 5.19 (b), these artifacts are, however, less annoying.

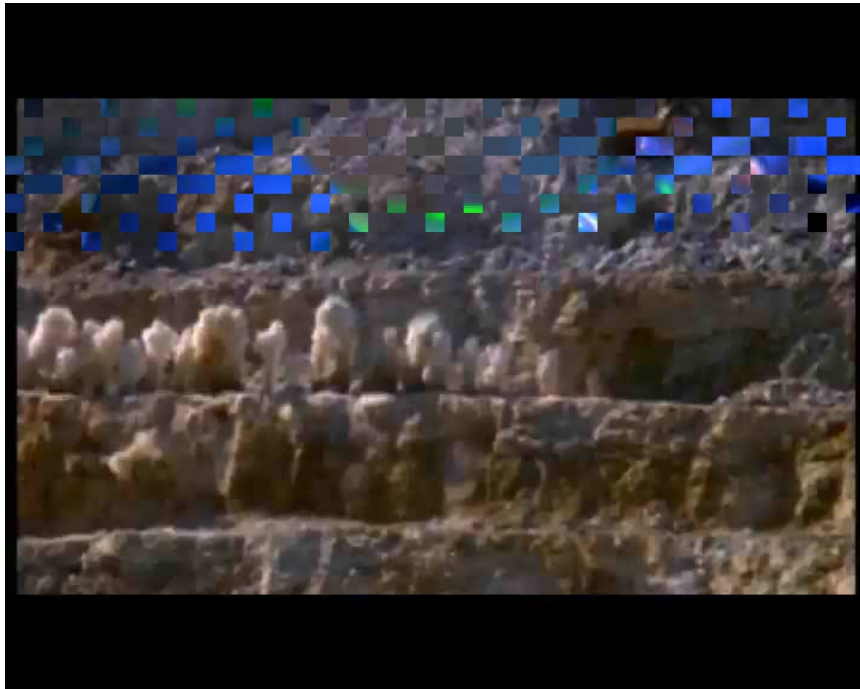


(a) Temporal method H.264/AVC [61], 31.65 dB



(b) 3-D extrapolation, 35.90 dB

Figure 5.18: Concealed frame 29, sequence *Crew*, mode *interleaved*; $1.4 \frac{\text{Mbit}}{\text{s}}$, 25 Hz, PAL.



(a) Temporal method H.264/AVC [61], 33.98 dB



(b) 3-D extrapolation, 36.99 dB

Figure 5.19: Concealed frame 11, sequence *Discovery*, mode *dispersed*; $1.7 \frac{\text{Mbit}}{\text{s}}$, 25 Hz, PAL.

5.4.5 Summary

In this section, both, 2-D and 3-D frequency selective extrapolation has successfully been integrated into a H.264/AVC decoder as concealment feature. The performance was compared to the non-normative concealment methods of the reference software of H.264/AVC for sequences in TV quality with PAL resolution.

The pixel interpolation method implemented in the H.264/AVC reference software results in blurring of the concealed parts, whereas the concealment performance of the 2-D extrapolation shows very convincing visual results due to its ability to conceal different textures like monotonous areas, edges, and structures. The method works especially successfull in combination with FMO using a checkerboard like pattern.

The 3-D extrapolation method is able to inherently compensate motion by exploiting spatio-temporal information without requiring motion vector search. Furthermore, changes in luminance occurring from frame to frame can be compensated inherently. The 3-D frequency selective extrapolation shows especially convincing results in combination with FMO applying a checkerboard like pattern, also compared to the motion vector recovery of the reference software. In case of a scene cut, the motion vector recovery approach results in severe visual degradations. The 3-D extrapolation performs significantly better, although slight artifacts maybe observed. In cases of non-reliable temporal information, the advantages of the 2-D extrapolation method could be exploited for Inter coded MBs. However, the motion of the sequence would have to be analyzed in advance for these cases. The 3-D extrapolation method can also successfully be applied to concealment of Intra coded MBs by exploiting temporal in addition to spatial information. The simulation results have shown that the 3-D extrapolation significantly improves the visual quality compared to the pixel based method of the reference software.

6 Spatial Concealment of Lost Wavelet Coded Data

In the previous chapter, frequency selective extrapolation applied to error concealment for lost block based coded data has been investigated. As already mentioned, there are coders which rely on wavelet based coding principles. Due to the different coding concepts, losses cause completely different artifacts compared to block based coding. Therefore, the impact of a loss on the data stream and error robustness tools offered by the decoder have to be investigated, before frequency selective extrapolation can be integrated into the JPEG2000 decoder as concealment feature.

In the beginning of 2001, the wavelet based image coding standard JPEG2000 Part 1 [1] was introduced in order to meet the requirements of new multimedia applications. Compressed bit streams obtained by redundancy and irrelevancy reduction are especially vulnerable to transmission errors. Therefore, special emphasis was put on error robustness which is of great importance in the case of image transmission over non-reliable channels as, e.g., the channels in mobile communications. JPEG2000 offers tools in order to increase the robustness against these errors.

However, these tools only allow detection of errors and resynchronization within the code-stream. JPEG2000 Wireless (JPWL) or Part 11 addresses therefore the issue of correcting transmission errors [15]. The applied techniques use forward error correction (FEC), data partitioning, interleaving, and unequal error protection (UEP). In case of non-correctable residual errors, the standard also includes a residual error descriptor specifying the locations of residual errors in the code-stream. This information can be exploited in order to conceal the lost data.

After presenting the fundamentals of JPEG2000 in Section 6.1, the error robustness tools of Part 1 are investigated in Section 6.2 in order to generate a robust bit stream. Visual distortions caused by packet losses correspond to lost wavelet coefficients. In Section 6.3, concealment using the 2-D frequency selective extrapolation technique is described. Based on the subsequent simulations and results presented in Section 6.4, the realization of a packet error robust JPEG2000 decoder using frequency selective extrapolation for concealment [33] is given in Section 6.4.4.

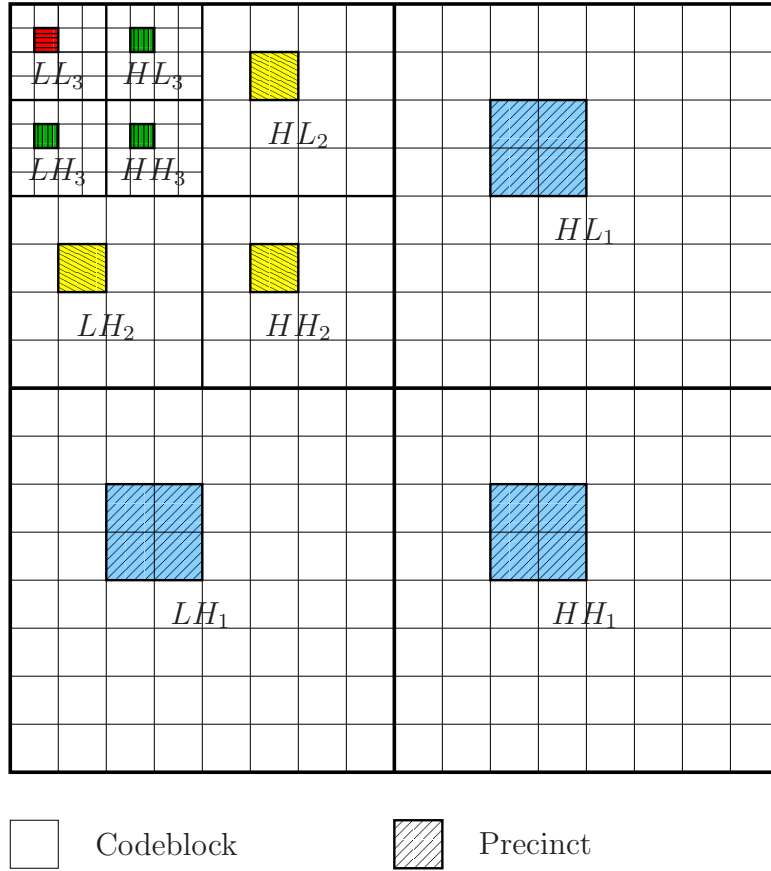
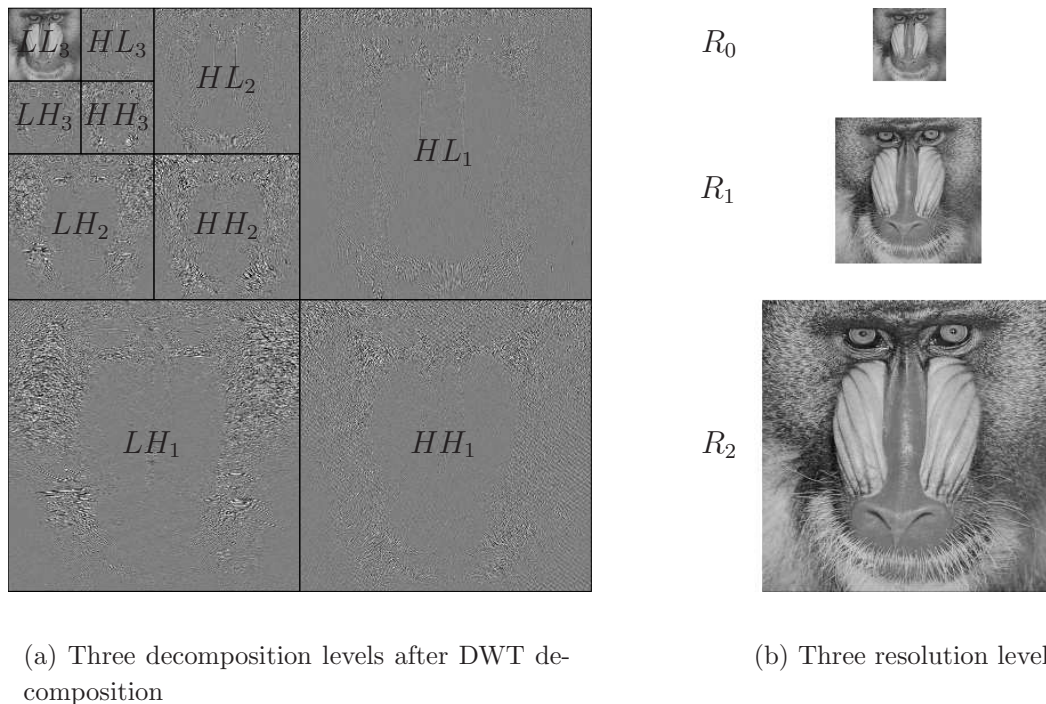


Figure 6.1: The decomposition in subbands, precincts, and codeblocks is shown.

6.1 Fundamentals of JPEG2000

In the following of this section, we briefly review the fundamentals of the JPEG2000 Part 1 coding standard. In the pre-processing step, the image can be split in independent non-overlapping rectangles called tiles. A level offset is applied to the image data in order to center the pixel amplitudes around zero. Further, an optional color transform converts RGB images into a luminance and two chrominance components. The 2-D discrete wavelet transform (DWT) decomposes the pre-processed image data into different decomposition levels. Each decomposition level consists of subbands with the corresponding wavelet coefficients. The subbands are referred to as LL , HL , LH , HH , where L denotes lowpass filtering and H highpass filtering. Fig. 6.1 shows an image transformed by three DWT decompositions generating three decomposition levels and four resolution levels. Each decomposition level consists of three subbands HL , LH , HH except for the lowest decomposition level which includes also the LL band. The corresponding index denotes the decomposition level.

The quantized coefficients are collected in code-blocks whereas the individual code-blocks are processed in bit-planes by an arithmetic coder. The desired bit-rate is obtained

Figure 6.2: Wavelet decomposition of image *Baboon*.

by rate-distortion optimized truncation of the code-block data streams by embedded block coding with optimal truncation (EBCOT). Optionally it is possible to form layers consisting of the entropy coded bit-plane data. Code-blocks from the same resolution level which are spatially located at the same position in the subbands are organized in precincts as illustrated in Fig. 6.1. For each layer, the data in each precinct constitutes a packet. In case of a packet loss, wavelet coefficients belonging to one precinct get lost.

An example of a 2-D wavelet decomposed image is given in Fig. 6.2. Three DWT decompositions are applied to the image *Baboon*. In Fig. 6.2(a), the corresponding subbands containing the wavelet coefficients are shown. The corresponding resolution levels are depicted in Fig. 6.2(b). As can be seen, the lowpass samples represent the original image with a lower resolution. The LL bands of the three DWT decompositions are referred to as R_0 , R_1 , and R_2 . Accordingly, the original is denoted by R_3 .

6.2 Tools for Error Control in JPEG2000

6.2.1 Effects of Transmission Errors

In the following, we look at the error control tools offered by the JPEG2000 coder. It should be pointed out that the standard specifies only the code stream syntax, whereas the use of error control mechanisms depends on the specific coder and decoder implemen-

tation. Therefore, we investigate the impact of transmission errors on the JPEG2000 bit stream first. Two types of transmission errors are considered, bit errors and packet losses.

The impact of bit errors strongly depends on the error location as investigations in [21, 9] show. For example, a bit error in the header may cause a complete data loss in contrary to an error occurring at the end of a code-block which might hardly be visible. Packet losses have a fatal result on the coded image because they lead to a synchronization loss between coder and decoder.

6.2.2 Error Robustness on Entropy Coding Level

Code-blocks are coded independently by the concept of embedded block coding with optimal truncation (EBCOT). Bit errors hence affect only the corresponding code-block. Additionally, JPEG2000 offers the opportunity to use two kinds of termination strategies of the arithmetic coder on bit plane level. Thereby, the detection of bit errors and the deletion of corrupted data is made possible. As already mentioned, one precinct constitutes a packet. Investigations in [21, 40] show that using smaller precincts increases the robustness against bit errors, but the coding efficiency is obviously decreased.

6.2.3 Error Robustness on Packet Level

Assuring the bit stream synchronization plays an important role in JPEG2000 coding. Start Of Packet (SOP) marker segments can be inserted into the data stream prior to each packet. The marker segment consist of 6 Bytes including a sequence number. The sequence number starts with zero for the first packet and increments by one with each following packet. If a SOP marker segment with the right sequence number prior to the packet header is not available during decoding, an error occurred in the data stream. A robust decoder can look for the next correct SOP marker with a higher sequence number and continue the decoding process.

Using small precincts increases the error robustness also in the packet loss scenario because the amount of lost data is reduced.

6.2.4 Concealment by Inserting Empty Packets

In order to assure synchronization, the sequence numbers contained in every SOP marker segment have to be counted. If the sequence number does not increment by one, a packet has been lost and is replaced by an empty packet with a correct sequence number which corresponds to inserting zeros instead of the lost wavelet coefficients. Note that due to the level offset in JPEG2000, the zeros correspond to an average gray value of the wavelet coefficients. In the case of layered transmission also the packets belonging to lower bit-planes in the same precinct have to be discarded. The idea of inserting empty packets

has been proposed in [9]. Additionally, the same technique can be applied when a SOP marker segment is not found at the expected position which means that a bit error has occurred in the previous packet header. As in the aforementioned case, the affected packet can be replaced by an empty packet.

6.3 Concealment by Frequency Selective Extrapolation

Error control is necessary in the presence of transmission errors as shown in the preceding section. So far we preserve synchronization by using error robustness mechanisms and obtain a decodable bit stream. However, the affected wavelet coefficients are lost and can cause a severe spatially limited degradation in visual quality. Our goal is to reduce the effects caused by packet losses. Therefore, we investigate the 2-D frequency selective extrapolation algorithm as an error concealment algorithm in order to conceal the lost wavelet coefficients in a subband. Fig. 6.1 shows an image transformed by a DWT generating four resolution levels. In case of a packet loss, wavelet coefficients belonging to one precinct get lost.

The proposed concealment algorithm uses frequency selective extrapolation which is exemplified for a packet loss in the lowpass band LL_3 in the following. The extrapolation task is illustrated in Fig. 6.3 showing a part of Fig. 6.1. The area \mathcal{L} in the LL_3 band consists of the missing wavelet coefficients, the area to be estimated \mathcal{B} (light gray), and the surrounding correctly received wavelet coefficients, the support area \mathcal{A} (dark gray). In this case, $f[m, n]$ denote the original wavelet coefficients of the considered subband. The parametric model $g[m, n]$ approximates the support area by a linear combination of weighted basis functions. In order to determine the expansion coefficients, the error criterion (3.2) between the original correctly received wavelet coefficients and the parametric model is minimized with respect to the support area. The missing wavelet coefficients are extrapolated from the surrounding correctly received wavelet coefficients of the same resolution level. The block \mathcal{L} as shown in Fig. 6.3 is transformed into the DFT domain where the extrapolation is performed according to Section 3.4.2.

The 2-D isotropic model introduced for the concealment of lost block coded data is also applied as weighting function (3.52) for lost wavelet coded data. Regarding the simulation results in [21], we use the following parameter set for the concealment approach: The size of the support area is set to 4 surrounding wavelet coefficients and the DFT-size equals 64×64 . The concealment performance of the frequency selective extrapolation is therefore considered in the next section with respect to:

- The decay parameter $\hat{\rho}$ as introduced in (3.52).
- The termination criterion maximum number of iterations.

