Diplomarbeit

Audio Inpainting

A Comparison between Methods of Autoregressive Modeling and Compressive Sampling

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Diplom-Ingenieurs

unter der Leitung von

Univ.Prof. Dipl.-Ing. Dr.-Ing. Norbert Görtz
Ass.Prof. Dipl.-Ing. Dr.techn. Gerhard Doblinger

Institute of Telecommunications

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Christian Hölzel
Matrikelnr.: 0626910
Hauptstraße 2
3122 Gansbach

Wien, April 2014

I hereby certify that the work reported in this thesis is my own, and the work done by other authors is appropriately cited.

Christian Hölzel
Wien, April 2014

Abstract

This diploma thesis evaluates the performance of restoration algorithms for the recovery of impulsively distorted samples and gaps in digital audio signals. The focus specifically lies on the comparison of methods from the recently very active field of compressive sampling with a classical technique from the 80s that is based on autoregressive modeling. Both approaches rely on the generation of a signal model from the reliable data to estimate the values of the defective samples. The latter are treated as missing and their locations are assumed to be known a priori. No additional information has to be incorporated. The corresponding theoretical basics of digital signal processing are outlined and some detailed insight into the specific algorithmic steps is given. For autoregressive modeling, we apply fast methods for the adaptive estimation of the model parameters and subsequent calculation of the unknown samples that utilize the Levinson-Durbin recursion and Cholesky decomposition. The compressive sampling methods are backed by the assumption that audio signals can be represented by a sparse vector in conjunction with a proper dictionary of basis functions. For this purpose, we employ a redundant discrete cosine transform dictionary. The examined reconstruction algorithms comprise of Orthogonal Matching Pursuit, Least Angle Regression and Iterative Soft Thresholding. In a series of extensive numerical experiments, the signal-to-noise-ratio of the resulting approximations is computed and compared, considering various kinds of error scenarios for multiple sets of speech and music signals. The developed software package for MATLAB is appended to allow for convenient reproducibility.

Keywords — Digital audio restoration, inpainting, autoregressive modeling, compressive sampling, sparse approximation.

Kurzfassung

Diese Diplomarbeit evaluiert die Leistungsfähigkeit von Algorithmen zu Rekonstruktion von impulsartig gestörten Samples und Aussetzern in digitalen Audiosignalen. Das Hauptaugenmerk liegt dabei auf dem Vergleich von Methoden aus dem relativ neuen Gebiet des Compressive Sampling mit einer klassischen Technik aus den 80ern, die auf autoregressiver Modellierung beruht. Bei beiden Ansätzen wird mittels der Informationen, die in den ungestörten Daten enthalten sind, ein Signalmodell generiert, welches anschließend für die Schätzung der unbekannten Samplewerte benutzt wird. Letztere werden als fehlend und ihre Positionen als von vornherein bekannt angenommen. Es müssen dabei keine weiteren Kenntnisse miteinbezogen werden. Die nötigen Grundlagen aus der Theorie der digitalen Signalverarbeitung werden umrissen und ein detaillierter Einblick in die speziellen algorithmischen Abläufe gegeben. Für die autoregressive Modellierung setzen wir schnelle Methoden zur adaptiven Schätzung der Modellparameter und der darauffolgenden Berechnung der unbekannten Samples ein, die die Levinson-Durbin-Rekursion und Cholesky-Zerlegung verwenden. Die Methoden des Compressive Sampling stützen sich auf die Annahme, dass Audiosignale in einer passenden Domäne durch einen schwachbesetzten Vektor repräsentiert werden können. Wir verwenden dazu ein redundantes Discrete Cosine Transform Dictionary. Die betrachteten Algorithmen beinhalten Orthogonal Matching Pursuit, Least Angle Regression und Iterative Soft Thresholding. In einer Reihe von umfangreichen computerbasierten Experimenten wird der Signal-Rausch-Abstand der resultierenden Näherungswerte berechnet und verglichen, wobei verschiedenartige Fehlerszenarien mit mehreren Sets von Sprach- und Musiksignalen getestet werden. Das zu diesem Zweck entwickelte Softwarepaket für MATLAB ist beigelegt, um eine einfache Reproduzierbarkeit zu ermöglichen.

Schlüsselwörter — Digitale Rekonstruktion von Audiosignalen, Inpainting, Autoregressive Modellierung, Compressive Sampling, Sparse Approximation.

Contents

1 Introduction

When a digital audio signal is transmitted or stored, chances are it will be disturbed in some way at some point. Besides the inevitable additive noise, the most common types of errors are impulsive distortions and gaps, both of which can lead to audible artifacts during playback. Although most communications and storage systems are equipped with error correction mechanisms such as channel codes [1], there can still occur errors that exceed their capabilities, so the values of the defective samples have to be accurately estimated afterwards. Hence, the challenge is to find replacements such that no more artifacts can be perceived or, in other words, to make the reconstructed signal virtually indistinguishable from the original. Such a procedure is called sample restoration [2] or audio inpainting [3]. The term inpainting originates from the field of image processing [4], where a typical task is the removal of an unwanted foreground object from an occluded image (see Fig. 1.1). Regarding audio signals, we typically have to deal with two kinds of error patterns. The disturbed samples can appear in a randomly scattered pattern, as shown in Fig. 1.2, whereas Fig. 1.3 illustrates the important special case of a burst.

The main focus of this thesis lies on the testing and comparison of two state-of-the-art time-domain restoration methods. One is based on autoregressive modeling (AR) as proposed by Janssen *et al.* in 1986 [5], while the other utilizes the rather recently established compressive sampling (CS) framework and is based on a publication by Adler *et al.* in 2012 [6]. Both approaches utilize the information contained in the reliable data to build a signal model and subsequently try to estimate the erroneous parts employing that model.

Since there is no additional information about the defective samples, they are treated as missing and their locations are assumed to be known a priori.

The thesis is organized as follows. In Chapter 2 we give an overview of some classical interpolation methods and comment on why the AR technique was picked as a representative. Basic definitions of the AR model and the derivation of the restoration algorithm are given in Chapter 3. The CS audio inpainting framework with three selected sparse approximation algorithms is introduced in Chapter 4. We present our experimental results in Chapter 5 and draw conclusions in Chapter 6. Details about the testing software can be found in Appendix A.

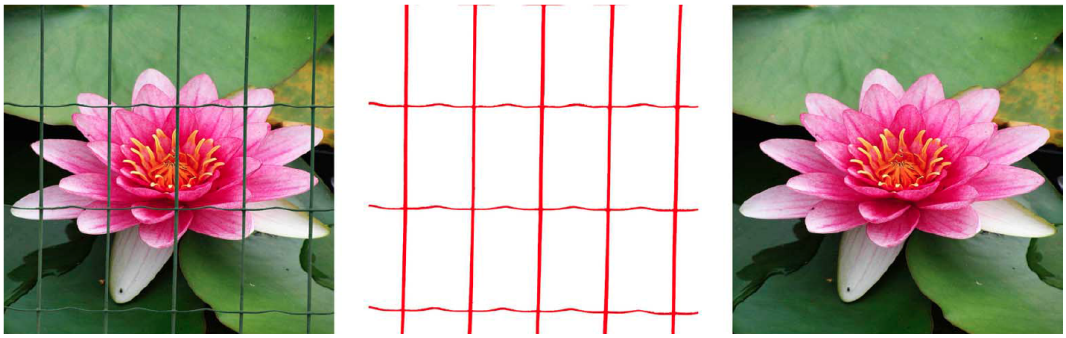


Figure 1.1: Image inpainting: an unwanted foreground object has to be removed from an occluded image [7]

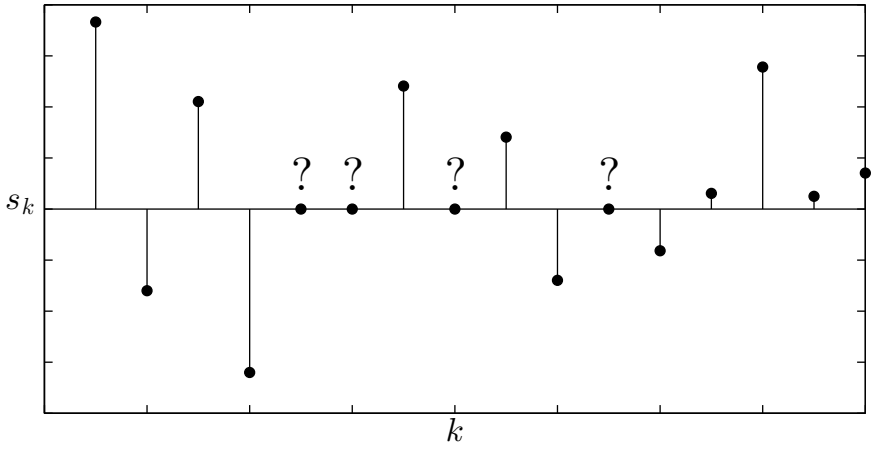


Figure 1.2: Randomly scattered pattern of unknown samples

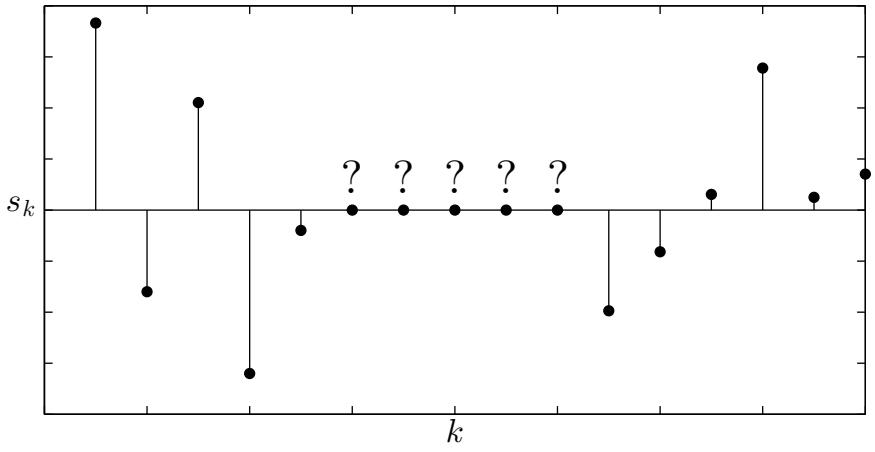


Figure 1.3: Burst of unknown samples

Nomenclature

$a, b, c, \ldots$	Scalars
$\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$	Random variables
$\hat{a}, \hat{b}, \hat{c}, \ldots$	Estimates of variables
$\underline{a}, \underline{b}, \underline{c}, \ldots$	Column vectors with scalar elements
$\underline{\mathbf{a}}, \underline{\mathbf{b}}, \underline{\mathbf{c}}, \ldots$	Column vectors with random variable elements
$A, B, C, \ldots$	Matrices with scalar elements
$\underline{a}^{(k)}$	Entity of vector $\underline{a}$ at iteration k
$\underline{0}$	Zero vector of appropriate size
$\underline{e}_k$	Unit vector with all elements zero, except $e_k = 1$
I_N	$N \times N$ identity matrix
$\langle \underline{a}, \underline{b} \rangle$	Inner product of vectors $\underline{a}$ and $\underline{b}$
$\lvert I \rvert$	Number of elements in the set I

Specific symbols

s_k	Segment of available data
$\underline{s}$	Vector of available data segment
N	Length of available data segment
R_N	$N \times N$ autocorrelation matrix
p	AR model order
$\underline{a}$	Vector of AR parameters
m	Number of unknown samples
$t(i)$	Time instants of unknown samples, $i = 1, \ldots, m$
$\underline{x}$	Vector of unknown samples
$\underline{y}$	Vector of reliable samples
$\underline{u}$	Sparse representation vector
K	Sparsity level
K_D	Dictionary size
ϵ	Approximation error threshold

2 Overview of classical interpolation methods

A variety of methods is available for the estimation of unknown sample values in discrete-time signals, but not all of them are suited for audio data containing predominantly harmonic components. Simple first-order linear interpolation is working only for bursts up to 5 samples without producing audible artifacts. Curve fitting methods utilizing Lagrange polynomials also give poor results if the number of unknown samples exceeds the number of samples in the periods of the harmonic components [2]. An approach where the lost parts of a transmitted audio signal are substituted by correctly received ones from the surroundings is pursued in [8,9], but for this method to work properly, a basic periodicity of the signal has to be assumed, which does not hold in general.