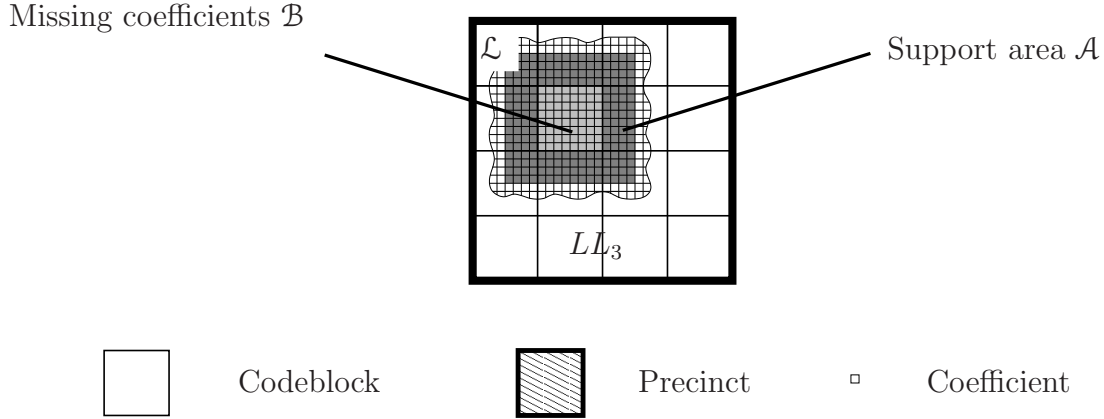


Figure 6.3: Concealment in case of packet loss in lowpass band LL_3 : The known area \mathcal{A} (dark gray) is approximated by a parametric model and the missing area \mathcal{B} (light gray) is obtained by extrapolation.

6.4 Simulations and Results

In [33], the standard decoder software JJ2000 [11] has been extended for the use in error prone environments by the following features: Decoding was enabled for data streams containing SOP markers including the insertion of empty packets according to [9]. In the following simulations, the concealment performance of frequency selective extrapolation is compared to the concealment method described in [9] and concealment by the mean of the adjacent wavelet coefficients.

We investigate the performance of the concealment algorithms for images with a size of 512×512 pixels. Four DWTs are applied at a bitrate of 0.5 bpp. A code block consists of 64×64 wavelet coefficients. The impact of a loss depends on the resolution level and is investigated in combination with subsequent concealment. In order to evaluate the performance, a lost packet corresponding to a precinct is simulated for the considered subband. The simulations are done for varying precinct sizes for the considered subband which are specified for each simulation. For the other subbands not evaluated in the respective simulation, one precinct per subband is chosen. The PSNR of the concealed image with respect to the original image is calculated. This procedure is repeated for each distinct precinct position in the subband. The PSNR is averaged over all blocks.

Based on the conclusions, the realization of a decoder robust against packet losses using frequency selective extrapolation is presented in Section 6.4.4. The concealment method is also applicable to colored images and is performed analogously to block coded data on each decorrelated YUV component separately.

6.4.1 Concealment of Lowpass Band Losses

A packet loss in the lowpass band of the layer consisting of the most significant bit planes causes the most severe visual degradation, since in general the lowpass components contain most of the image information. Because losses of wavelet coded data have completely different effects on the reconstructed images compared to block coded data, the behavior of frequency selective extrapolation using the isotropic weighting function is presented first for a loss in the lowpass band. The comparison of the concealment performances of different methods is given afterwards.

Concealment Performance of Frequency Selective Extrapolation

The error robustness is increased in the lowpass band LL_4 by choosing a smaller precinct size compared to the other subbands. A precinct size of 4×4 coefficients is selected which corresponds to 64×64 pixels in the reconstructed image. Although the codeblock size is chosen to 64×64 coefficients, the *effective* codeblock size is limited by the precinct size to 4×4 . For the other subbands, one precinct per subband is chosen. The results for the three test images *Lena*, *Peppers*, and *Baboon* are presented in Fig. 6.4. The evaluations of the parameters of frequency selective extrapolation are done for a rate of 8 bpp simulating uncoded images in order to exclude additional coding artifacts. The PSNR gains given in Fig. 6.4 denote the differences of the PSNR values of the extrapolation based concealed images and the images which are reconstructed by inserting zeros. Hence, the PSNR gain measures to what extent the extrapolation based concealed image is more similar to the original than the other one and, thus, expresses the quality of the extrapolation. In order to evaluate the performance, a lost packet in the lowpass band is simulated for each possible precinct position in the lowpass band. The PSNR gain is averaged over all blocks.

The results in Fig. 6.4 show that the maximum gain can be achieved for 4 to 7 iterations independent on the image and $\hat{\rho}$. The best results can be achieved for $0.35 \leq \hat{\rho} \leq 0.45$. In comparison to spatial concealment of block losses in Section 3.5.3, the optimum value of $\hat{\rho}$ is much lower. There, $0.74 \leq \hat{\rho} \leq 0.86$ leads to the best results. However, the LL_4 image is a subsampled version of the original. For a precinct size of 4×4 and four DWT decompositions, the loss corresponds to 64×64 pixels and the number of surrounding pixels of the support area to 64 known pixels in the reconstructed image. Therefore, the value of $\hat{\rho}$ has to be chosen lower for a loss in the lowpass band.

In summary, based on the evaluations in Fig. 6.4, $\hat{\rho} = 0.3$ and a maximum number of 4 iterations are selected as parameters. Using these parameters, very high PSNR gains of 6 dB, 3.6 dB, and 5.8 dB for the images *Lena*, *Peppers*, and *Baboon*, respectively, are obtained compared to the empty packet insertion.

So far, the evaluations are done for uncoded images only in order to avoid coding arti-

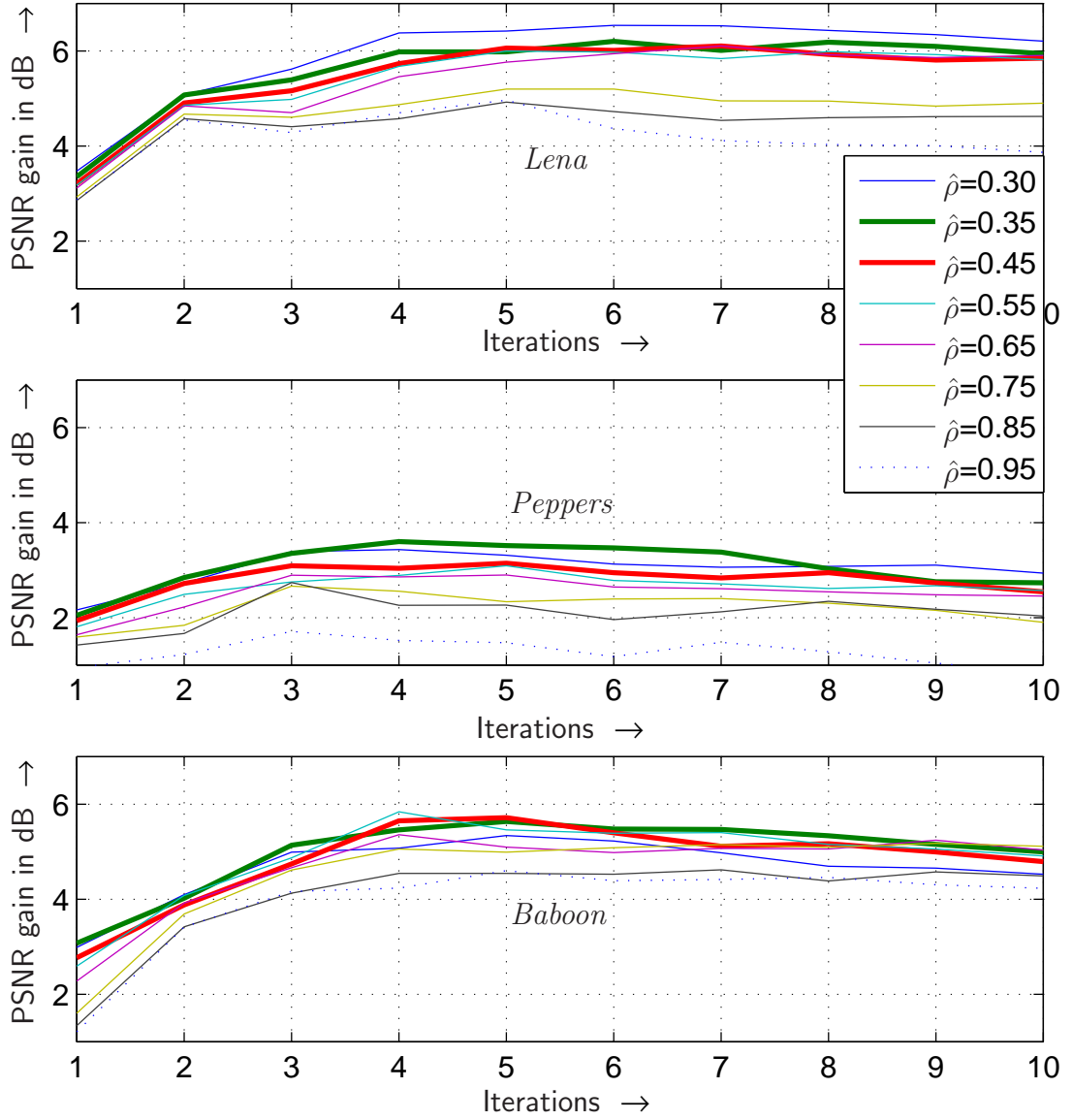


Figure 6.4: Average PSNR gain for frequency selective extrapolation of losses in the LL_4 band at a precinct size of 4×4 and 8 bpp coding rate.

facts. Next, results are presented for coded images at a rate of 0.5 bpp. The performance of frequency selective extrapolation applying the chosen parameters is evaluated. For the reconstructed images *Lena*, *Peppers* and *Baboon*, 32.70 dB, 32.93 dB and 25.28 dB, respectively, are obtained. The parameters are selected appropriately as the comparison to results achieved by the optimum parameters with 32.72 dB, 32.96 dB, and 25.30 dB for *Lena*, *Peppers* and *Baboon*, respectively, shows.

Table 6.1: Concealment of losses in LL_4 band for four DWT decompositions. The PSNR is calculated with respect to the original image. For extrapolation, results obtained with the optimum parameters and a set of parameters chosen for each image are presented.

Rate 0.5 bpp					
	Precinct size 4×4				
	coded	[9]	mean	extrap. opt.	extrap. set
Lena:	35.00	29.81	31.60	32.72	32.70
Peppers:	35.66	31.44	31.80	32.96	32.93
Baboon:	25.40	25.00	25.16	25.30	25.28

Table 6.2: Concealment of losses in LL_4 band for four DWT decompositions. The PSNR is calculated with respect to the original image.

Rate 0.5 bpp								
	Precinct size 2×2				Precinct size 8×8			
	coded	[9]	mean	extrap.	coded	[9]	mean	extrap.
Lena:	34.54	32.90	33.85	34.22	35.13	24.64	26.29	27.15
Peppers:	35.21	33.58	34.33	34.53	35.78	28.12	27.78	28.33
Baboon:	24.88	24.78	24.85	24.86	25.55	23.95	23.95	24.78

Comparison to Related Techniques

In the following, we compare the different concealment techniques for different precinct sizes containing 2×2 , 4×4 , or 8×8 coefficients in the lowpass band LL_4 which correspond to 32×32 , 64×64 , or 128×128 pixels in the reconstructed image, respectively. The images are coded at a bitrate of 0.5 bpp.

Table 6.1 shows the average concealment performance of the methods for a precinct size of 4×4 . A lost packet in the lowpass band is simulated and the PSNR of the concealed image with respect to the original image is calculated. This procedure is repeated for each possible precinct position in the lowpass band and the PSNR results are averaged. The results of frequency selective extrapolation are compared to the PSNR results for the coded and error free image, the concealment method described in [9], and concealment by the mean of the adjacent coefficients. In case of extrapolation, results obtained with the optimum parameters according to [21] and a set of parameters chosen for each image are presented.

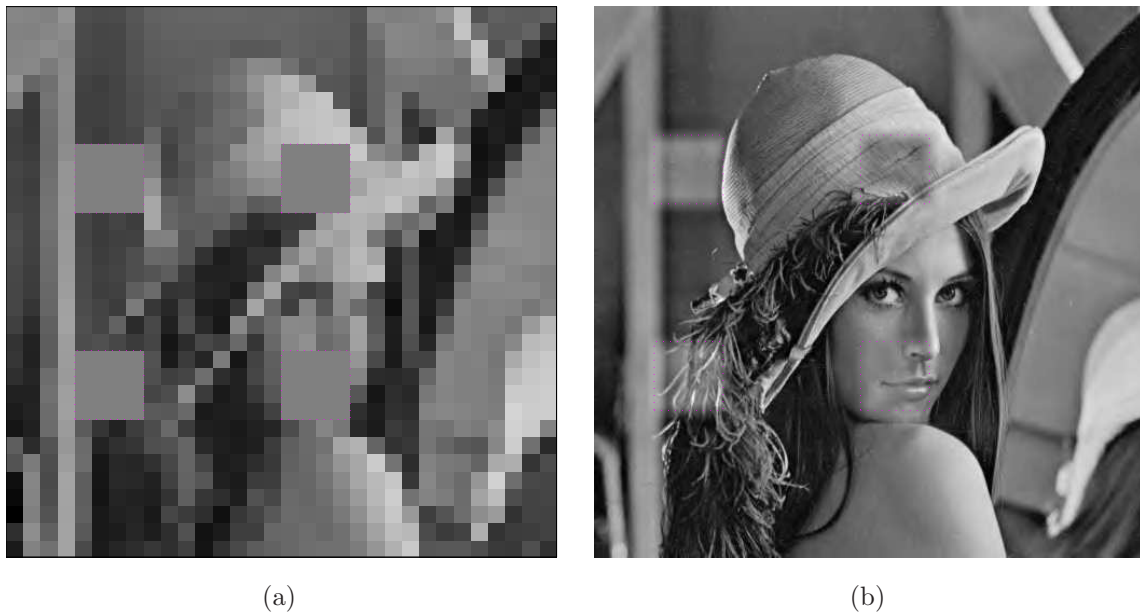


Figure 6.5: Concealment of four packet losses in the lowpass band (LL_4) by *empty packet insertion* [9] at a precinct size of 4×4 in the LL_4 band. The coded, error free image yields 35.00 dB. (a) LL_4 band with four concealed packet losses is shown. (b) The concealed and reconstructed image yields 25.33 dB.

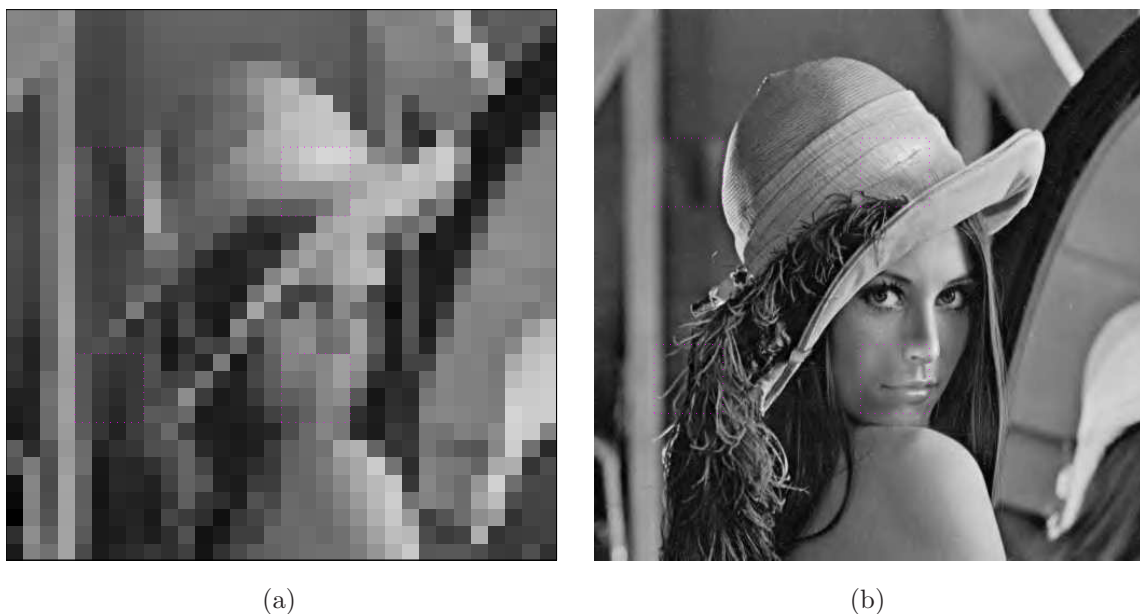


Figure 6.6: Concealment of four packet losses in the lowpass band (LL_4) by *frequency selective extrapolation* using $\hat{\rho} = 0.4$, maximally 6 iterations at a precinct size of 4×4 in the LL_4 band. The coded, error free image yields 35.00 dB. (a) LL_4 band with four concealed packet losses is shown. (b) The concealed and reconstructed image yields 31.70 dB.

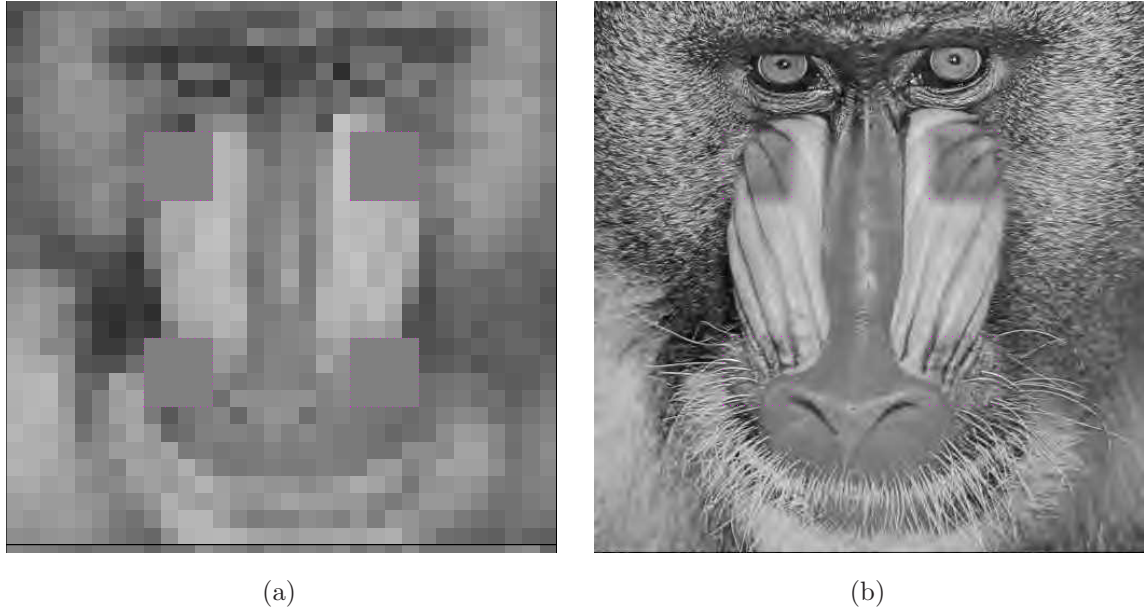


Figure 6.7: Concealment of four packet losses in the lowpass band (LL_4) by *empty packet insertion* [9] at a precinct size of 4×4 in the LL_4 band. The coded, error free image yields 25.40 dB. (a) LL_4 band with four concealed packet losses is shown. (b) The concealed and reconstructed image yields 23.99 dB.