Better results can be obtained by employing a statistical signal model for the data generation process. In [10], the assumed model is the band-limitedness of the signal. The restoration procedure, in theory, comes down to the minimization of the energy outside a prescribed baseband. Unfortunately, it is numerically unstable and oversensitive to out-of-band components and thus also only practicable for comparatively short intervals in addition to a well-defined baseband. Moreover, various concepts focussing on packet loss concealment (e.g. in VoIP systems) should be mentioned [11,12].

3 Autoregressive modeling

A signal model widely considered very suitable for audio applications is the autoregressive (AR) model. It is known to give expedient results for real-life signals such as music and speech [13–17].

3.1 Definitions

The AR model is a special case of the more general autoregressive moving-average (ARMA) model for time series [18]. In the latter, the observed data are described by the linear difference equation

$$\mathbf{s_k} = \sum_{l=1}^{p} a_l \mathbf{s_{k-l}} + \sum_{m=0}^{q} b_m \mathbf{e_{k-m}}. \tag{3.1}$$

Looking at the transfer function

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{m=0}^{q} b_m z^{-m}}{\sum_{l=1}^{p} a_l z^{-l}}, \tag{3.2}$$

it can be seen to consist of applying an IIR filter to the excitation sequence $\mathbf{e_k}$, which is independent, identically distributed noise [19]. Several reasons why it is usually not the model of choice as a basis for sample restoration are discussed in [2].

For the AR case, $B(z) = 1$, which corresponds to an all-pole filter with the difference equation

$$\mathbf{s_k} = \mathbf{e_k} - \sum_{l=1}^{p} a_l \mathbf{s_{k-l}}, \tag{3.3}$$

$$\mathbf{e_k} = \sum_{l=0}^{p} a_l \mathbf{s_{k-l}}, (a_0 = 1), \tag{3.4}$$

where $\mathbf{s_k}, k \in \{-\infty, \ldots, \infty\}$, is an autoregressive process, p is the model order, $a_l, l = 1, \ldots, p$ are called AR parameters or prediction coefficients ($a_0 = 1, a_l = 0$ for $l < 0$ and $l > p$) and $\mathbf{e_k}$ is white noise with mean $\mu_e = 0$ and variance ${\sigma_e}^2$. No further assumptions have to be made about the statistical properties of the noise process. Equation (3.3) can be regarded as a prediction of $\mathbf{s_k}$ based on p preceding values with a prediction error $\mathbf{e_k}$. Theoretically, any signal $\mathbf{s_k}$ can be modeled as an AR process, as long as it is invertible. To ensure wide-sense stationarity, all poles of $H(z)$ must lie within the unit circle [20]. The power spectral density is given by

$$S(\Theta) = \frac{{\sigma_e}^2}{\sum_{l=-p}^{p} b_l \, e^{-j\Theta l}}, \tag{3.5}$$

with

$$b_l = \sum_{k=0}^{p} a_k a_{k+l}. \tag{3.6}$$

In general, for a finite segment of data $s_k, k \in \{0, \ldots, N\}$, where a single unknown sample is estimated by

$$\hat{s}_k = \sum_{\substack{l=0 \\ l \neq k}}^{N} \beta_k(l) s_l, \tag{3.7}$$

$$\underline{\beta}_k = \frac{{R_N}^{-1} \underline{e}_k}{({R_N}^{-1})_{kk}}, \tag{3.8}$$

$$\underline{\beta}_k = [\beta_k(0), \ldots, \beta_k(N)]^T, \tag{3.9}$$

the interpolation error power

$$J_k = \frac{1}{(R_N{}^{-1})_{kk}} \tag{3.10}$$

is minimal at the midpoint of the segment if the signal is modeled as an AR process [14]. This suggests that the erroneous samples should be located approximately in the middle of the contemplated signal segment for more general error scenarios.

3.2 Sample restoration

A feasible method for the interpolation of bursts and general error patterns has been proposed by Janssen *et al.* [5]. It consists of estimating the AR parameters and the calculation of the unknown samples such that the restored signal pervades the model assumptions as well as possible. This is achieved by minimizing the cost function

$$Q(\underline{a}, \underline{x}) = \sum_{k=p}^{N-1} \left| \sum_{l=0}^{p} a_l s_{k-l} \right|^2, \tag{3.11}$$

where N is the length of the available data segment, $\underline{a} = [a_1, \ldots, a_p]^T$ is the vector of AR parameters and $\underline{x} = \left[s_{t(1)}, \ldots, s_{t(m)}\right]^T$ contains the unknown samples at time instants $t(i), i = 1, \ldots, m$. Since this function involves fourth-order terms, it would be a non-trivial task to find its minimum in one step. Instead, under the limitation that the number of unknown samples m has to be considerably smaller than N, it can be split into two quadratic problems, which are outlined below for the important special case of a stationary Gaussian AR process.

3.2.1 Parameter estimation

The cost function can be written as

$$Q(\underline{a}, \underline{x}) = \underline{a}^T C(\underline{x}) \underline{a} + 2\underline{a}^T \underline{c}(\underline{x}) + c_{00}(\underline{x}), \tag{3.12}$$

with

$$C(\underline{x}) = [c_{ij}(\underline{x})], \qquad i, j = 0, \ldots, p, \tag{3.13}$$

$$\underline{c}(\underline{x}) = [c_{01}(\underline{x}), \ldots, c_{0p}(\underline{x})]^T, \tag{3.14}$$

$$c_{ij}(\underline{x}) = \sum_{k=p}^{N-1} s_{k-i} s_{k-j}, \qquad i, j = 0, \ldots, p. \tag{3.15}$$

Because $C(\underline{x})$ is symmetric, the minimization

$$\frac{\partial}{\partial \underline{a}} Q(\underline{a}, \underline{x}) = 2C(\underline{x})\underline{a} + 2\underline{c}(\underline{x}) = 0 \tag{3.16}$$

leads to

$$C(\hat{\underline{x}})\hat{\underline{a}} = -\underline{c}(\hat{\underline{x}}). \tag{3.17}$$

This is known in literature as the autocovariance method, reasoned by the fact that $C(\underline{x})$ holds the properties of a covariance matrix [21]. The number of operations required to solve (3.17) is on the order of $\mathcal{O}(Np)$ and can become computationally quite intense for longer signal segments and/or high model orders. A possible enhancement can be achieved by replacing the entries of $C(\underline{x})$ and $\underline{c}(\underline{x})$ with estimates of the respective autocorrelation lags, leading to the autocorrelation method. The resulting system of equations is then solvable with only $\mathcal{O}(p^2)$ operations using the Levinson-Durbin recursion [22].

3.2.2 Calculation of the unknown samples

It can be shown that (3.12) can likewise be expressed as

$$Q(\underline{a}, \underline{x}) = \underline{x}^T B(\underline{a})\underline{x} + 2\underline{x}^T \underline{z}(\underline{a}) + d(\underline{a}), \tag{3.18}$$

where

$$B(\underline{a}) = \left[b_{t(i)-t(j)}(\underline{a})\right], \qquad i, j = 1, \ldots, m, \tag{3.19}$$

$$\underline{z}(\underline{a}) = [z_1(\underline{a}), \ldots, z_m(\underline{a})]^T, \tag{3.20}$$

$$z_i(\underline{a}) = \sum_{k=-p}^{p} b_k s_{t(i)-k}, \qquad i = 1, \ldots, m, \tag{3.21}$$

with b_k defined in (3.6) and $d(\underline{a})$ depending only on $\underline{a}$ and the known samples $s_k, k \notin \{t(1), \ldots, t(m)\}$. It is easily seen that $B(\underline{a})$ is also symmetric so, in analogy to (3.16), the minimization leads to

$$B(\hat{\underline{a}})\hat{\underline{x}} = -\underline{z}(\hat{\underline{a}}). \tag{3.22}$$

Since $B(\underline{a})$ is positive definite, an efficient implementation can be achieved by subjecting it to a Cholesky decomposition, facilitating (3.22) to be solved in $\mathcal{O}(m^3)$ operations [23]. One might also consider the generalized Levinson algorithm for this purpose, but a major drawback would be its strong dependence on the error pattern.

3.2.3 Adaptive interpolation

The presented steps can be implemented in an adaptive way by applying at each iteration the current estimates of the unknown samples $\hat{\underline{x}}^{(k)}$ for the estimation of the parameters $\hat{\underline{a}}^{(k)}$, which are then used for the calculation of the unknown samples $\hat{\underline{x}}^{(k+1)}$ in the subsequent step, and so forth. According to the convention that the erroneous samples are treated as missing, the initial estimate $\hat{\underline{x}}^{(0)}$ is set to $\underline{0}$. As mentioned above, the only necessary assumption is that $m \ll N$, which should be satisfied for most practical applications.

4 Compressive sampling

Over the past decade, significant advances have been made in the field of compressive sampling (a.k.a. compressed sensing) and sparse representation theory [24–26]. It has been demonstrated that techniques from this area can be employed to faithfully model audio signals [27–29].

4.1 Sparse approximation of audio signals

A segment of a signal $\underline{s} \in \mathbb{R}^N$ can be well approximated by a sparse linear combination

$$\underline{s} \approx D\underline{u}, \tag{4.1}$$

if the atoms (matrix columns) of the dictionary $D \in \mathbb{R}^{N \times K_D}$, $N \leqslant K_D$, represent a domain in which the signal can be considered sparse, i.e. the sparse representation vector $\underline{u} \in \mathbb{R}^{K_D}$ has only K non-zero entries, $K \ll N$, satisfying

$$\|\underline{s} - D\underline{u}\|_2^2 \leqslant \epsilon. \tag{4.2}$$

This means that the linear combination belongs to an ϵ-ball surrounding the signal $\underline{s}$, that is a region of $\mathbb{R}^N$ whose Euclidean distance from the signal is smaller than the approximation error threshold ϵ. The dictionary (see Section 4.3) is usually overcomplete and has full rank, implying that its columns

span the whole $\mathbb{R}^N$, which leads to an underdetermined linear system of equations with infinitely many solutions. In order to attain a well-defined solution, a sparsity-promoting regularization of the form

$$\min_{\underline{u}} \|\underline{u}\|_0 \;\; s.t. \;\; \|\underline{s} - D\underline{u}\|_2^2 \leqslant \epsilon \tag{4.3}$$

can be introduced, using the l_0 pseudo-norm $\|\cdot\|_0$ that counts the number of non-zero elements in $\underline{u}$ as a sparsity measure. Unfortunately, the extraction of the sparsest representation vector for this non-convex optimization problem is NP-hard and can't be solved directly in reasonable time. As a matter of fact, it would cost at least $\mathcal{O}(2^{K_D})$ flops [30], where $K_D > 10^3$ for typical applications! Hence, extensive research has been made in developing and adapting algorithms that are able to find near-optimal solutions. We outline three of them, that are expected to achieve satisfactory results for audio signal recovery, in Section 4.5.