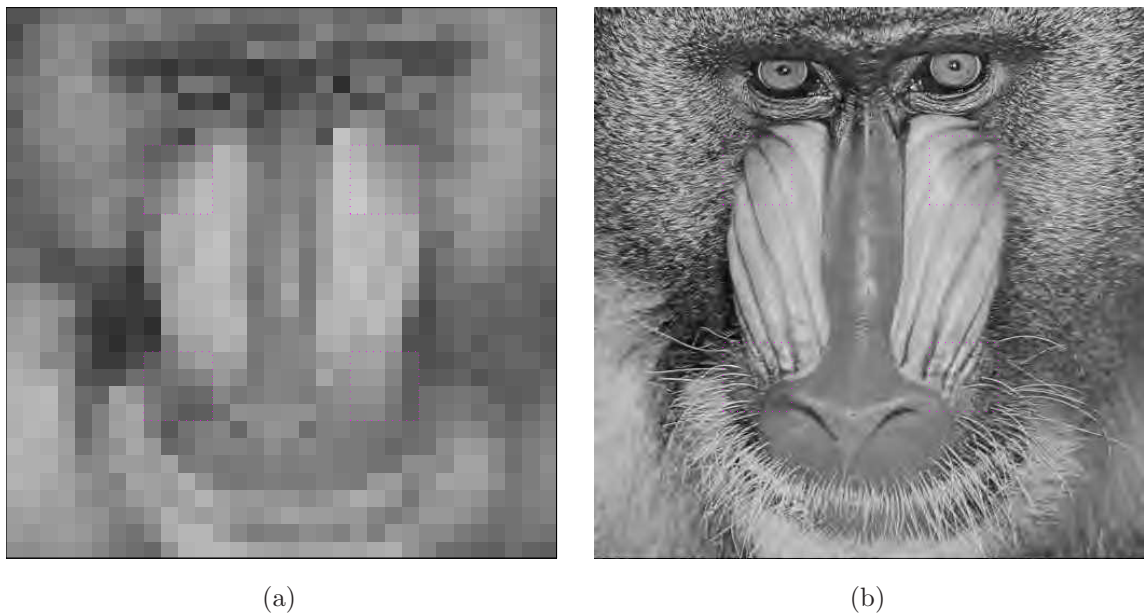


Figure 6.8: Concealment of four packet losses in the lowpass band (LL_4) by *frequency selective extrapolation* using $\hat{\rho} = 0.45$, maximally 4 iterations at a precinct size of 4×4 LL_4 band. The coded, error free image yields 25.40 dB. (a) LL_4 band with four concealed packet losses is shown. (b) The concealed and reconstructed image yields 24.97 dB.

For the coded and error free images, we obtain 35.00 dB for *Lena* and 35.66 dB for *Peppers* according to Table 6.1. The image *Baboon* has only 25.40 dB because of the noise like behavior of the fur, which concentrates more energy in the highpass bands than the other two images. The highpass bands are quantized coarser due to compression using irrelevancy reduction. The images *Lena* and *Peppers* suffer the most from packet losses in the lowpass band, the obtained image quality using zero insertion [9] is 29.81 dB and 31.44 dB, respectively.

In order to get an impression of the visual degradation caused by a packet loss, we present the image *Lena* containing four concealed packet losses. Fig. 6.5 (a) shows the concealed lowpass signal of four packet losses by inserting empty packets and in Fig. 6.5 (b) the reconstructed image. Severe visual distortions are clearly visible. Since the amplitudes of the image data are centered around zero, inserting an empty packet corresponds to an average gray level. In order to better adapt to the image, we used also the mean value of the correctly received adjacent coefficients for the concealment process. The PSNR result of the image *Lena* can be improved from 29.81 dB to 31.60 dB using this method.

However, using the frequency selective extrapolation method we can further improve the concealment result up to 32.72 dB using the parameters $\hat{\rho} = 0.4$ and a maximum number of 6 iterations. The visual result is shown in Fig. 6.6. Fig. 6.6 (a) shows the concealed lowpass image with the four corresponding packet losses as in Fig. 6.5 and in Fig. 6.6 (b) the reconstructed image. The severe visual degradation is well compensated. The frequency selective extrapolation method is more successful than the insertion of either an average gray level or the mean value of the surrounding coefficients. Note that the concealment performance of the proposed algorithm for the image *Baboon* shows a different behavior than for *Lena* and *Peppers*. The PSNR results do not reflect the achieved good subjective results. Therefore, the concealment results for the method [9] and the proposed algorithm of four block losses are compared in Fig. 6.7 and Fig. 6.8, respectively. On the left hand sides the concealed lowpass images are depicted and on the right hand side the corresponding reconstructed images. Using the proposed method we can improve the picture quality from 23.99 dB to 24.97 dB which is also clearly visible.

The results for 2×2 precincts are given in Table 6.2 as well as the results for 8×8 precincts. In case of smaller precincts, the coding efficiency is reduced because each SOP marker has a contribution of 6 bytes. However, the error robustness and hence the concealment performance is improved. For larger precincts the coding efficiency is improved at the expense of a reduced error robustness. The concealment performance is still improved by the frequency selective extrapolation technique compared to inserting zeros also for 8×8 precincts. However, it is less efficient in this case since more data is lost in the case of a packet loss. Hence, the data loss has to be limited by error robustness techniques in order to conceal a loss successfully by the proposed technique. In conclusion, we can state that depending on the considered application a suitable trade-off has to be

Table 6.3: Impact of a packet loss depending on the decomposition level for 4 DWT decompositions. The PSNR is calculated with respect to original image.

Rate 0.5 bpp, loss corresponds to 64×64 pixels in reconstructed image					
	Coded, error free	LH_4, HL_4, HH_4 band Precinct size 8×8		LH_3, HL_3, HH_3 band Precinct size 16×16	
Concealment by		[9]	extrapolation	[9]	extrapolation
Lena:	35.00	34.32	34.28	34.45	34.43
Peppers:	35.66	34.98	34.94	35.32	35.30
Baboon:	25.40	25.35	25.35	25.35	25.35

found between error robustness and coding efficiency.

6.4.2 Concealment of Highpass Band Losses

The performance of error concealment in highpass bands is also investigated. As each packet contains data of one precinct, missing wavelet coefficients in three subbands of the corresponding resolution level as illustrated in Fig. 6.1 have to be concealed. In Table 6.3, the concealment performance of empty packet insertion is compared to the frequency selective method for the highpass bands of the third and fourth decomposition level. Obviously, the performance of inserting zeros is comparable to the proposed method at a reduced computational complexity. In any case, a packet loss in the highpass subband affects the received image quality less than in the lowpass case since compressed images concentrate less energy in the highpass band. Therefore, the need for concealment is decreased and the JPEG2000 coder provides excellent inherent concealment strategies. Four packet losses in the HL_4, LH_4, HH_4 bands of the fourth decomposition level concealed by empty packet insertion are depicted in Fig. 6.9 (a) and in Fig. 6.9 (b) the reconstructed image is shown. Additionally, in order to successfully conceal from the surrounding, smaller precincts are required which reduces the coding efficiency noticeably. As can be seen in Table 6.3, the impact of a packet loss in the third decomposition level is even less strong than it has been already expected.

6.4.3 Concealment of Lower Layer Losses

The impact of a packet loss in a layer below the most significant one is evaluated next. For this purpose, we generated an image with 10 equally spaced layers numbered from 0 to 9 with a rate of 0.05 bpp per layer. Four packet losses were applied in the lowpass band to the first layer which is the layer below the most significant one. The coded, error free

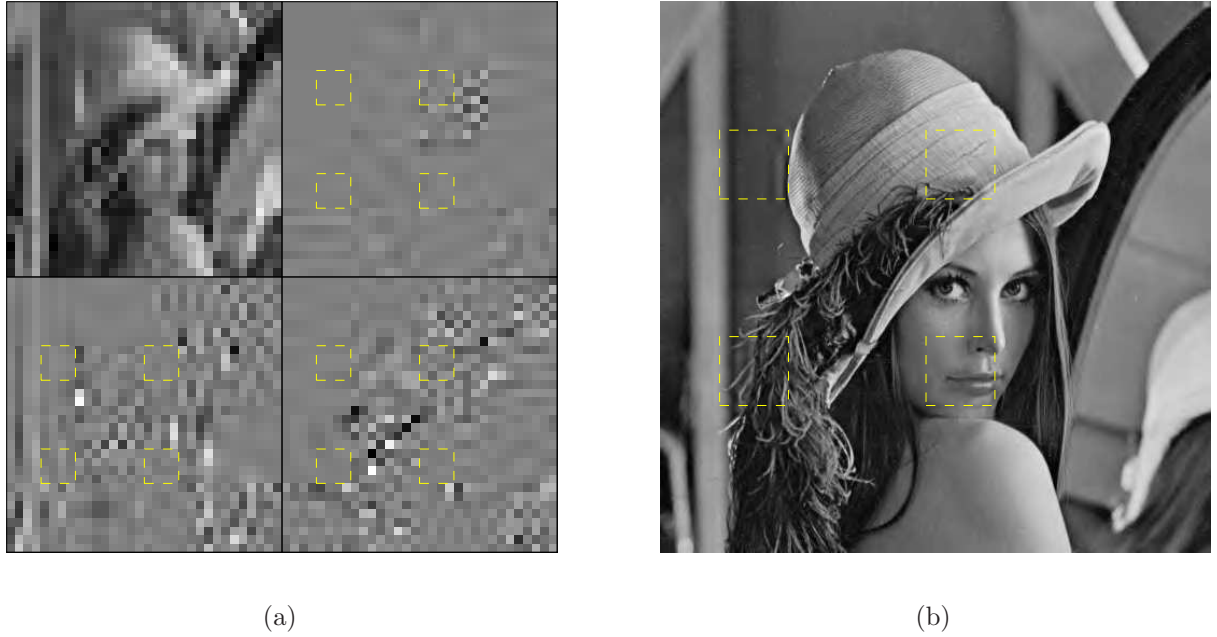


Figure 6.9: Concealment of four packet losses in the HL_4, LH_4, HH_4 bands by *empty packet insertion* [9] at a precinct size of 8×8 in the HL_4, LH_4, HH_4 bands. The coded, error free image yields 35.00 dB. (a) The HL_4, LH_4, HH_4 bands with four concealed packet losses are shown. (b) The concealed and reconstructed image yields 33.51 dB.

image yields 33.31 dB and the image including the four concealed packet losses according to [9] 33.13 dB. Since the impact is only marginal, we proceed as before. If the packet loss happens in the lowpass band at the most significant layer, we conceal it by the frequency selective extrapolation technique, otherwise the lost packet and the packets in the layers below are replaced by empty packets which corresponds to a coarser quantization of the wavelet coefficients belonging to the precinct of the packet.

6.4.4 Realization of Packet Error Robust JPEG2000 Decoder

Based on the above investigation results, we give an example of using the developed decoder which is robust against packet losses. The image *Lena* with a size of 512×512 pixels is transformed by four DWT decompositions and 10 layers are used. The applied code-block size is 64×64 coefficients. We use 16 precincts per resolution level except for the lowpass band which includes 64 precincts corresponding to a size of 4×4 coefficients. The precincts are numbered in raster-scan order. SOP marker segments are inserted before each packet. In order to test the decoder, packets from different resolution levels, color components and layers are discarded. Fig. 6.10 shows on top the corrupted image *Lena* concealed by inserting empty packets. On the bottom of the figure it shows the output of the proposed error robust JPEG2000 decoder. The remarkable image quality

improvement is due to the frequency selective extrapolation technique applied to the lowpass coefficients.

6.5 Summary

The coding standard JPEG2000 has been investigated with respect to error control features. By inserting markers offered by the JPEG2000 coder, it is possible to preserve synchronization and to generate a decodable bit stream also in error prone environments. The codestream generated by inserting empty packets for concealment [9] is fully decodable by every standard decoder as no changes to the standard JPEG2000 coder have to be applied. Since packet losses in the lowpass band cause the most severe visual distortion, smaller precinct sizes should be used in the lowpass band in order to increase the error robustness by limiting the data loss.

The severe visual degradations introduced by losses in the lowpass band are concealed using frequency selective extrapolation. The missing wavelet coefficients are replaced by extrapolating the surrounding correctly received wavelet coefficients of the same resolution level. Thereby, the image quality of the distorted areas due to a loss is remarkably improved up to 2.9 dB by using the frequency selective extrapolation technique instead of simple zero insertion. Losses occurring in the other layers and subbands do not cause such a degradation of the visual quality due to the decoder inherent error robustness. Therefore, in these cases it is appropriate to apply zero packet insertion and exploit the excellent inherent concealment properties due to the EBCOT principle of the JPEG2000 decoder.



Figure 6.10: Color image *Lena* (512×512). Rate: 1.5 bpp. Layer: 10. Precincts: 128×128 , 64×64 , 32×32 , 16×16 , 4×4 . Packet losses (*resolution level, component, layer, precinct number*): (0,0,0,34); (0,0,0,20); (0,1,0,17); (0,2,0,36); (0,0,0,51); (0,1,0,51); (0,2,0,51); (1,0,1,15). Top: Concealment by [9]. Bottom: Improved concealment by frequency selective extrapolation.

7 Further Applications

In this thesis, the estimation of unknown samples in a video signal is described as extrapolation of the spatio-temporal surrounding signal. Concealment can be seen as an extrapolation problem and has been already extensively discussed in the previous chapters. In the following, we consider the removal of TV logos and prediction in hybrid video coding as applications of the introduced frequency selective extrapolation technique.

7.1 TV Logo Removal

In the following of this section, we show that the removal of undesired TV logos in TV sequences can be also treated as an extrapolation problem. Broadcast stations often want to reuse TV sequences like sports sequences or news originally produced by another station. They seek to remove the embedded TV logo and to replace it by their own logo. A common technique is to blur the undesired logo and place the own one in a different corner. Examples for such blurred logos are given in Fig. 7.1 sampled from analog German television [39]. Another common method is to overlay multiple logos. The application of these methods leads to visually unsatisfying results.

In [64], a simple approach for logo removal is presented. The logo region is filled in by interpolation from adjacent neighboring pixels. The fill in causes blurring in case of larger logos, because details cannot be restored by this method. Other techniques which could be used for logo removal are methods commonly applied to the problem of *inpainting*. Possible applications are scratch removal from old photographs or the removal of writings. The most common inpainting algorithm is introduced in [10] which propagates the information surrounding the region to be inpainted to the inside in a special processing order along the so-called isophote lines at the region boundaries. The inpainted area of this spatial pixel based algorithm is obtained iteratively. In [30], it is used as spatial restoration method as part of a spatio-temporal method applied to the removal of texts in video sequences. The temporal restoration relies on block matching principles known from BMA for motion compensation. For stationary texts, the background is classified into two categories: stationary and moving background. In case of stationary background, the text is inpainted by the purely spatial inpainting technique of [10]. For moving backgrounds, all text pixels are replaced by the corresponding motion compensated pixels from the previous frame if they do not belong to the text in the previous frame. Except for very

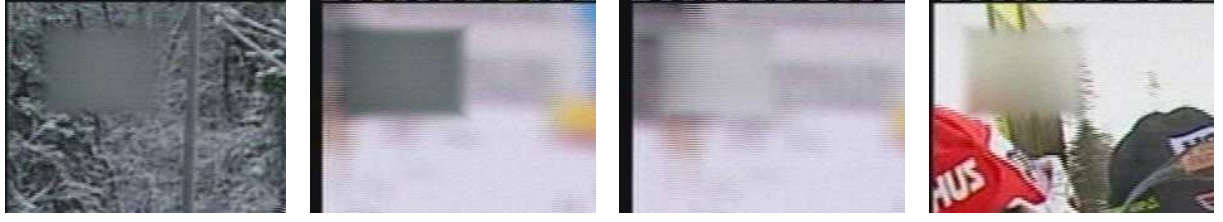


Figure 7.1: Example of common logo removal technique sampled from analog German television.

fast motion with an displacement as large as the text letters, only parts of the text fonts can be motion compensated. The remaining text pixels in the considered frame are inpainted according to [10].

Our approach replaces the undesired logo by extrapolating the surrounding signal into the logo area with the developed frequency selective method [38, 36]. The binary weighting function is determined by a pixel accurate logo mask which can be detected by appropriate algorithms as e.g. described in [38] based on change detection [3]. In order to achieve a good quality for the area of the removed logo, it is important that the logo is detected accurately. Otherwise artifacts are produced because the logo colours are propagated into the logo region.

In Section 7.1.1, logo removal using frequency selective extrapolation is described. Simulations and results are presented in Section 7.1.2 for both, 2-D and 3-D extrapolation. A comparison of the spatial logo removal technique proposed in [64] to 2-D frequency selective extrapolation is given in Section 7.1.3.