4.2 Measurement Matrix

We apply the measurement matrix $M^r \in \{0,1\}^{|I^r| \times N}$ to separate the reliable samples from the erroneous ones via the transformation

$$\underline{y} = M^r \underline{s}. \tag{4.4}$$

It is derived from the identity matrix I_N by selecting only the $|I^r|$ rows corresponding to the unimpaired samples. Thus, the sparse approximation (4.1) can be expressed as

$$\underline{y} \approx M^r D\underline{u}. \tag{4.5}$$

Similarly, we use $M^m \in \{0,1\}^{|I^m| \times N}$ to denote the support of the missing samples.

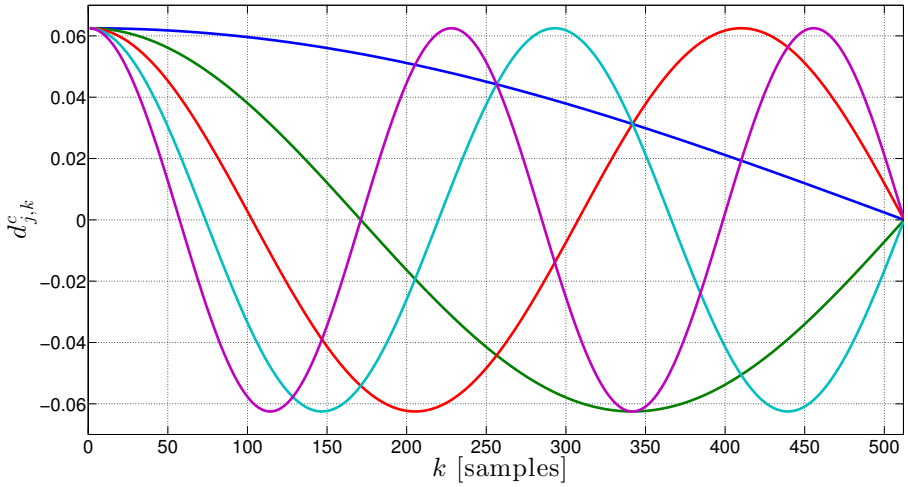


Figure 4.1: DCT atoms $j \in \{0, \ldots, 4\}$ with $N = 512$

4.3 Dictionary

We choose the discrete cosine transform (DCT) dictionary, which is widely used for sparse modeling of audio signals [28,29]. The atoms of D^c are defined as

$$d^c_{j,k} \stackrel{\text{def}}{=} w^d_k \cos\left(\frac{\pi}{K_D}\left(j + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right), \tag{4.6}$$

where $j \in \{0, \ldots, K_D - 1\}$ represents the frequency, $k \in \{0, \ldots, N - 1\}$ is the time index and w^d_k is a weighting window. For $K_D = N$, the matrix D^c is orthogonal, but one usually chooses $K_D > N$ to increase the probability of a sparser solution. Fig. 4.1 illustrates the first 5 atoms of the DCT dictionary with $N = 512$ and a rectangular weighting window.

For the examined reconstruction algorithms, it is requisite to normalize the atoms to unit norm by multiplying D with the diagonal matrix $W \in \mathbb{R}^{K_D \times K_D}$,

$$W = [w_{ij}] = \begin{cases} \frac{1}{\left\|M^r \underline{d}_j\right\|_2}\,; & i = j \\ 0\,; & i \neq j, \end{cases} \tag{4.7}$$

and selecting only the rows corresponding to the reliable samples

$$\tilde{D} = M^r D W. \tag{4.8}$$

4.4 Inpainting problem

The challenge is now to recover the missing samples $\underline{x}$ by estimating the sparse representation vector $\underline{u}$, such that

$$\underline{\hat{x}} = M^m D \underline{\hat{u}}, \tag{4.9}$$

given only the values and the positions of the reliable samples $\underline{y}$ and the dictionary D, as proposed by Adler *et al.* [6]. By using the relations (4.5) and (4.8), the optimization problem stated in (4.3) is therefore written as

$$\min_{\underline{u}} \|\underline{u}\|_0 \;\; s.t. \;\; \left\|\underline{y} - \tilde{D}\underline{u}\right\|_2^2 \leqslant \epsilon. \tag{4.10}$$

4.5 Reconstruction algorithms

As discussed in Section 4.1, finding the true sparsest representation in (4.10) is not feasible in practice. Thus, we employ suitable reconstruction algorithms to approximate the solution as well as possible.

4.5.1 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) is one of the earliest algorithms for sparse approximation [31,32]. It is an extension of the widely-used Matching Pursuit (MP) by Mallat and Zhang [33] and belongs to the family of greedy algorithms. "A greedy strategy abandons exhaustive search in favor of a series of locally optimal single-term updates. Starting from $\underline{u}^{(0)} = \underline{0}$ it iteratively constructs a k-term approximant $\underline{u}^{(k)}$ by maintaining a set of active columns – initially empty – and, at each stage, expanding that set by one additional column. The column chosen at each stage maximally reduces the residual l_2-error in approximating $\underline{s}$ from the currently active columns. After constructing an approximant including the new column, the residual l_2-error is evaluated; if it now falls below a specified threshold ϵ, the algorithm terminates" [26].

As detailed in Table 4.1, OMP identifies at every step the atom that is most strongly correlated with the current residual and then removes the projection of the residual onto the set of all previously selected atoms in order to get the

next residual. For this purpose, the LS problem (4.13) is solved by computing the Moore-Penrose pseudoinverse of the sub-dictionary $\tilde{D}_{\Omega_k}$,

$$\tilde{D}^{\dagger}_{\Omega_k} = (\tilde{D}^{T}_{\Omega_k}\tilde{D}_{\Omega_k})^{-1}\tilde{D}^{T}_{\Omega_k}. \tag{4.11}$$

This has to be done once per iteration and accounts for a large part of the computation time, which is on the order of $\mathcal{O}(K_D K_{max}\,|I^r|)$. This is more than for MP, but OMP converges in fewer iterations and leads to a smaller approximation error for a given number of atoms. Further advantages are its ease of implementation and the guarantee never to select the same atom twice, since the residual remains orthogonal to the active set at every iteration.

4.5.2 Least Angle Regression

It is possible to replace the l_0 pseudo-norm in (4.10) with the l_1 norm to obtain a convex optimization problem

$$\min_{\underline{u}} \|\underline{u}\|_1 \;\; s.t. \;\; \left\|\underline{y} - \tilde{D}\underline{u}\right\|_2^2 \leqslant \epsilon \tag{4.14}$$

that can be tackled in polynomial time [34]. This substitution is referred to as convex relaxation, with theoretical results closely related to those of OMP [30,35,36]. It is reasonable due to the fact that the l_1 norm is the convex function closest to l_0, since the l_p norm is non-convex for $p < 1$. Nevertheless, general-purpose LP solvers still require $\mathcal{O}({K_D}^3)$ flops to solve the full system (4.14).

We therefore utilze Least Angle Regression (LARS) [37] to approximately solve the Lagrangian equivalent of (4.14),

$$\min_{\underline{u}} \frac{1}{2}\left\|\underline{y} - \tilde{D}\underline{u}\right\|_2^2 + \lambda\|\underline{u}\|_1, \tag{4.15}$$

the so-called LASSO problem [38,39]. The parameter λ regulates the balance between the approximation error and the sparsity of the representation vector. The set $\{\hat{\underline{u}}_\lambda : \lambda \in [0,\infty)\}$ represents a solution path that converges to the solution of (4.14) as $\lambda \to 0$. It has been observed in [40] that this path is polygonal, piecewise linear and that its discretely numbered vertices correspond to solution subset models, which are vectors with non-zero elements only on a subset

of the potential candidate coefficients. This subset, the active set Ω, is augmented by one element at each step. To do this, LARS first determines the atom most correlated with the current residual and takes the largest step possible towards the direction of this atom until some other atom exhibits the same amount of correlation, corresponding to a vertex on the solution path. It then proceeds in a direction equiangular between the two atoms, the "least angle direction", until a third atom reaches as much residual correlation as the first two, and so forth. The detailed progression is presented in Table 4.2.

LARS and OMP share a similar structure, the only difference being that the LS problem that has to be solved at each iteration is l_1-penalized for LARS. Hence, LARS can be categorized "less greedy" than OMP. Its computational complexity is $\mathcal{O}(N^3)$.

4.5.3 Iterative Soft Thresholding

Iterative Soft Thresholding (IST) represents a different approach to surmount the optimization problem (4.14) when the solution is sufficiently sparse [41–43]. Starting from an initial solution $\hat{\underline{u}}^{(0)} = \underline{0}$, one iteratevely applies the rule

$$\hat{\underline{u}}^{(k+1)} = \eta_{t_k}(\hat{\underline{u}}^{(k)} + \kappa(\tilde{D}^T \underline{r}^{(k)})), \qquad \underline{r}^{(k)} = \underline{y} - \tilde{D}\underline{u}^{(k)}, \tag{4.18}$$

where the nonlinear soft thresholding operator

$$\eta_t(x) = \mathrm{sgn}(x)(|x| - t)_+ \tag{4.19}$$

is applied elementwise at each iteration k. It discards all elements smaller than the threshold value t and pulls the residual ones towards zero by the magnitude of t (see Fig. 4.2). The relaxation parameter $\kappa \in \{0, \ldots, 1\}$ is chosen as 0.6 according to the recommendation in [43]. At every step, the threshold value is calculated by leveraging the concept of interference heuristic known from statistical signal processing [44,45]. The marginal histogram of $\tilde{D}^T \underline{r}$ is assumed to be Gaussian for the estimation of the common standard deviation $\hat{\sigma}$. The threshold value is then set as a fixed multiple of $\hat{\sigma}$,

$$t = \xi\hat{\sigma}, \qquad \xi = \Phi^{-1}(0.9975 - 0.185\tau - 0.055\tau^2), \tag{4.20}$$

where $\tau = N/K_D$ and Φ^{-1} is the inverse of the standard normal distribution

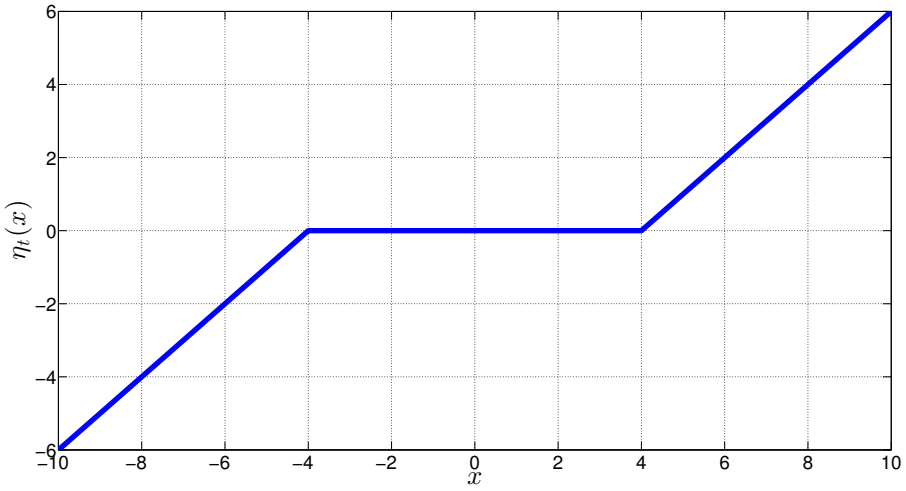


Figure 4.2: Soft thresholding operator for $t = 4$

function. The sequence of iterates $\underline{\hat{u}}^{(k)}$ approximately follows the LARS path at threshold t_k [46]. The complete algorithm is detailed in Table 4.3.