7.1.1 Logo Removal by Frequency Selective Extrapolation

In the following, the extrapolation task for logo removal is described for 3-D. The logo is always present at the same position for each frame of the sequence. In contrast to concealment with a missing 2-D area, the loss reaches over a 3-D volume for logo removal. Note that this implies that no simple copying from previous frames is possible. The modified situation is shown in Fig. 7.2. The actual frame τ is marked green. In this example, three previous frames $N_V = 3$ marked red and three following frames $N_N = 3$ marked yellow are taken into account. Note that the logo area has been already replaced in previous frames and can be used for extrapolation. On the other hand, the logo region in the future frames belongs to the missing area \mathcal{B} . The entire volume reaching from frame $\tau - 3$ to $\tau + 3$ is transformed to the DFT domain, where the frequency selective extrapolation is performed. As already mentioned, each successive approximation provides an extrapolation of all missing pixels due to the definition of the basis functions in the entire considered volume. Thus, an estimation for all unknown logo areas is given and the logo can be replaced in more frames than the actual frame. Here, the logo is replaced in a volume with $N_E = 3$ frames at once which are the frames τ to $\tau + 2$ bordered

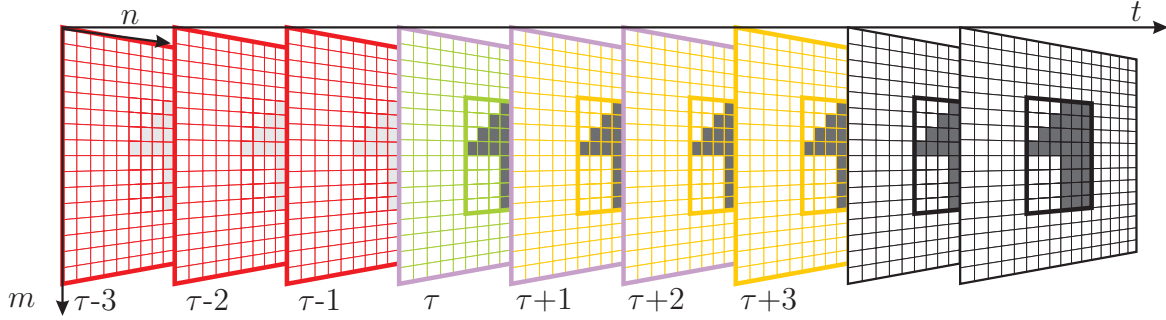


Figure 7.2: Example of logo removal with the logo replaced in $N_E = 3$ frames at once (purple bordered) with $N_V = 3$ previous (red) and $N_N = 3$ subsequent (yellow) frames.

by a purple colored line. After replacing the logos, the volume to be processed is not shifted frame by frame in this case, but by N_E frames. This reduces the computational complexity compared to a frame by frame processing. If the 2-D extrapolation is used for logo removal, the extrapolation task is reduced to frame τ .

The isotropic model introduced for concealment represents also a suitable model for logo removal. For 2-D extrapolation, the isotropic 2-D model as defined in (3.52) is used for the weighting function. In case of 3-D extrapolation, the weighting function relies on the 3-D model (4.27). An example for a weighting function using the 2-D isotropic model is given on the left hand side of Fig. 7.3 using the provided logo mask of the right hand side.

In case of large logos, the logo is partitioned into blocks and the different blocks are processed subsequently. The prespecified target size of a block determines the number of blocks to be processed for the removal of one logo. The investigations in [65] show that a suitable target size of a block to be processed equals the size of a MB in error concealment. A block to be processed should be surrounded by as much original data as possible. Therefore, the processing order of the blocks to be processed is aligned accordingly: first the corner blocks and then the inner blocks are inpainted. For the inner blocks already inpainted blocks are used but weighted by 0.1 in order to limit their influence as previously discussed in the concealment context. If no partitioning of the logo area is desired and the logo does not show a quadratic shape, separable models for the weighting function can be applied which allows for choosing the decay parameters in horizontal and vertical direction separately.

7.1.2 Simulations and Results

The logo removal algorithm was tested for sequences sampled from analog German television. The sampled deinterlaced test sequences have PAL format corresponding to 768×576 pixels at a length of 30 s with 25 frames per second. First, a pixel accurate logo mask

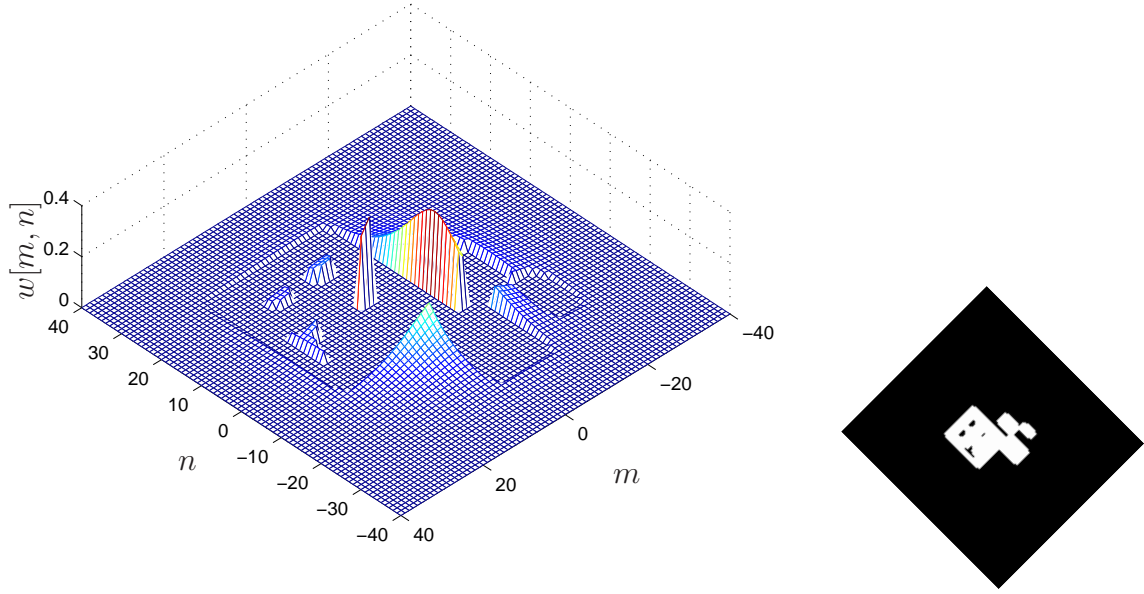


Figure 7.3: Weighting function using the 2-D isotropic model with $\hat{\rho} = 0.8$ for the provided logo mask.

is detected by the algorithm described in [57, 38] which is based on change detection. Subsequently, the logo is removed by frequency selective extrapolation applying both, the 2-D [38] and the 3-D approach [36]. For 2-D extrapolation, the parameters are chosen according to the investigations in [65] and for 3-D extrapolation according to [14]. Unless otherwise noted, they are equal to the parameters already used for concealment. The performance of 3-D extrapolation depends on the motion of the sequence and is discussed for sequences with no motion, moderate motion, fast motion and scene cuts.

First, results are presented for a sequence with moderate motion. Objective evaluations are not possible, because the original signal is unknown and covered by the logo. Therefore, five logos are placed in the surrounding of the original logo in each frame of the sequence as demonstrated for frame number 77 of the sequence *BR* in Fig. 7.4. The five inserted logos are subsequently removed and the performance is measured in terms of PSNR with respect to the original sequence for the extrapolated areas. In Fig. 7.5, the performance of 2-D to 3-D extrapolation is compared [36] for a scene of the sequence *RTL* with moderate motion. The decay parameter $\hat{\rho}$ equals 0.8 for both, 2-D and 3-D extrapolation. The performance of 3-D extrapolation is presented for a varying number of involved previous N_V and following frames N_N at one frame $N_E = 1$ with a replaced logo. The results show that two to three previous and following frames are reasonable as for concealment. According to the investigations in [14], the logos can be in general replaced at once in a volume of $N_E = 2$ or $N_E = 3$ frames. As can be seen from Fig. 7.5, the 3-D approach performs better than the 2-D approach. However, the objective improve-

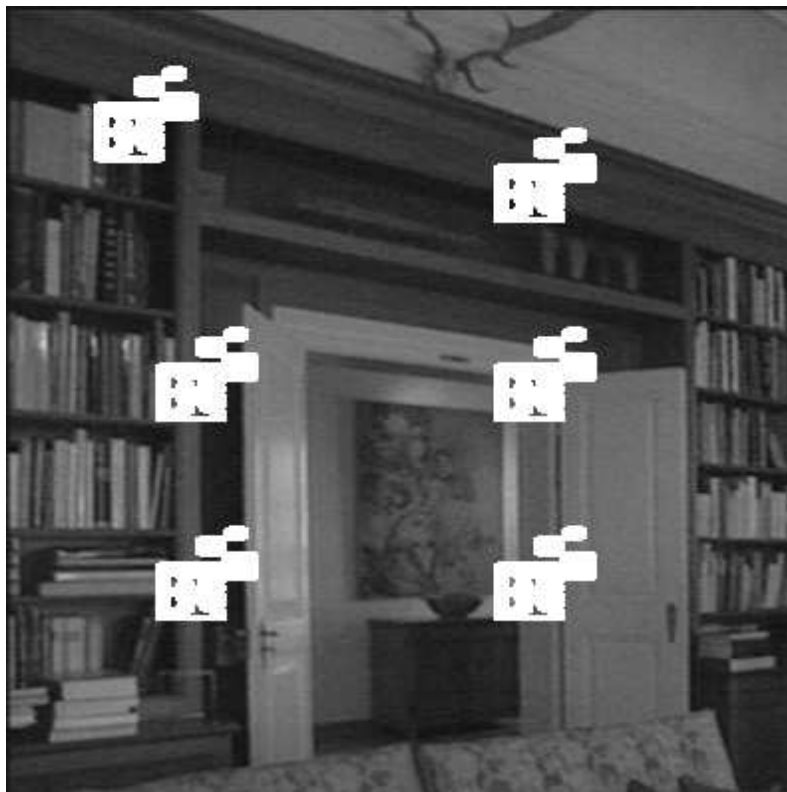


Figure 7.4: Five logos are placed in the surrounding of the original logo in the top left corner for evaluation purposes.

ment is only noticeable if the motion of the sequence is moderate and the displacement between two frames is within the volume described by the support area. Contents which are occluded by the logo in the actual frame appear in past and future frames and can be exploited for the estimation of the logo area. The improved quality is confirmed subjectively in Fig. 7.6 for frame number 61 of the sequence *DSF*. The binary pixel accurate logo mask is detected by [38] from Fig. 7.6 (a). The logo mask finally used as shown in Fig. 7.6 (b) is obtained after dilatation in order to account for artifacts like noise [57]. The external size of the logo *DSF* measures 20×89 pixels which leads to a partitioning of the logo into five blocks of approximately 20×18 pixels for separate processing. The 2-D result in Fig. 7.6 (c) is compared to the 3-D result in Fig. 7.6 (d) for the logo to be replaced in Fig. 7.6 (a). For detailed areas as the tribune, the 3-D result for the replaced logo appears sharper than the 2-D result.

If no motion is present, the 3-D approach can not benefit from any additional information and the objective result can not be improved. However, in [38] a flicker effect was reported for logo removal based on 2-D extrapolation, because the extrapolation algorithm selects slightly different basis functions for the parametric model for each frame which the human eye is sensitive to. This effect occurs only for non-moving sequences. It

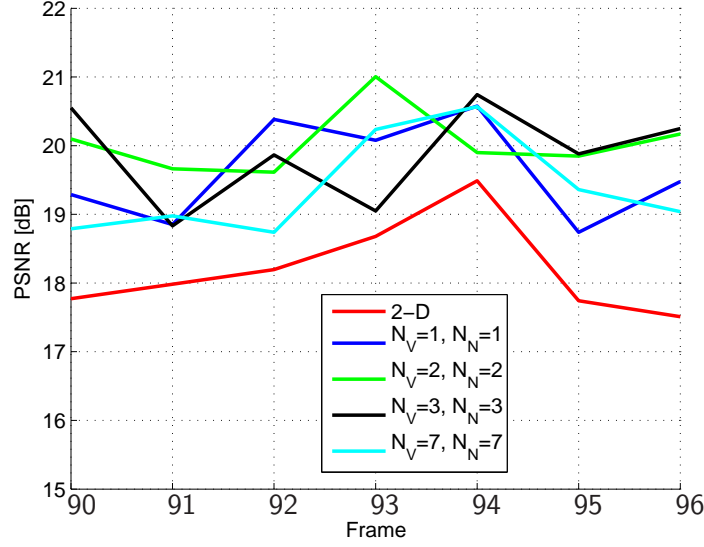


Figure 7.5: PSNR results for a scene with moderate motion of the sequence *RTL* comparing 2-D to 3-D extrapolation. The decay parameter $\hat{\rho} = 0.8$, the 2-D DFT size equals 64×64 and the 3-D DFT size $64 \times 64 \times 16$. The performance of 3-D extrapolation is plotted with respect to the number of involved N_V previous and N_N following frames for $N_E = 1$.

can be circumvented by collecting all frames with the same content in a volume, run the 3-D extrapolation algorithm and replace the logo in all considered frames at once [36]. Thus, the number of frames N_E with the logo replaced at once bordered by the purple line in Fig. 7.2 equals the number of frames in the volume. In this case, the decay of the 3-D weighting function based on the 3-D isotropic model along the time axis is undesired. Instead, a decay only in spatial direction is incorporated for the 3-D weighting function by applying the 2-D model to each time index separately. Hence, the loci of constant amplitudes are cylinders in contrast to spheres for the 3-D isotropic model.

If strong motion occurs such that the content within the support area changes significantly from frame to frame, the support area in temporal direction and the number of frames with logos replaced at once have to be chosen carefully for the 3-D approach. The performance of 3-D extrapolation decreases and might even be inferior to the performance of 2-D extrapolation as investigations in [14] show. Especially if image contents come into reach of the support area in past and future frames which do not belong to the area to be replaced, artifacts occur. The support area in temporal direction and the number of frames with logos replaced at once should then be chosen to e.g. $N_V = N_N = 1$ and $N_E = 2$. Another option is to apply the 2-D approach which also yields satisfying results.

The described problem arises even stronger for scene cuts. For the 3-D extrapolation of the last frame before the cut, frames of the new scene with a completely different content are taken into account. The same happens for the first frame of the new scene if frames



Figure 7.6: Comparison of the visual quality for replaced logos. (a) Original frame number 61 of sequence *DSF* with logo to be replaced. (b) Pixel accurate detected and dilated logo mask used for logo removal. (c) 2-D extrapolation. (d) 3-D extrapolation with $N_V = N_N = 3$.

of the past scene are included in the considered volume. Fig. 7.7 depicts visual results for logo removal for the last frame before the cut. The original plus five logos inserted into frame 77 of the sequence *BR* as illustrated in Fig. 7.4 are removed and their locations are emphasized by boxes. On the left hand side in Fig. 7.7, the extrapolated result without consideration of the scene cut is shown. The volume considered for extrapolation takes also the future frames into account. Strong patterns surrounding the logos in future frames lead to severe artifacts. Unlike the result on the right hand side obtained with consideration of the scene cut, because only previous frames are included into the extrapolation.

In summary, the 3-D extrapolation can be applied in general for logo removal. However, the support area in temporal direction has to be chosen carefully. In case of scene cuts, severe artifacts might occur. A possibility is to split the entire sequence in smaller scenes with constant motion and content and process the scenes separately. Algorithms which allow for analyzing and splitting a sequence into scenes automatically can be found

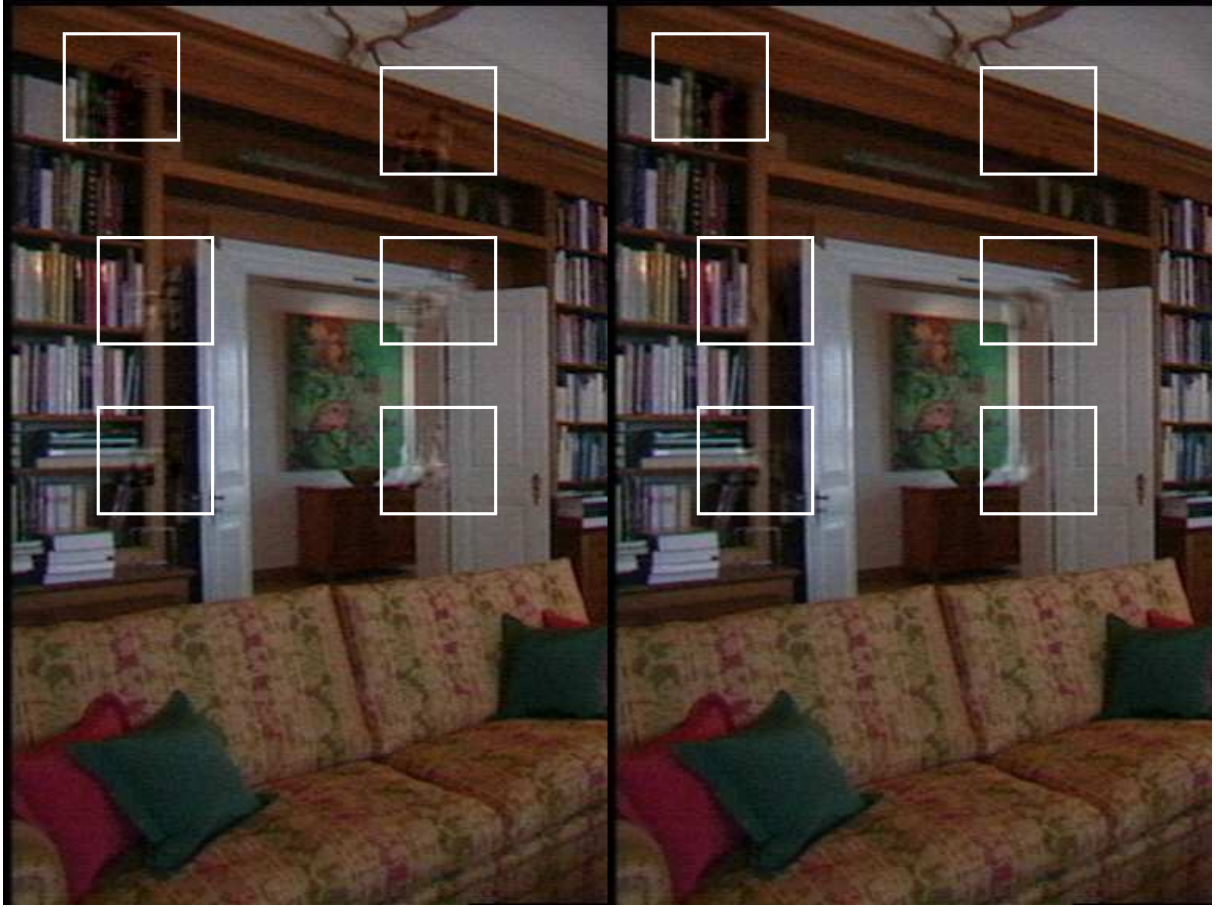
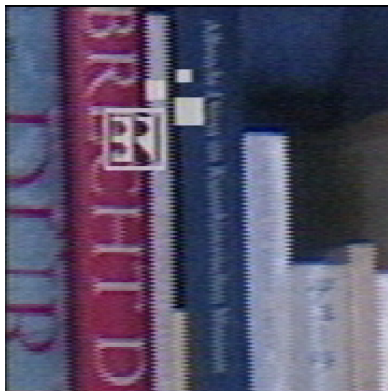


Figure 7.7: Impact of a scene cut in the following frame on the visual quality for 3-D extrapolation for frame 77 of sequence *BR*. Left: without scene cut detection. Right: with scene cut detection.