IST is easy to implement as it only requires two matrix-vector multiplications and some vector additions per iteration. The computational cost is on the order of $\mathcal{O}(N^2)$.

Input

- Reliable samples: $\underline{y} \in \mathbb{R}^{|I^r|}$
- Measurement matrix: $M^r \in \{0,1\}^{|I^r| \times N}$
- Dictionary: $D = \left[\underline{d}_j\right] \in \mathbb{R}^{N \times K_D}$
- Approximation error threshold: ϵ
- Maximum sparsity level: K_{max}

Initialization

- Normalized dictionary: $\tilde{D} = \left[\underline{\tilde{d}}_j\right] = M^r D W$
- Residual: $\underline{r}^{(0)} = \underline{y}$
- Active set: $\Omega_0 = \emptyset$
- Iteration counter: $k = 0$

Loop

- Increment iteration counter: $k = k + 1$
- Identify most strongly correlated atom:

$$j = \arg\max_j \left| \left\langle \underline{r}^{(k-1)}, \underline{\tilde{d}}_j \right\rangle \right| \tag{4.12}$$

- Augment active set: $\Omega_k = \Omega_{k-1} \cup j$
- Compute current estimate by solving the LS problem:

$$\underline{u}^{(k)} = \arg\min_{\underline{u}} \left\| \underline{y} - \tilde{D}_{\Omega_k} \underline{u} \right\|_2^2 \tag{4.13}$$

- Update residual: $\underline{r}^{(k)} = \underline{y} - \tilde{D}_{\Omega_k} \underline{u}^{(k)}$
- Stopping criteria: $k \geqslant K_{max}$ or $\left\| \underline{r}^{(k)} \right\|_2^2 \leqslant \epsilon$

Output

- Estimated SR vector: $\hat{\underline{u}} = W \underline{u}^{(k)}$

Table 4.1: OMP algorithm [6]

Input

- Reliable samples: $\underline{y} \in \mathbb{R}^{|I^r|}$
- Measurement matrix: $M^r \in \{0,1\}^{|I^r| \times N}$
- Dictionary: $D = \left[\underline{d}_j\right] \in \mathbb{R}^{N \times K_D}$
- Approximation error threshold: ϵ
- Maximum sparsity level: K_{max}

Initialization

- Normalized dictionary: $\tilde{D} = \left[\tilde{\underline{d}}_j\right] = M^r D W$
- Residual: $\underline{r}^{(0)} = \underline{y}$
- Residual correlations: $\underline{c}^{(0)} = \left\langle \underline{r}^{(0)}, \tilde{\underline{d}}_j \right\rangle$
- Active set: $\Omega_0 = \arg\max_j \left| c_j^{(0)} \right|$
- Initial solution: $\hat{\underline{u}}^{(0)} = \underline{0}$
- Iteration counter: $k = 0$

Loop

- Calculate update direction:

$$\underline{\delta} = \left(\tilde{D}_{\Omega_k}^T \tilde{D}_{\Omega_k}\right)^{-1} \operatorname{sgn}(\underline{c}_{\Omega_k}) \tag{4.16}$$

- Update parameter: $\lambda = \left\|\underline{c}^{(k)}\right\|_\infty$
- Calculate step size:[a]

$$\gamma = \min_{j \in \Omega_k^c}{}^{+} \left(\frac{\lambda - c_j^{(k)}}{1 - \tilde{\underline{d}}_j^{\,T} \underline{v}}, \frac{\lambda + c_j^{(k)}}{1 + \tilde{\underline{d}}_j^{\,T} \underline{v}} \right), \qquad \underline{v} = \tilde{D}_{\Omega_k} \underline{\delta} \tag{4.17}$$

- Increment iteration counter: $k = k + 1$
- Augment active set by minimizing index of (4.17): $\Omega_k = \Omega_{k-1} \cup j$
- Compute current estimate: $\hat{\underline{u}}^{(k)} = \hat{\underline{u}}^{(k-1)} + \gamma \underline{\delta}$
- Update residual: $\underline{r}^{(k)} = \underline{r}^{(k-1)} - \gamma \underline{v}$
- Update residual correlations: $\underline{c}^{(k)} = \underline{c}^{(k-1)} - \gamma \tilde{D}_{\Omega_k}^T \underline{v}$
- Stopping criteria: $k \geqslant K_{max}$ or $\left\|\underline{r}^{(k)}\right\|_2^2 \leqslant \epsilon$

Output

- Estimated SR vector: $\hat{\underline{u}} = W \underline{u}^{(k)}$

Table 4.2: LARS algorithm [36]

[a] The minimum is taken only over positive arguments.

Input

- Reliable samples: $\underline{y} \in \mathbb{R}^{|I^r|}$
- Measurement matrix: $M^r \in \{0,1\}^{|I^r| \times N}$
- Dictionary: $D = \left[\underline{d}_j\right] \in \mathbb{R}^{N \times K_D}$
- Approximation error threshold: ϵ
- Maximum sparsity level: K_{max}

Initialization

- Normalized dictionary: $\tilde{D} = \left[\tilde{\underline{d}}_j\right] = M^r D W$
- Initial solution: $\hat{\underline{u}}^{(0)} = \underline{0}$
- Relaxation parameter: $\kappa = 0.6$
- Threshold control parameter: $\xi = \Phi^{-1}(0.9975 - 0.185\tau - 0.055\tau^2)$
- Iteration counter: $k = 0$

Loop

- Update residual:

$$\underline{r}^{(k)} = \underline{y} - \tilde{D}\underline{u}^{(k)} \tag{4.21}$$

- Estimate standard deviation:

$$\hat{\sigma} = \frac{\kappa}{0.6745} \operatorname{median}\left(\left|\tilde{D}^T \underline{r}^{(k)}\right|\right) \tag{4.22}$$

- Update threshold value: $t = \xi\hat{\sigma}$
- Compute current estimate by applying the thresholding operator:

$$\hat{\underline{u}}^{(k+1)} = \eta_{t_k}(\hat{\underline{u}}^{(k)} + \kappa(\tilde{D}^T \underline{r}^{(k)})) \tag{4.23}$$

- Increment iteration counter: $k = k + 1$
- Stopping criteria: $k \geqslant K_{max}$ or $\left\|\underline{r}^{(k)}\right\|_2^2 \leqslant \epsilon$

Output

- Estimated SR vector: $\hat{\underline{u}} = W\underline{u}^{(k)}$

Table 4.3: IST algorithm [43]

5 Numerical experiments

The experimental tests are carried out with MATLAB. The source code and the test file are included on the enclosed DVD to allow for convenient reproducibility (see Appendix A for detailed information).

5.1 Test signals

All test signals are single-channel 16 bit WAV files with a length of 5 s, which seems to be a reasonable trade-off between an acceptable computation time and still providing enough duration for perceptually-based evaluation. They also get normalized to -1 dBFS. The files are sectioned into the following sets.

- Speech@8kHz - Excerpts of Harvard sentences in American English by male and female speakers[1]
- Speech@16kHz - ITU-T test signals for telecommunication systems in American English, German and Japanese by male and female speakers[2]
- Music@16kHz - Various material[3]

[1] http://www.voiptroubleshooter.com/open_speech/
[2] http://www.itu.int/net/itu-t/sigdb/
[3] see appendant text files on the DVD

Parameter	**Value**
Frame duration	64 ms
Frame overlap	50%
Analysis window w_k^a	rectangular
Synthesis window w_k^s	sine

Table 5.1: Parameter settings for frame-based processing

Parameter	**Value**
AR parameter estimation	Autocorrelation method
AR model order p	$\min(3m+2,50)$
Number of iterations	10

Table 5.2: AR parameter settings

5.2 Frame-based processing

A common frame-based method for local processing of the audio signal is employed for the experiments in Section 5.5.2. The full signal is segmented into overlapping frames of length N, weighted by an analysis window w_k^a. After inpainting each frame independently, the signal is reconstructed using the overlap-add (OLA) method with a synthesis window w_k^s [47]. The parameters are set according to Table 5.1.

5.3 Performance measure

The quality of the results is mathematically judged by means of the signal-to-noise ratio (SNR), defined by

$$\mathrm{SNR}(\underline{s},\underline{\hat{s}}) \stackrel{\mathrm{def}}{=} 10\log_{10}\frac{\|\underline{s}\|_2^2}{\|\underline{s}-\underline{\hat{s}}\|_2^2}. \tag{5.1}$$

5.4 Parameter settings

The parameter settings for the inpainting algorithms are listed in Table 5.2 and Table 5.3.

Parameter	Value
Dictionary size K_D	$2N$
Dictionary weighting window w_k^d	rectangular
Approximation error threshold ϵ	$10^{-6} \cdot \lvert I^r \rvert$
Maximum sparsity level K_{max}	$N/4$

Table 5.3: CS parameter settings

5.5 Experimental results

5.5.1 Randomly drawn frames

The overall performance of the examined inpainting methods is first evaluated on randomly drawn collections of 100 frames per set of test signals, respectively, to facilitate the computation of representative average SNR values shown in Figures 5.1 to 5.3.[4] Each frame is constrained to have a minimum mean energy of -6 dB with respect to the frame with maximum mean energy to avoid silences. The artificially introduced errors are located at random positions.

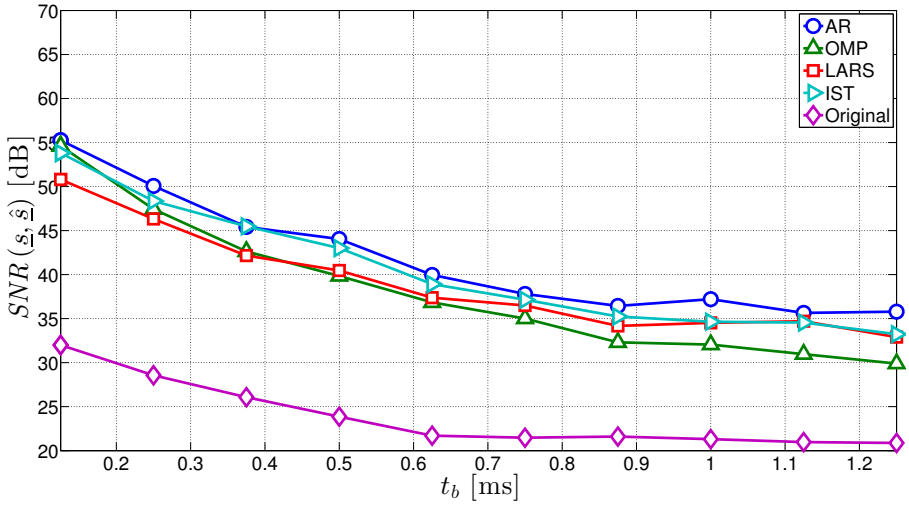
5.5.1.1 Short bursts

The case of short bursts spanning $m \in \{1, \ldots, 10\}$ samples is tested first, with the results pictured in Fig. 5.1. The AR approach yields the best performance for speech and is also only slightly subdued by IST for music. Overall, OMP performs worst and exhibits also the slowest computation speed, with AR being the fastest method. The average SNR improvement reaches from more than 20 dB for a single sample error to about 12 dB for a burst length of 10 samples.

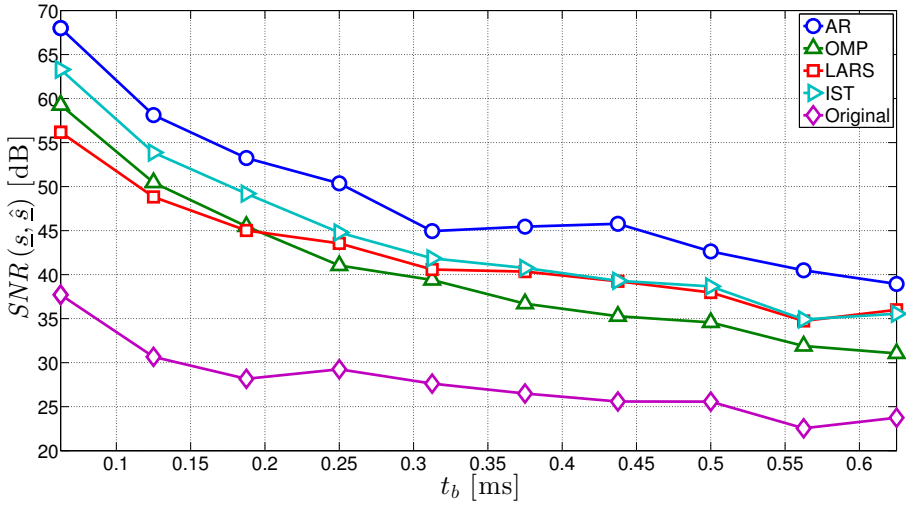
5.5.1.2 Long bursts

To simulate packet loss scenarios in communication systems, we introduce longer bursts in the range of $t_b \in \{1, \ldots, 10\}$ ms. The outcomes are shown in Fig. 5.2. AR outperforms the CS methods for 8 kHz speech and is on a par with LARS and IST for the 16 kHz sets. OMP is again behind by 3-8 dB for shorter intervals and disappoints especially for the longest ones, compared

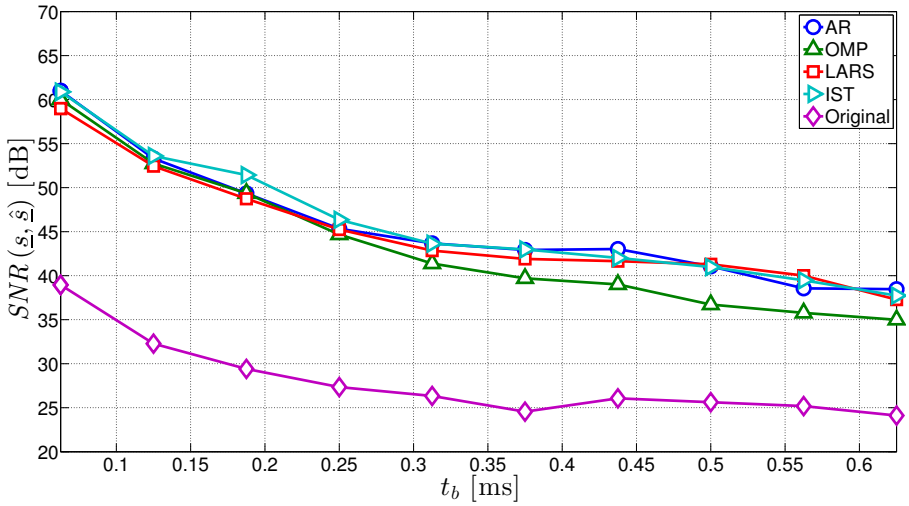
[4] Additionally, the figures contain the average SNR values of the crude signal as a reference.



(a) Speech@8kHz



(b) Speech@16kHz



(c) Music@16kHz

Figure 5.1: Average SNR for short bursts

to the other three algorithms, where it leads to no significant improvement anymore.

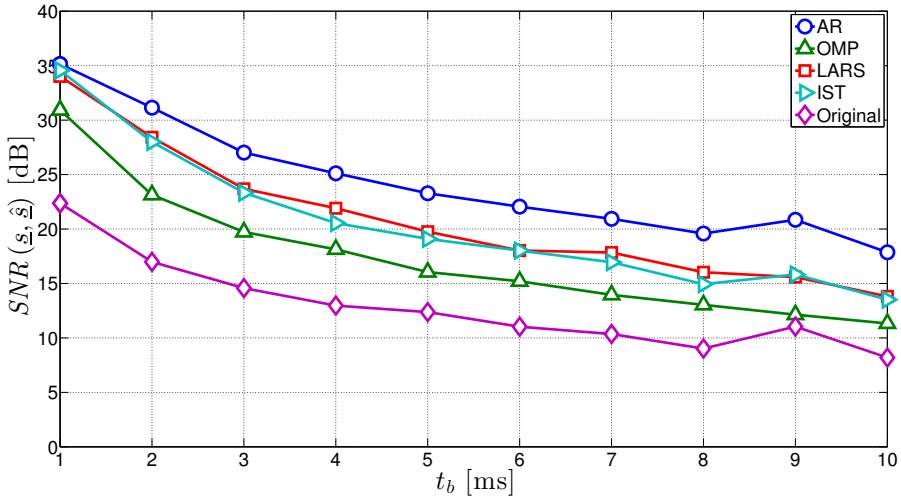
5.5.1.3 Scattered single errors

Impulsive distortions are mimicked by introducing a multitude of randomly scattered single errors. Fig. 5.3 illustrate the results, which are comparable with those of the short bursts. All methods perform better in this case than for long bursts with a equal amount of missing samples. For example, the differences for 8 kHz speech with $m = 80$, corresponding to $t_b = 10$ ms, are on the order of 10 dB. Similar empirical results have been found in [6]. Regarding the CS methods, one reason therefore is the fact that a higher randomness of the measurement matrix generally produces smaller errors in sparse approximations [24]. AR nonetheless outperforms CS by 1-4 dB, whereas the LARS algorithm seems to be the least suitable choice for recovering impulsively distorted signals, falling behind the others by about 1-3 dB for music and even worse with speech.

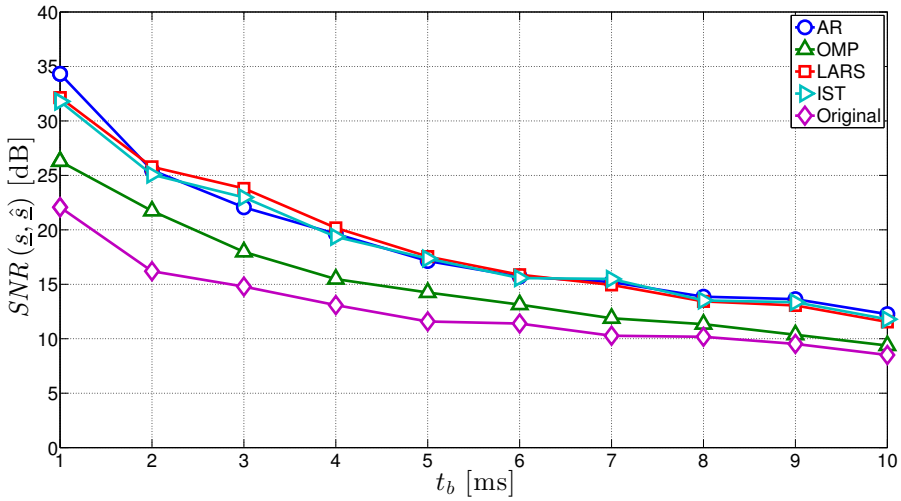
5.5.2 Specific signals

We now apply the inpainting algorithms framewise on 6 selected files from the 16 kHz sets, as described in Section 5.2. Table 5.4 lists the resulting values of SNR improvement for the case of periodically repeated bursts of $t_b = 5$ ms. The repetition time is 50 ms, yielding a total error ratio of 10%. The same signals are tested with randomly scattered single errors, which likewise account for 10% of the whole segment, with the results depicted in Table 5.5.

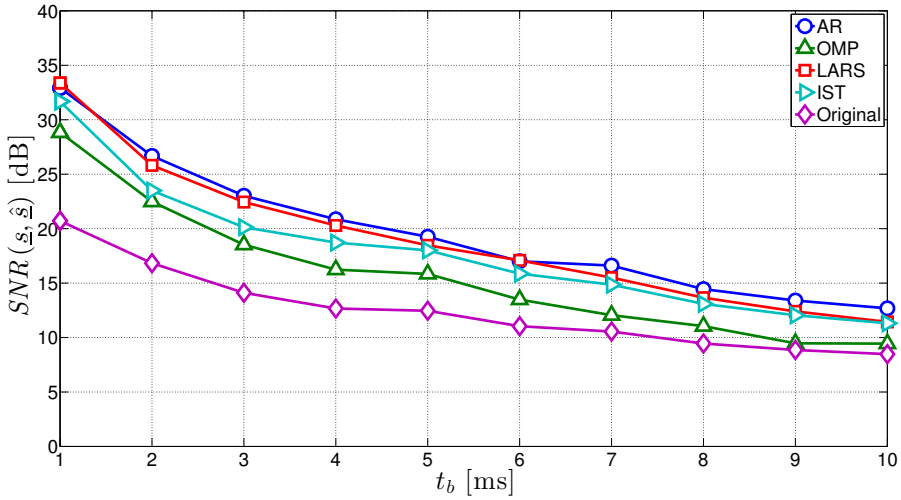
As alluded by the outcomes of the experiments in Section 5.5.1, all methods work best on signals composed predominantly of few harmonic components. Vivid examples are the female singing voice and the acoustic bass, both of which are largely made up of long sustained tones that are preceded by only short transients (e.g. in Fig. 5.4). Sudden transients are more often than not slightly under-approximated, particularly by IST. On the other hand, the performance on percussive sounds with plenty of transient components, plosives in speech and more complex signals, such as complete song fragments, tends to suffer from over-smooth approximations, especially with AR. The CS methods don't seem to prefer smooth solution that much, but in many cases they produce even worse mismatches, all with similar characteristics (see Fig. 5.5).