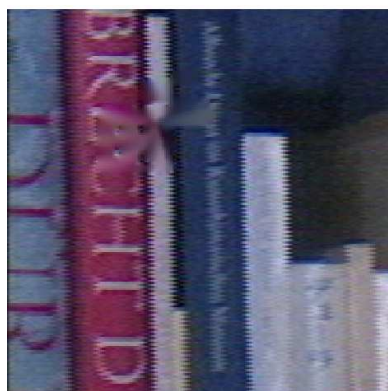
in [14]. For a scene with no motion, the visual impression can be improved using the 3-D method instead of the 2-D method, because the flicker effect is avoided. Therefore, the number of frames of the processed volume and of the volume with the logo replaced at once equals the length of the scene. Suitable parameters for scenes with moderate and strong motion are summarized in Table 7.1. If the motion is very fast and the displacement is not within the spatial dimension of the considered volume, the 2-D extrapolation should be applied.

motion	moderate	strong
$N_N = N_V$	3	1
N_E	4	2

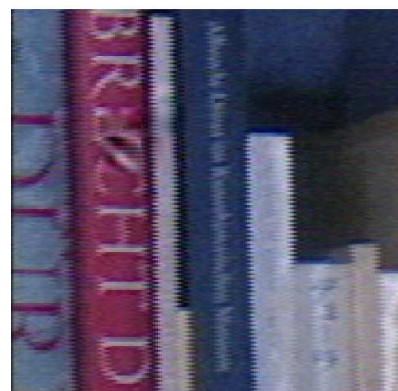
Table 7.1: Suitable parameters for scenes with moderate and strong motion.



(a) #90



(b) Inpainted [64]



(c) 2-D extrapolation



(d) #153



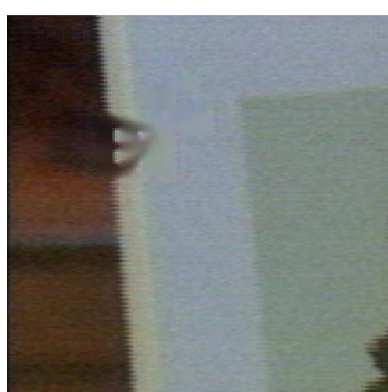
(e) Inpainted [64]



(f) 2-D extrapolation



(g) #272



(h) Inpainted [64]



(i) 2-D extrapolation

Figure 7.8: The original frame with the logo mask is shown in the left column. In the middle column, the results using pixel based removal according to [64] and in the right column the results using 2-D frequency selective extrapolation are depicted.

7.1.3 Comparison to Related Techniques

In the following, visual results are presented comparing the spatial removal technique of [64] to 2-D frequency selective extrapolation. The spatial domain method described in [64] computes each pixel as average of the eight nearest available neighbors. As available neighbors both, original samples and already inpainted logo pixels are considered as valid. The processing order moves in a circular way from the outside inwards.

In Fig. 7.8, visual results are shown for the removal of the TV logo *BR*. The original frames of the sequence *BR* with logo are shown in the left row. In the column in the middle, parts of three frames with the logo removed using the pixel based interpolation method [64] are depicted. The lowpass filtering leads to blurring and the processing order to error propagation. In the right column the corresponding results achieved using 2-D extrapolation are shown. The external size of the logo mask measures 49×53 pixels and the logo is therefore partitioned into 9 blocks of approximately 16×17 pixel for separate processing. The visual impression is clearly improved.

7.2 Prediction

The ability of 3-D frequency selective extrapolation to estimate unknown pixels is now applied to prediction in video coding. For the applications concealment and TV logo removal addressed so far, the finally achieved visual impression of the extrapolated area is the most important criterion. On the other hand, prediction for coding purposes aims at reducing the data rate of the prediction error as much as possible irrespective of the visual impression of the predicted signal. The application of frequency selective extrapolation to prediction is described in Section 7.2.1.

The 3-D extrapolation based prediction is integrated as coding mode into the H.264/AVC reference software version JM9.3 [54]. The performance in terms of coding efficiency is compared to *Intra* and *Inter* prediction of the coder H.264/AVC by rate distortion (RD) investigations. The coder decides RD optimized on MB level if a MB is Intra coded or Inter coded. Intra prediction takes place in the spatial domain by predicting from adjacent samples of previously coded blocks of the same frame that are located to the left and/or above the block to be predicted. Four prediction modes are available for blocks of size 16×16 pixels and nine modes for blocks of size 4×4 pixels. Inter prediction is based on motion compensation which is supported for blocks of size 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , and 4×4 pixels for five reference frames at quarter pixel accuracy. RD investigations comparing the H.264/AVC coding modes to the 3-D extrapolation based coding mode are presented in Section 7.2.2.

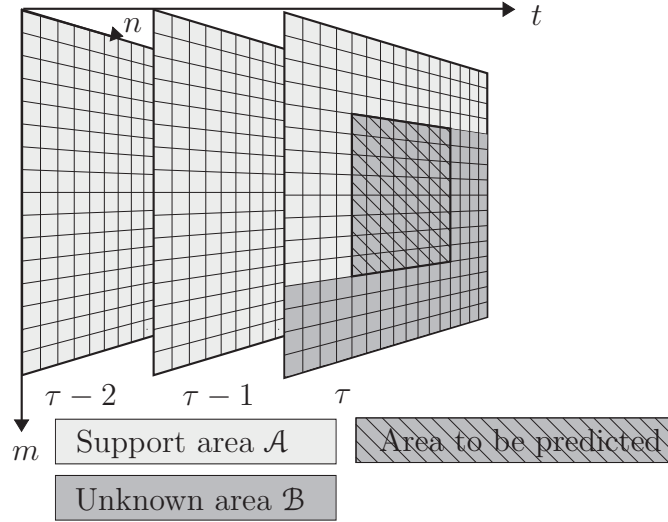


Figure 7.9: Image areas used for 3-D extrapolation in prediction consisting of the area to be estimated and its known surrounding.

7.2.1 Prediction by Frequency Selective Extrapolation

The block to be coded is predicted from its causal spatial as well as temporal surrounding pixels by 3-D frequency selective extrapolation. Fig. 7.9 shows schematically the volume considered for the extrapolation task. The causal pixels used for prediction are marked light gray. The unknown area is marked dark gray, whereas only the hatched part in frame τ forms the prediction signal. The number of surrounding MBs which can be used for prediction depends on the location of the MB within a frame. Here, the common case where the MB is not located at the border of the frame is depicted.

The processing order of the MBs to be predicted takes place in raster scan order as illustrated in Fig. 7.10. Furthermore, the MBs which can be used for prediction in the current frame is shown. For the first MB of a frame, no spatial neighboring MBs are available. Therefore, the MB is copied from the previous frame for prediction. For the MBs in the top row of a frame, only the MB to the left is available for prediction in the current frame. The number of available MBs to predict from is also limited for MBs located at the frame boundary as illustrated in Fig. 7.10 by the colored MBs. The majority of the MBs can be predicted from four surrounding MBs which is the case for the MBs marked dark gray.

Frequency selective extrapolation applied to prediction relies also on a weighting function $w[m, n, t]$ which is based on the 3-D isotropic model (4.27). The effect of the shape of the unknown area on the weighting function in 2-D is illustrated in Fig. 7.11. The parameters for 3-D frequency selective extrapolation applied to prediction are chosen according to the investigations in [32, 48]. The computational cost is limited compared to concealment by reducing the maximum number of iterations to 100 and $\Delta_{\min} = 0.1$. The

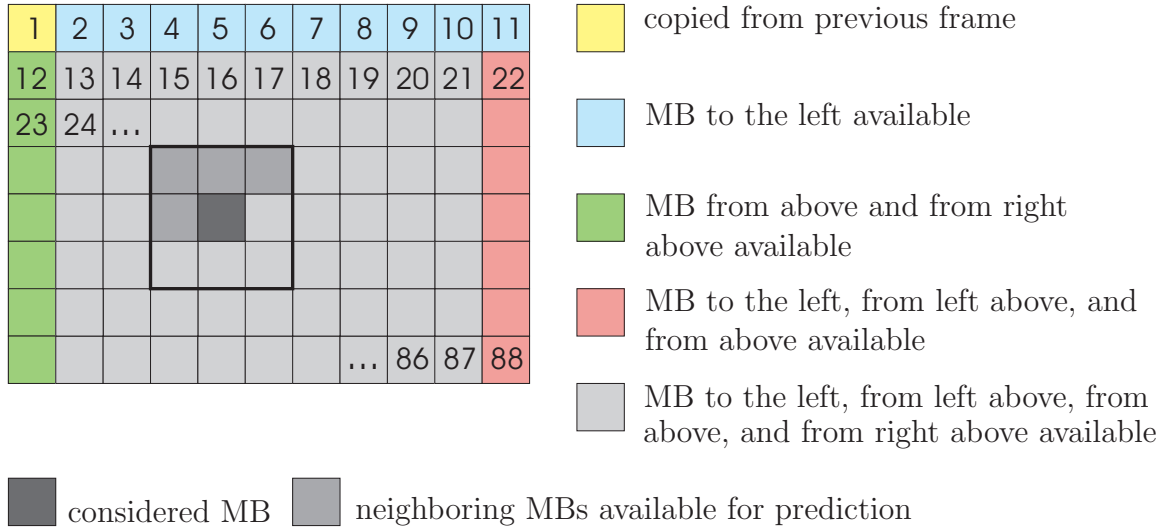


Figure 7.10: The processing order and the spatial neighboring MBs available for prediction depending on the location of the MB to be predicted are shown.

decay parameter of the 3-D isotropic weighting function (4.27) $\hat{\rho}$ equals 0.8. The support area in spatial direction consists of 16 pixels and in temporal direction of two previous frames. The sequences are coded with the Baseline profile which does not support B frames. Therefore, no future frames can be taken into account. The 3-D DFT size is adjusted to $64 \times 64 \times 16$ obtained by zero padding.

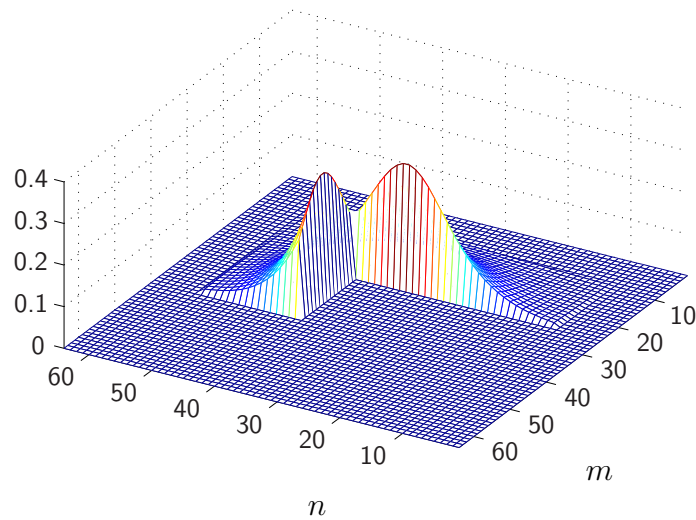


Figure 7.11: The effect of the shape of the unknown area in prediction on the weighting function in 2-D is shown.

7.2.2 Comparison to Related Techniques

The 3-D extrapolation is integrated into the reference software of the H.264/AVC coder version JM9.3 [54] as coding mode in order to evaluate the coding efficiency compared to the prediction modes of H.264/AVC. Twenty three frames of the test sequences *Crew* and *Flowergarden* are coded with the Baseline profile in the corresponding coding mode. The investigations are done for the 16×16 prediction modes, i.e., the MBs are not further partitioned. The following investigations are done with respect to their RD performance:

- The 3-D extrapolation based prediction is compared to the Intra modes of H.264/AVC.
- The 3-D extrapolation based prediction is compared to the Inter modes of H.264/AVC.
- The 3-D extrapolation based prediction mode is integrated into a RD optimized coder.

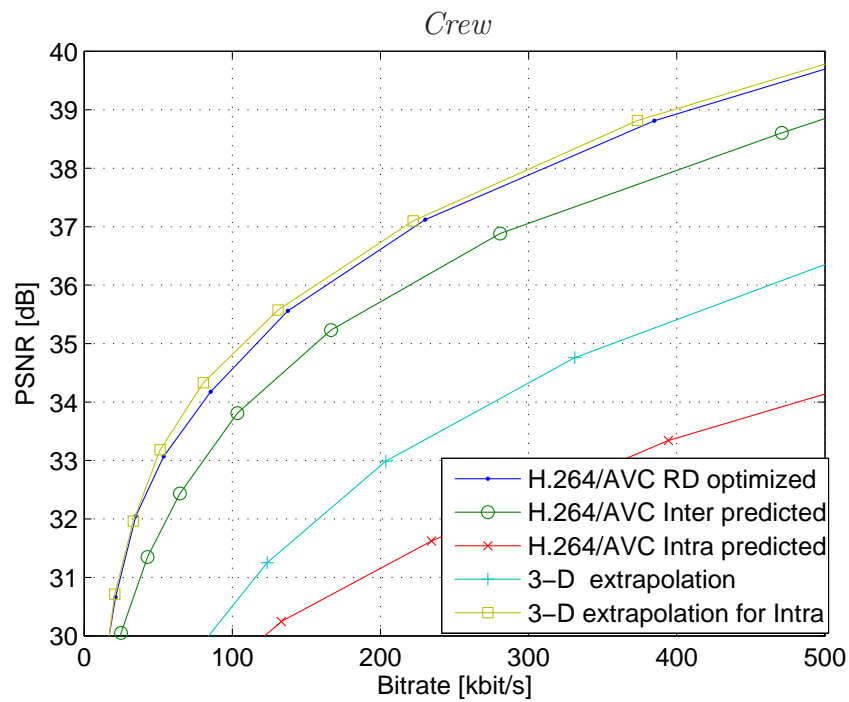
Fig. 7.12 (a) shows the results for the sequence *Crew* and Fig. 7.12 (b) for *Flowergarden*. First, the introduced coding mode 3-D extrapolation is compared to Intra prediction of the H.264/AVC coder. 3-D extrapolation based prediction performs significantly better than the H.264/AVC based Intra prediction due to the extrapolation within the spatio-temporal volume compared to the purely spatial method.

As can be seen, the H.264/AVC Inter prediction performs remarkably better than 3-D extrapolation, although no side information such as motion vectors has to be transmitted for the latter. The reason is that 3-D frequency selective extrapolation based prediction approximates the signal in the frequency domain. Small phase errors produce large prediction errors, even if they might not be visible. The prediction signals for *Crew* and *Flowergarden* are depicted in Fig. 7.13 and at least the result for *Flowergarden* looks visually very pleasing. In contrast, motion compensation provides a prediction signal at sub-pixel accuracy in the spatio-temporal domain. The signal to be predicted which is known at the encoder side is involved in the prediction process. Therefore, side information in terms of motion vectors has to be transmitted which leads, however, to a better overall coding efficiency.