(a) Speech@8kHz

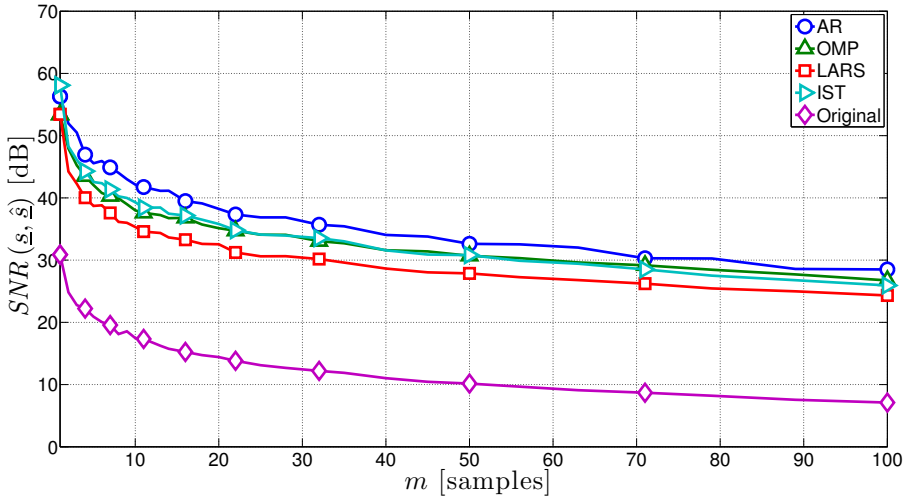


(b) Speech@16kHz

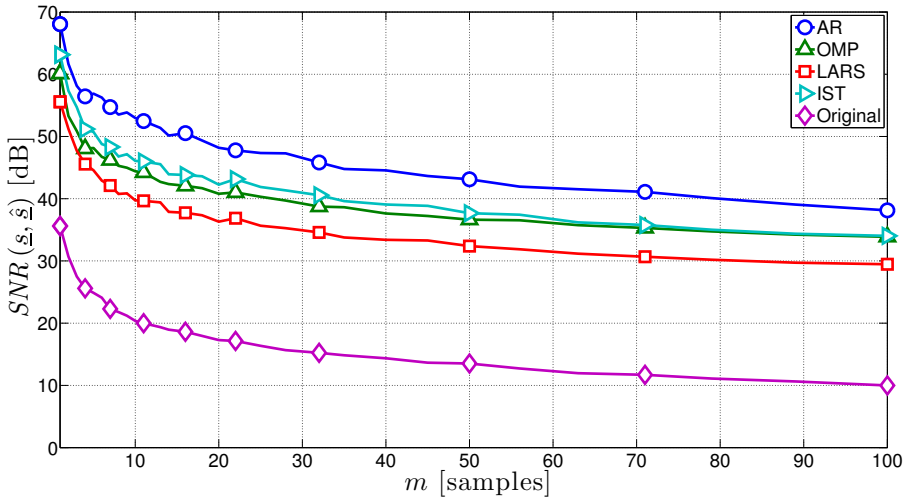


(c) Music@16kHz

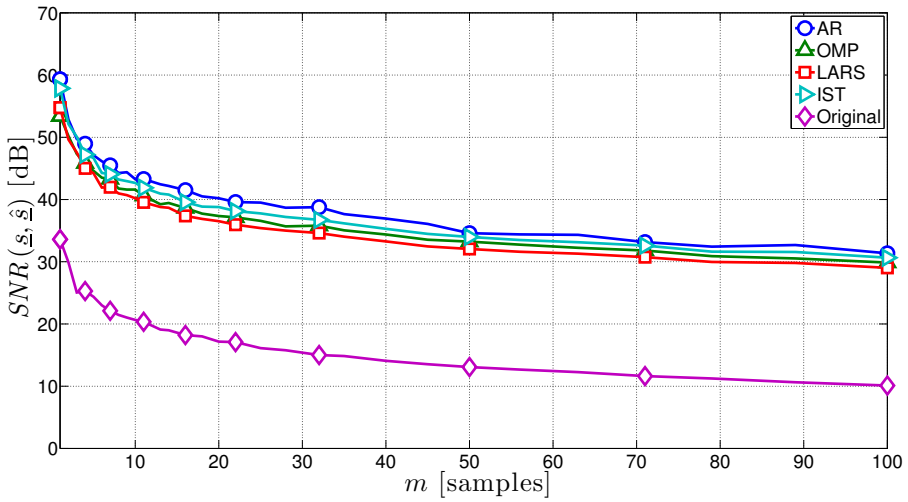
Figure 5.2: Average SNR for long bursts



(a) Speech@8kHz



(b) Speech@16kHz



(c) Music@16kHz

Figure 5.3: Average SNR for m randomly scattered single errors

	AR	**OMP**	**LARS**	**IST**
Female speech	**8.36 dB**	3.49 dB	1.97 dB	5.93 dB
Male speech	4.16 dB	3.07 dB	**5.56 dB**	5.14 dB
Female singing	**15.98 dB**	7.93 dB	7.66 dB	8.84 dB
Drums	**6.21 dB**	3.45 dB	5.15 dB	5.45 dB
Acoustic bass	7.49 dB	5.14 dB	**7.94 dB**	3.64 dB
Pop song	2.71 dB	2.15 dB	**4.37 dB**	4.24 dB

Table 5.4: SNR improvement for 16 kHz signals with periodic 5 ms bursts and a total error ratio of 10%

	AR	**OMP**	**LARS**	**IST**
Female speech	**13.99 dB**	13.57 dB	13.07 dB	13.58 dB
Male speech	**21.03 dB**	15.94 dB	15.00 dB	16.41 dB
Female singing	**21.68 dB**	20.23 dB	19.87 dB	20.47 dB
Drums	**19.42 dB**	16.42 dB	18.42 dB	18.61 dB
Acoustic bass	**23.81 dB**	23.75 dB	23.73 dB	23.78 dB
Pop song	**17.86 dB**	16.30 dB	16.18 dB	16.87 dB

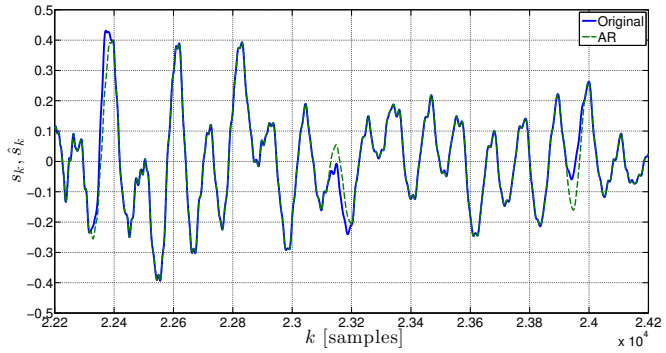
Table 5.5: SNR improvement for 16 kHz signals with 10% scattered single errors

5.5.2.1 Interpolation noise

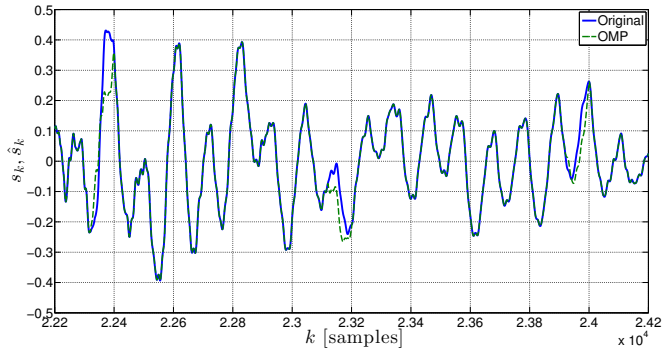
It is furthermore revealing to take a closer look at the behavior of the interpolation noise $s_k - \hat{s}_k$. We therefore investigate the scenario of the acoustic bass signal distorted by randomly scattered single errors, as pictured in Fig. 5.6. Although each method yields about the same amount of SNR improvement (23.73 - 23.81 dB), the interpolation noise of the reconstructed signal for the case of CS approximation (most notably LARS) exhibits a larger quantity of peaks. This is quite obvious when we look at the waveform as well as clearly audible during perceptually-based evaluation. AR produces a more steady sizzling noise that is barely audible in any of the reconstructed test signals.

5.5.2.2 Short-time stationarity

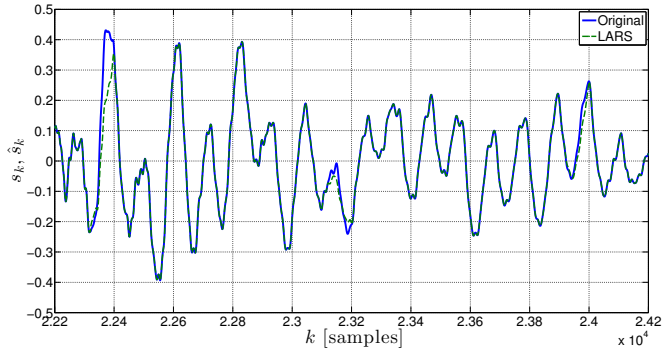
The short-time stationarity of the observed signal is oftentimes indicated to be a crucial factor affecting the performance of the inpainting algorithms on longer bursts [5,6]. To evaluate the influence, we handpick 10 non-stationary frames from each set of test signals and introduce single bursts of $t_b = 10$ ms



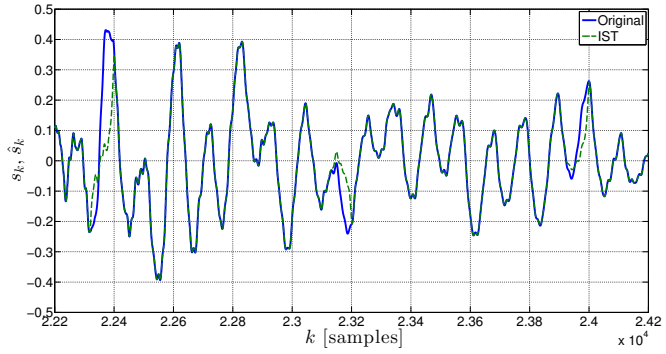
(a) AR



(b) OMP

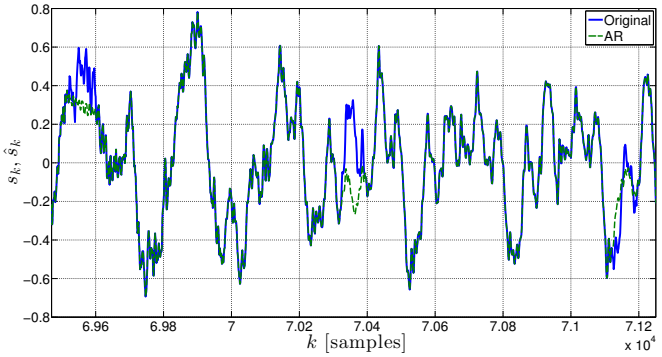


(c) LARS

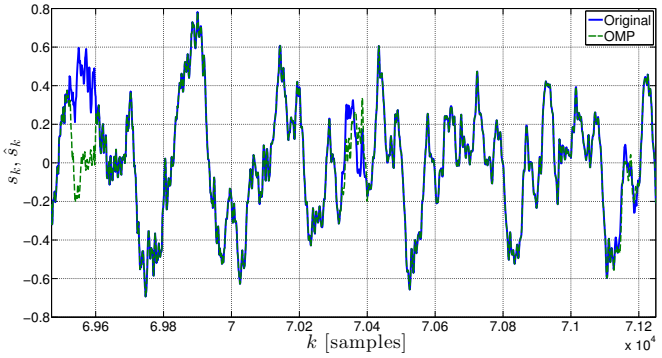


(d) IST

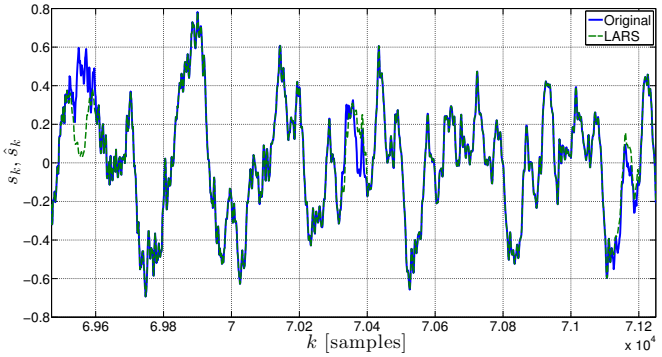
Figure 5.4: Extract of unimpaired and reconstructed 16 kHz acoustic bass signal with 5 ms (80 samples) bursts