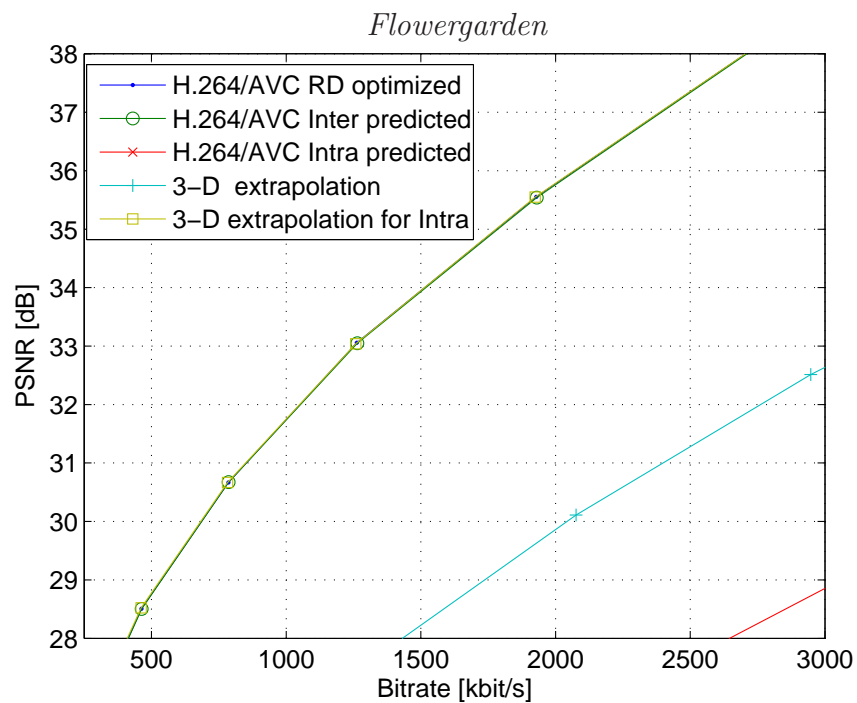
Since 3-D extrapolation performs significantly better than Intra coding provided by the H.264/AVC standard, we replaced the conventional Intra prediction by the 3-D extrapolation coding mode and ran the codec with RD optimized coding. The rate-distortion optimized prediction decides the coding mode on MB level for P frames. Intra coding might be chosen if the temporal information for prediction is not reliable. There, an adequate block can not be found in the previous frame as for example for occlusions. The results for *Crew* and *Flowergarden* are also presented in Fig. 7.12. Although only 2% of the MBs are Intra coded in case of the *Crew* sequence, savings of still 10 to 25 kbit/s data

rate can be achieved. The sequence *Flowergarden* can be predicted by the mode Inter due to the translational motion. Thus, the result can not be improved by 3-D extrapolation because only 0.02% of the MBs are chosen to be Intra.

In summary, the data rate can be reduced for sequences where temporal information is not sufficiently reliable for prediction by integrating the coding mode 3-D extrapolation into a RD optimized coder.



(a)



(b)

Figure 7.12: Prediction results for 23 frames of the test sequences in CIF format coded by H.264/AVC. (a) Sequence *Crew*. (b) Sequence *Flowergarden*.



(a)



(b)

Figure 7.13: Frame in CIF format predicted by 3-D extrapolation. (a) 9th frame of sequence *Flowergarden* predicted from two previous and two subsequent frames. (b) 316th frame of sequence *Crew* predicted from two previous frames.

8 Conclusions and Outlook

In this thesis, a generic method for estimating unknown signal samples by extrapolation from known signal data is derived. The presented concept is referred to as *selective extrapolation*. This very general approach allows for inserting problem-specific basis functions which makes the approach applicable to numerous extrapolation tasks.

Since image and video signals can be interpreted as a linear combination of multi-dimensional frequencies, DFT like basis functions are especially suited which leads to the *frequency selective extrapolation* approach. With the new concept of 3-D extrapolation, a video signal can be interpreted as a 3-D volume. Hence, *spatial and temporal* correlations of the video signal can be *simultaneously* used for extrapolation purposes.

The error criterion minimized during the extrapolation procedure allows for an application-specific adaptation due to an involved weighting function, i.e., subareas which are more important for the extrapolation can be included with more weight. For the applications considered in this thesis, where the missing area is connected, centrally located within the considered block, and surrounded by known data, it is reasonable to assume that the closer a known pixels is to the unknown area, the more important it is for the extrapolation. Therefore, the chosen weighting function is based on an isotropic model decaying radial symmetrically with increasing distance to the unknown area for both, image and video signals.

The derived frequency selective extrapolation approach was applied to several extrapolation tasks typically occurring in image and video communications. First, error concealment of lost block coded data is addressed. On the one hand, 2-D frequency selective extrapolation is applied to spatial concealment. On the other hand, spatio-temporal concealment is enabled by the 3-D approach. Both methods are successfully integrated into the H.264/AVC coder as concealment feature. In contrast to conventional temporal concealment methods, 3-D frequency selective extrapolation can be easily applied to concealment of Intra coded macroblocks, because it does not rely on motion vectors.

Other coders such as the JPEG2000 coder apply wavelet based coding principles. In this case, a data loss has completely different effects on the visual quality of the reconstructed image compared to block based coders. The investigations have shown that especially a loss in the lowpass band causes severe artifacts. Therefore, robustness tools provided by the coder are applied in order to assure that the loss of wavelet coefficients in the lowpass band is even further limited in case of an error compared to the other subbands. For concealing lowpass losses, the 2-D frequency selective extrapolation tech-

nique is integrated into the JPEG2000 decoder for spatial concealment of the lost wavelet coefficients.

Two further applications are addressed from the point of view of extrapolation: on the one hand TV logo removal and on the other hand prediction in hybrid video coding. For logo removal, the logo is located at the same spatial position for each frame of the sequence. Hence, simple copying from previous frames is not possible. Using 2-D extrapolation for each frame leads to a flicker effect for sequences with non-moving contents, which can be avoided by extrapolating the entire 3-D volume reaching over the entire scene at once. However, in case of applying 3-D extrapolation, the visual quality of the replaced areas depends significantly on the motion of the sequence. Scene cuts may cause severe artifacts. Therefore, the best results are achieved if the sequence is split into scenes with motion of similar type and contents which are then processed separately.

In prediction in hybrid video coding, the block to be predicted is estimated by 3-D extrapolation from already coded spatio-temporal pixels. The approach is integrated into a H.264/AVC(Advanced Video Coding) coder as prediction mode using rate distortion optimized coding principles. In cases where temporal information is not reliable as for occlusions or no purely translational motion, advantage can be taken from the 3-D extrapolation in terms of coding efficiency. Although commonly only a few macroblocks are not predicted by motion compensation, a gain in coding efficiency can be achieved by the new prediction mode.

In summary, frequency selective extrapolation provides a very generic approach for extrapolation tasks in image and video communications. The main advantage is the ability to inherently adapt to the local signal characteristics due to the frequency selectivity. In case of the 2-D approach, smooth areas, edges, patterns as well as noise-like areas can be extrapolated. Since the novel 3-D approach considers the video signal in a volume, additionally motion and variations in luminance occurring from frame to frame can be inherently compensated. These powerful extrapolation properties can be successfully exploited especially for applications where the visual impression of the extrapolated areas is significant as in concealment and logo removal.

The derivation of this generic technique and the results obtained in this thesis open up a wide range of future research directions. Conclusions drawn from the theoretical investigations illustrated by the geometrical interpretations can be the basis for further algorithmical developments. In each iteration, the residual vector in the approximation subspace is projected onto the selected basis function in the subspace of the support area. The basis functions are not mutually orthogonal in this subspace, i.e., the selected basis function includes also components from other basis functions. Hence, the expansion coefficient obtained by the projection results not only from the selected basis function, but also from other basis functions. This lack of orthogonality should be considered for the calculation of the expansion coefficient. It could be incorporated e.g. by attenuating

the calculated expansion coefficient assuming thus that the true value is lower.

The applications considered so far consist of connected known as well as unknown areas. Future research directions can involve other types of unknown areas and support areas, as e.g. in deinterlacing. There, a recorded sequence in interlaced format is to be converted for a progressive display. The two fields of a frame can be interpreted as a quincunx subsampling of this frame in column-time domain which leads to spectral replica in the frequency domain. The consequences of the replica have to be taken into account for the selection of a basis function and frequency, respectively. The area of frequencies to be selected from has to be limited in order to avoid aliasing.

Frequency selective extrapolation can be straightforwardly extended to more dimensions and used in corresponding applications. For instance, the coding and transmission of dynamic lightfields processes four or five dimensions for an efficient handling. The set of cameras recording such multi-view sequences provide the fourth and possibly fifth dimension. In case of errors while transmitting coded dynamic lightfield data in error-prone environments, the effects of losses should be concealed at the decoder side. Additional to spatial and temporal data, data from other cameras might be used for concealment applying a 4-D frequency selective extrapolation.

In this thesis, frequency selective extrapolation was applied to signals in image and video communications. The excellent extrapolation ability could be also used for other signals with different statistics which are out of the scope of this thesis. For instance, medical image signals exhibit a large amount of noise. For extrapolation tasks in medical imaging, it is very important to be able to imitate the noise in order to lead to a natural appearance of the images. There, the ability of extrapolating noiselike areas due to the frequency selectivity of the algorithm could be exploited. For instance, flat panel X-ray detectors provide images with defective areas amongst others due to manufacturing problems. Frequency selective extrapolation can be applied in order to restore these defects by extending the surrounding signal into the defective area. In [4], spectral deconvolution is applied which corresponds to frequency selective extrapolation using a binary weighting function showing already the great potential for medical applications. Benefit could be therefore derived from applying the 2-D frequency selective extrapolation using an adapted weighting function for static radiographs. Defects in sequences acquired by cine-angiography or fluoroscopy could be processed by 3-D extrapolation.

A Notations

A.1 Conventions

In general, the following conventions are used: Lower case boldface (**a**) denotes signal-domain vectors and matrices. Upper case boldface (**A**) denotes DFT-domain vectors and matrices.

A.2 Abbreviations

AVC	Advanced video coding
BMA	Boundary matching algorithm
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DMVE	Decoder motion vector estimation
EBMA	Extended boundary matching algorithm
FEC	Forward error correction
FFT	Fast Fourier transform
FMO	Flexible macro block ordering
IDFT	Inverse discrete Fourier transform
IP	Internet protocol
ITU	International telecommunication union
JPEG	Joint picture experts group
MB	Macro block
MPEG	Motion picture experts group
MTU	Maximum transfer unit
NAL	Network abstraction layer
OBMC	Overlapped block motion compensation
POCS	Projection onto convex sets
PSNR	Peak signal to noise ratio
VCL	Video coding layer
VQEG	Video quality experts group
TR	Temporal block replacement

A.3 Mathematical Symbols

Operators

$(\cdot)^{-1}$	inverse (\cdot)
$(\cdot)^*$	conjugate of (\cdot)
$(\cdot)^T$	transpose of (\cdot)
$(\cdot)^H$	conjugate transpose of (\cdot)
$ \cdot $	magnitude
$\text{diag}\cdot$	diagonal matrix formed by the listed entries
$\text{DFT}\{\cdot\}$	DFT transform of $\{\cdot\}$
$\text{IDFT}\{\cdot\}$	IDFT transform of $\{\cdot\}$

List of Symbols

\mathcal{L}	Entire area considered during extrapolation
\mathcal{A}	Support area
\mathcal{B}	Area to be estimated
\mathcal{K}	Set of indices of used basis functions
\mathcal{M}	Set of indices belonging to real-valued basis functions
M	Number of rows of entire area
N	Number of columns of entire area
T	Number of frames of entire area
N_V	Number of previous frames
N_N	Number of following frames
N_E	Number of frames with estimated area replaced at once
m	Row index
n	Column index
t	Time index
(k, l)	Index of 2-D basis function
(u, v)	Index of 2-D basis function
(k, l, p)	Index of 3-D basis function
(u, v, q)	Index of 3-D basis function
$f[m, n]$	Intensities of 2-D original signal
$f[m, n, t]$	Intensities of 3-D original signal
$g[m, n]$	2-D parametric model
$g[m, n, t]$	3-D parametric model
$r[m, n]$	2-D residual error
$r[m, n, t]$	3-D residual error
$r_w[m, n]$	2-D weighted residual error

$r_w[m, n, t]$	3-D weighted residual error
$b[m, n]$	2-D window function
$b[m, n, t]$	3-D window function
$w[m, n]$	2-D weighting function
$w[m, n, t]$	3-D weighting function
$x[m, n]$	Arbitrary 2-D function
$x[m, n, t]$	Arbitrary 3-D function
$\varphi_{k,l}[m, n]$	2-D basis function
$\varphi_{k,l,p}[m, n, t]$	3-D basis function
$c_{k,l}$	Expansion coefficient of 2-D basis function
$c_{k,l,p}$	Expansion coefficient of 3-D basis function
Δc	Update of expansion coefficient
$\hat{\rho}$	Decay parameter of isotropic model
$\Delta E_{\mathcal{A}}$	Residual error energy
ν	Counter of iterations
$W[k, l]$	2-D DFT of weighting function at (k, l)
$W[k, l, p]$	3-D DFT of weighting function at der Stelle (k, l, p)
$R_w[k, l]$	2-D DFT of $r_w[m, n]$ at (k, l)
$R_w[k, l, p]$	3-D DFT of $r_w[m, n, t]$ at (k, l, p)
$X[k, l]$	2-D DFT of $x[m, n]$ at (k, l)
$X[k, l, p]$	3-D DFT of $x[m, n, t]$ at (k, l, p)
$\Delta \mathbf{c}$	Vector of expansion coefficient updates
\mathbf{R}_w	Vector of weighted residual error in frequency domain
\mathbf{W}	Matrix of weighting function in frequency domain
\mathbf{W}^+	Pseudoinverse of \mathbf{W}
$\mathbf{f}_{\mathcal{L}}$	Vectorial interpretation of $f[m, n]$ corresponding to area \mathcal{L}
$\mathbf{f}_{\mathcal{A}}$	Vectorial interpretation of $f[m, n]$ corresponding to area \mathcal{A}
$\mathbf{g}_{\mathcal{L}}$	Vectorial interpretation of $g[m, n]$ corresponding to area \mathcal{L}
$\mathbf{g}_{\mathcal{A}}$	Vectorial interpretation of $g[m, n]$ corresponding to area \mathcal{A}
$\mathbf{x}_{\mathcal{L}}$	Vectorial interpretation of $x[m, n]$ corresponding to area \mathcal{L}
$\mathbf{x}_{\mathcal{A}}$	Vectorial interpretation of $x[m, n]$ corresponding to area \mathcal{A}
$\mathbf{r}_{\mathcal{A}}$	Vectorial interpretation of $r[m, n]$ corresponding to area \mathcal{A}
κ	counter
λ	counter

B Titel, Inhaltsverzeichnis, Einleitung und Zusammenfassung

The following german translations of the title (Section B.1), the table of contents (Section B.2), the introduction (Section B.3), and the summary (Section B.4) are a mandatory requirement for the dissertation at the Faculty of Engineering of the University of Erlangen-Nuremberg.

B.1 Titel

Selektive Signalextrapolation und ihre Anwendung in der Bild- und Videokommunikation

B.2 Inhaltsverzeichnis

1	Einleitung	1
2	Signalextrapolation	5
2.1	Anwendungen für Extrapolation in Bild- und Videokommunikation	5
2.2	Örtliche Signalextrapolation	7
2.2.1	Problemstellung der Extrapolation	7
2.2.2	Extrapolation durch spektrale Schätzung	8
2.3	Verwandte Techniken angewandt in der Bildcodierung	11
2.4	3D Signalverarbeitung	11
3	Örtliche selektive Extrapolation	13
3.1	Örtliche selektive Extrapolation für 2D Signale	13
3.1.1	Prinzip der selektiven Extrapolation	14
3.1.2	Erneuerung des selektierten Koeffizienten	15
3.1.3	Selektion der Basisfunktion	17
3.2	Geometrische Interpretation der selektiven Extrapolation	18
3.3	Geeignete Basisfunktionen	25
3.4	Frequenzselektive Extrapolation von Bildsignalen mittels DFT Basisfunktionen	26

3.4.1	Lösung im Ortsbereich	27
3.4.2	Lösung im Frequenzbereich	29
3.4.3	Zusammenfassung der frequenzselektiven Extrapolation	31
3.5	2D Gewichtungsfunktion	36
3.5.1	Extrapolationsergebnis für die rechteckige binäre Gewichtungsfunktion	39
3.5.2	Isotrope Gewichtungsfunktion	44
3.5.3	Extrapolationsergebnis für die isotrope Gewichtungsfunktion	45
3.5.4	Diskussion der Ergebnisse	50
3.6	Frequenzselektive Extrapolation interpretiert als spektrale Entfaltung	54
4	Örtlich-zeitliche selektive Extrapolation	61
4.1	Örtlich-zeitliche selektive Extrapolation für 3D Signale	61
4.2	Frequenzselektive Extrapolation von Videosignalen mittels DFT Basisfunktionen	64
4.2.1	Lösung im Orts-Zeitbereich	66
4.2.2	Lösung im Frequenzbereich	67
4.2.3	3D Isotrope Gewichtungsfunktion	68
4.2.4	Diskussion der Ergebnisse	76
4.3	Alternativer Ansatz: Örtlich-zeitliche Extrapolation mittels Best-Approximation	82
4.3.1	Grundprinzip der Best-Approximation	82
4.3.2	Geometrische Interpretation der Extrapolation durch Best-Approximation	84
4.3.3	2D Extrapolation durch Best-Approximation mittels DFT Basisfunktionen	88
4.3.4	3D Extrapolation durch Best-Approximation mittels DFT Basisfunktionen	91
4.4	Frequenzselektive Extrapolation im Vergleich zu Best-Approximation	93
4.4.1	Extrapolationsergebnis mittels 2D Best-Approximation	93
4.4.2	Extrapolationsergebnis mittels 3D Best-Approximation	96
4.4.3	Diskussion der Ergebnisse	97
5	Fehlerverschleierung verlorener blockcodierter Daten	101
5.1	Bekannte Methoden der Fehlerverschleierung verlorener blockcodierter Daten	102
5.1.1	Örtliche Verschleierung von Blockverlusten	102
5.1.2	Örtlich-zeitliche Verschleierung von Blockverlusten	103
5.2	Verschleierung mittels frequenzselektiver Extrapolation	105
5.2.1	Örtliche Verschleierung mittels 2D frequenzselektiver Extrapolation	105

5.2.2	Örtlich-zeitliche Verschleierung mittels 3D frequenzselektiver Extrapolation	109
5.3	Vergleich mit bekannten Methoden	116
5.3.1	Ergebnisse für Örtliche Verschleierung	116
5.3.2	Ergebnisse für Örtlich-zeitliche Verschleierung	119
5.4	Integration der frequenzselektiven Extrapolation in den H.264/AVC Decoder	124
5.4.1	Fehlerrobustheit des H.264/AVC	124
5.4.2	Fehlerverschleierung in der Referenzsoftware	125
5.4.3	Realisierung der Verschleierung mittels frequenzselektiver Extrapolation	126
5.4.4	Simulationen und Ergebnisse	128
5.4.5	Zusammenfassung	141
6	Örtliche Verschleierung verlorener Wavelet-codierter Daten	143
6.1	Grundlagen von JPEG2000	144
6.2	Werkzeuge zur Fehlerkontrolle in JPEG2000	145
6.2.1	Auswirkungen von Übertragungsfehlern	145
6.2.2	Fehlerrobustheit auf Entropiecodierungsebene	146
6.2.3	Fehlerrobustheit auf Paketebene	146
6.2.4	Verschleierung durch Einfügen leerer Pakete	146
6.3	Verschleierung mittels frequenzselektiver Extrapolation	147
6.4	Simulationen und Ergebnisse	148
6.4.1	Verschleierung von Verlusten im Tiefpassband	149
6.4.2	Verschleierung von Verlusten im Hochpassband	155
6.4.3	Verschleierung von Verlusten in den unteren Layer	155
6.4.4	Realisierung eines paketfehlerrobusten JPEG2000 Decoders	156
6.5	Zusammenfassung	157
7	Weitere Anwendungen	159
7.1	TV Logoentfernung	159
7.1.1	Logoentfernung mittels frequenzselektiver Extrapolation	160
7.1.2	Simulationen und Ergebnisse	161
7.1.3	Vergleich mit bekannten Techniken	168
7.2	Prädiktion	168
7.2.1	Prädiktion mittels frequenzselektiver Extrapolation	169
7.2.2	Vergleich mit bekannten Techniken	171
8	Schlussfolgerungen und Ausblick	175

A	Notationen	179
A.1	Konventionen	179
A.2	Abkürzungen	179
A.3	Mathematischen Symbole	180
B	Titel, Inhaltsverzeichnis, Einleitung und Zusammenfassung	183
B.1	Titel	183
B.2	Inhaltsverzeichnis	183
B.3	Einleitung	186
B.4	Zusammenfassung und Schlussfolgerungen	189
	Literaturverzeichnis	193

B.3 Einleitung

In vielen Anwendungen der Signalverarbeitung besteht das Problem, unbekannte Signalwerte aus bekannten Daten zu schätzen. Diese Aufgabenstellung kann als die Extrapolation bekannter Signallbereiche in unbekannte Bereiche interpretiert werden. In der medizinischen Bildverarbeitung tritt diese Fragestellung beispielsweise bei der Defekt-Pixel-Interpolation von Röntgen-Flachdetektoren auf. Diese Detektoren erlauben die sofortige Anzeige der aufgenommenen Bilder, können jedoch auf Grund von herstellungsbedingten Schwierigkeiten Regionen inaktiver Elemente enthalten. Diese inaktiven Elemente verursachen in den aufgenommenen Röntgenbildern unbekannte Stellen. Die entsprechenden Defekte können in den Bildern durch das aus benachbarten Bildpunkten extrapolierte Signal ersetzt werden [4], da die Positionen der inaktiven Elemente durch Kalibrierungsmessungen vorab lokalisiert werden können.

Die Modellierung des menschlichen Gehörs einschließlich einer hochauflösenden Spektralanalyse [50] stellt ein Beispiel für eine Anwendung aus der Audiosignalverarbeitung dar. Da nur Signalblöcke von kurzer Dauer zur Verfügung stehen, basiert die verwendete Methode zur hochauflösende Kurzzeit-Spektralanalyse auf einer Extrapolation. Der Einfluss der zeitbegrenzten Beobachtung des Audiosignals wird im Spektralbereich entfernt, was einer Extrapolation des Audiosignals über den beobachteten Zeitraum hinweg entspricht.

Ein ähnliches Problem tritt in dem völlig andersartigen Gebiet der Astrophysik auf. Zur Apertur-Synthese werden die Einschränkungen der Teleskope auf die aufgenommenen Messwerte eliminiert [22]. Die aufgenommenen, so genannten “verschmutzten Aufnahmen” (“Dirty Map”) werden von den sogenannten “verschmierten Keulen” (“Dirty Beams”) der aufnehmenden Teleskope “gesäubert”. Da der Ausschnitt des Himmels, der durch die Teleskope aufgenommen werden kann, begrenzt ist, entspricht die Methode einer

Extrapolation der Intensitätswerte der aufgenommenen Objekte über den beobachteten Teil des Himmels hinweg.

In der Nuklearphysik stellt die Prädiktion nuklearer Massen in instabilen Bereichen eine weitere Extrapolationsfragestellung dar [17]. Die Differenz zweier gegebener Modelle zur Prädiktion nuklearer Massen ist in Abb. 1.1 in Abhängigkeit der Anzahl der Neutronen auf der X-Achse und der Anzahl der Protonen auf der Y-Achse gegeben. Die nuklearen Massen in den instabilen Regionen können basierend auf der Extrapolation des gegebenen Muster über seine Grenzen hinweg prädiziert werden.

Das Problem der Schätzung unbekannter Signalwerte tritt auch in verschiedenen Anwendungen der Bild- und Videokommunikation auf. In der Praxis besonders wichtige Beispiele werden in dieser Arbeit aus dem Blickwinkel der Signalextrapolation interpretiert. Die Problemstellung der Verschleierung gestörter Videodaten, die durch Übertragungsfehler in der mobilen Videokommunikation verursacht wurden, kann z.B. als Extrapolation des vorhandenen umgebenden Videosignals in den Fehlbereich verstanden werden. Die Entfernung unerwünschter TV Logos kann auch im betrachteten Extrapolationszusammenhang gedeutet werden, in dem die durch das Logo verdeckten Bildpunkte aus dem umgebenden Videosignal geschätzt werden. In der hybriden Videocodierung wird zur Erhöhung der Codiereffizienz das Videosignals prädiziert. Dieser Signalverarbeitungsschritt kann auch als Extrapolation der bereits codierten Bildpunkte zur Prädiktion der darauf folgenden Bildpunkte interpretiert werden.

Es gibt eine Reihe von Methoden, die die Problemstellung der Extrapolation zweidimensionaler Signale durch spektrale Schätzung lösen. Methoden der bandbegrenzten Extrapolation [44, 45, 53, 56] verursachen bekanntermaßen Artefakte im Signalbereich und sind daher zur Signalextrapolation in größere Bereiche nicht geeignet. Methoden der spektralen Entfaltung [18, 50, 12, 4] erzielen weitaus bessere Ergebnisse. Da sie jedoch die Basisfunktionen der Diskreten Fourier-Transformation (DFT) verwenden, sind sie auf den Fourier-Bereich beschränkt. Andere für bestimmte Anwendungen geeignete Basisfunktionen, wie z.B. die der Diskreten Cosinus-Transformation (DCT) oder Polynome, können bei diesem Ansatz nicht verwendet werden.

Die grundlegenden Prinzipien dieser Arbeit werden nicht nur für Extrapolationsaufgaben angewandt, sondern auch zur Codierung benutzt. In [26] wird die auf beliebigen Basisfunktionen aufbauende sukzessive Approximation zur objektbasierten Bildcodierung eingesetzt. Zur Extrapolation von Videosignalen werden hingegen normalerweise keine spektralen Schätzmethoden verwendet. Unbekannte Abtastwerte in Videosignalen werden herkömmlicherweise durch Hybrid-Techniken unter der Verwendung von entweder örtlicher oder zeitlicher Information geschätzt, wie z.B. die Prädiktion in der hybriden Videocodierung. Am häufigsten wird die Bewegungskompensation angewandt, um zeitliche Information zur Prädiktion zu nutzen.

In dieser Arbeit wird eine generische Schätzmethode entwickelt, die als *Selektive*

Extrapolation bezeichnet wird. Im Gegensatz zu früheren Arbeiten zur Extrapolation [18, 50, 12, 4], beruht dieser allgemeine Ansatz auf beliebigen Basisfunktionen, die eine Anpassung an die spezifischen Eigenschaften der betrachteten Anwendung ermöglicht. Basierend auf dem Prinzip der sukzessiven Approximation [26] wird der den zu schätzenden Bereich umgebende Unterstützungsbereich durch eine Linearkombination *selektierter* Basisfunktionen approximiert. Die für die Approximation verwendeten Basisfunktionen sind in einem Bereich definiert, der auch das Gebiet der unbekannten Werte umschließt. Deswegen liefert die Approximation des bekannten Signalbereichs gleichzeitig eine *Extrapolation* in den *unbekannten* Bereich. Das während der Extrapolation minimierte Fehlerkriterium verwendet eine Gewichtungsfunktion, die eine weitere anwendungsspezifische Anpassung des allgemeinen Ansatzes ermöglicht. Für die in dieser Arbeit betrachteten Anwendungen wird eine geeignete Gewichtungsfunktion spezifiziert. Da das präsentierte Konzept unabhängig von den Dimensionen des betrachteten Signalraums ist, kann der zunächst für zweidimensionale Signale hergeleitete Ansatz der selektiven Extrapolation direkt auf dreidimensionale Signale erweitert werden. Dies wird durch die Interpretation eines Videosignals als 3D Volumen erreicht. So können örtliche und zeitliche Korrelationen des Videosignals gleichzeitig zur Schätzung der unbekannten Signalbereiche genutzt werden. Diese Interpretation eines Videosignals ermöglicht neue Erkenntnisse im Vergleich zu konventionellen Hybrid-Ansätzen, die entweder örtlich oder zeitlich präzisieren.

Diese Arbeit ist wie folgt gegliedert. Zunächst wird in Kapitel 2 ein Überblick über spektrale Schätzmethoden zur Signalextrapolation und für verwandte Gebiete gegeben. In Kapitel 3 wird die Herleitung des generischen Konzepts zur selektiven Extrapolation für den 2D Fall dargestellt. Zur Extrapolation von Bildsignalen sind DFT ähnliche Basisfunktionen besonders geeignet, deren Einsatz zu dem Konzept der *frequenzselektiven Extrapolation* führt. In Kapitel 4 wird dieses Konzept für 3D Signale erweitert.

Den theoretischen Herleitungen folgend, wird die frequenzselektive Extrapolation zur Lösung typischer Fragestellungen in der Bild- und Videokommunikation angewandt. In Kapitel 5 wird die Anwendung des entwickelten Ansatzes zur Fehlerverschleierung verloren gegangener blockcodierter Daten bei der Videokommunikation in fehleranfälligen Umgebungen beschrieben. Nach detaillierten Untersuchungen wird die frequenzselektive Extrapolation als Verschleierungsverfahren in die Referenzsoftware [54] des H.264/AVC (Advanced Video Coding) Coders [23] integriert.

Andere Coder wie der JPEG2000 Coder [1] arbeiten mit Wavelet-basierten Codierprinzipien. Da sich dieses Codierprinzip von den blockbasierten Ansätzen unterscheidet, hat ein Datenverlust völlig andere Auswirkungen. Nach Untersuchungen in Kombination mit den durch den Standard zur Verfügung stehenden Robustheitstechniken, wird die 2D frequenzselektive Extrapolation in Kapitel 6 in den JPEG2000 Coder als Verschleierungsverfahren integriert.

In Kapitel 7 werden zwei weitere interessante Anwendungsbeispiele für die entwickelte

frequenzselektive Extrapolationsmethode untersucht: einerseits wird die Entfernung unerwünschter TV Logos in Abschnitt 7.1 und andererseits die Prädiktion in der hybriden Videocodierung in Abschnitt 7.2 betrachtet. TV Logos befinden sich gewöhnlich während der ganzen Sequenz zeitunveränderlich an der gleichen Position. Das Konzept, ein Videosignal als Volumen zu interpretieren, ermöglicht das Signal sowohl aus der örtlichen als auch der zeitlichen Umgebung in den Logobereich zu extrapolieren. Die Prädiktion in der hybriden Videocodierung wird als Extrapolation bereits codierter Daten zur Erhöhung der Codiereffizienz interpretiert. Deswegen wird die 3D Extrapolation in den H.264/AVC Coder als alternativer Codier Modus integriert.

Die wichtigsten Ergebnisse dieser Arbeit werden abschließend in Kapitel 8 zusammengefasst. Zusätzlich wird ein Ausblick auf vielversprechende zukünftige Forschungsrichtungen gegeben.

B.4 Zusammenfassung und Schlussfolgerungen

In dieser Arbeit wird eine generische Methode zur Schätzung unbekannter Signalwerte durch die Extrapolation bekannter Signaldaten hergeleitet. Das vorgestellte Konzept wird als *selektive Extrapolation* bezeichnet. Durch die Verwendung problemspezifischer Basisfunktionen kann dieser sehr allgemeine Ansatz auf eine Vielzahl von Extrapolationsaufgaben angewandt werden.

Da Bild- und Videosignale als eine Linearkombination mehrdimensionaler Frequenzen aufgefasst werden können, sind DFT ähnliche Basisfunktionen besonders geeignet, was zu dem Ansatz der *frequenzselektiven Extrapolation* führt. Mit dem neuen Konzept der 3D Extrapolation kann ein Videosignal als 3D Volumen interpretiert werden. Daher können *örtliche und zeitliche* Korrelationen des Videosignals *gleichzeitig* für die Extrapolation genutzt werden.

Das während der Extrapolation minimierte Fehlerkriterium ermöglicht durch die Verwendung einer Gewichtungsfunktion eine anwendungsspezifische Anpassung, d.h. dass Teilbereiche, die für die Extrapolation wichtiger sind als andere, durch eine größere Gewichtung stärker einbezogen werden können. Für die in dieser Arbeit betrachteten Anwendungen, die durch einen zusammenhängenden Fehlbereich gekennzeichnet sind, der zentral innerhalb des betrachteten Blocks lokalisiert und von bekannten Daten umgeben ist, ist es sinnvoll anzunehmen, dass ein Bildpunkt umso wichtiger für die Extrapolation ist, je näher er sich am zu schätzenden Bereich befindet. Deswegen basiert die gewählte Gewichtungsfunktion sowohl für Bild- als auch für Videosignale auf einem isotropen Modell, welches radialsymmetrisch mit zunehmender Distanz zum unbekannten Bereich abnimmt.

Die hergeleitete Methode der frequenzselektiven Extrapolation wurde auf mehrere Fragestellungen angewandt, wie sie in der Bild- und Videokommunikation typischerweise

auftreten. Zunächst wird die Fehlerverschleierung von verloren gegangenen blockcodierten Daten behandelt. Einerseits wird die 2D frequenzselektive Extrapolation zur örtlichen Verschleierung angewandt und andererseits örtlich-zeitliche Verschleierung durch den 3D Ansatz ermöglicht. Beide Ansätze werden erfolgreich als Verschleierungsmethoden in den H.264/AVC Coder integriert. Im Gegensatz zu konventionellen Methoden der zeitlichen Verschleierung, kann die 3D frequenzselektive Extrapolation leicht zur Verschleierung Intra codierter Makroblöcke angewendet werden, da sie nicht auf Bewegungsvektoren basiert.

Andere Coder wie der JPEG2000 Coder verwenden Wavelet-basierte Codierprinzipien. In diesem Fall hat ein Datenverlust völlig andere Auswirkungen auf die visuelle Qualität des rekonstruierten Bildes als im Vergleich zu blockbasierten Codern. Untersuchungen haben gezeigt, dass ein Verlust im Tiefpass Band besonders schwerwiegende Artefakte zur Folge hat. Deswegen werden vom Coder zur Verfügung gestellte Werkzeuge zur Erhöhung der Fehlerrobustheit eingesetzt, um im Falle eines Übertragungsfehlers den Verlust an Wavelet-Koeffizienten im Tiefpass-Band im Vergleich zu den anderen Teilbändern weiter einzuschränken. Um Verluste im Tiefpass-Band zu verschleiern, wird die Technik der 2D frequenzselektiven Extrapolation in den JPEG2000 Decoder zur örtlichen Verschleierung verlorener Wavelet-Koeffizienten integriert.

Zwei weitere Anwendungen werden aus dem Blickwinkel der Extrapolation betrachtet: einerseits die TV Logoentfernung und andererseits die Prädiktion in der hybriden Videocodierung. Bei der Logoentfernung ist das Logo für jedes Bild der Sequenz an der gleichen örtlichen Position platziert. Daher ist ein einfaches Kopieren aus dem vorangegangenen Bild nicht möglich. Die Verwendung der 2D Extrapolation auf jedes einzelne Bild führt zu einem Flacker-Effekt bei Sequenzen mit unbewegtem Inhalt, der dadurch vermieden werden kann, dass das sich über die gesamte Szene erstreckende 3D Volumen auf einmal extrapoliert wird. Jedoch hängt die visuelle Qualität der ersetzten Bereiche beim Einsatz der 3D Extrapolation maßgeblich von der Bewegung der Sequenz ab. Szenenschnitte können schwerwiegende Artefakte hervorrufen. Die besten Ergebnisse können daher erzielt werden, wenn die Sequenz in Szenen gleichgearteter Bewegung und Inhalte aufgeteilt wird, die dann separat verarbeitet werden.

Bei der Prädiktion in der hybriden Videocodierung wird der zu prädizierende Block durch die 3D Extrapolation von bereits codierten örtlich-zeitlichen Bildpunkten prädiziert. Der Ansatz ist in einen H.264/AVC Coder als Prädiktionsmodus integriert worden, der auf Rate-Distortion optimierenden Codierprinzipien basiert. In Fällen, in denen die zeitliche Information nicht zuverlässig ist, wie bei Verdeckungen oder nicht ausschließlich translatorischer Bewegung, kann die Codiereffizienz von der 3D Extrapolation profitieren. Obwohl normalerweise nur einige Makroblöcke nicht durch Bewegungskompensation prädiziert werden, kann ein Gewinn an Codiereffizienz durch den neuen Prädiktionsmodus erzielt werden.