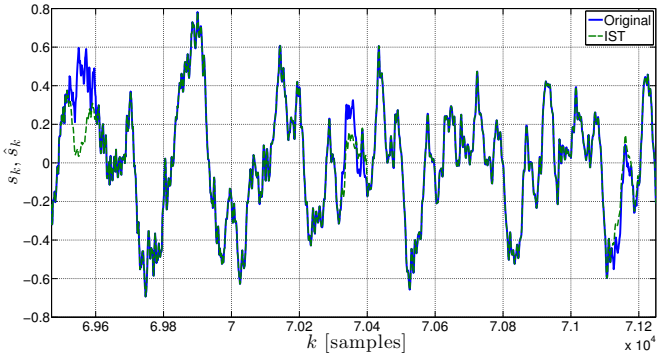
(a) AR



(b) OMP

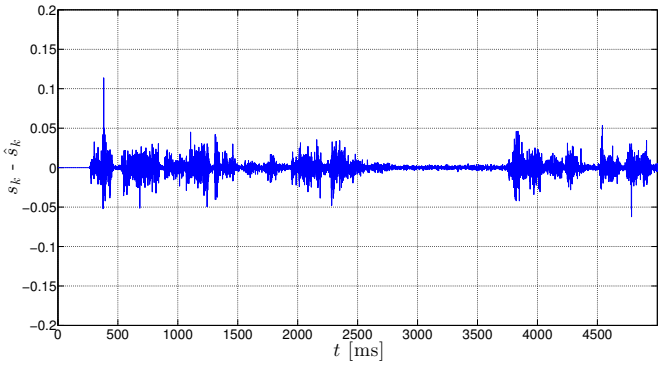


(c) LARS

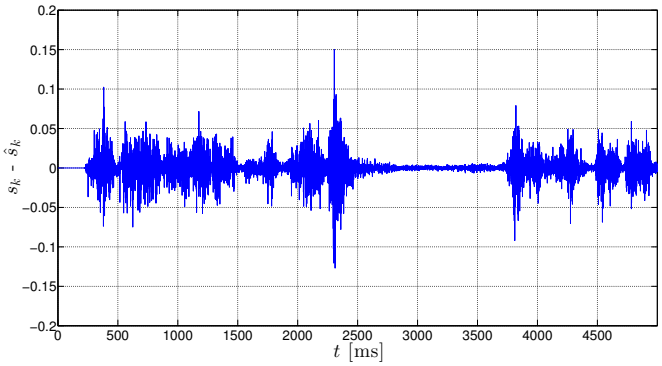


(d) IST

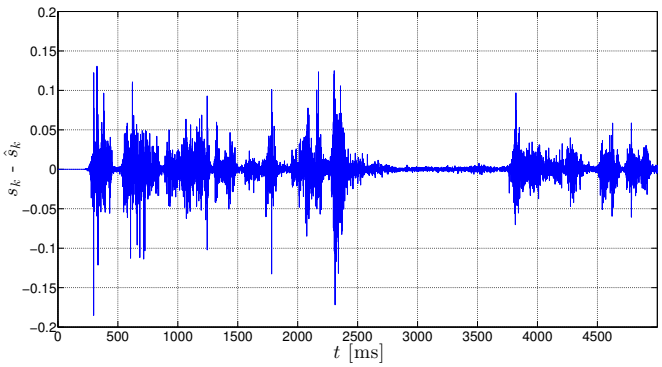
Figure 5.5: Extract of unimpaired and reconstructed 16 kHz pop song fragment with 5 ms (80 samples) bursts



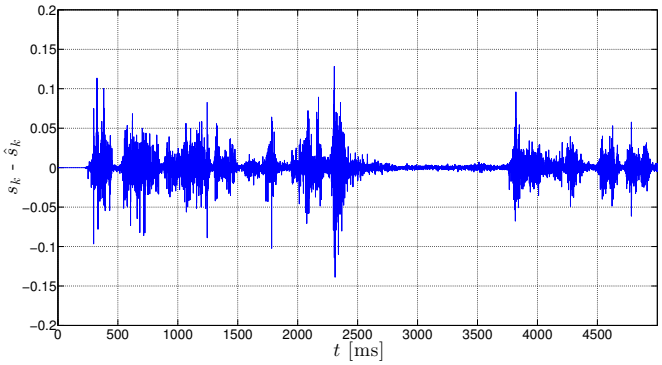
(a) AR



(b) OMP



(c) LARS



(d) IST

Figure 5.6: Interpolation noise of 16 kHz acoustic bass signal with 10% scattered single errors

	AR	**OMP**	**LARS**	**IST**
Speech@8kHz	**6.61 dB**	1.38 dB	3.91 dB	4.36 dB
Speech@16kHz	**6.63 dB**	2.04 dB	4.59 dB	4.21 dB
Music@16kHz	3.60 dB	1.02 dB	**4.14 dB**	3.69 dB

Table 5.6: Average SNR improvement for selected non-stationary fragments with 10 ms bursts

at transition instants (e.g. in Fig. 5.7). The results are listed in Table 5.6. AR performs best for speech nonetheless and only up to 1 dB worse than LARS and IST for music. Regarding the overall improvements, there can only be identified a considerable setback (2-3 dB) for the case of 8 kHz speech, compared to those of the tests with randomly drawn frames (Fig. 5.2).

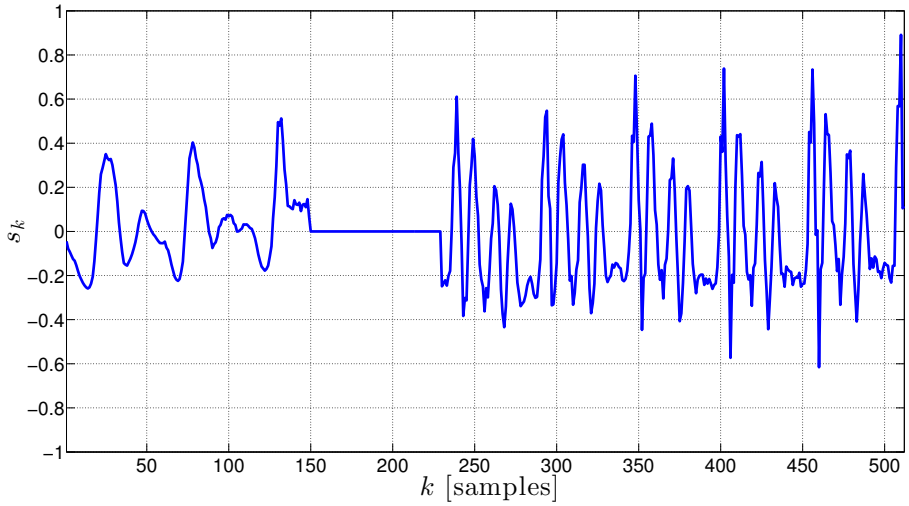


Figure 5.7: Example of non-stationary frame with 10 ms gap

5.5.3 Computational aspects

As we already noted, AR is by far the most efficient technique in terms of processing time due to the employment of the Levinson-Durbin recursion during the parameter estimation stage and the Cholesky decomposition for the calculation of the unknown samples. The CS methods – in their basics forms, as we apply them – all have to execute one or more time-consuming matrix operations at every iteration. In addition to that, AR gets by with 10 steps, while the CS methods require up to $N/4$ iterations to achieve comparable SNR improvements. Table 5.7 shows the average processing times normalized to those

	OMP	**LARS**	**IST**
Bursts	294	135	122
Scattered errors	237	106	94

Table 5.7: Average processing times normalized to those of AR

of the AR method for the experiments in Section 5.5.2.[5] Regarding stability, AR has the disadvantage of being sensitive to gaps embedded in a neighborhood of low-amplitude samples. In this case, the matrix $B(\hat{\underline{a}})$ in (3.22) can become almost singular, which leads to inaccurate results.

[5] All tests have been performed on an Intel Core 2 Duo processor clocked at 2.26 GHz.

6 Conclusion and outlook

In this diploma thesis, we evaluated the performance of two types of state-of-the-art time-domain restoration algorithms for the recovery of impulsively distorted samples and gaps in digital audio signals. After an overview of the broad spectrum of existing approaches, we specifically discussed techniques based on autoregressive modeling and compressive sampling as a basis for a thorough comparison during a series of extensive numerical experiments. Similarly to related inpainting problems in image processing, the examined audio inpainting methods rely on the generation of a signal model from the reliable data to estimate the values of the defective samples, implying that the positions of the errors are known a priori.

In contrast to the autoregressive model, compressive sampling requires the allocation of a proper dictionary to be able to represent audio signals in a sparse domain. For this purpose, we chose a fixed redundant discrete cosine transform dictionary, backed by suggestions in previous related works. From the substantial selection of available sparse approximation algorithms, we considered the greedy Orthogonal Matching Pursuit (OMP) as well as Least Angle Regression (LARS) and Iterative Soft Thresholding (IST), which both belong to the family of l_1-minimization techniques.

The alluded error types have been mimicked by the insertion of missing samples of different lengths and distributions and can be categorized into randomly scattered errors and bursts. We assembled several sets of test signals, containing speech and music fragments, that have been used throughout the experiments.

Somewhat surprisingly, we found that the classical autoregressive modeling approach outperforms the rather recently established compressive sampling methods for almost every contemplated scenario in terms of the SNR as well as the required computation time. It provides a particular advantage for speech, although more transient-heavy and complex signals, such as music, can be recovered more accurately with LARS and IST in some cases. In consideration of the fact that the latter are computationally significantly more intensive, the slight gain in SNR might not be worth the longer processing time in many practical applications. On average, the OMP algorithm has been found to perform worst for most cases, in addition to being the computationally most intensive one. The already known fact that the audio inpainting problem is easier to deal with for randomly scattered errors than for burst has been vividly reassured.

Thus in conclusion, we have clearly demonstrated that the examined inpainting methods from the field of compressive sampling are not able to compete with the classical autoregressive modeling approach if they are implemented without any further algorithmic enhancements. Based on our findings, a number of future directions may be explored. First of all, even though it is convenient and simple to employ a fixed dictionary, a promising trend is the topic of dictionary learning, which has received a lot of attention over the last years [48–51]. In particular, the K-SVD algorithm has been shown to yield auspicious results for this purpose [52]. There is also, obviously, a lot of room for improvement regarding computational efficiency, which might be reduced by using fast transforms for critical matrix multiplications and dictionary handling. This will become particularly important if one attempts to apply compressive sampling methods to data at higher sampling rates, e.g. CD-quality audio.

Appendix A

MATLAB software

The software developed for the experimental tests in Chapter 5 is based on the freely available[1] Audio Inpainting Toolbox [6]. To be able to use it, just copy the main folder *AudioInpainting* to your local hard drive and add it to the MATLAB search path, including all subfolders.

We have included 3 ready-to-run experiments, which can be executed without the need of specifying any parameters. However, it is possible to customize all parameters, as explained in the following sections. The test data listed in Section 5.1 is part of the package as well.

A.1 Burst experiment

```
[MeanSNR,MeanTime] = BurstExperiment(expParam)
```

The function *BurstExperiment* can be applied to reproduce the tests from Sections 5.5.1.1 and 5.5.1.2. A list of random frames is drawn from one set of test signals, each of which gets impurified with bursts of ascending length. After inpainting the defective frames with the examined methods, the average

[1] http://small-project.eu/software-data/

Parameter	Function	Default value
SoundDir	Test data set	Speech@8kHz
NFrames	Number of random frames	5
tFrames	Frame duration [ms]	64
MaxDiffE2m	Max. frame-energy difference [dB]	6
BurstSizes	Range of burst durations [ms]	$\{1, \ldots, 5\}$
NBursts	Number of bursts per frame	1
ARIterations	Fixed number of iterations for AR	10
ErrThreshold	Approximation error threshold factor	10^{-6}
MaxSparsity	Max. sparsity level as a multiple of N	0.25
Dictionary	Dictionary function	DCT
DictWind	Dictionary weighting window	rectangular
DictRedundancy	Dictionary redundancy K_D/N	2
ShowPlot	Plot SNR figure	true

Table A.1: Parameters and default values of *BurstExperiment*

SNR values are computed and can be plotted. The return values include the SNR as well as the average processing times of the different algorithms. The parameters and their default values are listed in Table A.1.

A.2 Scattered experiment

```
[MeanSNR,MeanTime] = ScatteredExperiment(expParam)
```

The function *ScatteredExperiment* can be applied to reproduce the tests from Section 5.5.1.3. A list of random frames is drawn from one set of test signals, each of which gets impurified with an ascending number of randomly scattered single errors. After inpainting the defective frames with the examined methods, the average SNR values are computed and can be plotted. The return values include the SNR as well as the average processing times of the different algorithms. The parameters and their default values are listed in Table A.2.

A.3 Frame-based experiment

```
[AudioData,IMiss,SNRImp,CompTime] = ...
FrameBasedExperiment(expParam)
```

Parameter	**Function**	**Default value**
SoundDir	Test data set	Speech@8kHz
NFrames	Number of random frames	5
tFrames	Frame duration [ms]	64
MaxDiffE2m	Max. frame-energy difference [dB]	6
ErrorSize	Size of errors [samples]	1
NErrors	Range of error numbers	1
ARIterations	Fixed number of iterations for AR	10
ErrThreshold	Approximation error threshold factor	$\{1, \ldots, 10\}$
MaxSparsity	Max. sparsity level as a multiple of N	0.25
Dictionary	Dictionary function	DCT
DictWind	Dictionary weighting window	rectangular
DictRedundancy	Dictionary redundancy K_D/N	2
ShowPlot	Plot SNR figure	true

Table A.2: Parameters and default values of *ScatteredExperiment*

The function *FrameBasedExperiment* can be applied to reproduce the main experiment from Section 5.5.2. The positions of the errors can be specified by a logical vector, which is set to the case of randomly scattered single errors by default. The return values include the SNR, computation times, error position vector and a structure containing the normalized audio data (see Table A.4). The parameters and their default values are listed in Table A.3.

Parameter	**Function**	**Default value**
SoundFile	Test file	Speech@8kHz
tFrames	Frame duration [ms]	64
IMiss	Error position vector	10% scattered
ARIterations	Fixed number of iterations for AR	10
ErrThreshold	Approximation error threshold factor	$\{1, \ldots, 10\}$
MaxSparsity	Maximum sparsity as a multiple of N	0.25
Dictionary	Dictionary function	DCT Dictionary
DictWin	Dictionary weighting window	rectangular
DictRedundancy	Dictionary redundancy K_D/N	2
AnalysisWin	Analysis window function	rectangular
SynthesisWin	Synthesis window function	sine
FrameOverlap	Overlap factor for OLA	2

Table A.3: Parameters and default values of *FrameBasedExperiment*

Field	**Function**
xClean	Original signal
xDist	Defective signal
xEst	Estimated signals in columns
fs	Sample rate

Table A.4: Returned audio data structure of *FrameBasedExperiment*

Acronyms

AR Autoregressive

ARMA Autoregressive moving-average

CS Compressive sampling

DCT Discrete cosine transform

IIR Infinite impulse response

IST Iterative Soft Thresholding

LARS Least Angle Regression

LASSO Least absolute shrinkage and selection operator

LP Linear programming

LS Least squares

MP Matching Pursuit

OLA Overlap-add

OMP Orthogonal Matching Pursuit

SNR Signal-to-noise ratio

SR Sparse representation

List of Figures

List of Tables

Bibliography

[1] A. J. Viterbi and J. K. Omura, **Principles of digital communication and coding**. Courier Dover Publications, 2009.

[2] R. Veldhuis, **Restoration of lost samples in digital signals**. Prentice-Hall, Inc., 1992.

[3] J. Le Roux, H. Kameoka, N. Ono, A. De Cheveigne, and S. Sagayama, "**Computational auditory induction as a missing-data model-fitting problem with bregman divergence**," *Speech Communication*, vol. 53, no. 5, pp. 658–676, 2011.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "**Image inpainting**," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[5] A. Janssen, R. Veldhuis, and L. Vries, "**Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes**," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 2, pp. 317–330, 1986.

[6] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "**Audio inpainting**," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 922–932, 2012.

[7] J. F. Gemmeke, H. Van Hamme, B. Cranen, and L. Boves, "**Compressive sensing for missing data imputation in noise robust speech recognition**," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 272–287, 2010.

[8] H.-J. Platte and V. Rowedda, "**A burst error concealment method for digital audio tape application**," in *Audio Engineering Society Convention 77*. Audio Engineering Society, 1985.

[9] D. Goodman, G. Lockhart, O. Wasem, and W.-C. Wong, "**Waveform substitution techniques for recovering missing speech segments in packet voice communications**," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 6, pp. 1440–1448, 1986.

[10] A. Janssen and L. Vries, "**Interpolation of band-limited discrete-time signals by minimizing out-of-band energy**," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, vol. 9, pp. 515–518. IEEE, 1984.

[11] C. Perkins, O. Hodson, and V. Hardman, "**A survey of packet loss recovery techniques for streaming audio**," *Network, IEEE*, vol. 12, no. 5, pp. 40–48, 1998.

[12] A. Ito, K. Konno, S. Makino, and M. Suzuki, "**Packet loss concealment for mdct-based audio codec using correlation-based side information**," in *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on*, pp. 612–615. IEEE, 2008.

[13] J. Makhoul, "**Linear prediction: A tutorial review**," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[14] S. Kay, "**Some results in linear interpolation theory**," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, no. 3, pp. 746–749, 1983.

[15] S. Vaseghi and P. Rayner, "**Detection and suppression of impulsive noise in speech communication systems**," *IEE Proceedings I (Communications, Speech and Vision)*, vol. 137, no. 1, pp. 38–46, 1990.

[16] W. Etter, "**Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters**," *Signal Processing, IEEE Transactions on*, vol. 44, no. 5, pp. 1124–1135, 1996.

[17] J. S. Erkelens, **Autoregressive modelling for speech coding: estimation, interpolation and quantisation**. Delft University Press, 1996.

[18] H. Madsen, **Time series analysis**. CRC Press, 2008, vol. 72.

[19] S. J. Godsill and P. J. Rayner, "**Digital audio restoration-a statistical model based approach**," 1998.

[20] G. E. Box, G. M. Jenkins, and G. C. Reinsel, **Time series analysis: forecasting and control**. Wiley. com, 2013.

[21] S. M. Kay and S. L. Marple Jr, "**Spectrum analysis—a modern perspective**," *Proceedings of the IEEE*, vol. 69, no. 11, pp. 1380–1419, 1981.

[22] B. R. Musicus, **Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices**. Citeseer, 1988.

[23] G. H. Golub and C. F. Van Loan, **Matrix computations**. JHU Press, 2012, vol. 3.

[24] D. L. Donoho, "**Compressed sensing**," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[25] E. J. Candès and M. B. Wakin, "**An introduction to compressive sampling**," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, 2008.

[26] M. Elad, **Sparse and redundant representations: from theory to applications in signal and image processing**. Springer, 2010.

[27] M. G. Christensen, S. Jensen, *et al.*, "**On compressed sensing and its application to speech and audio signals**," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pp. 356–360. IEEE, 2009.

[28] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "**Sparse representations in audio and music: from coding to source separation**," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[29] C. Kereliuk and P. Depalle, "**Sparse atomic modeling of audio: A review**," in *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11), Paris, France*, 2011.

[30] D. L. Donoho, M. Elad, and V. N. Temlyakov, "**Stable recovery of sparse overcomplete representations in the presence of noise**," *Information Theory, IEEE Transactions on*, vol. 52, no. 1, pp. 6–18, 2006.

[31] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, "**Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition**," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pp. 40–44. IEEE, 1993.

[32] J. A. Tropp and A. C. Gilbert, "**Signal recovery from random measurements via orthogonal matching pursuit**," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[33] S. G. Mallat and Z. Zhang, "**Matching pursuits with time-frequency dictionaries**," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[34] S. P. Boyd and L. Vandenberghe, **Convex optimization**. Cambridge university press, 2004.

[35] J. A. Tropp, "**Just relax: Convex programming methods for identifying sparse signals in noise**," *Information Theory, IEEE Transactions on*, vol. 52, no. 3, pp. 1030–1051, 2006.

[36] D. L. Donoho and Y. Tsaig, "**Fast solution of l1-norm minimization problems when the solution may be sparse**," *Information Theory, IEEE Transactions on*, vol. 54, no. 11, pp. 4789–4812, 2008.

[37] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "**Least angle regression**," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[38] R. Tibshirani, "**Regression shrinkage and selection via the lasso**," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[39] S. S. Chen, D. L. Donoho, and M. A. Saunders, "**Atomic decomposition by basis pursuit**," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[40] M. R. Osborne, B. Presnell, and B. A. Turlach, "**On the lasso and its dual**," *Journal of Computational and Graphical statistics*, vol. 9, no. 2, pp. 319–337, 2000.

[41] D. L. Donoho, "**De-noising by soft-thresholding**," *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, 1995.

[42] M. Elad, "**Why simple shrinkage is still relevant for redundant representations?**" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5559–5569, 2006.

[43] A. Maleki and D. L. Donoho, "**Optimally tuned iterative reconstruction algorithms for compressed sensing**," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 330–341, 2010.

[44] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "**Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit**," *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 1094–1121, 2012.

[45] I. Selesnick, "**A derivation of the soft-thresholding function**," *Polytechnic Institute of New York University (2009/2010)*, 2009.

[46] I. Drori, "**Fast l1 minimization by iterative thresholding for multidimensional nmr spectroscopy**," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007.

[47] A. V. Oppenheim, R. W. Schafer, J. R. Buck, *et al.*, **Discrete-time signal processing**. Prentice-hall Englewood Cliffs, 1989, vol. 2.

[48] M. Aharon, M. Elad, and A. Bruckstein, "**K-svd: Design of dictionaries for sparse representation**," *Proceedings of SPARS*, vol. 5, pp. 9–12, 2005.

[49] M. G. Jafari and M. D. Plumbley, "**Fast dictionary learning for sparse representations of speech signals**," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 5, pp. 1025–1031, 2011.

[50] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "**Domain adaptive dictionary learning**," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 631–645.

[51] D. Barchiesi, "**Sparse approximation and dictionary learning with applications to audio signals**," Ph.D. dissertation, Queen Mary, University of London, 2013.

[52] V. Mach and R. Ozdobinski, "**Optimizing dictionary learning parameters for solving audio inpainting problem**," *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, vol. 2, no. 1, pp. 39–44, 2013.