Zusammenfassend lässt sich feststellen, dass die frequenzselektive Extrapolation einen sehr allgemeinen Ansatz zur Lösung von Extrapolationsfragestellungen in der Bild- und Videokommunikation darstellt. Der Hauptvorteil besteht in der Eigenschaft des Algorithmus, sich auf Grund der Frequenzselektivität an lokale Signalcharakteristiken anzupassen. Im Falle des 2D Ansatzes können sowohl glatte Flächen, Kanten als auch rauschartige Bereiche extrapoliert werden. Da der neuartige 3D Ansatz das Videosignal in einem Volumen betrachtet, können zusätzlich Bewegungen und Luminanzänderungen, die von Bild zu Bild auftreten, kompensiert werden. Diese leistungsfähigen Extrapolationseigenschaften können besonders erfolgreich für Anwendungen genutzt werden, bei denen vor allem der visuelle Eindruck der extrapolierten Bereiche von Bedeutung ist, wie es z.B. bei der Fehlerverschleierung und der Logoentfernung der Fall ist.

Die allgemeine Herleitung dieses Ansatzes und die damit erzielten Ergebnisse, eröffnen eine große Bandbreite zukünftiger Forschungsaktivitäten. Die Schlussfolgerungen aus den theoretischen Untersuchungen, die anhand der geometrischen Interpretationen gewonnen wurden, können die Grundlage für weitere algorithmische Entwicklungen darstellen. In jeder Iteration wird der Restfehlervektor des Approximationsunterraums auf die selektierte Basisfunktion aus dem Unterraum des Unterstützungsbereichs projiziert. Die Basisfunktionen in diesem Unterraum sind nicht orthogonal zueinander, d.h. die selektierte Basisfunktion beinhaltet auch Komponenten anderer Basisfunktionen. Die nicht vorhandene Orthogonalität sollte bei der Berechnung des Expansionskoeffizienten, z.B. durch eine entsprechende Dämpfung, berücksichtigt werden.

In den bisher betrachteten Anwendungen sind sowohl der Unterstützungsbereich als auch die zu schätzenden Bereiche zusammenhängend. Zukünftige Forschungsaktivitäten können andere Formen von bekannten und unbekannten Bereichen betrachten, wie sie z.B. beim Deinterlacing auftreten. Dort wird eine im Interlaced-Format aufgenommene Sequenz für die Ausgabe auf einem Bildschirm mit Progressiv-Format konvertiert. Die zwei Halbbilder eines Vollbildes können als eine Quincunx-Unterabtastung dieses Vollbildes in Zeilen-Zeit-Richtung interpretiert werden, die zu spektralen Wiederholungen im Frequenzbereich führen. Die Auswirkungen dieser Wiederholungen müssen bei der Selektion der Basisfunktion bzw. der Frequenz berücksichtigt werden. Um Aliasing zu vermeiden, muss deshalb der zur Selektion verfügbare Frequenzbereich eingeschränkt werden.

Die frequenzselektive Extrapolation kann direkt auf mehrere Dimensionen erweitert und für entsprechende Anwendungen genutzt werden. Die Codierung und Übertragung dynamischer Lichtfelder verwendet z.B. vier Dimensionen für eine effiziente Verarbeitung, wobei die Menge der aufnehmenden Kameras solcher Multi-View Sequenzen die vierte Dimension darstellen. Während der Übertragung solcher codierter dynamischer Lichtfeld-daten auftretende Fehler sollten auf der Decoderseite verschleiert werden. Zusätzlich zu örtlichen und zeitlichen Daten könnten Daten der anderen Kameras für die Verschleierung mittels 4D frequenzselektiver Extrapolation genutzt werden.

In dieser Arbeit wurde die frequenzselektive Extrapolation auf Signale der Bild- und Videokommunikation angewandt. Die hervorragenden Extrapolationseigenschaften könnten auch für Signale mit anderen Statistiken genutzt werden, die nicht im Anwendungsbereich dieser Arbeit liegen. Medizinische Bildsignale beinhalten z.B. einen großen Rauschanteil. Für Extrapolationsprobleme medizinischer Bildsignale ist es außerordentlich wichtig, dieses Rauschen nachbilden zu können, um ein natürliches Erscheinungsbild der Bilder zu gewährleisten. Auf Grund der Frequenzselektivität des Algorithmus wäre es möglich, diese rauschartigen Bereiche geeignet zu extrapolieren. Röntgen-Flachdetektoren erzeugen Bilder mit defekten Bereichen, die unter anderem auf Grund prinzipieller Schwierigkeiten bei der Herstellung entstehen. Die frequenzselektive Extrapolation kann zur Entfernung dieser Defekte durch Extrapolation des umgebenden Signals in den defekten Bereich genutzt werden. In [2] wird bereits das große Potenzial für medizinische Anwendungen durch die Verwendung der spektralen Entfaltung gezeigt, die der frequenzselektiven Extrapolation mittels einer rechteckigen binären Fensterfunktion entspricht. Für statische Röntgenbilder könnte deshalb eine 2D frequenzselektive Extrapolation mit einer angepassten Gewichtungsfunktion von Vorteil sein. Der 3D Ansatz könnte bei der Bearbeitung von Defekten in durch Cine-Angiographie oder Fluoroskopie aufgenommenen Sequenzen verwendet werden.

Bibliography

- [1] ISO/IEC 15444-1. *The JPEG2000 Image coding system-Part 1: core coding system*. ISO/IEC/ITU-T, 2000.
- [2] T. Aach. Missing data interpolation by transform-based successive approximation. In *Proc. Workshop on Spectral Methods and Multirate Signal Processing (SMMSP-2001)*, pages 3–10, Pula, Croatia, June 2001.
- [3] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal Processing*, 31:165–180, March 1993.
- [4] T. Aach and V. Metzler. Defect interpolation in digital radiography - how object-oriented transform coding helps. In *SPIE Medical Imaging 2001*, volume 4322, San Diego, CA, Feb. 2001.
- [5] Til Aach and Dietmar Kunz. A lapped directional transform for spectral image analysis and its application to restoration and enhancement. *Signal Processing*, 80(11):2347–2364, 2000.
- [6] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. Motion field interpolation for temporal error concealment. *IEE Proc.-Vis. Image Signal Process.*, 147(5):445–453, Oct. 2000.
- [7] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull. Multiple-reference temporal error concealment. In *IEEE Int. Symposium on Circuits and Systems*, pages V149–V152, May 2001.
- [8] Z. Alkachouh and M. Bellanger. Fast DCT-based spatial domain interpolation of blocks in images. *IEEE Trans. Image Process.*, 9(4):729–732, April 2000.
- [9] P. Amon. Error resilience for JPEG2000. In *Proc. Picture Coding Symposium (PCS)*, pages 243–246, France, April 2003.
- [10] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Siggraph 2000, Computer Graphics Proceedings*, pages 417–424, USA, July 2000.

- [11] Canon, EPFL, and Ericsson. JJ2000 An implementation of the JPEG standard in Java. <http://jj2000.epfl.ch/index.html>.
- [12] A. A. Clark, B. T. Thomas, N. W. Campbell, and P. Greenway. Texture deconvolution for the Fourier-based analysis of non-rectangular regions. In *British Machine Vision Conference*, pages 193–202, Sept. 1999.
- [13] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of the complex fourier series. In *Mathematics of Computation*, pages 297–301, April 1965.
- [14] U. L. Dang. *Entfernung von TV-Logos mittels dreidimensionaler frequenzselektiver Kompensation*. Student Research Project. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany, 2006.
- [15] F. Dufaux, G. Baruffa, F. Frescura, and D. Nicholson. JPWL - an Extension of JPEG 2000 for Wireless Imaging. In *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Greece, May 2006.
- [16] H. G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numerische Mathematik*, 69:423–440, 1995.
- [17] J. C. L. Barea J. Hirsch J. G. Velazquez V. Van Isacker P. Frank, A. Vieyra. Nuclear forecasting as pattern recognition: Can we predict nuclear masses? In *American Institute of Physics (AIP) Conference Proceedings*, volume 819, pages 151–158, USA, 2006.
- [18] U. Franke. *Regionenorientierte Bildbeschreibung - Algorithmen und Möglichkeiten*. Fortschritt-Berichte VDI, Reihe 10: Informatik/Kommunikationstechnik, Nr. 101, 1989.
- [19] M. Friebe and A. Kaup. 3D-deblocking for error concealment in block-based video decoding systems. In *Proc. Picture Coding Symposium (PCS)*, Beijing, April 2006.
- [20] M. Friebe and A. Kaup. Spatio-temporal fading scheme for error concealment in block-based video decoding systems. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 2237–2240, Atlanta, Oct. 2006.
- [21] J.-U. Garbas. *Fehlerkontrolle und Fehlerverschleierung bei gestört empfangenen JPEG2000 codierten Bildern*. Diploma Thesis. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany, 2004.
- [22] J. A. Högbom. Aperture synthesis with a non-regular distribution of interferometer baselines. *Astron. Astrophys. Suppl.*, 15:417–426, 1974.

- [23] Joint video team (JVT) of ISO/IEC MPEG and ITU-T VCEG. *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264—ISO/IEC 14496-10 AVC)*. ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, March 2003.
- [24] A. Kaup. *Modelle zur regionenorientierten Bildbeschreibung*. PhD thesis, Universität RWTH Aachen, Düsseldorf: VDI-Verlag, Fortschritts-Berichte VDI Reihe 10, no. 381, 1995.
- [25] A. Kaup and T. Aach. Efficient prediction of uncovered background in interframe coding using spatial extrapolation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume V, pages 501–504, Adelaide Australia, April 1994.
- [26] A. Kaup and T. Aach. Coding of segmented images using shape-independent basis functions. *IEEE Trans. Image Process.*, 7(7):937–947, 1998.
- [27] A. Kaup, K. Meisinger, and T. Aach. Frequency selective signal extrapolation with applications to error concealment in image communication. *Int. J. Electron. Commun. (AEÜ)*, 59:147–156, June 2005.
- [28] A. K. Kokaram. Reconstruction of severely degraded image sequences. *Image Analysis and Processing*, 2:719–725, Sept. 1997.
- [29] W.-M. Lam, A. R. Reibman, and B. Liu. Recovery of lost or erroneously received motion vectors. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages V417–V420, April 1993.
- [30] C. W. Lee, K. Jung, and H. J. Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24:2607–2623, 2003.
- [31] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3414, Dec. 1993.
- [32] S. Martin. *Dreidimensionale frequenzselektive Signalextrapolation und ihre Anwendung in der Videokommunikation*. Diploma Thesis. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany, 2005.
- [33] K. Meisinger, J.-U. Garbas, and A. Kaup. Error control and concealment of JPEG2000 coded image data in error prone environments. In *Proc. Picture Coding Symposium (PCS)*, San Francisco, Dec. 2004.
- [34] K. Meisinger and A. Kaup. Minimizing a weighted error criterion for spatial error concealment of missing image data. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 813–816, Singapore, Oct. 2004.

- [35] K. Meisinger and A. Kaup. Spatial error concealment of corrupted image data using frequency selective extrapolation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume III, pages 209–212, Montreal Canada, May 2004.
- [36] K. Meisinger and A. Kaup. Spatio-temporal selective extrapolation for 3-D signals and its applications in video communications. *IEEE Trans. Image Process.*, accepted for publication 2007.
- [37] K. Meisinger, S. Martin, and A. Kaup. Spatio-temporal selective extrapolation for 3-D signals applied to concealment in video communications. In *Proc. European Signal Processing Conference (EUSIPCO)*, Italy, Sept. 2006.
- [38] K. Meisinger, T. Troeger, M. Zeller, and A. Kaup. Automatic TV logo removal using statistical based logo detection and frequency selective inpainting. In *Proc. European Signal Processing Conference (EUSIPCO)*, Turkey, Sept. 2005.
- [39] K. Meisinger, T. Troeger, M. Zeller, and A. Kaup. Automatische TV-Logo-Entfernung durch statistische modellbasierte Detektion und frequenzselektive Kompensation. In *Elektronische Medien: 11. Dortmunder Fernsehseminar, ITG-Fachbericht 188*, Dortmund, Sept. 2005.
- [40] A. Natsu and D. Taubmann. Unequal protection of JPEG2000 code-streams in wireless channels. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, volume 1, pages 534–538, Nov. 2002.
- [41] R. Neff and A. Zakhor. Matching pursuit video coding – part I: Dictionary approximation. *IEEE Trans. Circuits Syst. Video Technol.*, 12(1):13–26, Jan. 2002.
- [42] J.-R. Ohm. *Multimedia Communication Technology*. Springer-Verlag, Berlin, Germany, 2004.
- [43] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with H.264/AVC: Tools, performance, and complexity. *IEEE Circuits and Systems Magazine*, 4(1):7–28, 2004.
- [44] A. Papoulis. A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Transactions on Circuits and Systems*, 22(9):735–742, 1975.
- [45] A. Papoulis and C. Chamzas. Detection of hidden periodicities by adaptive extrapolation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(5):492–500, Oct. 1979.
- [46] R. Penrose. A generalized inverse for matrices. In *Proc. Cambridge Phil. Soc.*, volume 51, pages 406–413, 1955.

- [47] A. Rahmoune, P. Vandergheynst, and P. Frossard. MP3D: Highly scalable video coding scheme based on matching pursuit. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 133–136, May 2004.
- [48] J. Seiler. *Schätzung unbekannter Bildbereiche in Videosignalen mittels zeitlich-örtlicher selektiver Extrapolation*. Diploma Thesis. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany, 2006.
- [49] S. Shirani, F. Kossentini, and R. Ward. Reconstruction of baseline JPEG coded images in error prone environments. *IEEE Trans. Image Process.*, 9(7):1292–1299, July 2000.
- [50] R. Sottek. *Modelle zur Signalverarbeitung im menschlichen Gehör*. PhD thesis, University RWTH Aachen, Germany, 1993.
- [51] R. Sottek, K. Illgner, and T. Aach. An efficient approach to extrapolation and spectral analysis of discrete signals. *Informatik-Fachberichte 253, ASST 90*, pages 103–108, 1990.
- [52] T. Stockhammer and M. H. Hannuksela. H.264/AVC video for wireless transmission. *IEEE Wireless Communications*, 12(4):6–13, Aug. 2005.
- [53] T. Strohmer. On discrete band-limited signal extrapolation. *Contemp. Math*, 190:323–337, 1995.
- [54] K. Suehring. H.264/AVC Reference Software. <http://iphome.hhi.de/suehring/tml/download/>.
- [55] G. J. Sullivan and T. Wiegand. Video compression - from concepts to the H.264/AVC standard. *Proceedings of the IEEE*, 93:18–31, Jan. 2005.
- [56] H. Sun and W. Kwok. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Trans. Image Process.*, 4(4):470–477, April 1995.
- [57] T. Tröger. *Automatische Erkennung von Senderlogos in Videos zur Logoentfernung basierend auf einem statistischen Modell*. Student Research Project. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany, 2005.
- [58] V. Varsa, M. Karczewicz, G. Roth, R. Sjöberg, T. Stockhammer, and G. Liebl. *Common Test Conditions for RTP/IP over 3GPP/3GPP2*. ITU-T SG.16/VCEG-N80, Dec. 2001.
- [59] Y. Wang and Q.-F. Zhu. Error control and concealment for video communication: a review. *Proceedings of the IEEE*, 86(5):974–977, May 1998.

- [60] Y. Wang, Q.-F. Zhu, and L. Shaw. Maximally smooth image recovery in transform coding. *IEEE Trans. on Commun.*, 41(10):1544–1551, Oct. 1993.
- [61] Y.-K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj. The error concealment feature in the H.26L test model. In *Proc. Int. Conf. on Image Processing (ICIP)*, volume 2, pages 729–732, Sept. 2002.
- [62] S. Wenger. H.264/AVC over IP. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):645–656, July 2003.
- [63] L. Xin and M. T. Orchard. Novel sequential error-concealment techniques using orientation adaptive interpolation. *IEEE Trans. Circuits Syst. Video Technol.*, 12(10):857–864, Oct. 2002.
- [64] W.-Q. Yan and M. S. Kankanhalli. Erasing video logos based on image inpainting. In *Proc. Int. Conf. on Multimedia and Expo (ICME)*, volume 2, pages 521–524, Lausanne, Switzerland, Aug. 2002.
- [65] M. Zeller. *Entfernen von Senderlogos aus Videos basierend auf einer frequenzselektiven Fehlerverschleierungstechnik*. Student Research Project. Chair of Multimedia Communications and Signal Processing, University Erlangen-Nuremberg, Germany, 2005.
- [66] J. Zhai, K. Yu, J. Li, and S. Li. A low complexity motion compensated frame interpolation method. In *The 2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 4927–4930, Kobe, Japan, May 2005.
- [67] J. Z. Zhang, J. F. Arnold, and M. R. Frater. A cell-loss concealment technique for MPEG-2 coded video. 10(4):659–665, June 2000.

Curriculum Vitae

Name:	Katrin Meisinger	
Birth:	May 6, 1975	Hilpoltstein, Germany
Nationality:	German	
School education:	Sept. 1981 – July 1985	Grundschule, Heideck
	Sept. 1985 – July 1994	Gymnasium, Hilpoltstein
	July 1994	Abitur
University education:	Oct. 1994 – July 1997	Student of electrical engineering, University of Erlangen-Nuremberg
	Aug. 1997 – Sept. 1998	Student of electrical engineering, Royal Institute of Technology Stockholm
	Oct. 1998 – Sept. 2001	Student of electrical engineering, University of Erlangen-Nuremberg
	Sept. 2001	Dipl.–Ing. degree
Internship:	July 2000 – Nov. 2000	Ericsson Nürnberg, Research Mobile Communications
Professional experience:	Feb. 2002 – Feb. 2007	Research assistant at the Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg