TUM School of Computation, Information and Technology
Technische Universität München

Vibrotactile Communication: Signal Compression, Quality Assessment and Enhancement, Actuator Equalization

Andreas Noll, M.Sc.

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr.-Ing. Dr. rer. nat. Holger Boche

Prüfer*innen der Dissertation:
1. Prof. Dr.-Ing. Eckehard Steinbach
2. Prof. Dr.-Ing. habil. Ercan Altinsoy

Die Dissertation wurde am 17.06.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 05.01.2023 angenommen.

Abstract

The research area of haptics aims to achieve the virtualization and digital representation of the human sense of touch, similar to the processes that we have seen for the visual and auditory senses. The incorporation of touch in internet applications allows for the creation of an entirely new level of immersion. There are two main research directions in haptics. The kinesthetic sense refers to the forces and torques acting on our bodies and has received the most attention so far in terms of signal acquisition devices, compression algorithms and display devices. The other aspect of human haptic perception, the so-called tactile modality, deals with all the impressions from surface material interactions. As such, it can be subdivided into the five categories of macroscopic roughness, microscopic roughness, softness, friction, and warmth.

In tactile research, so-called vibrotactile signals have received the most attention so far. These signals resemble the skin vibrations that are ellicited when interacting with a textured surface. Interactions include, for example, tapping or sliding. Thus, vibrotactile signals essentially capture microscopic roughness and friction surface properties. In this research field, devices for signal acquisition and display already exist. The existence of such devices and methods enables the next stage of development, which is to lift the existing simple state-of-the-art frameworks of signal recording, transmission and recreation to new heights in terms of quality and fidelity.

To achieve this next stage in sophistication for the vibrotactile domain, we introduce four main methods that form a signal processing pipeline between existing signal acquisition and display solutions. First, we present two lossy compression schemes for data reduction of the acquired vibrotactile signals. These two codecs build upon each other. The first is crafted to compress single-channel vibrotactile signals. It uses findings on human perceptual limitations to reduce the data requirement of vibrotactile signals while maximizing perceptual signal quality. Then, the second codec extends the developed framework for multi-channel vibrotactile signals. By leveraging inter-channel redundancies, the multi-channel codec achieves substantially higher compression while maintaining high perceptual quality.

After developing the codecs to compress vibrotactile signals, we present methods to evaluate the perceptual quality of these compressed signals. First, we design an experiment to acquire perceptual quality ratings with human assessors. The method takes into consideration many perceptual and psychological aspects to produce highly reliable results. We then present two computable perceptual quality metrics that predict the experimentally-measured ratings from signal data.

Next, we shift focus onto the decoder side, where we develop a neural network-based quality enhancement method. By training a recurrent neural network to learn the relationship between a compressed signal and its original counterpart, we use the trained model to map other compressed signals onto enhanced signals that are closer to their original. Through the inclusion of side information and signal preprocessing techniques, we increase the enhancement performance and tailor the method to vibrotactile signals and codecs.

Finally, we investigate the behavior of vibrotactile actuators used to display signals to a human user. When the transduction from digital signal to real vibration takes place, unavoidable distortions are introduced. To counter this, we develop a framework with an adaptive filter to equalize the actuator. By choosing a novel nonlinear adaptive filter model, we reduce distortions more efficiently than with previously employed filter models.

Acknowledgement

In order to conduct the work presented in this dissertation, I was fortunate enough to be a member of the academic staff at the Chair of Media Technology (LMT) at the Technical University of Munich. Many people have supported me, personally as well as professionally, during the past years and have helped me to achieve this.

First of all, I would like to express my deep gratitude and appreciation to my supervisor Prof. Eckehard Steinbach for providing the excellent and interesting topic as well as the opportunity to conduct research in his group. I am grateful of the trust he placed in me and my abilities and his guidance that helped me to better develop my research skills. A lasting impression was left on me by his outstanding ability to take his time for discussion and answering all of my questions promptly despite his busy schedule.

In addition, I would like to thank Prof. Ercan Altinsoy for happily agreeing to become the second examiner of this dissertation and Prof. Holger Boche for agreeing to chair the examination committee.

I feel incredibly lucky to have met my former and current colleagues at LMT. The team spirit at this chair is truly unrivaled – I was fulfilled both professionally and personally, having received a large amount of support and greatly enjoying my time here respectively. All my colleagues at LMT contributed to making this a great time in my life, however, I want to highlight some colleagues in particular. First, I would like to thank Basak Gülecyüz and Dr. Christoph Bachhuber, who mainly shared the office with me for their advice, topic discussions and just generally brightening my days with lots of fun. Additionally, I would like to thank Dr. Tamay Aykut, Edwin Babaians, Dr. Rahul Chaudhari, Kai Cui, Markus Hofbauer, Hasan Furkan Kaynar, Christopher Kuhn, Martin Oelsch, Martin Piccolrovazzi, Dr. Matti Strese, Dr. Xiao Xu, and Alexandra Zayets for their help, answering questions, inspiring interesting discussions over coffee and creating a great work atmosphere. My deep gratitude goes also to the colleagues for the administrative side of the chair, namely Marta Giunta, Simon Krapf, Dr. Martin Maier, and Etienne Mayer for their smooth running of all operations and making no one forget to have fun at the same time.

I also want to thank the students that I was lucky to supervise for their theses, in particular Christian-Daniel Curiac, Ayten Gürbüz, and Lars Nockenberg. Many thanks also go to Dr. Evelyn Muschter and Prof. Shu-Chen Li from TU Dresden for the great collaborative work together and sparking fascinating interdisciplinary ideas.

I would like to thank my family, especially my parents, brother, and sister who always believed in my abilities, pushed me and supported me. Finally, many thanks go out to many of my wonderful friends, particularly Reggie and Cory for providing lots of cheering up during the lockdown times and Benedikt and Lasse for always being there when I need them.

Andreas Noll, Munich, June 15, 2022

Funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden.

Contents

Abbreviations

Acronyms

AC	arithmetic coding . . . 16, 43, 45, 55, 95
ATH	absolute threshold of hearing . . . 11, 16, 38
ATV	absolute threshold of vibration . . . 11 f., 17, 20, 38–41, 53, 60, 67, 96 f.
AWGN	additive white Gaussian noise . . . 86
BiLSTM	bidirectional long short-term memory . . . 22, 73 f., 78
BiRNN	bidirectional recurrent neural network . . . 22, 75, 78
BMF	benchmark model filter . . . 86 f.
BVF	bilinear Volterra filter . . . 83–87, 90, 92
CDF	Cohen-Daubechies-Feauveau . . . 35, 37
CNN	convolutional neural network . . . 98
CR	compression ratio . . . 18, 29, 44 f., 51 ff., 55 ff., 59 ff., 63 ff., 67 f., 70, 74–77
CS	compressed signal 13, 18–21, 23, 29, 44 f., 53, 55, 57, 59 ff., 64 f., 67 f., 73, 75 ff.
CTR	compressed-then-reference . . . 59 f.
DB	Daubechies . . . 36, 103 f., 106
DCT	discrete cosine transform . . . 14, 17, 36, 40 f., 46, 67
DFT	discrete fourier transform . . . 14
DWT	discrete wavelet transform . . . 3 f., 15, 17, 35 ff., 44 ff., 49, 95 f., 102–106
EDS	energy distribution score . . . 37
ERM	eccentric rotating mass . . . 10
ES	enhanced signal . . . 75 f.
EVUQ	embedded values uniform quantizer . . . 41, 50, 53 f.
EZTW	embedded zero-tree coding of wavelet coefficients . . . 15 f.
FC	fully connected . . . 73 f., 79
GFT	Graph Fourier Transform . . . 17
GMTh	global masking threshold . . . 41
GRU	gated recurrent unit . . . 21 f., 78
GUI	graphical user interface . . . 19, 95
HC	hierarchical clustering . . . 47, 95, 97
HP	high-pass . . . 15, 37
i.i.d.	independently, identically distributed . . . 13
ISI	interstimulus interval . . . 59
LDV	laser-doppler vibrometer . . . 7 f., 11
LEA	linear electromagnetic actuator . . . 10, 83
LF	linear filter . . . 27, 83 f., 86 f., 90
LP	low-pass . . . 15, 37
LSB	least significant bit . . . 15
LSTM	long short-term memory . . . 21 f., 78 f.
MAD	median absolute deviation . . . 61
MAE	mean absolute error . . . 73, 80 f., 98
MApp	model application . . . 40 f.

MDS multidimensional scaling . . . 61
MGen model generation . . . 40 f.
MNR mask-to-noise ratio . . . 16, 41
MRA multiresolution analysis . . . 101 ff.
MSB most significant bit . . . 15
MSE mean square error . . . 18 f., 23, 67–70, 80, 84 f., 90, 92
MUSHRA multi-stimulus test with hidden reference and anchor 4, 19 f., 59 f., 71, 96
MVibCode multi-channel vibrotactile codec . . . 45, 51–54, 56 f., 65, 71, 95 ff.
NEA non-electromagnetic actuator . . . 10
NLMS normalized least mean squares . . . 26, 87, 90 f., 97
NN neural network . . . iii, 5, 21 ff., 73–77, 79 ff., 98
NSNR normalized signal-to-noise ratio . . . 69 f.
PC Pearson correlation . . . 20, 67–70
PCM pulse code modulation . . . 13 f., 33 f.
PSD power spectral density . . . 31 ff.
PSNR peak signal-to-noise ratio . . . 18 f., 55 ff., 65, 71
PVC-SLP perceptual vibrotactile codec based on sparse linear prediction 17, 57, 61, 63 ff., 68 ff.
RBF radial basis function . . . 70
REA rotary electromagnetic actuator . . . 10
RefC reference channel . . . 47, 49, 54
ResC residual channel . . . 45, 47, 49 ff., 54
RL residual learning . . . 22 f., 73, 79, 81, 96
RNN recurrent neural network . . . iii, 5, 21 f., 73–76, 78 f., 81, 96 ff.
RoC root channel . . . 45, 47, 49 f., 53
RS reference signal . . . 18–21, 45, 53, 59 ff., 64, 67, 73
RTC reference-then-compressed . . . 59 f.
SMR signal-to-mask ratio . . . 16, 40 f., 46, 49 f., 97
SNR signal-to-noise ratio . . . 16, 19, 41, 51 f., 55 ff., 65, 69, 71, 74 ff.
SOV-IIRF second order Volterra with infinite impulse response filter . 27, 83 f., 86 f., 90
SOVF second order Volterra filter . . . 27, 92
SPIHT set partitioning on hierarchical trees . . . 16, 41 ff., 45, 55, 95
SPQI spectral perceptual quality index . . . 4, 67–71, 95 f., 98
ST-SIM spectral-temporal similarity . . . 20, 65, 67–70
SVM support vector machine . . . 4, 69 ff.
THD total harmonic distortion . . . 23
TOE time order effect . . . 20, 59
VC-PWQ vibrotactile codec with perceptual wavelet quantization . . 34, 37, 43, 45 ff., 50–53, 55, 57, 61, 63 ff., 68–71, 95 ff.
VibroMAF vibrotactile multi-method assessment fusion . . . 69 ff., 95 f., 98
VM vanishing moment . . . 36, 103–106
VMAF video multi-method assessment fusion . . . 69
VPC-DS vibrotactile perceptual codec with DWT and SPIHT 57, 61, 63 ff., 68 ff.
VPL vibrotactile pressure level . . . 38
VQA vibrotactile quality assessment . . . 59 ff., 63, 65, 67, 69, 71, 96, 98
VR virtual reality . . . 1 f.
ZTC zero-tree coding . . . 15 f.

Notation

Numbers, Vectors and Matrices

$\boldsymbol{A}$	Matrix
$\boldsymbol{x}$	Vector
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	Euclidian inner product of $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^d$
$\|\boldsymbol{c}\|$	Length of bitstream $\boldsymbol{c}$
$\|x\|$	Absolute value of x
$\mathrm{Im}(z)$	Imaginary part of $z \in \mathbb{C}$
$\mathrm{Re}(z)$	Real part of $z \in \mathbb{C}$

Signals and Systems

$\overline{x}$	Mean value of $x[n]$
$\tilde{x}[n]$	Mirrored signal of $x[n]$, i.e., $\tilde{x}[n] = x[-n]$
$x(t)$	Continuous signal as a function of time t
$x[n]$	Discrete signal with sample index n
$x * y$	Convolution of $x[n]$ and $y[n]$

Operators and Functions

$\frac{\partial}{\partial x}$	Partial derivative with respect to x
$\hat{w}$	Quantized value of w, i.e., $\hat{w} = \mathcal{Q}(w)$
$\lceil \cdot \rceil$	Ceiling operation
$\lfloor \cdot \rfloor$	Floor operation
$\mathcal{C}(\cdot)$	Coding operator
$\mathcal{Q}(\cdot)$	Quantization operator
$\mathcal{T}(\cdot)$	Transform operator
$\mathrm{E}[\cdot]$	Expectation operator
$\mathrm{round}(\cdot)$	Rounding to the nearest integer
$\mathrm{sgn}(w)$	Sign function, outputs 1 for $w > 0$ and -1 for $w < 0$
$H(X)$	Shannon Entropy of random variable X
$p_X(x)$	Probability of realisation x of the random variable X

Sets

$[a, b)$	The real interval including a and excluding b
$[a, b]$	The real interval including a and b
$\mathbb{B}$	Binary set $\{0, 1\}$
$\mathbb{C}$	Set of complex numbers
$\mathbb{N}$	Set of positive integers
$\mathbb{N}_0$	Set of nonnegative integers
$\mathbb{R}$	Set of real numbers
$A \setminus B$	Set of elements in A that are not part of B

Units

bit/S	bits per sample
dB	decibel
Hz	Hertz
s	seconds

Chapter 1

Introduction

Human curiosity for technological innovations and advancements is virtually limitless. This curiosity has inspired extensive research, giving rise to developments that have truly transformed our world. An area where this fact is seen very evidently is in multimedia via the electronification of human senses. The human visual and auditory senses have been made electronically acquirable, transmittable, and recreatable by cameras and microphones, efficient codecs, and displays and loudspeakers.

With the emergence of the internet, humans also soon started aiming for enabling immersive telepresence experiences over vast distances. As a result, we are now able to communicate immersively with loved ones and collaborate effectively with others without ever being in the same room. Furthermore, we are able to record audio content that is able to create the illusion of being in a concert hall, even when listened to from one's living room. Recently, these technologies have enabled truly remarkable, realistic and immersive experiences like virtual reality (VR).

Even in the most rich VR experiences, however, the sense of touch in the form of haptic sensations is missing, limiting the level of immersion. Humans rely very heavily on haptic impressions of their environment. Being able to deliver such cues in VR would enable groundbreaking new applications. Online shopping could benefit from lower return rates, because customers are able to better review products before purchase. At the workplace or in factories, an entirely new level of collaboration could be enabled, where people work on projects requiring physical interaction from remote locations. Video gaming could benefit from completely new experiences both in solo VR games and when playing together in multiplayer scenarios.

In very recent years, the COVID-19 pandemic has made evident how much humans need physical interaction and how difficult it is to replace such sensations with online telecommunication and similar technologies, regardless of their quality and sophistication. At the same time, it brought to the surface entirely new use cases and applications, where delivering haptic sensations would be highly useful. For example, when thinking about homeschooling that has posed significant stress on students, teachers and parents, it would ease a lot of the problems if physical interaction were possible even when everyone is participating from their homes. The same holds true for other pandemic-related applications, such as its utility in the creation of a home office. All in all, we see that being able to deliver haptic sensations over distance - via the internet, say - could be a game changer for many research fields and everyday situations and enable new exciting applications.

Unlike video or audio, the haptic modality is incredibly diverse and complex. While sound is picked up by the ear and visual input by the eyes, we humans feel haptic sensations with our entire body. The way sensations are perceived differs significantly depending on which part of the body is being acted upon. Adding to that, haptics consists of two very different submodalities, namely the kinesthetic and the tactile modalities [10].

The kinesthetic modality refers to muscle, joints and tendon movement and the forces and torques humans perceive acting upon them. Examples of kinesthetic sensations include hitting a wall with your arm, sitting down on a chair or picking up a small object and feeling its size, shape and weight [11]. Generally, kinesthetic sensations are ever-present on the human body, as even standing on the ground already constitutes a kinesthetic sensation.

The tactile modality covers the perception of object surfaces and textures. Examples of such sensations include sitting down on a couch and feeling the properties of the fabric or when perceiving an artificially created click sensation when pressing virtual buttons on a touchscreen. As they do for kinesthetic sensations, humans are always perceiving tactile cues on their body, be it only the texture of the ground they are walking on. The tactile submodality is then further categorized into five different sensations that only together allow for a realistic perception of textures. These categories are: macroscopic roughness, microscopic roughness, hardness, friction, and warmth [12].

Thus, we see that while it is possible for humans to receive no visual stimulus in a totally dark room (or, similarly, no auditory stimulus in a totally silent room), it is impossible to remove the haptic modality entirely. This ever-presence of haptic stimuli makes it particularly challenging to achieve a high level of realism and immersion with artificial haptic sensations. If we mount a robot arm or some other wearable to the body, humans already have haptic cues from those devices themselves.

Because of the widely different properties of haptic sensations and the novelty of the research field, it is vital to focus on only a few different research directions at once. Today, haptics research is not sophisticated enough to provide the level of immersion that its audio and video counterparts are able to achieve (such as VR). This is also evident from the large amount of work in haptics research so far that has focused primarily on the kinesthetic sense with teleoperation and skill-transfer [10], [13], [14]. By comparison, the tactile sense has received less attention up to now [10]. This has started to change more recently however, as haptics is evolving from being focused on robot teleoperation or very simple tactile cues like clicks to conveying rich, artificial touch sensations over distance [15].

In light of these developments, we choose to focus on the tactile modality, as it as a research area is the least mature. Currently, combining multiple categories of tactile sensation is not yet feasible to a satisfactory extent. In [1] a computer mouse was presented that was purportedly able to display all those subcategories; however its fidelity was clearly not able to meet the requirements for a truly realistic sensation. The particular study could therefore be regarded as a proof-of-concept, but first the categories need to be researched individually to optimize the realism of each sensation before they can be combined efficiently.

We focus on the microscopic roughness and friction properties of surface textures, since these are a vital part of how humans perceive surface materials [11]. The corresponding research domain is called the vibrotactile domain [16]. This is due to the fact that when sliding over a textured surface, vibrations are ellicited in the skin. These vibrations carry microscopic roughness and friction properties of the surface to the mechanoreceptors in the skin where they are translated into nerve signals that can be processed by the human brain to create an impression. Thus, by capturing these vibrations and playing them back to the human user on their skin, we should in theory be able to recreate the texture surface impression [17].

For the vibrotactile modality, we can base our research on recording and display hardware and methods that have been developed previously [16], [5], [18]. As such, it is already possible to record a microscopic roughness signal [5], [19] and play it back to a human user [16]. However, transmitting such raw signals over the internet efficiently can be challenging as described in detail in Sec. 3.2.4. Additionally, the playback quality needs to be optimized for a higher level of realism. We can therefore see the need for methods that solve these challenges and in turn, get this category of the tactile modality to a level of sophistication that makes it fit for widespread adoption and application.

In summary, our goal is to develop methods that can be employed to compress such tactile signals efficiently, assess their quality, enhance their quality after transmission through the internet, and improve the accuracy of display devices to ensure true-to-life recreation of signals. This endeavor is visualized in Fig. 1.1. Here we show the already existing remote touch framework in grey, consisting of signal acquisition, transmission, and display. With our major contributions (highlighted in a different color each), we are lifting this framework to a state where the fidelity of the experience is significantly increased. Our tools resemble the recreation of a process that we have seen already in the early stages of research on audio and video signal transmission.

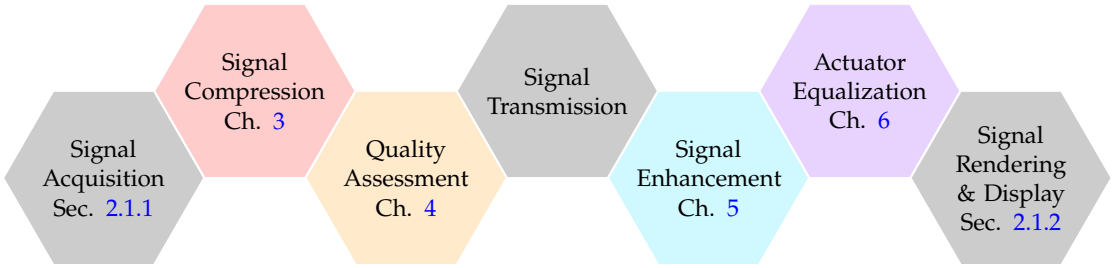


Figure 1.1 Tactile signal processing chain for rich remote touch experiences.

1.1 Main Contributions

Now, we want to describe the contributions of this thesis in detail. These are referred to throughout the thesis by their number.

1. **Analysis of vibrotactile signal properties:** In order to develop efficient signal processing, compression, and enhancement methods, one needs to first understand the properties of vibrotactile signals. For that we conduct an analysis of signals from a publicly available reference dataset. We examine the sampling frequency, dynamic range, frequency content, and data rate. These properties provide us with first insights on how we could use the spectral properties of signals to compress them and allow us to derive hints on potential challenges.

2. **Analysis of human vibrotactile perception:** For perceptual compression and quality assessment, we need to understand how humans perceive vibrotactile signals. This is especially important since our goal is to develop a compression scheme that introduces minimal perceivable distortions to signals. For that, we study the perceptual threshold functions found in literature. We do so both for the absolute threshold of perception as well as for spectral masking thresholds. By combining multiple findings and extracting important features from them, we can later develop perceptual models.

3. **Single-channel vibrotactile codec:** We aim to develop a vibrotactile codec that is able to compress a single-channel vibrotactile signal. Since vibrotactile signals and perception differ significantly to other modalities, we are not able to use existing audio codecs as-is. Instead, we need to develop an optimized compression scheme tailored specifically to vibrotactile signals. The compression should be perceptual, leveraging the effects we found in the previous analysis. With this, we aim to achieve the goal of perceptual transparency, i.e., the codec should not introduce perceivable distortions into signals. For very high compression, the distortions should be kept as small as possible. In other words, this means that we strive to maximize the data reduction capability of the codec while simultaneously maintaining a high level of perceptual signal quality. Simultaneously, the developed codec should be easy to enhance in the future by having a modular structure. It should be executable fast and encode signals efficiently without large delay. Finally, it should be flexible enough to cope with different signals and different application scenarios. We achieve this development endeavor by completing the following main targets:

    a) **Psychohaptic model:** In order to incorporate the perceptual limitations of humans for the codec, we have to develop a computable model of human vibrotactile perception. For that we leverage the features of the perceptual thresholds that we found as part of contribution 2. For the absolute threshold of perception in particular, we develop a parametrized function that can be used to compute this threshold over frequency for any given sampling frequency. For the masking effect, we use the found features for a dynamic model taking into account the signal spectrum at hand, extracting maskers from it and computing their respective masking thresholds.

b) **Discrete wavelet transform on vibrotactile signals:** The discrete wavelet transform (DWT) is a very powerful transform that has found wide application across many signal processing fields [20]. Because of its advantageous properties, we aim to employ it for our codec as decorrelating transform. In order for it to work efficiently, we analyze different wavelets on vibrotactile signals to find the best performing ones.

c) **Combination and joint optimization:** We combine the DWT and the psychohaptic model together with the other necessary elements of a compression system, i.e., block splitter, quantizer and entropy coder. On one hand, we need to ensure that the different elements are compatible and design their inputs and outputs accordingly. On the other hand, we also optimize each individual component, as well as their combination in order to tailor the codec to the properties of vibrotactile signals. Here, we particularly take into account the signal properties from contribution 1 that show that there can be a great variability between signals. The codec therefore needs to be flexible enough to be able to cope with a large variety of signals.

4. **Multi-channel vibrotactile codec:** After the development of the single-channel codec, we seek to extend it for multi-channel signals. When processing many signals jointly, additional gains can be leveraged by exploiting inter-channel redundancies.

   a) **Perceptual channel clustering:** To exploit inter-channel redundancies, we develop a clustering method that dynamically clusters channels when they are similar. For that, we first develop a model with which we are able to quantify the similarity of different channels. We do so by computing a measure for the coding gain that can be expected when we group two signals together. This information is then used to cluster similar channels together and keep dissimilar channels separate. The signals within a cluster are then encoded with residual encoding methods. This means that some signals are encoded fully and serve as reference signals, while the others are encoded by first subtracting a reference signal from them and then merely encoding the remaining difference, the so-called residual. Through this clustering approach, we can greatly improve on the compression performance of the codec.

   b) **Parameter optimization:** In order for the clustering to work as efficiently as possible, we need to optimize the decision parameters. This holds especially true for the threshold that determines the line of decision to group two different signals into a cluster.

5. **Perceptual quality assessment:** In order to evaluate the performance of our compression schemas, we need methods that can measure the perceptual quality of signals. Perceptual quality, in this context, means the similarity of compressed and original signal as human observers would rate it. This can differ significantly from the computed objective quality that usually is based on the mathematical difference between signals. In order to grasp the perceptual quality, we propose two methods, one a human user experiment and the other an automated assessment with computable metrics:

   a) **Quality evaluation experiment:** In order to measure the perceptual similarity of signals, we develop an experimental procedure. We base our method on the well-established multi-stimulus test with hidden reference and anchor (MUSHRA) from the audio domain. Since the MUSHRA is optimized for audio signals and requires expert listeners, we cannot use it directly in the vibrotactile domain. Therefore, we make important adjustments to the method that are based on psychological principles and vibrotactile perceptual effects. Here, the findings from contribution 2 play a crucial role.

   b) **Perceptual quality metrics:** After having measured perceptual quality scores in the human experiment, we strive to develop a method that is able to recreate the results from signal data. This would allow us to skip the time-consuming experiments and assess the perceptual quality automatically. In order to achieve that, we develop a perceptual quality

score called the spectral perceptual quality index (SPQI) which takes the frequency content of signals and weights it by their perceivability. Here, again the found perceptual effects from contribution 2 are used. In a second step, we present a framework based on support vector machines to combine multiple metrics into a final score that is able to more accurately reflect the actual measured scores. The framework is built in a way that makes it easily extendable with more metrics as they are being developed in the future.

6. **Quality enhancement:** As described in contributions 3 and 4, we aim to design the codecs to introduce minimal perceivable distortions. However, especially for very aggressive compression, the compressed signals might contain perceivable coding artifacts that deteriorate the human user experience. Therefore, we aim to develop a method to enhance the compressed signals after they have been decoded on the receiver side. To do so, we employ a recurrent neural network (RNN). By training the neural network with the compressed and original signals, we are able to reverse some of the distortions in the compressed signals. Additionally, we use signal information and processing techniques to showcase the possibilities of optimizing performance and tailoring the method to vibrotactile signals.

7. **Actuator equalization:** To display vibrotactile signals to humans, so-called vibrotactile actuators are used. These actuators come in very different forms and sizes and with different fidelity. However, a common property of all actuators is that they are not able to recreate the vibrotactile signal exactly and there will always be signal distortions when transferring from digital signals to real ones. These distortions also come from the amplifiers and signal processing methods before the actuators. In order to mitigate these distortions, we propose an equalization setup based on adaptive filtering. For that, we develop a nonlinear adaptive filter model inspired by vibrotactile actuators. We show that this adaptive filter model as part of an equalization setup is able to reduce distortions more reliably than previous approaches.

1.2 Thesis Organization

This thesis is structured as follows: Chapter 2 provides further overview into vibrotactile signal processing by providing background information and discussing related work in the areas of vibrotactile research, lossy coding, quality assessment, quality enhancement, and adaptive filter equalization. Chapter 3 describes the developed single-channel and multi-channel codecs after an analysis of signal properties and human perceptual limitations and thus adresses contributions 1, 2, 3, and 4. Chapter 4 describes the experiments and methods developed to assess the performance of the codecs, therefore adressing contribution 5. Chapter 5 describes our method to enhance distorted signals after compression, which is the subject of contribution 6. Chapter 6 contains the equalization methods to enhance the playback quality of vibrotactile actuators, which is contribution 7. Finally, Chapter 7 concludes this thesis and outlines possible future resarch directions.

Parts of the work presented in this thesis have been published in [2], [3], [4], [6], [7], [8], and [9]. The copyright of those publications belongs to the publishers.

Chapter 2

Background and Related Work

2.1 Vibrotactile Haptics

In vibrotactile haptics, research has so far focused mainly on acquiring signals and then recreating them after transmission or storage. In this section, we aim to provide an overview of the current methods and best practices for vibrotactile signal acquisition and display.

2.1.1 Signal Acquisition

In order to acquire vibrotactile signals, we use certain devices integrated into a vibrotactile signal acquisition setup. Depending on the application, one aims either at capturing signals of interactions like tapping or sliding or more rich sensations from sliding motion over different textures. The suitability of acquisition devices for different kind of tasks varies.

2.1.1.1 Acquisition Devices

There are several devices that are able to acquire vibrotactile signals. In the following, we review the four mainly used types:

- **Laser-Doppler Vibrometer:** A laser-doppler vibrometer (LDV) is probably the most precise way of measuring skin vibrations [21]. An LDV is a device that points onto a surface with a laser and measures vibrations of the surface via the doppler effect. Usually, vibrations are measured in terms of the skin displacement in this case.

  The LDV has many advantages on the quality side. The noise in the measurements is very low and the frequency range that can be measured is very high. It also allows for contactless measurements, since it does not have to be mounted on the human skin.

  However, the point that is measured on the skin should be static and therefore it is important that human subjects do not move at all when measuring vibration data. This limitation can already be a quite challenging problem, because it significantly limits the ability of an LDV to be used in applications outside a lab. Since the finger has to refrain from moving, one is not able to record vibrotactile signals from interactions like taps. Therefore, this device is to be used exclusively to capture rich textured surface vibrations and no interaction patterns. Additionally, the LDV is very expensive with costs of tens of thousands, if not hundreds of thousands of dollars. It also requires trained staff to control it. All in all, it is apparently not a good tool for easy-to-use everyday applications.

- **Tribometer:** A tribometer consists of a horizontal textured surface, mounted on some suspension with a force sensor attached at the end [22]. When the user slides his finger over the textured surface in the direction of the force sensor, the sensor measures the friction force between the finger and the surface. Now, depending on the texture of the surface, the friction force will

be slightly oscillating. These oscillations resemble the vibrotactile signal from the vibrations between skin and texture.

The approach is suitable for recording signals with sufficient quality, is fairly simple and low-cost to build. The users can move their finger on the presented material with different speeds. By mounting different materials onto the suspended surface, we are able to record signals for different textured surfaces easily.

However, there are serious limitations to tribometry. First, it is challenging to measure signals for soft materials like foams. Second, the mounting mechanism itself distorts the recorded signals as the vibrations are guided through it. This means that the signal measured by the force sensor does not exactly resemble the vibration at the fingertip. Third, even though the finger can move, it can only do so in one direction. Extending such a device to two dimensions is challenging. Finally, captured signal quality depends highly on the chosen force sensor quality. Also, it is generally hard to avoid variability between measurements, because the sliding of the finger over the surface can never occur with the same speed and uniformity.

- **Piezoelectric Actuator:** These measurement devices are based on the piezoelectric effect. When a piezoelectric material is placed under voltage, it changes its shape. This process works in both ways, so by changing the shape of a piezoelectric material, it induces a voltage. Thus, much like piezoelectric microphones [23], one can record vibrotactile vibrations with such devices.

  In general, these piezoelectric devices have only rarely been used for recording vibrotactile signals. They have found wider use as actuators (see Sec. 2.1.2). Thus, their up- and downsides as recording devices are mostly unknown and therefore should be investigated in the future.

- **Accelerometer**: An accelerometer is able to measure acceleration usually in three dimensions. By attaching the accelerometer to a tooltip or the skin near the fingertip, we are able to capture the vibrations ellicited with the sliding motion over a textured surface as acceleration signal [5]. The three-channel signal is then reduced to one channel with a suitable algorithm or by choosing one channel and discarding the others, e.g., when the accelerometer was placed in a specific controlled way.

  The accelerometer has a wide range of benefits over the other approaches. First, it is a very affordable device and simple to use and delivers signals of sufficient quality [5]. Second, it can be attached to the human finger, which then allows for free-hand movement and recording. Thus, with this device, we are also able to record signals from real-life interaction scenarios. Third, accelerometers are usually very small and can also be bought in very miniature versions, making them a great choice for attaching many of them to the human skin at the same time [24]. Finally, there is minimal distortion between the actual signal at the fingertip and the recorded signal since the accelerometer is placed near the fingertip and has no large body attached to it [21].

  On the downside, for one accelerometers have a higher noise floor than LDVs. Also, with the unconstrained free-hand movement when measuring data, having controlled experimental conditions, e.g., a steady sliding speed, can be challenging. Overall, accelerometers are the most widely used vibrotactile signal acquisition device because they give a good tradeoff between simplicity, ease of use, realism, and quality [15], [5], [24]–[26].

2.1.1.2 Acquisition Setups

In order to conduct measurements of vibrotactile signal data, we need to integrate the devices just described into signal acquisition setups. Only the tribometer is directly usable as acquisition setup.

When using an LDV, the textures for which signals are to be acquired are usually mounted on a rotating drum [19], [21]. Thus, the finger of a human participant is kept still, with the laser focused on the area of the fingertip, while the textured surface slides over the fingertip as the drum rotates.

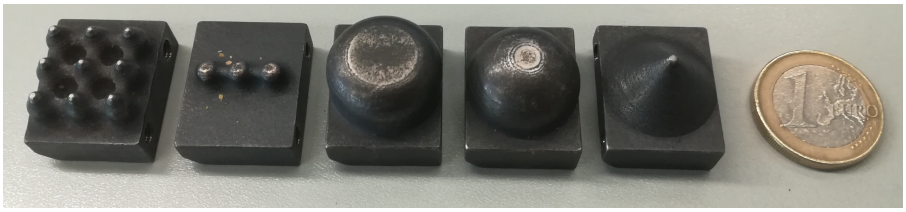
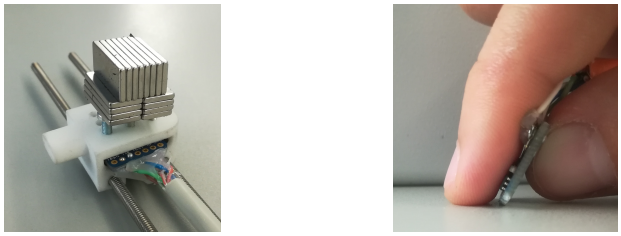


Figure 2.1 Tooltips for attaching to an acceleration sensor to acquire vibrotactile signals by sliding over a surface texture. Adopted from [5] © 2018 IEEE.



(a) **(b)**

Figure 2.2 Measurement setups with acceleration sensor: (a) measurement with a metal tooltip and weights for controlled contact force, (b) measurement with fingertip. Adopted from [5] © 2018 IEEE.

The rotation of the drum through a motor creates vibrations itself, which are almost impossible to eliminate completely. Thus, even though the LDV offers pristine measurement quality, the acquired signal will contain artifacts and noise from the drum rotation.

The piezoelectric actuators are usually used on their own or in arrays. Especially the setup in [27] is exciting in this context, because it is actually built as display setup. However, due to the reversibility of the piezoelectric effect, it can also be used to acquire signals. Thus, we can play back the acquired signals on the same device.

For the accelerometers, so far three kinds of setups have been presented. First, one can attach these sensors to different tooltips. The metal tooltips presented in [5] have very different shapes as shown in Fig. 2.1 The accelerometer is then attached to the tooltip in a setup as shown in Fig. 2.2a. By sliding the setup over a textured surface, the vibrations between surface and tooltip are recorded by the accelerometer. The tooltip shape has a large influence on the properties of the acquired signal, as shown in [5] briefly. We analyze this in detail in Sec. 3.2.3. The second accelerometer-based acquisition setup is for fingertip measurements. Here, the accelerometer is attached to the fingertip close to the point of interaction while it slides over the textured surface, as can be seen in Fig. 2.2b. Finally, for multi-channel signal acquisition setups, accelerometers can be attached to the upper hand and the sides of the fingers [24]. Because the skin is not highly conductive for vibrations from the palm to the upper hand, such a setup is more suitable for simple interaction patterns like taps rather than a rich surface texture sliding interaction.

2.1.1.3 Reference Datasets

We base our work mainly on two signal datasets that have been acquired with two different setups. One dataset contains single-channel signals, while the other consists of multi-channel ones.

The first dataset is named LMT reference dataset in this work. It contains 280 vibrotactile signals acquired with an accelerometer with various tooltips and the fingertip with the measurement procedure described in [5]. In Table 2.1 we summarize the materials, tooltips and speeds for and with which the signals were acquired. Thus, this dataset contains solely rich vibrotactile signals from surface textures and no signals from interactions like tapping.

The second dataset is referenced as CEA reference dataset in this work. It contains 25 vibrotactile signals with 8 channels each. The signals are recorded with the setup described in [27], i.e., with

Materials
aluminium grid
antivib pad
baltic brown
bamboo
cork
felt
foam
polyester pad
rubber

(a)

Tooltips
3x1 round
3x3 round
3x1 spike
3x3 spike
finger
half round
round
spike

(b)

Speed Labels	Speeds (mm/s)
fast	130 – 160
medium	100 – 130
slow	70 – 100
slower	40 – 70
too slow	< 40

(c)

Table 2.1 Materials (a), tooltips (b) and scan speeds (c) for and with which the signals from the LMT reference dataset were recorded in [5].

piezoelectric actuators. These signals are mostly recorded from interactions, e.g., tapping with one or more fingers and sliding over the glass surface. Some signals are recorded from sliding a finger over a fabric that is placed on top of the glass plate. Thus, a few signals also resemble rich textured surfaces. The sampling frequency of these signals was 10 kHz originally. We reduced this frequency to 2.5 kHz by low-pass filtering and downsampling. Thus, we have a more meaningful range of frequency content in the signals, because human vibrotactile perception goes up to 1 kHz at most [28].

2.1.2 Vibrotactile Display

To convey vibrotactile signals to human users, we use vibrotactile actuators that are parts of vibrotactile displays. Numerous actuators of different sizes and properties have been introduced in the past as described in the overview [16]. All these actuators can be sorted into three main categories: linear electromagnetic actuators (LEAs), rotary electromagnetic actuators (REAs), and non-electromagnetic actuators (NEAs).

The LEAs are based on the physical principle that conducting wires transporting an electric charge create a magnetic field. By wrapping a wire into the shape of a coil, one can then create a magnet whose strength and direction depends on the electric current. Then a permanent magnet is placed next to this coil. When the wire is now driven with a voltage that changes over time, the coil will be attracted to or repulsed by the permanent magnet according to the waveform of that voltage. With this, we are able to recreate a vibrotactile signal by translating it into an electric voltage that is then translated into movement. The class of LEAs constist mostly of so-called solenoids and voice coils. The first describes an actuator where the wire is wrapped around a moving piece of ferromagnetic material, whereas the second contains a wire wrapped around a moving permanent magnet. The design of LEAs allows for displaying amplitude and frequency independently from each other.

REAs usually consist of a DC motor that is driven by an oscillating voltage. When a DC motor gets a steady voltage it rotates, so when the voltage oscillates, the motor vibrates. Typically, an eccentric mass is attached to the motor, which gives us a so-called eccentric rotating mass (ERM) motor. When these actuators display vibrotactile signals, the amplitude and frequency are always coupled. This means, for low voltages, the vibration is weak and slow, while for high voltages, the vibration is strong and fast. This effect limits the level of realism when vibrotactile signals are displayed to human users by these actuators. Due to their mechanical symplicity, they nonetheless have seen wide adoption in vibrotactile applications, where the richness of tactile cues is not as important.

Finally, the NEAs are based on the piezoelectric effect. On the upside, these actuators are very responsive and can display arbitrary waveforms. However, since they often require input voltages in the area of hundreds of volts, they are hard to integrate into systems safely and can be quite energy hungry.

All actuators can be used to create vibrotactile displays. This is done either by attaching the actuator to a surface or tool or by combining multiple actuators to display multiple points to the human skin.

The second approach has a signficantly higher level of realism and will be necessary in the future to create truly immersive experiences. Thus, we work on enabling such multi-point setups with appropriate multi-channel signal compression techniques in Chapter 3 as part of contribution 4.

2.2 Human Vibrotactile Perception

Humans perceive vibrotactile impressions on their skin through a variety of mechanoreceptors. This distributed processing of skin vibrations leads to some frequencies being better perceivable for humans than others. Therefore, researchers have aimed at measuring this frequency dependency of vibrotactile signals and establish corresponding threshold functions, as described in the following.

2.2.1 Mechanoreceptors

Humans are believed to have four primary mechanoreceptor cells in their skin that are responsible for capturing skin vibrations. In short, the mechanoreceptors transduce the mechanical vibrations on the skin into electrical nerve signals. Each mechanoreceptor is set to capture different parts of the frequency range. The four mechanoreceptors are named Merkel disk, Meissner corpuscle, Pacinian corpuscle and Ruffini ending [29], [30].

The Merkel disks are responsible for capturing pressure sensations of close-to-zero frequency. As such, they do not convey detailed surface information, but merely the amount of pressure force acting on the skin. In contrast, the Meissner corpuscles and Pacinian corpuscles capture the relevant skin vibrations for the perception of surface textures. The Meissner corpuscle has a peak sensitivity at around 50 Hz, while the Pacinian corpuscle has its sensitivity peak at around 250 Hz [30]. The general sensitivity of the Pacinian corpuscle is higher than that of the Meissner corpuscle. Finally, the Ruffini endings are believed to play an important role in the percepion of shear forces [29], [30].

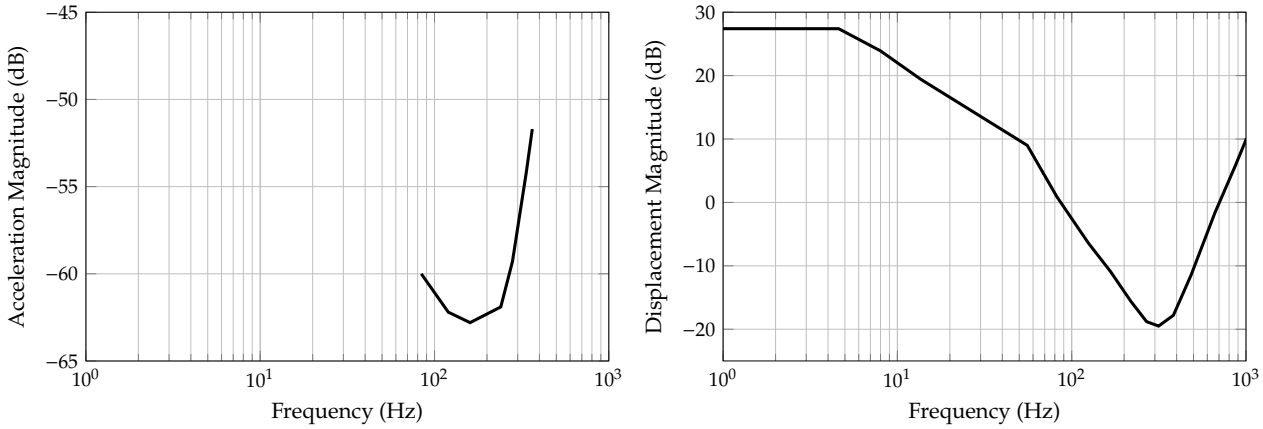
The distribution of mechanoreceptors is highly non-uniform on the human body. In general, highest mechanoreceptor density and therefore sensitivity is found on the fingers and hand, while it is lowest on the back [31].

2.2.2 Absolute Threshold of Vibration

The absolute threshold of vibration (ATV) resembles a frequency-dependent function that dictates how perceivable sinusoidal signals are at any given frequency without the presence of any other signals. This is analog to the audio domain, where the absolute threshold of hearing (ATH) exists [32]. The ATV is different for every human, so when modeling this threshold, an average function has to be used. Thus, the ATV is defined as the signal level, where 50% of human users would be able to perceive a sinusoidal signal.

So far, the ATV has usually been measured by delivering sinusoidal stimuli of increasing or decreasing amplitude to human users [33]. In the first scenario, a stimulus starts at very low amplitude that increases until the human users are able to perceive it. This is indicated by the user through a button. When that button is pressed, the stimulus amplitude decreases again. By continuously lowering the step size of those increases or decreases, one is able to obtain an arbitrarily accurate estimate of the ATV at a particular frequency. The experiment is then repeated for multiple frequencies and also in the converse way of a stimulus starting very high and decreasing, with the push of the button increasing the amplitude again.

There are two mainly used ways to measure the ATV, which differ in the physical definition of the underlying signals. For one, one can measure the ATV in terms of displacement required at a certain frequency. Otherwise, the ATV can be based on the required acceleration. This often also depends on the kind of measurement device used. When using a LDV, one is most likely to use displacement as a frame of reference for the ATV. On the other hand, when using an accelerometer,



(a) **(b)**

Figure 2.3 (a) absolute threshold of vibration measured in terms of acceleration reproduced from [33], (b) absolute threshold of vibration measured in terms of displacement reproduced from [35].

an acceleration-based ATV will most probably be the preferred choice as the acceleration signals are directly obtainable.

So far, ATVs have mostly been measured for displacement signals, while there are fewer measurements of ATVs for acceleration. The ATV has usually been measured for the palm of the hand, a hand gripping a tool or the fingertip. We exemplify the shape of the ATV by reproducing two measurements visually, one for acceleration and one for displacement. For that, we choose two measurement that were presented in dB, since the unit of measure is widely used in perceptual models, e.g., of audio codecs [34]. We reproduce the acceleration ATV measurement from [33] that was measured for the human hand gripping a tool in Fig. 2.3a. A very complete displacement ATV measurement for the human fingertip was presented in [35] that we reproduce in Fig. 2.3b. Additional displacement ATV measurements have been presented in [28], [36] with a very comprehensive overview in [28, Fig. 3] of the displacement thresholds measured in [37]–[42]. In [28] additionally acceleration ATVs were measured and in [43] an acceleration ATV was calculated.

All the measured ATV curves have the same qualitative shape. There is a minimum at frequencies between 150 and 430 Hz. Towards higher frequencies the threshold increases rapidly, leading to a maximum frequency of roughly 1000 Hz that humans are able to perceive. Towards lower frequencies, the ATV is also increasing.

However, despite the consensus on the qualitative shape of the ATV curve, there is no consensus on its quantitative feats. The frequency, where the minimum of the curve occurs is around 180 Hz in [33] and some measurements of [28], around 250 Hz in [36]–[42] and some other measurements of [28], around 300 Hz in [35], and as high as 430 Hz in [43, Fig. 2b]. Additionally, the difference in dB between the minimum value of the ATV curve and the value at zero frequency is not consistent either. Some publications indicate a difference of only 10 to 30 dB [28], [36], [37], [40], [42], [43], while others have differences of 50 to 80 dB [28], [35], [38], [39], [41].

2.2.3 Vibratory Masking

In human perception, the phenomenon of masking refers to the effect that a stimulus is rendered imperceivable by a stronger nearby stimulus. In this case, "nearby" refers to a stimulus of similar frequency (frequency masking) or close in time (temporal masking) or spatially close to the location of the weaker stimulus (spatial masking). For audio, auditory masking models enabled the MP3 codec to achieve high performance, leading to its wide-spread adoption [34].

In the vibrotactile domain, we also have masking occuring as temporal masking [44], spatial masking [36] and frequency masking [33], [36]. In general, masking phenomena for vibrotactile

signals have not been studied as thoroughly as for other domains so far. Out of the three masking possibilities, frequency masking has been studied the most [33]. As we find later in Sec. 3.2.3, a large amount of signal information can be gathered from signal spectra, which puts an emphasis on leveraging frequency masking in human perceptual models. Also, many relevant early codecs in other domains, e.g., MP3, utilized only frequency masking in their perceptual models at first [34]. Thus, in this work we focus on frequency masking and develop a perceptual model based on it, since we can base its properties on enough research to have fair confidence in it. The development of our frequency masking model is conducted in Sec. 3.3.3.

2.3 Lossy Compression

In order to develop a well-performing codec for vibrotactile signals, we first need to understand how codecs work in general. Codec is short for "code-decode". Therefore, a codec consists of an encoder, which processes the input signal into a compressed bitstream, and a decoder, whose purpose it is to reconstruct the signal waveform from the compact bitstream representation. The bitstream representation is often also called coded representation, since it describes the signal in a coded form that is not human-readable.

2.3.1 Lossless Compression Limitations for Vibrotactile Signals

Lossless compression aims at reducing the amount of data needed to represent a signal by exploiting statistical redundancies. As such, lossless codecs compress signals without introducing any distortion, i.e., when decoding the compressed signal (CS) we receive a signal that is mathematically identical to the original signal. Thus, it is clear that lossless compression is desired because it allows for high fidelity display of signals.

However, there are significant limitations to lossless compression techniques, which have prevented it from being adopted widely in the past. Only very recently, lossless audio has seen wide-spread adoption for consumers as internet data rates have gotten sufficiently high [45]. Lossless video applications are still not feasible over the internet.

The main limitation of lossless compression techniques is the meager compression capability. To illustrate this, we want to analyze the compression capability of lossless techniques on our vibrotactile signal data from the LMT reference dataset. We assume that the vibrotactile signal samples over time are realizations of independently, identically distributed (i.i.d.) random variables. The minimum data rate required to represent a signal can be found by computing the Shannon entropy

$$H(X) = -\sum_{i=1}^{m} p_X(x_i) \log_2(p_X(x_i)) \text{ bit/S}, \tag{2.1}$$

for a discrete random variable X, where p_X is the probability distribution of X and the convention $0 \cdot \log_2(0) = 0$ is used. This holds under the assumption that each signal sample is encoded individually.

We now assume again that our original data is encoded with pulse code modulation (PCM) with $B = 16\,\text{bit/S}$. Every signal sample is a realization of our random variable X, which now has $2^{16} = 65536$ possible values. We compute the entropy of X for each signal individually and average over all signals to receive $H_{mean}(X) \approx 10.03\,\text{bit/S}$. The minimum entropy occurring in the dataset is $H_{min}(X) \approx 7.81\,\text{bit/S}$. Overall, this means that we would be able to compress by a factor of approximately 1.59 on average and of approximately 2.05 at best for this dataset. By performing joint or conditional encoding, we could in theory descrease the encoded rate, but a substantially higher data rate reduction is not to be expected. Overall, the data rate reduction is not sufficient to transmit a large number of channels reliably.

2.3.2 Compression System Elements

Since lossless compression cannot meet the demands for high data compression, we resort to lossy data compression techniques. As described in [20, Sec. 1.3], a general lossy compression system can be regarded as a mapping $\boldsymbol{c} = M(\boldsymbol{x})$ of the input signal $\boldsymbol{x} \in \mathbb{R}^N$ into the bitstream $\boldsymbol{c} \in \mathbb{B}^{\|\boldsymbol{c}\|}$. In this case, the signals are represented as vectors, since they have finite length. This mapping is practically a look-up table with NB entries, where N is the signal length and B is the number of bits per sample. For our vibrotactile signals we have $B = 16\,\text{bit/S}$ in the reference datasets. This mapping then is essentially a vector quantizer, which maps the signal vector onto a quantized vector that requires fewer bits to represent because it is part of a limited set of possible output vectors. However, a vector quantizer grows exponentially in complexity with the signal length and therefore this approach is not a viable option.

To solve the problem of vector quantizer complexity, codecs are further decomposed into substeps [20], [34]. First, a decorrelating transforms maps the input signal onto transform coefficients as $\boldsymbol{y} = \mathcal{T}(\boldsymbol{x})$. This transform serves mainly two purposes. First, the signal samples in one block that are normally highly correlated are then decorrelated. Through this, redundancies are automatically reduced and we can quantize each coefficient individually with scalar quantizers. Second, the transform compacts the signal energy into fewer coefficients. Therefore, we are able to discard or quantize coarsely some of the lower-energy coefficients without losing a lot of information.

After the transform, we quantize the transform coefficients as $\boldsymbol{q} = Q(\boldsymbol{y})$. The quantizer Q is now a scalar quantizer. Because the quantization uses generally fewer bits than the original PCM encoding, the entropy of $\boldsymbol{q}$ is lower than that of $\boldsymbol{y}$. Therefore, the data rate can be further reduced with entropy coding methods, which resembles another mapping $\boldsymbol{c} = C(\boldsymbol{q})$. The vector $\boldsymbol{c}$ is then the bitstream that can be transmitted efficiently.

At the decoder side all the steps are then inverted, although only $\mathcal{T}$ and C are invertible. For Q we only have the approximate inverse operator $\overline{Q^{-1}}$. The reconstructed signal $\hat{\boldsymbol{x}}$ is a distorted version of $\boldsymbol{x}$. In the following, we present the relevant transforms, quantizer structures and entropy coding methods that we need to develop our vibrotactile codec.

2.3.2.1 Block-based Processing

Usually, in codecs signals are processed in a block-wise manner as a standard practice for many compression methods [20], [34]. This is because all operations are then able to operate on a certain defined length of signal portion and we are able to use powerful block transform methods. As such a block splitter is employed in the beginning of the encoder that is responsible for separating the input signal into consecutive blocks of equal length named block length L_{block}.

The operation of the block splitter varies slightly depending on whether the application is offline or online. In an offline application, since the entire signal is available at once, the block splitter simply separates the signal into blocks. If the total length of the signal is not a multiple of the block length, we pad with zeros at the end. In case of an online application, the block splitter must contain a buffer in which the incoming signal samples are stored in the order in which they arrive. Once the given block length has been reached, the buffered signal content is forwarded at once to the subsequent processing methods and the buffer is reset to store the new incoming signal samples.

2.3.2.2 Decorrelating Transforms

There are essentially two transforms that have found wide-spread adoption in lossy codecs. First, the discrete cosine transform (DCT) has been used in modified form in MP3 [34]. It transforms a signal block by taking the discrete fourier transform (DFT) of the mirrored signal from the block. As such, the received coefficients resembly solely frequencies. It has excellent decorrelation and energy compaction capabilities for highly correlated data. On the downside, temporal information in the signal block is lost after transform. Thus, it is not easy to asses how small changes of coefficients like

quantization will affect the original signal. It also does not have a structure between coefficients as every coefficient is computed independently from each other.

The discrete wavelet transform (DWT) has found wide adoption in modern lossy codecs, most prominently in JPEG2000 [20]. An additional reason for this is that the DWT can be applied with an enormous variety of different wavelets with widely different properties. Therefore, it is an extremely versatile tool that can be fitted to work best with different kinds of signal data and therefore we can optimize the design of our codec for vibrotactile signals. Finally, with very efficient lifting implementations, the DWT is easily computable, which has contributed to its wide-spread adoption.

The DWT can be performed numerous times to achieve a separation into multiple frequency bands. The number of times a DWT is applied to a signal block is called level l_{DWT}. A DWT with $l_{\mathrm{DWT}} = 1$ splits the signal block into a low-pass (LP) (or approximation) and high-pass (HP) (or detail) band of equal width. The subsequent DWT level is then applied on the LP band. This form of DWT is named dyadic. The maximum possible level $l_{\mathrm{DWT},max}$ of DWT for a block of length L_{block} is defined as

$$l_{\mathrm{DWT},max} = \log_2(L_{block}). \tag{2.2}$$

Through cascaded application of the DWT, the computed coefficients are structured in a tree-like way [46].

2.3.2.3 Scalar Quantizer

A scalar quantizer is the centerpiece of every codec. It quantizes the transform coefficients and is thus responsible for the main part of the data reduction. As described in [20], scalar quantizers are characterized by their quantization intervals $\mathcal{I}_i$ and quantization values $q_i \in \mathcal{I}_i, i = 1, 2, \ldots, Q$. The simplest quantizer is the uniform quantizer, where all $\mathcal{I}_i$ have the same width Δ and the q_i are in the middle of their quantization intervals. The mapping of an input value w onto its respective quantized output $\hat{w}$ is characterized by

$$\hat{w} = \mathrm{sgn}(w) \left\lfloor \frac{|w|}{\Delta} + 0.5 \right\rfloor. \tag{2.3}$$

In this work, we only examine binary uniform quantizers. They are characterized by their number of bits $bits$ and have $Q = 2^{bits}$ possible quantization values.

When quantizing with different number of bits on the same set of coefficients, it can be advantageous to use an embedded uniform quantizer. As defined in [20], this quantizer is characterized by the property that quantization intervals of finer quantizers, i.e., with more bits, are always subsets of quantization intervals of coarser quantizers. Through this, a multi-staged implementation of quantization is possible. The quantization values are different between embedded quantizers of different number of bits.

2.3.2.4 Entropy Coding Methods

The purpose of entropy coding is to exploit the knowledge of signal statistics to code signal values in a more optimal way. For example, take a signal of three different values denoted by the symbols A, B and C. If we code these with the same number of bits, we need 2 bits per symbol. However, if we know that the probability of A occuring is twice as high as that of B and C, then we can code A with 1, B with 01 and C with 00. With this we only need $0.5 \cdot 1 + 0.5 \cdot 2 = 1.5$ bits per symbol.

This example illustrates the concept behind variable length coding like Huffman coding or Colomb coding [20]. Especially Huffman coding is extremely widely known and used. The general idea is assigning shorter codewords to more probable symbols and vice versa. These coding methods produce codes that are always tailored to the particular signal statistics.

In contrast to that, zero-tree coding (ZTC) methods are independent of signal statistics. Instead, they leverage the structural dependencies of transform coefficients and therefore only work with the DWT. This is a significant advantage of the property of the DWT that it preserves temporal

information in its coefficients. ZTC iterates through the wavelet coefficients by bitplanes, starting with all most significant bits (MSBs) of each wavelet coefficient down to all least significant bit (LSB). For the DWT, wavelet coefficient bits of lower frequency bands are highly correlated with respective groups of coefficient bits from higher frequency bands. Thus, if a lower frequency coefficient bit is zero, it is likely that the higher frequency coefficient bits are zero as well and that is coded very efficiently in embedded zero-tree coding of wavelet coefficients (EZTW), which was the first major ZTC method [47].

The basic EZTW method was extended by set partitioning on hierarchical trees (SPIHT) [20], [46]. SPIHT extends the ZTC concept with some extra features that make it more powerful. With it, lossless compression closely up to entropy can be achieved. SPIHT is producing an embedded bitstream, just like EZTW. This means that truncating a coded bitstream at the end, results in a bitstream that is still decodable and only has lower quality.

Arithmetic coding (AC) is an entropy coding method that has been widely used due to its many advantages [20]. AC is based on Elias coding. However, Elias coding requires infinite precision and is therefore impractical. This was solved with AC, which works with finite precision. As to its advantages, for one, it is very powerful in achieving a high level of compression. Furthermore, it is an incremental coding method, which codes an input sequence as it arrives. Therefore, we do not need to store large codebooks and do not have additional delay added by the coding mechanism. At the decoding side, the arithmetic decoder works incremental as well, so it can decode the coded bitstream as it arrives on the other side.

The AC operates by dividing a coding interval $[0, 1]$ according to an incoming sequence of bits. The probablities $p(0)$ and $p(1)$ occuring in the bit sequence have to be known. Then, if a 0 comes in, the interval $[0, 1]$ is reduced to $[0, p(0)]$, otherwise to $[p(0), 1]$. This subdivision is continued iteratively, where the interval becomes smaller with each incoming bit. The achieved compression is higher, the larger the inequality of $p(0)$ to $p(1)$ is. Additionally, the estimation accuracy of the probabilities is an important factor for the efficiency of the AC.

2.3.3 Perceptual Compression

The primary goal in any lossy codec is to have it be perceptually transparent. This means, even though the coded signals are distorted and information is discarded, this should not be perceivable to humans. The way this is achieved is by adapting the quantization appropriately.

Intuitively, the quantizer should be configured to quantize more finely in frequency ranges where humans perceive well and conversely, quantize more coarsely at frequencies where they do not. This can primarily be done in two different ways. First, one can have the quantizer be fixed and quantize all transform coefficients equally. Then, by weighting the transform coefficients, e.g., with human sensitivity functions [43], a perceptual quantization can be achieved. In most audio codecs and especially MP3, the converse approach was chosen. Here, the transform coefficients are not weighted and instead the quantizer is adapted perceptually.

Of particular interest for vibrotactile codecs is the approach that was utilized in the first two versions of MP3 [34]. It works by iteratively allocating bits to different frequency bands and therefore making the quantization finer in the appropriate frequency ranges. First, a perceptual model determines a threshold function over frequency by combining the ATH with masking thresholds. Then the signal energy in each frequency band (MP3 splits the input signal blocks into 32 frequency bands with a filter bank) is compared to this threshold. In particular, the so-called signal-to-mask ratio (SMR) is calculated, i.e., the ratio of signal energy to threshold energy in each band. Then, the signal-to-noise ratio (SNR) is calculated in each frequency band with the current bit allocation of the quantizer. Finally, the mask-to-noise ratio (MNR) is calculated as MNR = SNR − SMR. Then, one bit is allocated to the frequency band with the lowest MNR. The reasoning behind this allocation is that when a band has very low MNR, this means that the noise introduced by the quantization is above the threshold and therefore perceivable to humans. Thus, by allocating a bit and making the quantization finer, one reduces the noise and therefore increases the MNR. With this procedure, bits are allocated to

different frequency bands until a predefined bit budget has been reached. In later versions of MP3, this procedure was replaced by noise-shaping methods, which are more complex.

2.3.4 Vibrotactile Codecs

Despite the clear necessity for a vibrotactile codec, the development of such has only gained traction very recently. In particular, the emergence of standardization activities in IEEE P1918.1.1 [10] and MPEG-H have given a significant boost to the conception of new vibrotactile compression methods.

In the early works [48] and [49], compression methods were introduced that are capable of compressing signals by a factor of 4 without signficantly impacting perceptual quality. The compression of vibrotactile signals was conducted using a DCT and perceptual properties, in particular the ATV. In [50] a speech codec was adapted to the vibrotactile domain and then enhanced in [33]. The final codec employed masking thresholds as well as the ATV. The principle of this codec is to analyze the signal coming in and extract a set of parameters through that. These parameters are then used to synthesize an artificial signal. This artificial signal is subtracted from the input signal. Then, by transmitting the obtained parameters and the quantized residual between artificial and original signal, the decoder is able to reconstruct the signal at the other end.

Most recently, in [43] the codec named perceptual vibrotactile codec based on sparse linear prediction (PVC-SLP) was introduced. It employs the ATV to produce a sensitivity function. Then, the input signal is analyzed with sparse linear prediction to be split into coefficients and residual. Then, the coefficients and residual are quantized perceptually. Finally, Huffman coding compresses the quantized values losslessly.

So far, no codec has been able to achieve the requirements to achieve a breakthrough for vibrotactile signal compression. First, the resulting compression performance is not yet sufficient for highly sophisticated applications. The works in [49], [33] and [43] achieved data reduction by a factor of 4 [49], 8 [33], and 15 [4] without perceptual impairments, respectively. Second, only the compression scheme in [33] leveraged masking effects in addition to the ATV. It is to be assumed that a codec that includes masking in its perceptual analysis should be able to perform better. Third, the best performing codec so far, presented in [43] contains an optimization problem and therefore has high computational complexity and a large algorithmic delay. Finally, all introduced codecs are quite monolithic in their design. The codec from [43] in particular is optimized on the LMT reference dataset with respective Huffman coding tables. This makes it very hard to enhance these codecs in the future as more signal data become available, perceptual models are fine-tuned and coding methods improve. We aim to overcome these outlined challenges with our vibrotactile codecs in Chapter 3.

2.3.5 Multi-Channel Codecs

To date, there is no vibrotactile codec that is able to leverage redundancies between different signal channels. Without such multi-channel codecs, the compression performance for multi-channel setups will probably be insufficient. This is because for multi-channel signals, all the developed single-channel codecs will merely apply the coding to each signal channel individually. This is especially critical since more sophisticated multi-channel hardware setups have emerged recently [24], [27].

A particular challenge for the multi-channel coding of vibrotactile signals is that there is no established consensus on structural aspects of recording and display hardware. We see this already with the two example setups from [24], [27]. In [27], the hardware setup is built as a regular grid of piezoelectric actuators. On the other hand, in [24] the setup consists of irregularly spaced accelerometers on the human hand. Therefore, a straightforward adaptation of surround audio codecs for example, is not possible since these are based on universally agreed upon structures and protocols, e.g., 5.1 Surround Sound [51]. Therefore, a multi-channel codec for vibrotactile signals needs to be developed to be flexible enough to incorporate many different hardware configurations.

Multi-channel codecs that are able to handle irregularly spaced data, have mostly been developed for image and video compression so far [52]. In particular, the Graph Fourier Transform (GFT) has found wide adoption here. It takes graphs with nodes and weighted edges and computes a decorrelating transform on them by adapting for the edge weights. However, for compression, the GFT is not very suitable, because the adapted transform coefficients need to be transmitted as well [53]. Other transforms, such as the 2D DWT, assume there is always correlation between channels, which is not always the case.

An approach that allows for more flexibility as is required for vibrotactile signals is clustering. An interesting work in [54] used graph construction algorithms to form prediction relationships between nodes (correspond to different channels). For that, the set of nodes was initialized as a full graph where all nodes are connected. Then, virtual nodes were generated, after which a minimum spanning tree was calculated from them. Finally, the virtual nodes are removed and some nodes will have been formed into a group called a cluster. This algorithm however, is quite high in computational cost. Also, it follows a top-down approach that starts with a full graph and then forms clusters. We believe that for vibrotactile signals a bottom-up approach is more suitable, where channels are separate and then clustered one by one to allow for flexibility and a certain sparseness in the formed clusters.

2.4 Quality Assessment

2.4.1 Rate-Distortion Evaluation

To assess the performance of lossy codecs, they are usually evaluated in terms of rate and distortion jointly since these entities depend on each other. For lower rate, the distortion increases as we have to discard larger amounts of signal information. Conversely, to lower the distortion, we have to spend more bits to preserve much of the signal content. Thus, one will usually plot the mean curves of distortion over rate (or some derived quantities from them) and evaluate a codecs performance by comparing these curves. This assessment can be done to find optimal parameters or compare different codecs to each other.

The resulting data rate of a lossily coded signal can be computed simply by

$$R = \frac{\|c\| f_S}{N_S}, \tag{2.4}$$

where $\|c\|$ is the number of bits in the bitstream, N_S is the number of total samples of the signal and f_S is the sampling frequency. Usually, we are interested in the change in data rate, the lossy encoding has produced, which is expressed by the compression ratio (CR) computed as

$$\mathrm{CR} = \frac{R_{orig}}{R}, \tag{2.5}$$

where R_{orig} is the rate of the original PCM encoded signal. The CR gives the factor by which the rate has been reduced by the encoding.

The distortion is usually measured in terms of mean square error (MSE), which is

$$\mathrm{MSE} = \frac{1}{N_S} \sum_{n=0}^{N_S-1} (\hat{x}[n] - x[n])^2, \tag{2.6}$$

where $\hat{x}[n]$ is the CS waveform and $x[n]$ is the original signal.

2.4.2 Objective Quality Metrics

The purpose of objective quality metrics is to give us an estimate on the mathematical similarity of the CS to the reference signal (RS). As such, objective quality metrics are based on the MSE, mapping this error measure to a quality score.

Perhaps the simplest objective quality metric is the peak signal-to-noise ratio (PSNR). It is defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{I_{max}^2}{\text{MSE}} \right) \text{dB}, \tag{2.7}$$

where I_{max} is the maximum possible amplitude range of the signals to be examined. Since the accelerometer used for measuring the signals in the LMT reference dataset has the output range of -3 to 3, we have $I_{max} = 6$. The signals from the CEA reference dataset have a range of -1 to 1, so then we have $I_{max} = 2$. Thus, the PSNR directly maps the MSE to a quality score by inverting it and displaying it in dB.

Additionally to the PSNR, often the SNR is used. Generally, it is defined as the ratio of signal energy to noise introduced into the signal, i.e.,

$$\text{SNR} = 10 \log_{10} \left(\frac{E_{signal}}{E_{noise}} \right) \text{dB}, \tag{2.8}$$

where E_{signal} is the average energy of the signal and E_{noise} is the average energy of the noise. In our case, the noise energy is calculated by the MSE and thus the SNR becomes

$$\text{SNR} = 10 \log_{10} \left(\frac{E_{signal}}{\text{MSE}} \right) \text{dB}. \tag{2.9}$$

Thus, similarly to the PSNR the SNR inverts the MSE and displays it in dB, but is normalized for each signal individually by the signal energy. Consequently, this leads to a better estimation of the significance of the introduced distortions relative to the signal energy.

2.4.3 Subjective Evaluation through Experiments

While objective quality metrics are simple and straightforward to calculate, they are not able to grasp the perceptual quality of CSs sufficiently. For example, when humans perceive a sinusoidal signal with a single frequency, they will not report any perceptual impairment if the signal is shifted by one sample. However, the SNR and PSNR will be quite low, even if the signals are almost identical. Similarly, for a well designed lossy coding scheme, the objective quality metrics can be quite low but humans might not perceive any impairments because the distortions were inserted smartly in ways humans would not perceive them.

In summary, the only way to obtain accurate and reliable ratings of perceptual quality of CSs is through human user experiments. A widely used experimental procedure for quality assessment that is originally coming from the audio domain is multi-stimulus test with hidden reference and anchor (MUSHRA) [55]. Generally, MUSHRA operates as follows:

- Human assessors are presented with a graphical user interface (GUI) with buttons that play back different signals each. One button plays the RS, while the others play different CSs. The assessors can choose to play any signal in any order and as many times as they want.
- Associated to each signal is a scale from 0 to 100 with a slider, with which the assessors can rate the similarity of each CS to the RS. Additionally, the scale is labeled with subjective quality descriptions, e.g., *excellent* or *fair*.
- After the scores have been set, the assessors save them and are finished. When all the ratings from all assessors have been recorded, a post-screening takes place to exclude ratings from assessors that are outliers (see [55]).

MUSHRA contains two additional key elements that aid the validation of the reliability of the gathered rating data by serving as catch trials. These two elements are the hidden reference and the anchor signals, which are mixed into the set of CSs in the GUI. The hidden reference is equal

to the uncompressed RS. Assessors are informed of its existence, i.e., they know that one of the CSs is actually perfectly identical to the reference. With the hidden reference, we are able to observe whether assessors are able to detect the differences between signals accurately. If an assessor gives a low rating to the hidden reference, it means that his ratings are probably unreliable. The anchor signals are signals with controlled impairments inserted into them. By designing these impairments appropriately, we are able to create signals that we roughly know the quality of beforehand. Thus, the anchor signals can help to calibrate the rating scale. For example, when we design a medium quality anchor, we expect it to have a rating around $40 - 60$. So if the rating of that anchor is way off, we can detect possible mistakes in our experiment and analyze their cause. Additionally, both the hidden reference and the anchor signals serve in the post-screening of the rating data. In particular, assessors are usually excluded if they rate the hidden reference below 90 in 25% of their ratings, except if a significant number of the other assessors have done the same.

While MUSHRA has found wide-spread use and adoption, producing generally reliable results, it is still a method that is primarily tuned for audio signals. An adoption for the vibrotactile domain is not straightforward. First, as mentioned, the MUSHRA is completely unconstrained regarding the order and frequency of the signals played back to the assessors. The general idea behind such an unconstrained assessment is to give the assessors the chance to find also slight differences in the signals according to their preference. However, the freedom of this self-pacing comes with significant caveats. In particular, the measured quality ratings are often subject to non-systematic individual differences. These individual differences arise since there are always masking [56] and temporal integration effects [57]. Adding to that are other confounding factors steming from variations in brain state [58], attention [59], or habituation [60]. It also has been found that the human somatosensory system exhibits a sensory persistence around $500 - 1200$ ms. This means that, when two sequential stimuli are perceived, the second stimulus will be more intense subjectively [44], [61], [62]. This effect is one aspect of time order effects (TOEs) that arise in vibrotactile perception. In general, TOEs describe effects where the order in which signals are perceived changes their perception [63]. Therefore, the timing and order of signals need to be tightly controlled. Since a MUSHRA session is never exactly reproducible, all these effects add up and can distort ratings in an uncontrollable way. Therefore, an experimental procedure with precise timing is to be preferred. Finally, the self-pacing often leads to long durations of the assessment experiment, which can lead to fatigue for the assessors. An additional aspect of MUSHRA is that it requires trained expert assesors [64]. However, experts usually rate quality different than the broad population, which means that the obtained quality ratings might be very different to the average experience of an unskilled human user when it comes to assessing the performance of codecs in everyday use [65], [66]. In the vibrotactile domain, there are only very few experts available, which is in sharp contrast to the audio domain. Overall, while MUSHRA is widely used in the audio domain, it is not directly applicable in the vibrotactile domain as its experimental procedure is not carefully controlled and may be subject to experimental confounds and biases that can have a detrimental impact in experiments with non-expert assessors.

2.4.4 Subjective Metrics

The idea behind subjective metrics is to compute a score from signal data that reflects perceptual signal quality. Thus, with such a metric one could skip time-consuming and tedious human user experiments and directly assess perceptual quality in an automated fashion. In CS quality assessment, the calculated scores describe the perceptual similarity of a CS to the RS similar to the ratings recorded with human user experiments.

To the best of our knowledge, for compressed vibrotactile signals, one relevant perceptual metric has been presented: the so-called *spectral-temporal similarity (ST-SIM)* [67]. The ST-SIM is a combined metric from two individually computed scores. First, the spectral score is calculated from signal spectra and the ATV. The ATV is hereby subtracted from the spectra of blocks of the original and CSs respectively. Then this difference is mapped to a range between 0 and 1 with the sigmoid function. Then the spectral score S-SIM can be calculated by the overlap of the two obtained functions. Second,

the temporal score T-SIM is calculated from the time-domain signals with a formula similar to Pearson correlation (PC). The combination of both scores is then done by multiplying them as

$$\text{ST-SIM} = \text{S-SIM}^{(1-\eta)} \cdot \text{T-SIM}^{\eta}, \tag{2.10}$$

with a weighting parameter η. Usually, we have $\eta = 2/3$, i.e., the temporal score is twice as relevant as the spectral score, which is supported by [68]. It also is justified by the fact that the spectral score is quite coarse since it examines only the perceivable frequency overlap and not the exact differences.

2.5 Postprocessing and Quality Enhancement

After transmission and decoding of a compressed vibrotactile signal, enhancing its quality to reduce some of the artifacts introduced by lossy coding leads to a better user experience. Methods of decoder-side quality enhancement have been studied extensively for multiple types of CSs like audio signals, images, and videos. Thus, we aim to develop our methods inspired by these findings. In the following, we summarize the most important aspects of decoder-side quality enhancement and methods that lay the foundation for our work on enhancing vibrotactile signal quality.

2.5.1 Quality Enhancement Principles

The principal idea of decoder-side quality enhancement is to filter the decoded signal in a way that removes some of the coding artifacts. Thus, the conventional approach is to design filters and convolve them with the signal. However, this approach has not led to satisfactory results overall. This is mostly due to the fact that the filters used are hand-crafted and therefore usually suboptimal for the signal data at hand and also convolution is a purely linear method. It has been shown that with nonlinear methods based on neural networks (NNs), signficantly better results are achievable. As such, multiple NN-based quality enhancement approaches have been proposed for image, video and audio processing [69]–[71]. These approaches bring significant performance improvements for both subjective and objective evaluation metrics compared to conventional approaches.

In general, the employment of a NN for quality enhancement follows a simple idea. To effectively enhance a signal with coding artifacts, we need to process it in a way that makes it more similar to its RS again. Therefore, we need a structure that can be trained to map the CS to the RS. A NN is exactly such a structure that can be trained to learn an arbitrary relationship between any two signals.

2.5.2 Recurrent Neural Networks

So-called recurrent neural networks (RNNs) are one of the most popular NN types used for quality enhancement. This is due to their good ability to exploit temporal correlations in sequential signal data [72]. The general form of an RNN with one layer is depicted in Fig. 2.4. The RNN takes the input signal $\boldsymbol{x}[n]$ and maps it onto the output signal $\boldsymbol{o}[n]$ via the hidden state $\boldsymbol{h}[n]$. The mapping is characterized by the weight matrices $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ as [72]

$$\boldsymbol{h}[n] = \tanh\left(\boldsymbol{U}\boldsymbol{x}[n] + \boldsymbol{W}\boldsymbol{h}[n-1] + \boldsymbol{a}\right) \tag{2.11}$$

$$\boldsymbol{o}[n] = \boldsymbol{V}\boldsymbol{h}[n] + \boldsymbol{b}, \tag{2.12}$$

where the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are additional bias parameters. It is possible to use other nonlinearities than tanh. By concatenating multiple hidden state vectors, we can form RNNs with multiple layers. The number of entries in $\boldsymbol{h}[n]$ is usually referred to as the number of neurons.

The output $\boldsymbol{o}[n]$ of the RNN is compared to the desired output signal $\boldsymbol{y}[n]$ via the loss function $L[n]$. The network is trained using the available data with a gradient descent algorithm to minimize $L[n]$. In our quality enhancement application $\boldsymbol{x}[n]$ is the CS and $\boldsymbol{y}[n]$ is the RS.

RNNs often exhibit the problem of exploding or vanishing gradients. This means, in its basic form the gradient in RNNs grows or shrinks uncontrollably, which can make training nearly impossible. To

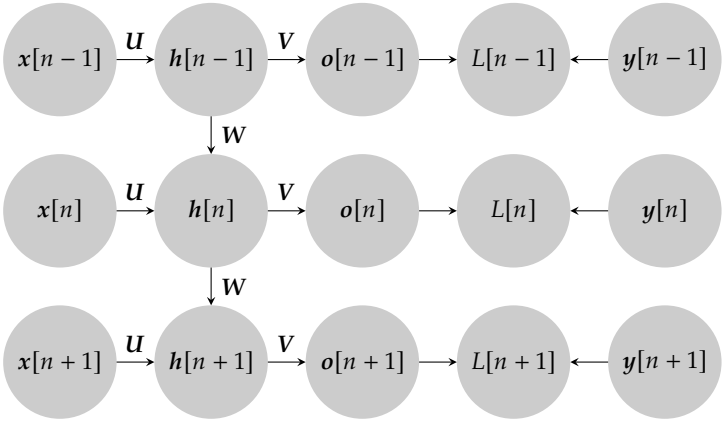


Figure 2.4 Structure of a general recurrent neural network with one layer around time sample n.

solve this, the long short-term memory (LSTM) [73], [74] and gated recurrent units (GRUs) [75]–[77] were introduced.

In LSTM cells the equations (2.11) and (2.12) are extended by so-called gates. In total there are the input, forget, and output gates. These gates map the weighted combination of input and hidden state to a range between 0 and 1 with a sigmoid function [72, Eq. 10.40-10.44]. The LSTM cell has an additional state variable between the input and hidden state. Then the input gate controls how the input can influence the state of the LSTM cell and in the extreme case shut it off by becoming 0. The forget gate acts on the cell state and by becoming 0, it would erase the state and hence forget past learned parameters. The output gate controls how much the cell state can translate to the hidden state in the same manner as the input gate. Thus, it is intuitive to see how the limiting of certain weights between 0 and 1 is able to keep the magnitude range of gradients in check.

The widely used alternative to the LSTM is the GRU. It is very similar, but less complex by using only two gates, namely the so-called update and reset gates [72, Eq. 10.45-10.47]. The update gate controls the computation of the hidden state. In the extreme case, it either copies the previous hidden state or it uses the input to compute an entirely new hidden state like in (2.11). The reset gate then determines to what extent the previous hidden states can influence the new one in the case of a computation of a new hidden state.

When blocks of signals are available at once, causality is not required in signal processing applications. Thus, we are then able to use more powerful bidirectional networks like the bidirectional recurrent neural network (BiRNN). In a BiRNN, we simply extend (2.11) with an additional term consisting of another weight matrix multiplied with $\boldsymbol{h}[n+1]$. Since it is able to exploit dependencies in both time directions, it often exhibits higher performance compared to the RNN. Again, in order to avoid exploding or vanishing gradient effects, the bidirectional long short-term memory (BiLSTM) was introduced in [78], [79].

In order to enhance signals after decoding, RNNs have been successfullly used in the image, video, and audio domains. To enhance human speech signals, in [80] a RNN-based method was introduced in which a network was trained to reconstruct missing high-frequency information that was left out when encoding. For enhancing audio signals encoded with MP3, quite recently a LSTM RNN was introduced in [81]. In the video domain, [82], [83] proposed RNN-based methods to enhance the quality of encoded video sequences. Such enhancement of videos becomes subjectively visible as deblurring.

2.5.3 Residual Learning

When enhancing signals with NNs, a high difference in dynamic range between signals can often pose a challenge. To overcome this, *residual learning (RL)*, which was first introduced in [84] can be used. Many types of distortions introduced into signals by compression algorithms have zero-

centering distributions, e.g., Gaussian white noise or quantization distortion. By applying RL, the NN is trained to model and reconstruct a residual signal to compensate for the distortion, which is easier than reconstructing the high-quality signal itself. This is implemented by adding shortcut connections (see [84, Fig. 2]). RL has been shown to achieve a significant improvement in performance, especially when the signals at hand have a high difference in dynamic range, like the vibrotactile signals in the LMT reference dataset from Sec. 2.1.1.3. Additionally, it has been observed that RL counteracts the effect of gradient vanishing and explosion as well as performance degradation when training a very deep NN.

RL was quickly adopted for numerous tasks, e.g., denoising, undersampled signal reconstruction, and CS quality enhancement. In [85], a convolutional autoencoder with shortcut connections was developed that was able to enhance compressed speech signals. The works [86], [87] introduced RL-based NNs for enhancement of compressed images. Finally, [88] further extended the concept of RL by adding local residual dense connection to the present global residual shortcut connections. Through this, a further boost in performance was gained.

2.6 Adaptive Filter Equalization

When displaying vibrotactile signals to human users via actuators, the signals will not remain undistorted [16], [89]. The introduced distortions stem from mechanical imitations, imperfections and noise in the signal transmission, and nonlinearities in the amplifiers and the actuators themselves.

The introduced distortions can be highly detrimental to the human user experience in some applications. For example, in [90], [91] an artificial pulling sensation is created by conveying waveforms to the human skin. In order for the effect to be realistic and controllable, the waveforms have to be displayed very precisely. This can only be achieved by reducing the actuator distortions, since even buying a very high-quality actuator and equipment, if available, is usually high in cost and it is not possible to eliminate all factors leading to distortions. In general, when reducing distortions, we aim for reducing the difference between input and output signals of the actuators as much as possible. This difference can be measured by the MSE over a certain time frame. The process of reducing this type of distortions is called *equalization*.

A few works have investigated the possibilities of reducing the actuator distortions so far. Namely, [89], [92], [93] presented linear offline equalization schemes, i.e., linear filters were applied before the actuators to change the signals appropriately for the actuator to output the correct signal. In [89] additionally to the equalization scheme, the actuator mechanics were changed.

These approaches suffer from significant drawbacks. First, with only linear filters being applied, the equalization methods are unable to cope with nonlinear behavior of most actuators and amplifiers [16]. In [93] the total harmonic distortion (THD) of actuators was determined, which is a measure for the degree of nonlinearity. The result confirmed that nonlinearities are certainly present in common vibrotactile actuators. Second, the approaches are purely offline, meaning the linear filters are crafted and then unchanged. However, actuators are usually dynamic and their behavior changes, e.g., with changing contact force of the human finger on them. Therefore, it is essential to use an automated method, where the equalization scheme can adapt to changes in actuator behavior.

2.6.1 Equalization Setups with Adaptive Filters

To develop an equalization method that is both able to cope with nonlinearities as well as dynamic actuators, we resort to adaptive filter equalization [94]. Basically, an adaptive filter is a digital filter with the capability to adapt its filter coefficients by itself in real time [95]. A major advantage of adaptive filtering is that the method is applicable to any kind of vibrotactile actuator without requiring knowledge of its mechanics or transfer functions. We also favor adaptive filtering over a machine learning approach with NNs, since an adaptive filter is able to quickly adjust to a changing actuator behavior, whereas in machine learning the pre-trained model would have to be retrained.

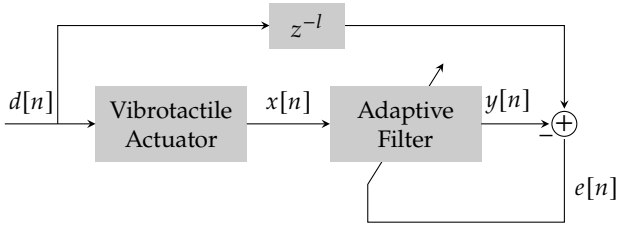


Figure 2.5 Postdistortion setup to equalize a vibrotactile actuator by adapting a filter to correct the actuator output $x[n]$ to better resemble the desired signal $d[n]$. Adopted from [2] © 2020 IEEE.

In vibrotactile applications the actuator behavior changes very frequently, e.g., by users changing the pressure their finger exerts on the actuator. Additionally, adaptive filters are less computationally expensive than NNs.

Adaptive filtering has seen wide adoption in many fields, e.g., echo cancellation [96], biomedical imaging [97], and noise cancellation, channel equalization or signal prediction [94], [98]. In the following, we present the two most common equalization setups employing adaptive filters.

2.6.1.1 Postdistortion Setup

A very widely used adaptive filter equalization setup, the so-called *postdistortion setup*, is shown in Fig. 2.5. In this configuration, $d[n]$ is the desired vibrotactile signal that is put into the actuator. The actuator then produces the signal $x[n]$ at its output that can be felt physically. This signal is similar to $d[n]$ but is distorted by the actuator. Then, the actuator is equalized by the subsequent adaptive filter, which takes $x[n]$ to output $y[n]$. By adapting the filter parameters appropriately, the difference signal $e[n] = d[n-l] - y[n]$ is minimized. In this difference, we need to account for a delay of l since the actuator and filter are causal systems that delay the signal.

On the upside, this postdistortion setup has a high simplicity of implementation. By measuring the actuator output with an accelerometer attached to the fingertip as described in Sec. 2.1.1, the influence of the human finger can also be taken into account.

However, the postdistortion setup is not practical for a real application of actuator equalization to ensure that the actuator output is less distorted. This is because the equalized signal $y[n]$ is only available digitally and not physically at the actuator ouput. In order to have the actuator output an equalized signal, the adapted filter needs to be placed before the actuator. Therefore, we are only able to equalize offline with this setup. That means, the entire signal $d[n]$ is displayed by the actuator and the output $x[n]$ is recorded as a whole. Then, these gathered signal data are used to train the adaptive filter, which is then placed before the actuator to filter its output. For an audio setup, this approach is usually sufficient, because the properties of speakers do not change over time. However, in the vibrotactile domain, there is constant interaction with the actuators and as, e.g., contact forces change, so will the actuator properties. Adding to that, as described previously, the actuators are usually nonlinear and therefore nonlinear adaptive filter models are used. Thus, these two blocks usually do not commute.

A straightforward solution is to simply exchange the order of actuator and adaptive filter. This then yields a so-called *predistortion setup* [99]–[101]. As described before, since both blocks are nonlinear, we cannot assume a commutative property. This is why we are then not able to apply the same adaptive algorithms to train the adaptive filters. Instead, we need to have knowledge of the actuator behavior, which we can then take into consideration in the adaptation. We show this in Sec. 2.6.2.1 as well. To estimate the behavior of the actuator, we need a second adaptive filter whose parameters can then be fed into the adaptation algorithm. Therefore, we would require signficantly higher computational effort and a more advanced and complex algorithm. Also, since the estimation will have imperfections, the equalization performance will be lowered.

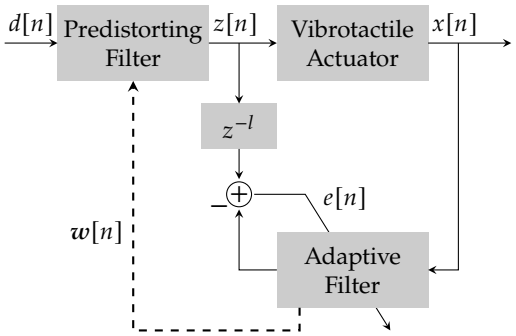


Figure 2.6 Postdistortion setup to equalize a vibrotactile actuator by adapting a filter to correct the actuator output $x[n]$ to better resemble the desired signal $d[n]$. The dashed arrow illustrates that the adaptive filter coefficients are copied to the predistorting filter. Adapted from [2] © 2020 IEEE.

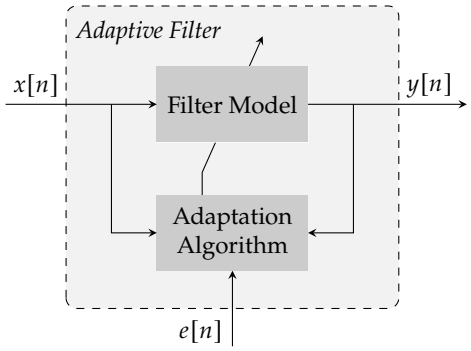


Figure 2.7 Configuration of a general adaptive filter with filter model and adaptation algorithm. Adapted from [2] © 2020 IEEE.

2.6.1.2 Postdistortion and Translation Setup

In order to allow for online equalization, while retaining the ability to use simple algorithms, the *postdistortion and translation setup* has been developed. It has found wide adoption in equalization applications [102]–[106].

The setup is an extension of the postdistortion setup, where the adaptive filter is copied to the front, as illustrated in Fig. 2.6. With the predistorting filter, the desired signal is first filtered to receive $z[n]$. Then, $z[n]$ is processed by the postdistortion setup as before. Thus, instead of exchanging the order of actuator and adaptive filter, the adapted filter is simply copied to filter the input signal before the actuator.

This setup again exchanges the order of blocks that are usually nonlinear. That means, it also is not entirely mathematically proper since the blocks are not commutative in general. Nonetheless, it has been shown numerous times that the performance obtained with this setup can be highly satisfactory [102]–[107]. Whether the violation of the commutative property deprecates performance to an extend that makes this setup a bad choice for vibrotactile actuators, needs to be examined in experiments.

While in theory the adaptive filter coefficients are copied with every signal sample, this is not feasible in practice. This is due to the delay of the overall system. Therefore, the predistorting filter coefficients are updated block-wise.

2.6.2 Adaptive Filters

An adaptive filter is made up of a filter model and an adaptation algorithm as shown in Fig. 2.7. The filter model serves the purpose of mapping the input signal $x[n]$ to the output $y[n]$. Then, input and

output are used together with the externally computed error signal $e[n]$ in the adaptation algorithm. This algorithm adjusts the coefficients of the filter model over time to minimize the signal $e[n]$.

2.6.2.1 Adaptation Algorithm

The adaptation algorithm that we focus on in this work assumes that the filter model is linear in its parameters. That means, it can be described by

$$y[n] := \boldsymbol{w}^\top[n] \cdot \boldsymbol{\varphi}[n], \tag{2.13}$$

where $\boldsymbol{w}[n]$ is the so-called weight vector containing the filter model parameters and $\boldsymbol{\varphi}[n]$ is the so-called regression vector, which is built up from previous input or output signal values.

The adaptation algorithm is practically an optimization algorithm to solve the problem

$$\min_{\boldsymbol{w}[n]} \mathrm{E}\left[e^2[n]\right]. \tag{2.14}$$

A very commonly used algorithm is the normalized least mean squares (NLMS) algorithm [108]. It assumes that the expectation of the error signal can be approximated by its current value, i.e., $\mathrm{E}\left[e^2[k]\right] \approx e^2[k]$. In [109] it was shown that this is an unbiased estimate of $\mathrm{E}[e^2[n]]$. Then, the algorithm tries to find an optimal value by performing gradient steps as

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] - \mu_n \frac{\partial e^2[n]}{\partial \boldsymbol{w}[n]}. \tag{2.15}$$

With the filter model equation (2.13), the gradient is calculated as

$$\frac{\partial e^2[n]}{\partial \boldsymbol{w}[n]} = -2e[n] \cdot \frac{\partial y[n]}{\partial \boldsymbol{w}[n]}. \tag{2.16}$$

This shows clearly why in a predistortion setup, an additional filter is required to estimate the behavior of the actuator. In this setup, the filter output $y[n]$ is further altered by the actuator. Therefore, without knowledge of the actuator behavior, it is not possible to calculate $\frac{\partial y[n]}{\partial \boldsymbol{w}[n]}$ when the actuator is placed after the adaptive filter.

For the postdistortion and translation setup, however, (2.13) is valid, so the gradient is then

$$\frac{\partial e^2[n]}{\partial \boldsymbol{w}[n]} = -2e[n] \cdot \boldsymbol{\varphi}[n]. \tag{2.17}$$

The NLMS algorithm has a normalized step size that is

$$\mu_n = \frac{\mu}{\boldsymbol{\varphi}^\top[n] \cdot \boldsymbol{\varphi}[n] + \psi}, \tag{2.18}$$

where μ is the step size scaling factor and ψ is a small positive constant to avoid numerical instabilities.

2.6.2.2 Filter Models

Although most systems are nonlinear, e.g., audio speakers and vibrotactile actuators [16], often linear adaptive filter models are employed since their theory is well known and relatively simple [94], [98]. In order to sufficiently equalize nonlinear distortion parts, a nonlinear filter model should be chosen. There is an enormous variety of nonlinear filter models, most prominently including spline filters [110], kernel filters [111], simple multilinear functionals [112], and Volterra filters [113].

One of these filter models clearly stands out in terms of adoption and established theoretical foundation: Volterra filters [96], [97], [114]–[120]. These filters are a type of polynomial filters [113]. They are linear in their filter parameters and can therefore be analyzed analogously to linear filters [121].

The Volterra filter can have any order of nonlinearity in theory, but complexity increases rapidly with the order. Therefore, mostly second order Volterra filters have been employed so far [96], [97], [114]–[116], which in turn means the filter is unable to model complex nonlinearities of higher order [117].

Even though the adaptive filter equalization setup is universally applicable to any actuator, the choice of the best performing filter model can differ for different actuators. For example, if an actuator existed that was entirely linear, then a linear filter model would in all likelihood be the best choice. In general, all actuators have nonlinear effects and are similar in their behavior [16]. In the following, we present two state-of-the-art filter models that serve as comparison for our proposed method.

1. **Linear Filter:**
   The linear filter (LF) is characterized by the difference equation

$$y[n] = \sum_{i=0}^{N-1} a_i[n]x[n-i] + \sum_{j=1}^{M} b_j[n]y[n-j], \tag{2.19}$$

   where $a_i[n], i = 0, \ldots, N-1$ and $b_j[n], j = 1, \ldots, M$ are the coefficients of the feedforward and feedback part, respectively. They have n as index, indicating that they are time-dependent thus part of an adaptive filter. Rewriting (2.19) into the form from (2.13), we get

$$\boldsymbol{\varphi}[n] := \left[\boldsymbol{\varphi}_a^\top[n] \;\; \boldsymbol{\varphi}_b^\top[n]\right]^\top \tag{2.20}$$

$$\boldsymbol{\varphi}_a[n] := \left[x[n], x[n-1], \ldots, x[n-N+1]\right]^\top \tag{2.21}$$

$$\boldsymbol{\varphi}_b[n] := \left[y[n-1], y[n-2], \ldots, y[n-M]\right]^\top, \tag{2.22}$$

   and

$$\boldsymbol{w}[n] := \left[a_0[n], \ldots, a_{N-1}[n], b_1[n], \ldots, b_M[n]\right]^\top. \tag{2.23}$$

2. **Second Order Volterra Filter:**
   The second order Volterra filter (SOVF) is described by

$$y[n] = \sum_{i=0}^{N-1} a_i[n]x[n-i] + \sum_{i=0}^{N-1}\sum_{j=i}^{N-1} b_{i,j}[n]x[n-i]x[n-j], \tag{2.24}$$

   where $a_i[n]$ and $b_{i,j}[n]$ with $i, j = 0, \ldots, N-1$ are the coefficients for the linear and quadratic part, respectively. With the quadratic part, the SOVF can display only up to second order nonlinearities. Therefore, we consider an extended version of the SOVF, which is called second order Volterra with infinite impulse response filter (SOV-IIRF) and was introduced in [115]. It is given as

$$\begin{aligned} y[n] = &\sum_{i=0}^{N-1} a_i[n]x[n-i] + \sum_{i=0}^{N-1}\sum_{j=i}^{N-1} b_{i,j}[n]x[n-i]x[n-j] \\ &+ \sum_{i=1}^{M} c_i[n]y[n-i] + \sum_{i=1}^{M}\sum_{j=i}^{M} d_{i,j}[n]y[n-i]y[n-j], \end{aligned} \tag{2.25}$$

   where $a_i[n]$ and $b_{i,j}[n]$ with $i, j = 0, \ldots, N-1$ are the coefficients for the linear and quadratic feedforward part and $c_i[n]$ and $d_{i,j}[n]$ with $i, j = 1 \ldots, M$ for the linear and quadratic feedback part, respectively. The SOV-IIRF is now able to behave as a nonlinear filter of arbitrarily high order. However, the filter model is also more prone to instabilities because of the nonlinear feedback loop.

We can reformat (2.23) in the form (2.13) as

$$\boldsymbol{\varphi}[n] := \left[\boldsymbol{\varphi}_a^\top[n] \;\; \boldsymbol{\varphi}_b^\top[n] \;\; \boldsymbol{\varphi}_c^\top[n] \;\; \boldsymbol{\varphi}_d^\top[n]\right]^\top \tag{2.26}$$

$$\boldsymbol{\varphi}_a[n] := \left[x[n], x[n-1], \ldots, x[n-N+1]\right]^\top \tag{2.27}$$

$$\begin{aligned}\boldsymbol{\varphi}_b[n] := \big[&x[n]x[n], \ldots, x[n]x[n-N+1],\\ &x[n-1]x[n-1], \ldots, x[n-1]x[n-N+1],\\ &\ldots, x[n-N+1]x[n-N+1]\big]^\top\end{aligned} \tag{2.28}$$

$$\boldsymbol{\varphi}_c[n] := \left[y[n-1], y[n-2], \ldots, y[n-M]\right]^\top \tag{2.29}$$

$$\begin{aligned}\boldsymbol{\varphi}_d[n] := \big[&y[n-1]y[n-1], \ldots, y[n-1]y[n-M],\\ &y[n-2]y[n-2], \ldots, y[n-M]y[n-M]\big]^\top,\end{aligned} \tag{2.30}$$

and

$$\begin{aligned}\boldsymbol{w}[n] := \big[&a_0[n], \ldots, a_{N-1}[n], b_{0,0}[n], \ldots, b_{0,N-1}[n],\\ &\ldots, b_{N-1,N-1}[n], c_1[n], \ldots, c_M[n],\\ &d_{1,1}[n], \ldots, d_{1,M}[n], \ldots, d_{M,M}[n]\big]^\top.\end{aligned} \tag{2.31}$$

Chapter 3

Vibrotactile Signal Compression

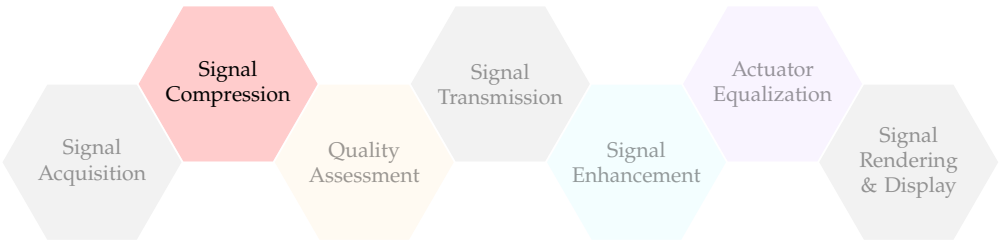
There is an evident need in reducing the transmission rate requirements of vibrotactile signals as multi-channel setups become available that increase signal data rates substantially. This calls for efficient, high-performing compression methods for vibrotactile signals. High-performing in this context means that they should allow both for high compression ratio (CR), but also preserve high signal quality for human users. To this end, we aim to use findings from psychophysical experiments to develop perceptual models that can be used in the compression methods, to maintain high signal quality and a pleasant user experience.

In this chapter, we present the development of our vibrotactile signal compression framework, which is the first part of the overall vibrotactile communication pipeline as shown in the figure below. The compression scheme development is structured into four parts. First, we define the key points of the properties the developed codecs should have in Sec. 3.1. Then, we analyze the available vibrotactile signal data to extract their properties in Sec. 3.2. These properties will give us insights on points we need to consider to tailor our coding methods to vibrotactile signals specifically. Then, we present our two developed codecs. The first, is a single-channel codec for respective signals recorded with one point of interaction presented in Sec. 3.3. Here, we first analyze human perception to develop suitable models. Then, we use the developed models and coding techniques to develop our single-channel codec. Finally, we move on to examine the coding of multi-channel signals and present our developed coding scheme for such in Sec. 3.4. Parts of this chapter have been published in [6], [7], [9].

3.1 Desired Codec Capabilities

The design of the vibrotactile codec was conducted with certain desired features in mind. By designing the codec around these principles, we achieve a robust and efficient codec framework with high performance and flexibility. The desired features that the codec must have, are described in detail in the following.

1. **Rate-Scalability:** The designed codec framework should allow for flexible scaling of the output rate and therefore the resulting CR. As it is typical, the scaling of the target output rate should be done by varying exactly one codec parameter.



2. **Perceptual Transparency:** Essential to any lossy, perceptual codec is its capability of compressing signals in a way that introduces minimal perceivable distortions to the human user. A codec that introduces no perceivable distortions for humans is called perceptually transparent. Our goal is to have the codec to be perceptually transparent for as many CRs as possible. In other words, we aim to maximize the value range for CR, for which the compressed signals (CSs) are equal to the original in terms of perception for humans. Beyond that range, the perceptual quality should decrease as slowly as possible.

3. **Fast Execution:** The codec should be able to encode and decode signals fast. In particular, the algorithmic delay should be small enough to enable online applications over the internet. Since the internet connection as well as hardware on both sides of the pipeline already introduce delays, this requirement on the codec can be very strict. In practical terms, it means that the codec should not utilize complex optimization algorithms or machine learning, since these methods typically tend to introduce a rather long algorithmic delay.

4. **Modularity:** Since haptics in general is a quite new research field, the potential for future enhancement is vital in methods being developed today. The design of the codec should therefore be conducted with ease of enhancement in mind. This can be accomplished with a modular structure. The modular structure of MP3 allowed users to enhance individual components, especially the psychoacoustic model, which contributed to the wide adoption of this codec [34]. We aim to follow this philosophy in the design of the vibrotactile codecs as well.

5. **Versatility:** Unlike the audio, image or video domains, where established norms exist, for vibrotactile domain there is no such consensus. The areas where no gold standard has been established include but are not limited to the used setups for acquisition and display, the number of channels, the sensor placement on the human body, the actuators to be used, or the amplitude range of measured signals. Adding to that, as we show in Sec. 3.2, vibrotactile signals have a very large variability. Finally, vibrotactile applications can be offline, where signals are stored and rendered at a different time, or online, i.e., signals are recorded and streamed in real-time. Thus, the designed vibrotactile codecs should have appropriate parametrizations and mechanisms to adapt to a multitude of different applications, signal properties and hardware setups. As part of this capability, we also highlight that it is desirable to have a multi-channel codec that is fully backwards-compatible to the preceding single-channel codec. With this compatibility, users can switch between multi-channel and signal-channel processing as desired.

3.2 Signal Properties

In order to optimize the codecs for vibrotactile signals, it is vital to analyze the properties of the signals at hand. In the following, we conduct such an analysis in terms of sampling frequency, dynamic range, frequency content and compressability. We do so mainly for the LMT reference dataset, since we opt to optimize our vibrotactile codec for accelerometer-recorded signals. The other dataset will be used to showcase the generalization ability of our methods towards other kinds of signals.

3.2.1 Sampling Frequency

When sampling a vibrotactile signal, we have to make sure to capture all the perceivable essence of a signal without violating Nyquist criterion. It is widely accepted that the maximum frequency of vibrations humans are able to feel is around 1 kHz [28]. Thus, the minimum allowed sampling frequency to adhere to Nyquist criterion is 2 kHz. In practice, we will use a higher frequency to allow for sufficient frequency buffer zone and avoid the introduction of perceivable aliasing artifacts. The two most commonly used sampling frequencies are 2.8 kHz in the LMT reference dataset and 8 kHz in other non-publicly available datasets such as that under consideration by the MPEG group.

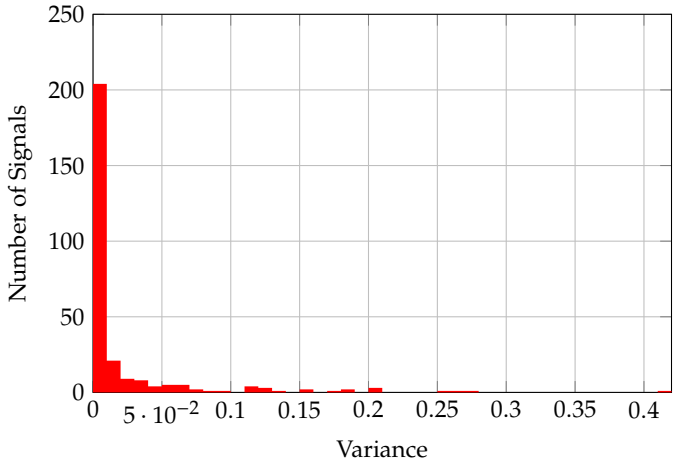


Figure 3.1 Histogram of the variances of the vibrotactile signals in the LMT reference dataset.

3.2.2 Dynamic Range

To assess the dynamic range of vibrotactile signals, we compute the signal energy, i.e., the variance, since the signals in the LMT reference dataset are zero-mean. Overall, the highest variance in the LMT reference dataset is approximately 0.416, the lowest is roughly $2.93 \cdot 10^{-5}$. This implies in total a dynamic range of roughly 41.5 dB. This means, any codec to compress vibrotactile signals has to be able to cope with signals of very different amplitude.

Next, we examine the distribution of variances. For that we plot the histogram of the variances for all 280 signals in the reference dataset. The resulting histogram is depticted in Fig. 3.1. More than 200 (approximately 73%) of all signals have a variance lower than 0.01. Thus, even though the dynamic range in general is high, most of the signals are fairly low in energy. This constitutes an additional challenge for a coding system, since there is a tradeoff between optimizing for low-energy signals and being able to compress the rarer high-energy signals efficiently as well.

3.2.3 Frequency Composition

The frequencies f in our signals can be linked to spatial distances d with

$$d = \frac{v}{f}, \tag{3.1}$$

where v is the scanning velocity. This links the frequency content to the material texture characteristics that are translated into vibration signals via the different tooltips. To analyze the frequency content of the signals in the LMT reference dataset, we compute the power spectral density (PSD) function for different materials, tooltips and speeds.

First, we choose the *3x1 spike* tooltip and the *fast* speed, since these signals are very distinctive according to [5] and compute the PSD for all different available materials. The resulting PSD functions are depicted in Fig. 3.2 where they are grouped by similar amplitude. We see that the frequency content can differ widely depending on the material. For the analysis, we group the materials by their hardness and texture coarseness. We observe:

- Hard, coarsely textured materials (*antivib pad*, *cork*, *rubber*, and *aluminium grid*): The PSD spectra of the first three materials mostly consist of a dominant frequency peak around 55 to 75 Hz. For these signals recorded at the *fast* speed, the frequency range of 55 to 75 Hz corresponds to roughly 1.7 to 2.9 mm. The three materials have fairly coarse macroscopic structures of this size. Therefore, the dominant frequency peaks most probably stem from these macroscopic texture elements. Since they are hard materials, the frequency peaks are distinctive and not dampened.

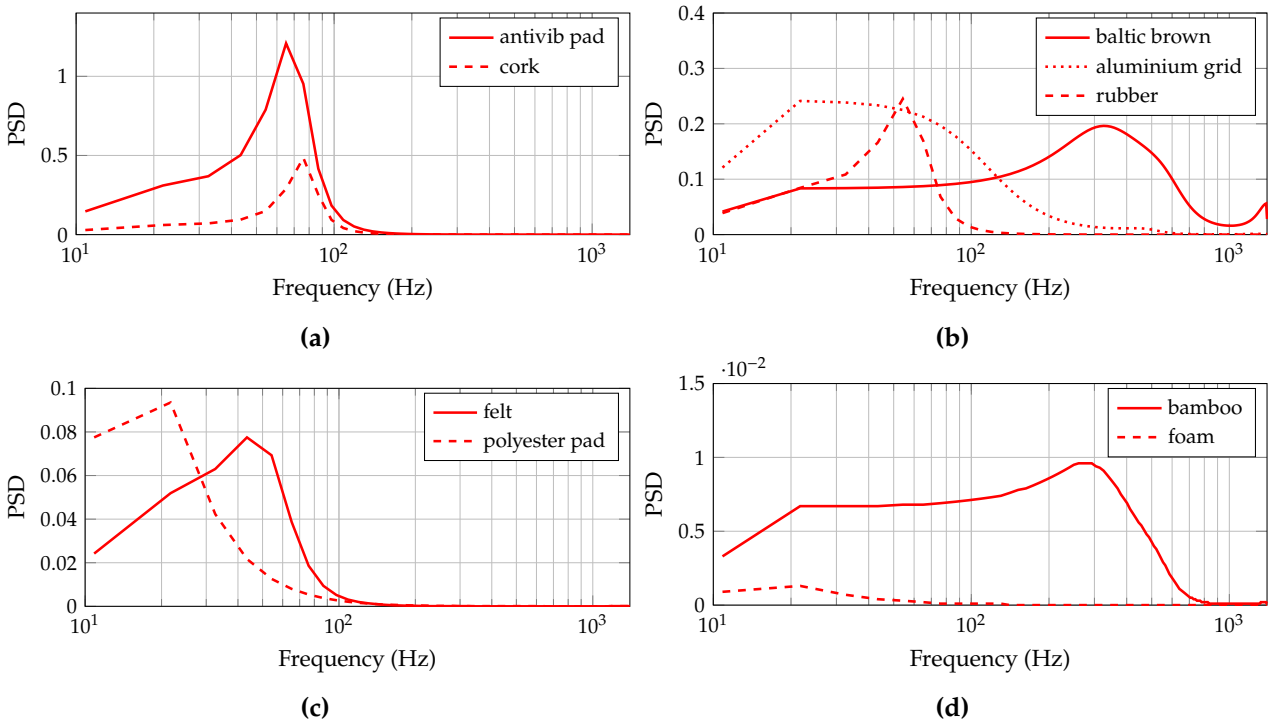


Figure 3.2 Power spectral density functions of signals from the LMT reference database for different materials, recorded with the *3x1 spike* tooltip and at *fast* speed.

The material *aluminium grid* produces a wider sprectrum, because the grid-like structure leads to a spikey signal that results in a wider frequency range.

- Hard, finely textured materials (*baltic brown* and *bamboo*): These materials produce much more wideband PSD spectra, whose dominant frequency lies around 300 Hz, which corresponds to much finer structures of around 0.5 mm in size. The wide spread of structure size in such smooth materials explains the wide frequency range present in these vibration signals.
- Soft, coarsely textured materials (*felt* and *polyester pad*): These materials produce PSDs with low amplitude in general and dominant frequencies around 20 to 45 Hz. This most probably comes from both the macroscopic structures with the added effect of the softness dampening the vibration signals overall.
- Soft, finely textured material (*foam*): The spectrum has very low amplitude and thus this corresponding signal is very smooth and low in energy. Thus, vibration is significantly dampened by the material softness.

Again, we observe the high dynamic range of around two orders of magnitude in these PSDs. So, one factor contributing to the high dynamic range surely is the vastly different smoothness and softness properties of the materials.

Next, we choose the material *rubber* at the speed *fast* and compute the PSDs for different tooltip choices. The results are depicted in Fig. 3.3 grouped by tooltip shape as also in the following analysis:

- Spike shaped tooltips (Fig. 3.3a; *spike*, *3x1 spike*, *3x3 spike*): The shape of the PSD is the same for all tooltips. The maximum frequency peak is around 55 to 65 Hz. Also, the dynamic range is in the same magnitude. Thus, the spike-shaped tooltips seemingly lead to quite similar signals.
- Small round array tooltips (Fig. 3.3b; *3x1 round*, *3x3 round*): Again, the shape of the PSDs is the same for both tooltips. The peak frequency is 55 Hz. However, compared to the spike tooltips,

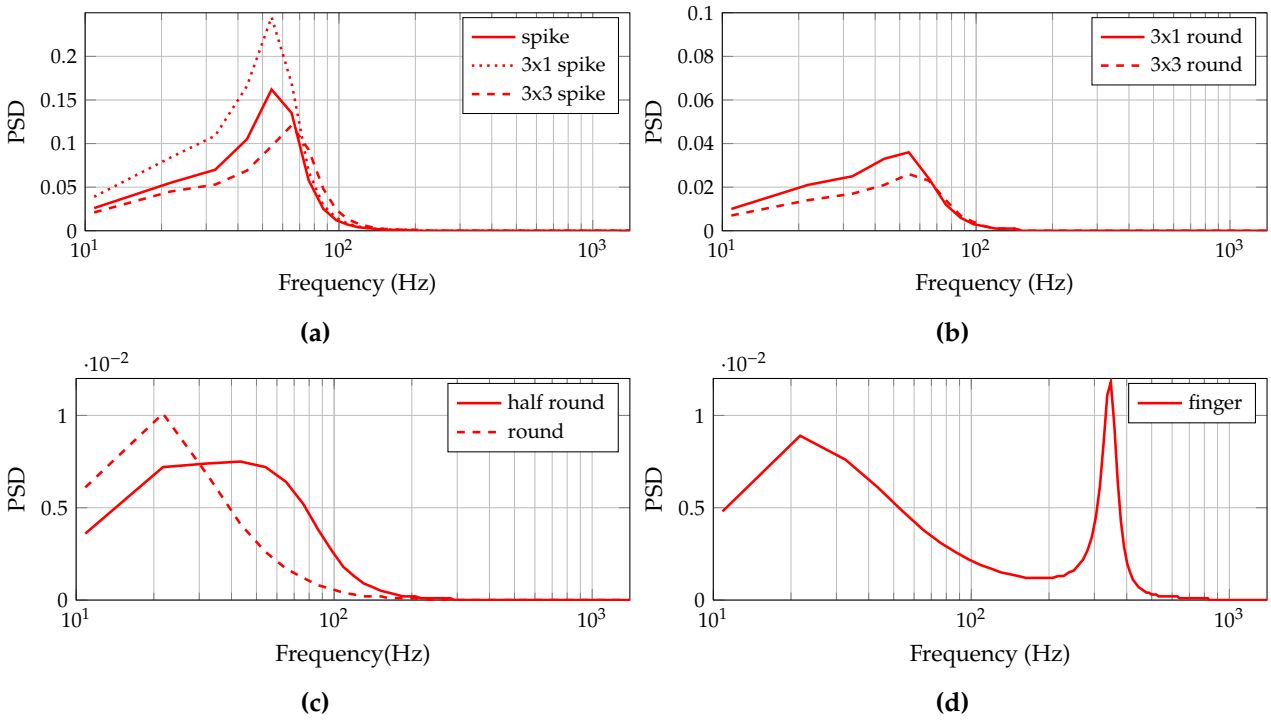


Figure 3.3 Power spectral density functions of signals from the LMT reference database for different tooltips, recorded for the *rubber* material and at *fast* speed.

the peak is less pronounced. Also, the dynamic range is lower than for the spike-shaped tooltips, because the round shape dampens the vibration more significantly.

- Big round tooltips (Fig. 3.3c; *half round*, *round*): For these two tooltips, the peak frequency is reduced to 10 to 30 Hz. This most probably stems from the tooltip shape having a size of roughly 1 cm, corresponding to roughly 15 Hz as frequency contribution of the tooltip vibrating itself. This contribution is probably overshadowing the peak from the material texture. Additionally, the size and shape of the tooltips dampens the signal energy overall with a very low dynamic range in the order of 10^{-2}.

- Fingertip (Fig. 3.3d; *finger*): In general, the fingertip signal is very similar structurally to the *round* tooltip. This is not surprising, given the similar shape and size. However, the fingertip signal has a very distinct additional peak around 350 Hz corresponding to roughly 450 µm. This frequency peak comes from the fingerprint ridges, as has also been shown in [19].

Again, we observe a very high dynamic range, which implies that the choice of tooltip is also a major factor in determining how high the signal energy is going to be. The spike shaped tooltips lead to the highest signal energy.

Finally, we compute the PSDs for the different scan speeds with the material being *rubber* and the tooltip *3x1 spike*. The resulting curves are shown in Fig. 3.4. We observe that for the three higher speeds, the shape of the PSD is practically equal. For the two lowest speeds *slower* and *too slow*, the curves shift towards lower frequencies, probably because of the fact that the speeds are so low that structural features of the texture translate to lower frequencies overall. The signal energy scales with the scanning speed.

In total, we have that the signals in our reference database can differ significantly from each other in terms of frequency composition. This poses a high challenge for a coding system, having to deal with both low and high frequency signals and very different signal energies. Additionally, we see that a lot of the material properties translate into the signal spectra. Therefore, the perceptual models used in the codecs should be based on spectral information.

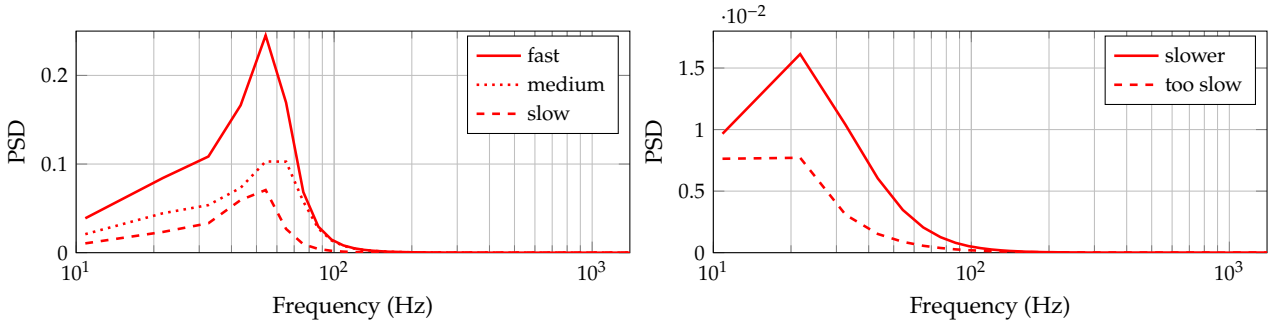


Figure 3.4 Power spectral density functions of signals from the LMT reference database for different speeds, recorded for the *rubber* material and with the *3x1 spike* tooltip.

3.2.4 Data Rate Considerations

Typically, when aquiring vibrotactile signals, they are processed using pulse code modulation (PCM) with 16 bits per sample. With the signal range between -3 and 3 as in the LMT reference dataset, this gives a precision of roughly $9 \cdot 10^{-5}$. For the CEA reference dataset, with signal range $[-1, 1]$, this precision becomes even $3 \cdot 10^{-5}$.

With the sampling frequency of $f_S = 2.8\,\text{kHz}$, the PCM encoding leads to a raw data rate of 44.8 kbit/s for one signal channel. This data rate is sufficiently low to be transmitted over the internet fast and with low delay. However, for a truly immersive experience, multiple points of interaction need to be rendered on the human skin and therefore multiple signal channels have to be transmitted.

Covering the human hand with actuators spaced roughly 1 cm apart gives around 60 points of interaction per hand. Thus, the raw data rate to transmit all channels for both hands would be approximately 5.4 Mbit/s. Extending the skin area that receives tactile feedback even further and increasing the display precision by placing actuators closer can easily lead to data rates of 100 Mbit/s and beyond.

Transmitting data at such a rate over the internet in real time is very challenging, especially alongside accompanying video and audio channels. Therefore, the necessity for compressing the vibrotactile signals becomes evident once again.

3.3 Single-Channel Vibrotactile Codec

In this section, we describe the developed codec named *vibrotactile codec with perceptual wavelet quantization (VC-PWQ)*. The working principle of the codec is inspired by the original MP3 audio codec [34]. This means first, the signals are processed block-wise. Second, a frequency band subdivision is performed. Thirdly, the transformed blocks are quantized, where the quantization is steered to deliver minimal perceptual impairments. Lastly, the quantized coefficients are compressed further by lossless compression algorithms. Nonetheless, all components have been altered significantly, so the processing is optimized for vibrotactile signals.

The encoder structure of the VC-PWQ is shown in Fig. 3.5. The encoder is formed out of individual building blocks to satisfy the modularity requirement described in Sec. 3.1. Also, each block is free of optimization problems and mostly consists of feedforward calculations. For many of the building blocks, efficient software implementations are available. With this, we aim to satisfy the fast execution requirement. In the following, the function of each building block is described in detail.

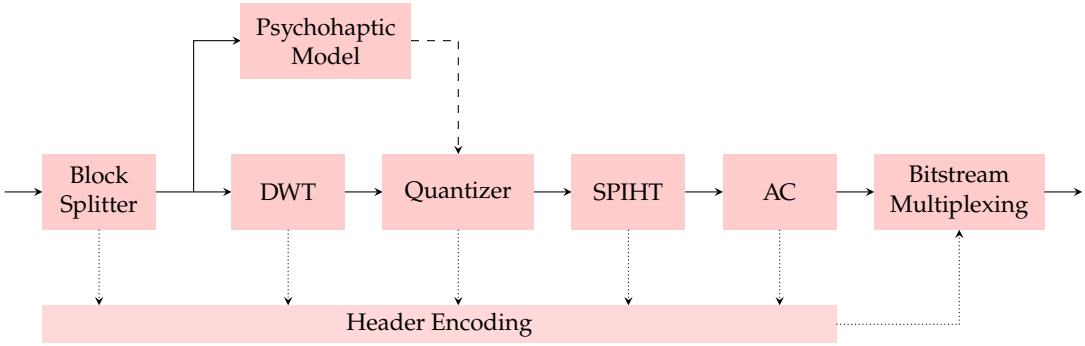


Figure 3.5 Encoder structure of the vibrotactile codec with perceptual wavelet quantization (VC-PWQ). The input signal path is shown by solid arrows, the control signals are shown as dashed arrows and side information is shown as dotted arrows.

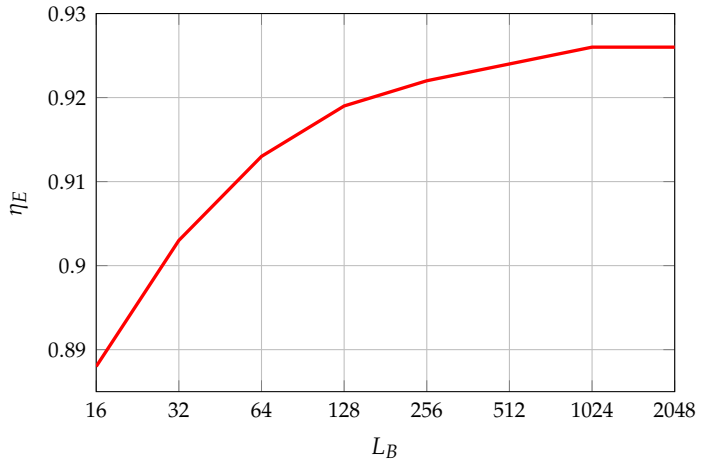


Figure 3.6 Energy compaction efficiency η_E for different block lengths L_{block} averaged over all 280 test signals from the LMT reference dataset.

3.3.1 Block Splitter

As first element of our encoder, we need to design the block splitter. Most relevant in this context is the correct choice of L_{block}. The choice of L_{block} on one hand depends on the restrictions placed by the application at hand. In general, having a longer L_{block} is favorable, because block transforms have better decorrelation and energy compaction capabilities when operating on a higher number of samples. This effect saturates, however, and increasing L_{block} beyond a certain point leads to no practical gain, meaning there is a highest favorable L_{block}. We call this value the cut-off block length $L_{block,c}$.

The value of $L_{block,c}$ depends on the data used for compression. Thus, in order to determine this value, we analyze the available data in our LMT reference dataset. First, we split all available signals into blocks of lengths between 16 and 2048 samples. Then, we compute a single level discrete wavelet transform (DWT) of all obtained blocks using the Cohen-Daubechies-Feauveau (CDF) 9/7 filters. Then, we compute the ratio of the signal energy of all coefficients in the low-pass band to the total signal energy of all coefficients. The computed ratio is called energy compaction efficiency η_E.

In Fig. 3.6 we show the resulting η_E for different block lengths. We can clearly validate the property that increasing L_{block} brings better energy compaction. However, increasing L_{block} beyond 1024 leads to practically no gain. Thus, for these data we determine $L_{block,c} = 1024$. This is the maximum L_{block} we should choose.

Aside from the gain in energy compaction through longer blocks, we should also consider that each block length corresponds to a duration connected by the sampling frequency f_S. In Table 3.1 the

L_{block} \ f_S	2.8 kHz	8 kHz
16	5.7 ms	2 ms
32	11.4 ms	4 ms
64	22.9 ms	8 ms
128	45.7 ms	16 ms
256	91.4 ms	32 ms
512	182.9 ms	64 ms
1024	365.7 ms	128 ms
2048	731.4 ms	256 ms

Table 3.1 Time duration of signal blocks for different block lengths L_{block} and sampling frequencies f_S. Recommended combinations of L_{block} and f_S for online (shorter duration) and offline (longer duration) applications highlighted.

resulting durations for different block lengths for the two most commonly used f_S are given. This block duration can play a crucial role in the codec depending on the application scenario.

On one hand, in an offline application, the block splitting does not introduce any considerable delay, since the signal is available all at once and as the splitting is a fairly simple operation the computational cost is very low. Also, we do not have any buffer delay, since all signal samples are available at once. Thus, we have no upper limit on L_{block} placed by the application in this case. However, too long blocks should still be avoided due to the overhead that comes from the zero padding.

On the other hand, for online applications, the block duration introduces a buffer delay, since a block can only be forwarded after all its samples have arrived. Thus, the block duration in Table 3.1 is added to the other delays of the communication network. Typically, online application scenarios are delay sensitive, because humans can only tolerate a certain amount of delay before having a deprecated experience. Therefore, a smaller L_{block} has to be chosen, despite reduced performance.

Overall, to satisfy the versatility requirement, we choose the codec to allow for the usage of all possible block lengths between 32 and 1024 samples. The author's recommendation is indicated by the shading in Table 3.1. The offline scenario can use longer block lengths, while keeping in mind that block durations over 200 ms can lead to high overhead due to the padding, e.g., when short signals have to be encoded. On the other hand, in an online scenario, the buffer delay should be kept low, in order to avoid fatigue because of noticeable delay.

3.3.2 Discrete Wavelet Transform

For the design of a modern codec, we opt for the DWT, because it allows for a signal analysis in both time and frequency. The DWT processes the signal in the temporal domain. Thus, it is designed to preserve temporal information in its coefficients, i.e., it is not invariant to circular shifting of signal data within a block. This is in contrast to block transform methods like discrete cosine transform (DCT) that are insensitive to spatial information within a block.

At the same time, the DWT separates signal content into different frequency bands, called wavelet bands. Such frequency band analysis is inspired by human perception. Through the different mechanoreceptors in the skin, signal content is processed differently depending on its frequency. Thus, by splitting the signal into different frequency bands and processing them individually, we are able to leverage perceptual effects more efficiently (see also Sec. 3.3.3).

In order to maximize performance, we find the best fitting wavelets for vibrotactile signals. To do this, we develop a scoring method that gives us an estimate on how good the respective wavelets perform on these signals.

From the detailed analysis described in Appendix A, we can select wavelets that are the most promising candidates. The first one is the Haar wavelet. Despite sobering results in the analysis, it is the only orthogonal wavelet that can be extended symmetrically and has linear phase. Since Daubechies (DB) wavelets with more than 9 vanishing moments (VMs) perform well we select the

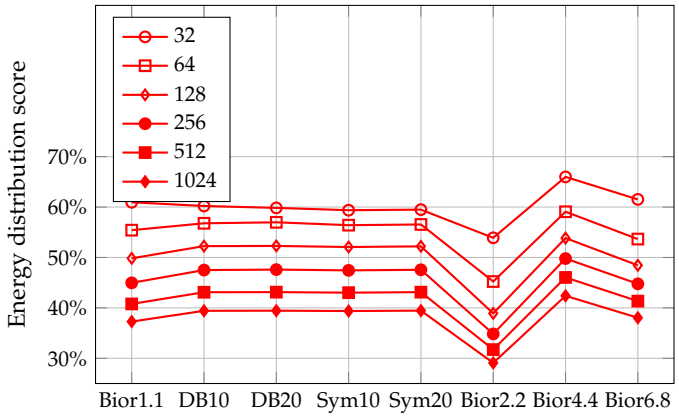


Figure 3.7 Energy distribution score for various wavelets (horizontal axis) and block lengths (legend).

DB10 wavelet as well as the DB20 wavelet for validation purposes. To assess the impact of the phase shift, we include Sym10 and Sym20. Additionally, we include three biorthogonal wavelets. First, the Bior2.2 wavelet that showed good energy compaction behavior. Likewise, we choose the Bior4.4 and Bior6.8 wavelets since they have a low amplification of the input signal and are almost orthogonal.

For the analysis, we always take the highest DWT level possible for the respective block length L_{block}. Then, we compute the so-called energy distribution score (EDS) that considers the energy of the individual signals and the significance of lower level bands as they contain more coefficients. The EDS is computed as follows:

1. We calculate the energy for each band and divide it by the total energy of all bands.
2. We apply the cumulative summation over all wavelet bands starting with the highest band (approximation band) and going down. The last value is always 1 and is removed.
3. We calculate the mean over the cumulative sums from before.
4. Finally, the amplification of biorthogonal wavelets has to be considered. Thus, we divide the mean by the energy amplification, which is calculated by dividing the wavelet coefficient energy by the signal energy of the block.
5. We average over all blocks from the 280 test signals.

We compute the EDS for all 280 test signals in the LMT reference dataset for L_{block} between 32 and 1024. The resulting EDS values averaged over all signals are shown in Fig. 3.7. As can be seen, the Bior4.4 (CDF 9/7) wavelet clearly outperforms the other candidate wavelets. Therefore, this wavelet is chosen for the VC-PWQ. Its coefficients for the low-pass (LP) and high-pass (HP) analysis filters are shown in Table 3.2. The additional benefit of these wavelets is that they are almost orthogonal.

The level of DWT in the codec depends on L_{block} and should allow for sufficient number of samples in each wavelet band to provide a meaningful analysis in later processing steps. This is especially critical for the lowest bands as those are the ones with the fewest samples. In our case, we choose to have at least 4 samples in the lowest wavelet band, to have meaningful energy computation results. However, for the smallest L_{block} of 32, this would imply the lowest wavelet band covering the frequency range lower than 175 Hz for $f_S = 2.8$ kHz. Thus, for this block length, we do an additional level of DWT. All in all, the desired l_{DWT} is then a function of L_{block} as

$$l_{\mathrm{DWT}} = \begin{cases} 4 & L_{block} = 32 \\ \log_2(L_{block}) - 2 & L_{block} \in \{64, 128, 256, 512, 1024\} \end{cases}. \tag{3.2}$$

The total number of wavelet bands is $B = l_{\mathrm{DWT}} + 1$.

n	LP	HP
−4	0.0378	
−3	−0.0238	−0.0645
−2	−0.1106	0.0407
−1	0.3774	0.4181
0	0.8527	−0.7885
1	0.3774	0.4181
2	−0.1106	0.0407
3	−0.0238	−0.0645
4	0.0378	

Table 3.2 Filter coefficients of the CDF 9/7 low-pass (LP) and high-pass (HP) analysis filters. Adapted from [6] © IEEE 2020.

3.3.3 Psychohaptic Model

The psychohaptic model is essential for ensuring the perceptual transparency of the codec. By analyzing the incoming signal blocks through perceptual models, we gain insight on where the most perceivable signal contents are and where pieces of the signal can be neglected without provoking perceptual degradation to the human user. As such, the psychohaptic model aims to provide a computable, machine-readable model of human vibrotactile perception that can be used in algorithms to steer the quantizer accordingly. Overall, the model is the centerpiece for achieving perceptual transparency.

3.3.3.1 Threshold Model Functions

The first part of our perceptual model is the absolute threshold of vibration (ATV). We aim to develop a computable function $t(f)$, which resembles the ATV. From the review in Sec. 2.2.2, we extract that the minimum $t(f_{min})$ should occur at a frequency between 150 and 430 Hz. As mentioned, there is no consensus as to the exact location of that miminum. We choose to build an ATV function with a minimal value at $f_{min} = 250\,\text{Hz}$, since most measurements found it to occur here as explained in Sec. 2.2.2. The function $t(f)$ also can be chosen with different offset overall. This offset is equivalent to an assumed playback volume when the signal is displayed in an actuator. After analyzing spectra of signals from the LMT reference dataset, assuming that all signals should be perceivable, the offset is set to $t_{min} = -77\,\text{dB}$. Additionally, the difference in level of ATV between that minimum value and the one at zero frequency is assumed to be in the higher range between 50 and 70 dB. Therefore, we choose the level of the ATV at $f = 0\,\text{Hz}$ to be $-15\,\text{dB}$, leading to a difference of $t_0 = 62\,\text{dB}$ from the minimum value.

In summary, the ATV function is then computed by

$$t(f) = \left| \frac{t_0}{\left(\log_{10}\left(\frac{f_{sharp}}{f_{sharp}+f_{min}}\right)\right)^2} \left[\log_{10}\left(\frac{f+f_{sharp}}{f_{min}+f_{sharp}}\right)\right]^2 \right| + t_{min}, \tag{3.3}$$

where f_{sharp} controls the sharpness of the overall curve. A smaller f_{sharp} leads to a steeper curve and a more pronounced minimum. In our case we choose $f_{sharp} = 300\,\text{Hz}$ by visual inspection to match the measured threshold shape while taking a conservative approach in that the threshold curve is rather broad to assume more frequencies being perceivable.

Additionally, for higher frequencies the threshold of damage needs to be accounted for. The function $t(f)$ from (3.3) grows to very large values for frequencies above 800 Hz. This does not reflect human perception well, since even for very high frequencies, if the signal amplitude is sufficiently high, this can cause damage or discomfort. For audio signals, the threshold of damage is found to be about 90 dB above the minimum of the absolute threshold of hearing (ATH) [122]. To the best of our knowledge, there are no findings available for the vibrotactile domain on this. Again, we

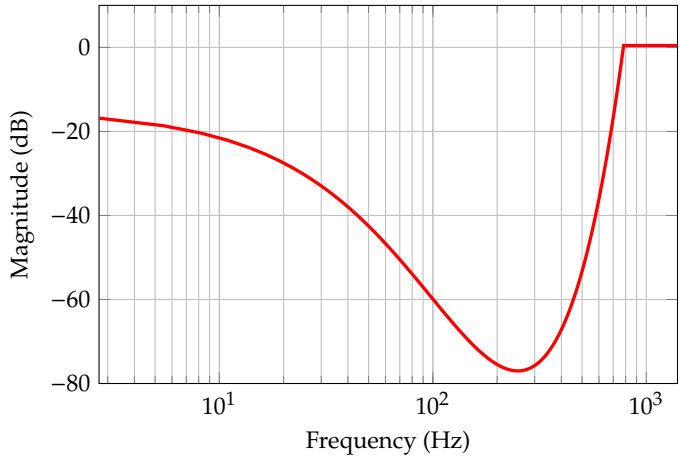


Figure 3.8 Model function $t(f)$ of the absolute threshold of vibration (ATV) on the index fingertip for sinusoidal vibrotactile signals.

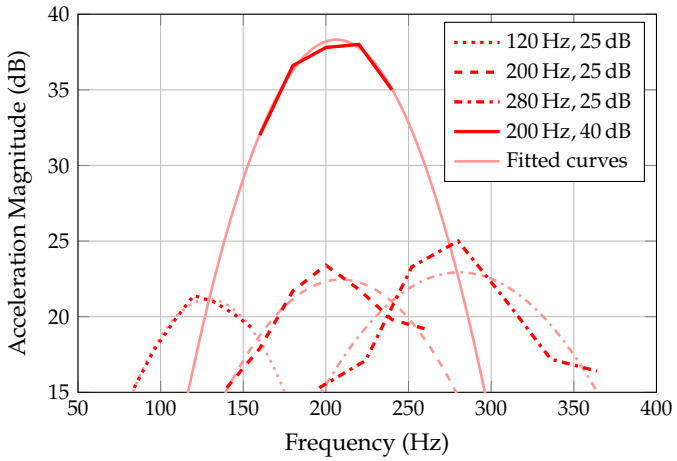


Figure 3.9 Masking thresholds reproduced from [33, Fig. 10b] for four different value pairs of masker frequency f_m and masker level a_m and respective fitted quadratic curves (light red).

therefore choose a conservative approach with a relatively low assumed threshold of damage. Since the minimum of $t(f)$ is $-77\,\mathrm{dB}$ we therefore assume the threshold of damage to be around $0\,\mathrm{dB}$ vibrotactile pressure level (VPL). We alter $t(f)$ to have a cut-off at this value, i.e., $t(f)$ is set to 0 for $f \geq 784\,\mathrm{Hz}$, since we find that $t(784\,\mathrm{Hz}) \approx 0$. The final curve of the computed ATV function is shown in Fig. 3.8. Here, we compute the ATV for frequencies up to $1400\,\mathrm{Hz}$, which is half the sampling frequency of the signals in the LMT reference dataset.

For computations on discrete signal samples, we have a discretized version of $t(f)$, namely $t[m]$. This allows the generation of vectors containing the sampled ATV for different sampling frequencies f_S. This discretized threshold over a frequency grid with N samples is defined as

$$t[m] := t\left(m\frac{f_S}{2N}\right), \quad m \in \{0, 1, \ldots, N-1\}. \tag{3.4}$$

To develop a suitable perceptual masking model, we study the masking thresholds measured in [33] in detail. We reproduce the measured thresholds given in [33, Fig. 10b] in Fig. 3.9. The thresholds were measured for narrowband noise maskers with center frequencies f_m of 120, 200, and 280 Hz with masker level $a_m = 25\,\mathrm{dB}$ above ATV, respectively and additionally with $f_m = 200\,\mathrm{Hz}$ and $a_m = 40\,\mathrm{dB}$ above ATV. The measured curves in Fig. 3.9 give the increase in threshold on top of the ATV.

We fit quadratic curves onto the measured thresholds in order to extract their properties. We see that the masking thresholds are always centered around the masker frequency f_m. With increasing f_m

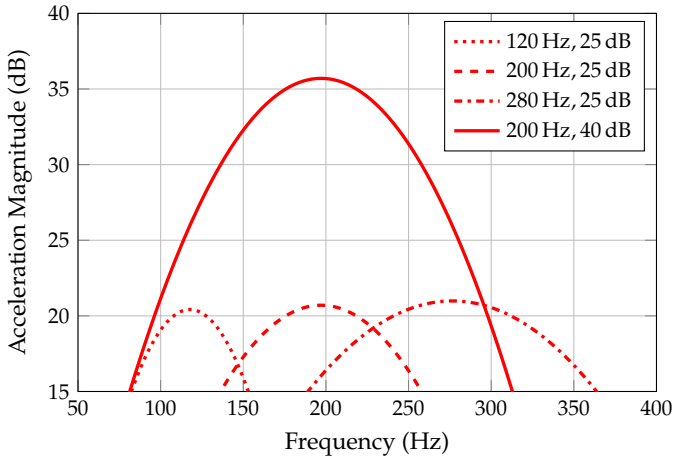


Figure 3.10 Computed model functions for the masking thresholds for sinusoidal masker signals with the four different (f_m, a_m) value pairs as examined in [33, Fig. 10].

and constant masker level a_m, the masking thresholds increase in level and also become wider. From the measurements with $f_m = 200\,\text{Hz}$, we observe that the masking threshold level directly translates with a_m although the shape is mostly unchanged. The fitted quadratic curve is a bit narrower in this case for the higher a_m, but that most likely comes from the two outermost frequencies less that were measured, compared to the curve of $a_m = 25\,\text{dB}$. Therefore, we assume that the masking threshold does not change shape when a_m increases. In general, the maximum masking threshold level is always 2 to 4 dB lower than its respective a_m, however, in [36] it was found to be as much as 10 dB lower.

We take the extracted properties of the masking thresholds to compute masking threshold functions $t_m(f)$ for a masker with frequency f_m and level a_m. We determine $t_m(f)$ as

$$t_m(f) = a_m - 5\,\text{dB} + 5\,\text{dB}\frac{2f_m}{f_S} - \frac{30\,\text{dB}}{f_m^2}(f - f_m)^2\,. \tag{3.5}$$

The first term a_m is responsible for ensuring that the masking threshold translates in vertical direction proportionally to a_m. Then, the term $-5\,\text{dB} + 5\,\text{dB}\frac{2f_m}{f_S}$ leads to the masking threshold maximum value $t_m(f_m)$ being 5 dB lower than a_m for $f_m = 0$. Then, the difference between a_m and $t_m(f_m)$ decreases linearly and reaches zero for $f_m = \frac{f_S}{2}$. Finally, the last term $-\frac{30\,\text{dB}}{f_m^2}(f - f_m)^2$ creates a parabola centered around f_m. We therefore see that (3.5) creates curves that fulfill all the properties of masking thresholds extracted before.

To further validate the developed masking threshold model, we plot the masking threshold curves for the same value pairs (f_m, a_m) as in Fig. 3.9. The resulting curves are depicted in Fig. 3.10. We observe that the computed masking thresholds match the measured ones quite well. The computed thresholds are lower generally, which again resembles a conservative approach, where we place the thresholds at the lower ends of the confidence intervals from [33, Fig. 10].

3.3.3.2 Application of the Psychohaptic Model

Now that we have obtained the ability to compute the ATV and masking thresholds, we can develop the psychohaptic model module. As such, the psychohaptic model should take an input signal block and provide us with the signal-to-mask ratio (SMR) values in the different wavelet bands so they can be used to steer the quantizer appropriately.

The processing structure of the psychohaptic model is shown in Fig. 3.11. First, the incoming signal block is transformed into frequency domain with the DCT. The result is represented in dB. Then, the

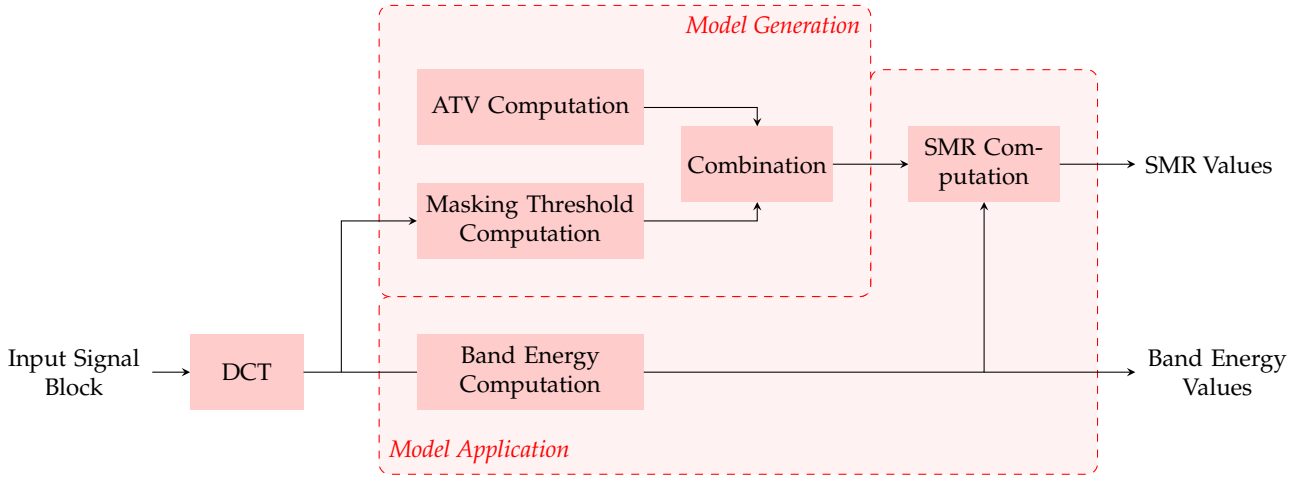


Figure 3.11 Block processing structure of the psychohaptic model.

main perceptual part of the psychohaptic model can be separated into the blocks model generation (MGen) and model application (MApp).

The MGen is responsible for providing a global masking threshold (GMTh) function from the model functions before. The GMTh is the combination of ATV and different masking thresholds from all maskers present in the current signal block. Thus, on one hand the ATV is calculated from the sampling frequency and block size. On the other hand, from the DCT spectrum, peaks are extracted. We assume that peaks in the spectrum with a certain prominence act as maskers. Therefore, we choose a minimum prominence of 15 dB by visual inspection of multiple spectra from the LMT reference dataset. With this we ensure that only quite prominent peaks are identified as maskers. A minimum masker level of −42 dB and a minimum separation of 10 Hz between extracted peaks can optionally also be used to limit the number of extracted peaks somewhat further but are not necessary. The frequencies f_m and levels a_m of all identified peaks are then used to compute masking thresholds according to (3.5). Then all computed thresholds are combined to the GMTh denoted by $t_G(f)$ by taking the maximum across all threshold functions at each frequency.

The MApp then takes the obtained GMTh to compute the SMR. In parallel to the computation in the MGen, the signal energy $E_{S,b}$ in each wavelet band b is computed from the DCT spectrum. Then the SMR is computed for each wavelet band individually. First, we compute the energy $E_{M,b}$ of the obtained GMTh in each wavelet band. Then for each b we compute

$$\mathrm{SMR}_b = \frac{E_{S,b}}{E_{M,b}} \tag{3.6}$$

and transfer the result into dB. The values of the SMR and $E_{S,b}$ for all bands are then passed on to steer the quantizer.

3.3.4 Quantizer

The quantizer in our codec quantizes the incoming wavelet coefficients perceptually. The perceptual quantization is achieved by allocating different number of bits to each wavelet band. For the bit allocation, we follow the same approach based on the SMR, signal-to-noise ratio (SNR) and mask-to-noise ratio (MNR) used in the MP3 codec [34] and described in Sec. 2.3.3. The number of bits that can be allocated to a wavelet band is limited to 15 plus one sign bit. This is done to ensure that the codec will produce signals that have a data rate that is at most as high as the original data rate. The bit allocation process terminates when the sum of allocated bits reaches a predefined bit budget N_{bits}. By scaling the value of N_{bits}, we are able to freely scale the output rate of the coding, achieving the rate-scalability requirement.

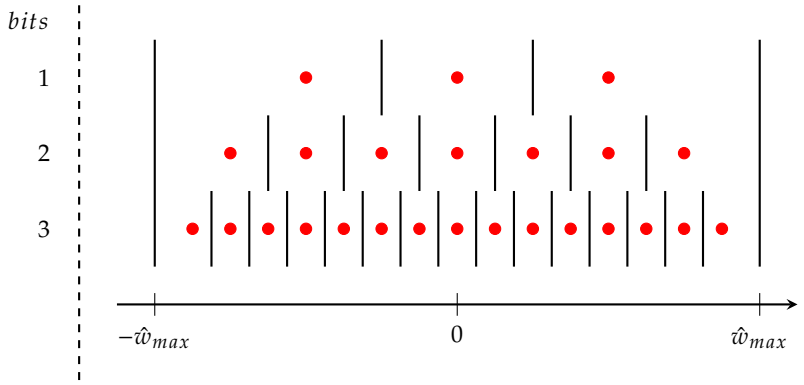


Figure 3.12 Illustration of the quantization characteristic of the embedded values uniform quantizer (EVUQ) with different number of quantization bits. The vertical solid lines denote the quantization intervals and the red dots the quantization levels.

With the perceptual steering, the quantization will produce a different number of quantizer bits in each wavelet band. Therefore, care needs to be taken to design the quantizer appropriately, so that the following set partitioning on hierarchical trees (SPIHT) algorithm works efficiently. For that, we design the embedded values uniform quantizer (EVUQ) as a modified version of embedded quantizers described in [20]. Instead of embedding the quantization intervals, the EVUQ is embedding the quantization values. This behavior is illustrated in Fig. 3.12. By embedding the quantization values, we achieve that for wavelet bands with fewer bits, there are no additional possible quantization values generated than those given by the wavelet band with the maximum number of allocated bits. Thus, the SPIHT algorithm is able to form bitplanes directly across all wavelet bands. The embedding corresponds to an appending of zeros to quantizer bits to reach the same number of bits in all wavelet bands. The added zeros mean that SPIHT is then able to exploit a large number of correlations with zero trees.

The quantizer design also has to be flexible enough to cope with the large difference in dynamic range between signals. For that, we design the quantizer in a way that applies scaling to the signals and outputs the quantized values as integers. For that, we calculate the value w_{max} of the maximum wavelet coefficient in the current block. This obtained value is quantized as a fixed-point number. For $w_{max} < 1$, 7 fraction and zero integer bits are used. For $w_{max} \geq 1$, 3 integer and 4 fraction bits are used. Then, 1 extra bit is used for indicating which mode was used for the quantization of w_{max}. w_{max} is quantized through a ceiling operation, yielding $\hat{w}_{max}$. This quantized maximum value scales the entire quantization range of the quantizer as indicated also in Fig. 3.12.

Overall, the quantizer then operates as follows. First, the quantization interval for each wavelet band Δ_b is determined by

$$\Delta_b = \frac{\hat{w}_{max}}{2^{bits_b}}, \tag{3.7}$$

where $bits_b$ denotes the number of bits allocated to the b-th wavelet band. The calculated wavelet coefficients w can then be quantized to the allowed quantization values with

$$\hat{w} = \operatorname{sgn}(w) \left\lfloor \frac{w}{\Delta_b} + 0.5 \right\rfloor \Delta_b. \tag{3.8}$$

Here, we also add one sign bit implicitly with the $\operatorname{sgn}(\cdot)$ function. The quantization itself is performed without changing the overall range of the coefficients. In the end, all quantized wavelet coefficients are rescaled to integers with

$$\hat{w}_{int} = \hat{w} \frac{2^{bits_{max}}}{\hat{w}_{max}}, \tag{3.9}$$

where $bits_{max}$ is the maximum number of bits allocated over all bands. This formula again shows the embedded principle, where all coefficients are quantized to integers with $bits_{max}$ bits. Then, the

different resolution in each wavelet band is coded by the number of zeros counted from the right in the bit representation of each quantized coefficient. Besides the advantage of SPIHT being able to work with the quantized values directly, we also have that the only side information we need to recover quantized coefficients in their original range at the decoder are $\hat{w}_{max}$ and $bits_{max}$. This is significantly more efficient than having to signal the different numbers of allocated bits for all wavelet bands to the decoder.

3.3.5 Entropy Coding

For losslessly compressing the wavelet coefficients after quantization we use the well-established SPIHT algorithm with subsequent arithmetic coding (AC). SPIHT can be used directly without changes due to the smart design of the quantizer. The SPIHT algorithm then outputs a bitstream that can be further compressed with the AC.

For the VC-PWQ, we implement the AC inspired by [123]. The design of this implementation is advantageous, because many necessary multiplication operations are substituted by more efficient bit shifts. Additionally, efficiency is further enhanced by a sophisticated rescaling method of the coding interval, when it becomes too small. In our implementation, we also check for trailing zeros output by the AC and remove them.

The AC can be designed to be even more efficient by being context sensitive [123], [124]. This means that bits can have different probabilities depending on their semantic meaning, which is called a context. In particular, the SPIHT algorithm has three contexts of its output bits, namely sign bits, refinement bits, and significance map coding bits. The latter is further subdivided into three different cases of significance maps. Each of these different class of bits has a different probability distribution. Thus, the AC encodes them while assuming different probabilities for each of them. This method was presented in [124].

For the VC-PWQ, we extend the approach with an additional context. In particular, we add a context for the side information of signals. In order to encode efficiently, the SPIHT algorithm provides a second output with labels on the context of each bit. Thus, the AC straightforwardly knows, which context is to be assumed. At the decoder side, this information is not available and thus the arithmetic decoder and inverse SPIHT have to work hand in hand to always get the correct context.

In order to obtain the probabilities of the input bits for different contexts, we have to calculate estimates from signal data. For that, we have two options. The first is to calculate probabilities from signal data as a whole, for example from an entire signal block. However, then the calculated probabilities have to be transmitted for the decoder to be able to have this information. This is clearly inefficient, since the data rate increases unnecessarily. The second option, which we go for here, is to calculate probabilities adaptively. For that, counters are implemented that count the incoming zeros and ones. In mathematical terms, we have the counter functions $c_i(0)$ and $c_i(1)$, where i indicates the context in this case. Thus, when a new bit is received from the SPIHT algorithm, the respective counter is incremented by 1. Then, the probability for a zero is estimated as

$$p_i(0) = \frac{c_i(0)}{c_i(0) + c_i(1)}, \tag{3.10}$$

individually for every context i. Then, we have $p_i(1) = 1 - p_i(0)$.

In order to start the process off, we need to initialize the counters. This is to keep the influence of the first bits in check, which would otherwise lead to a highly varying probability in the beginning. Thus, we initialize with $c_i(0) = c_i(1) = 8$ for all i. This, means that initially, we assume $p_i(0) = p_i(1) = 0.5$.

For every new signal block, the probabilities $p_i(0)$ and $p_i(1)$ are carried over. With this, the subsequent blocks can profit from a more accurate estimation of the probabilities. However, since the counters grow larger with each sample, after even one signal block, new incoming bits will have practically no influence on the probabilities. This is especially critical, if the signal statistics change

L_{block}	Code	Total Cost (bit/S)	Theor. Maximum CR
32	1	0.7188	22.26
64	01	0.3906	40.96
128	001	0.2109	75.87
256	0001	0.1133	141.22
512	00001	0.0605	264.46
1024	00000	0.0313	511.18

Table 3.3 Coding of L_{block} in the block header, cost in terms of bits per sample of the entire header and theoretical maximum compression ratio for each of the available block lengths.

over time. In order to solve this, we carry over the probabilities $p_i(0)$ and $p_i(1)$ but not the counters. Instead, the counters are reset to

$$c_i(0) = \text{round}(c_{reset} \cdot p_i(0)) \tag{3.11}$$

$$c_i(1) = \text{round}(c_{reset} \cdot p_i(1)). \tag{3.12}$$

The value of c_{reset} dictates how much the estimation will be receptive for changes in signal statistics. The lower, the more of an influence new incoming bits of a new signal block have. In our implementation of the codec, we choose $c_{reset} = 32$, since we empirically found it to lead to good compression performance using the LMT reference dataset.

3.3.6 Header Encoding

The decoder requires certain side information passed along with the coded bitstream in order to be able to recover the waveform of the CS from it. More specifically, the decoder needs to be able to recover the quantized wavelet coefficients $\hat{w}$ exactly and from them perform an inverse DWT. The required side information is placed in the header of each block. Of course, it is desirable to have the header be as short as possible, because the header bits constitute the minimum amount of data the codec outputs for one block. This leads to a theoretical maximum in CR for the case that just the header and no other bits are transmitted. Thus, decreasing the header size, increases that theoretical maximum CR of the codec. Therefore, we analyze the contribution of the block header towards the data rate with the cost in terms of bits per sample.

In our single-channel codec, the header length L_{header} ranges from 23 bits for $L_{block} = 32$ up to 32 bits for $L_{block} = 1024$. These bits are distributed as follows:

- $1 - 5$ bits: Block length L_{block}
- $10 - 15$ bits: Integer number coding the number of subsequent bits that belong to the current block
- 4 bits: Integer number coding $bits_{max}$
- 8 bits: Fixed point number coding $\hat{w}_{max}$ (and 1 additional signal bit).

The first 1 to 5 bits of the header signal the used block length. This information is coded in a way that makes for a shorter header for lower block lengths. In Table 3.3 we give the codes for the 6 different block length options of our codec. Naturally, with increasing block length, the cost for this part of the header decreases. Therefore, coding in this way that gives the shortest code to the shortest block length is vital.

After the block length, we code the number of bits that follow, which belong to the block currently under consideration. This is coded via a binary unsigned integer. For the shortest block length 32, we use a 10 bit integer, which leads to a cost of 0.3125 bit/S. Then, as the block length is doubled, we increase the number of bits for the integer number by 1 each time. For the highest block length, we then have 15 bits for the integer number, corresponding to a cost of 0.014 64 bit/S.

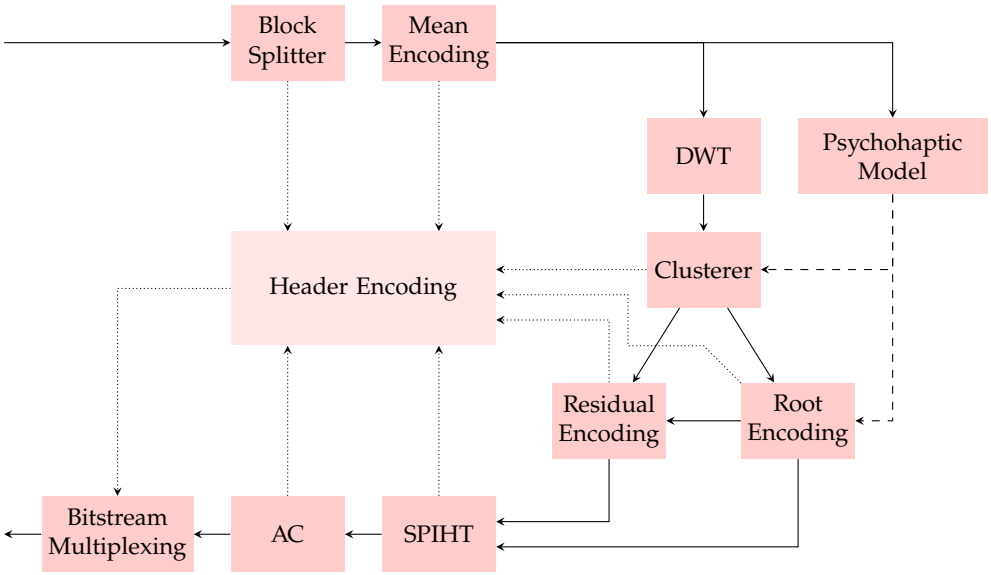


Figure 3.13 Encoder structure of the multi-channel vibrotactile codec (MVibCode). The input signal path is shown by solid arrows, the control signals are shown as dashed arrows and side information is shown as dotted arrows.

The last two parts of the header are the same for every block length. For one, we code $bits_{max}$ with a 4-bit unsigned integer. We can limit this to 4 bits since $bits_{max}$ is limited to 15. Then, 8 bits are used to code $\hat{w}_{max}$ as described in Sec. 3.3.4.

Finally, the total cost of the header and the theoretical maximum CR are also shown in Table 3.3. We see that the longer the block length, the higher the reachable CR is. Therefore, aside from the fact that with longer blocks we can leverage correlation between samples better (see Sec. 3.3.1), this gives another reason why one should always choose the highest L_{block} possible under the given delay requirements.

3.3.7 Single-Channel Compressed Signal Reference Dataset

For the further evaluation, especially in terms of quality assessment, we create a reference dataset of CSs. For that, we compress all 280 signals from the LMT reference dataset with the VC-PWQ. We do so with 17 bit budgets between 8 and 120. The maximum bit budget of 120 leads to exactly 16 bit/S in the quantizer. Thus, in this case we have only practically lossless compression with SPIHT and AC.

For every reference signal (RS), we first store the respective bitstream. From it, we compute the CR and store that as well. Finally, we decode the bitstream to obtain the waveform of the CS. All three quantities are stored together, so for each CS, we can easily obtain the bitstream, the CR and the decoded waveform.

3.4 Multi-Channel Vibrotactile Codec

In this section we introduce MVibCode, which is short for *multi-channel vibrotactile codec*. It builds upon the single-channel VC-PWQ and adapts it to multi-channel signals. Through leveraging inter-channel redundancies, we are able to leverage an impressive boost in performance.

The encoder structure of the MVibCode is shown in Fig. 3.13. After the block splitter, we now introduce a mean encoding to be able to cope with signal data that is not zero-mean. Then, after the DWT, we now perform a perceptual clustering method that groups similar channels together to be encoded jointly. In each cluster, one channel is encoded as so-called root channel (RoC), while for others we encode only residual signal information, therefore calling them residual channels (ResCs).

The RoCs are encoded with the VC-PWQ in full. The ResCs contain difference signals and can therefore be encoded with a different method. In the following, we describe the methods in detail.

3.4.1 Mean Encoding

The test signals from the LMT reference dataset are close to zero-mean and therefore the VC-PWQ did not have to deal with the consequences of having a significant zero-frequency component after DCT. The major problem arising from such a DC component is that the SMR values get distorted and the DWT produces coefficients that are very unbalanced. Since the quantization of the wavelet coefficients scales with the maximum wavelet coefficient, a high zero-frequency coefficient can lead to the quantization scale being much coarser than otherwise.

In order to avoid this problem, we have two options. The first is to subtract the mean or zero state value of each signal before going into the codec. This is fine in most cases because most actuators are not able to display zero-frequency elements anyway. However, in case the mean cannot be removed from the signal, we propose a mean encoding method that extracts the information of the signal mean into the header and subtracts the mean from the signal so it can be coded as a zero-mean signal. Then, the decoder is able to recover the mean from the header and add it to the signal after decoding. The mean encoding block is optional and should be used for non-zero-mean signals only. Otherwise, the extra mean encoding increases only the bitrate without gain.

In order to encode the mean values of a multi-channel signal, we perform three steps. First we calculate the means of each channel $\overline{s}_i$ as

$$\overline{s}_i = \frac{1}{L_{block}} \sum_{n=0}^{L_{block}-1} s_i[n], \tag{3.13}$$

where $s_i[n]$ are the signal samples from the i-th channel. Then, we find the maximum of the computed means and quantize it, i.e.,

$$\overline{s}_{max,q} = Q\left(\max_{i \in I} \overline{s}_i\right). \tag{3.14}$$

Then, for all means, we quantize them with reference to this maximum value as

$$\overline{s}_{i,q} = Q\left(\frac{\overline{s}_i}{\overline{s}_{max,q}}\right). \tag{3.15}$$

In our implementation, we used 8 bits for the quantizer $Q(\cdot)$ when mean encoding was used. All the quantized mean values are placed in the header.

After the means have been calculated and quantized, we compute the nearly zero-mean signals for all channels of the current block as

$$s_i'[n] = s_i[n] - \overline{s}_{i,q} \cdot \overline{s}_{max,q}. \tag{3.16}$$

These signals are then coded.

3.4.2 Perceptual Clustering

For leveraging inter-channel correlations to compress multi-channel signals efficiently, we opt for a clustering approach. This clustering works by identifying channels that are similar and grouping them together. Then, the signals in a cluster are encoded jointly, while the signals that are on their own, are encoded separately.

Our approach meets the demand for high flexibility of multi-channel vibrotactile compression methods as part of the versatility requirement outlined in Sec. 3.1. The flexibility is achieved by a custom clustering algorithm that takes separate channels and groups them together in steps.

3.4.2.1 Hierarchical Clustering Approach

We start by defining the set C which contains the indices for all the channels of a multi-channel vibrotactile signal. The idea of the clustering is to group channels into different clusters, when it is beneficial for coding. On the other hand, if there is no benefit to be gained from clustering, then channels should remain separate. In mathematical terms, the set C is to be divided into subsets that are then encoded jointly.

As a starting point, joint encoding of two channels means that one channel is categorized as the reference channel (RefC) and the other as the residual channel (ResC). Then, the RefC is subtracted from the ResC, i.e., the RefC is used to predict the ResC. The RefC is encoded with the VC-PWQ. The remaining residual in the ResC is quantized. This process is described in detail in Sec. 3.4.3.

The joint encoding method with prediction establishes a hierarchy between channels in a cluster. Thus, it is beneficial to perform the clustering not only by separating channels into subsets, but by directly establishing a prediction hierarchy between them through an iterative process.

This results in a method called *hierarchical clustering (HC)* [125]. In HC, we connect the channels in the set C by directed edges. These directed edges then constitute the edge set $E \subseteq \{(i,j)|i,j \in C\}$. Here, a directed edge (i,j) specifies that channel i predicts channel j. In the beginning of a HC algorithm the edge set E is initialized as an empty set. Then, edges are added to the set one by one.

In order for the encoding and decoding to work properly, we have to place two restrictions on the clustering algorithm:

1. The edges in E cannot form a cycle.
2. Every ResC has exactly one RefC.

These two restrictions together have the consequence that every cluster will have exactly one so-called RoC that is not predicted by any other channel. Therefore, we have two additional sets that aid us in the HC algorithm. First, the set $E_{max} = \{(i,j)|i,j \in C\}$ forms the set of all possible edges that can be formed from the channels in C. Then, we constitute the forbidden edge set E_f. In this set, we store all the edges that have already been added to E or that would violate the restrictions if they were added to E. Thus, with every addition to E, we search for new forbidden edges and add them to E_f.

The decision which edge to add next to E is based on a metric $g_{i,j}$. This metric is generally based on some measure of the distance between the channels i and j. This distance can be with respect to many different quantities, e.g., inter-channel correlation, signal differences, energy considerations or even perceptual aspects. The design of our metric is described in detail in Sec. 3.4.2.2. No matter the specific design, since $g_{i,j}$ resembles the distance of two channels, we can base the decision for clustering on a threshold value g_{thr}.

First, all the metrics $g_{i,j}$ between all possible pairs of i and j are calculated as long as $(i,j) \in E_{max} \setminus E_f$. In every iteration of the clustering algorithm, we find the channels i and j that produce the minimal $g_{i,j}$, denoted as g_{min}. If then $g_{min} < g_{thr}$, that means the respective channels i and j are sufficiently similar to each other, they can be clustered and an edge (i,j) is added to E. This is done until all channels, whose metric is below threshold, have been clustered.

To summarize, we visualize the overall clustering algorithm in Fig. 3.14. The HC algorithm as designed here, processes the channel index set C and outputs a corresponding edge set E where the performed clustering is encoded as edges (i,j). This set E is also forwarded to the header encoding.

The stopping criterion of the algorithm in this case is that there cannot be found any more channel pairs for which the metric is below threshold. This can optionally be further supplemented with a constraint on the cluster size. For example, by restricting the cluster size to 1, the algorithm will not cluster and each signal will be encoded separately with the VC-PWQ.

3.4.2.2 Perceptual Clustering Metric

In order for the HC algorithm to cluster signals correctly, we need to define an appropriate metric $g_{i,j}$. Ideally, the metric should give indication on whether the prediction operation between two signals

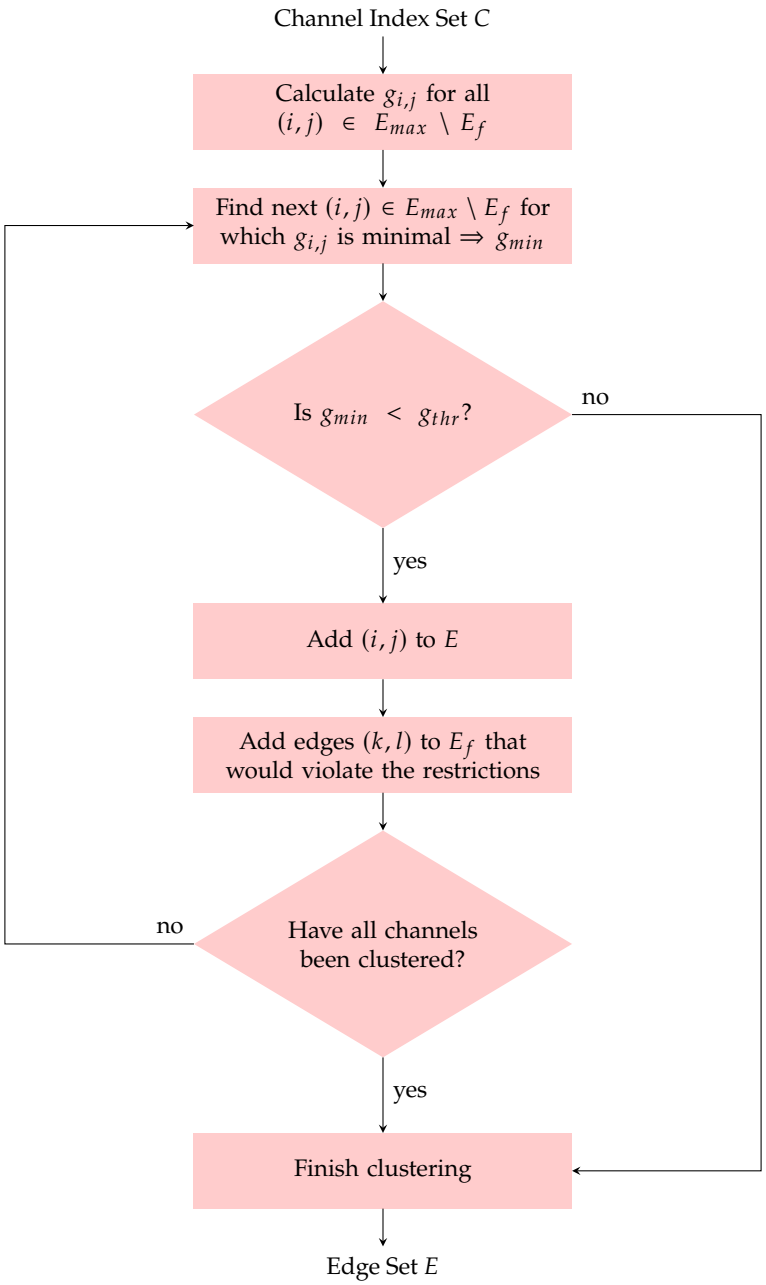


Figure 3.14 Hierarchical clustering algorithm for multi-channel vibrotactile signals in the MVibCode.

would be advantageous. It should be designed to fit to the codec structure and perceptual aspects to deliver an optimal result in terms of perceptual transparency.

We observed that the inter-channel cross-correlation as metric does not lead to satisfying results. Therefore, we develop a gain-oriented metric that directly assesses how the signal energy can be reduced with the prediction operation. We assume that channel i predicts channel j and start by defining the metric as

$$g_{i,j} := \frac{E_{i,j}^{diff}}{E_j}. \tag{3.17}$$

Here, E_j is the signal energy of the j-th channel computed from its wavelet coefficients and $E_{i,j}^{diff}$ is defined as

$$E_{i,j}^{diff} := \sum_{n=0}^{L_{block}-1} (w_i[n] - w_j[n])^2, \tag{3.18}$$

where $w_i[n]$ denotes the n-th wavelet coefficient of the signal block in the i-th channel. Since we divide by E_j in (3.17), we have $g_{i,j} \neq g_{j,i}$.

Intuitively, the metric gives a ratio for the energy gain to be expected by the prediction of channel j from channel i. If the two channels are identical, we have $g_{i,j} = 0$. This means, the gain from the prediction is maximal. If we have $g_{i,j} = 1$, then the prediction does not change the signal energy at all. For a $g_{i,j} > 1$, the energy of the difference is higher than the energy of the RefC itself. Thus, we see that by clustering channels for which $g_{i,j} < g_{thr}$, we group channels for which the prediction provides an advantage. The minimum advantage we require for the decision to cluster two channels depends on the chosen g_{thr}.

Now, the metric so far is only based on the objective similarity of channels. This however, is suboptimal for the codec, because rather than clustering objectively similar channels, we need to cluster perceptually similar channels to satisfy the perceptual transparency capability. In order to modify the metric to be perception-based, we leverage the structure of the codec, more precisely the structure gained from the DWT that separates the signal block into wavelet bands in time domain. Since we have different wavelet bands, we can first split the computation of the metric into each wavelet band, i.e.,

$$g_{i,j,b} := \frac{E_{i,j,b}^{diff}}{E_{j,b}}. \tag{3.19}$$

Here, b denotes the index of the respective wavelet band. Then the individually computed metrics are combined with a weighted sum to the now perceptual metric as

$$g_{i,j} = \frac{1}{\sum_{b=1}^{B} a_b} \sum_{b=1}^{B} a_b g_{i,j,b}, \tag{3.20}$$

where $a_b > 0$ denotes the weight for the b-th wavelet band and B is the total number of wavelet bands. The normalization with the sum of all a_b is necessary to keep the value range of the metric unchanged.

The perceptual aspect of the metric is established by the calculation of the weights a_b. For this, we leverage the information from the psychohaptic model. Specifically, we employ the computed SMR values. As described, the SMR for each wavelet band describes how perceivable the signal is in said band. Thus, a wavelet band with highly perceivable signal should receive a higher weighting than another wavelet band, where signal content is not as perceivable. We found empirically that the SMRs of both channels should contribute to a_b. We choose a_b as the geometric average between both SMRs, i.e.,

$$a_b = \sqrt{\mathrm{SMR}_{i,b} \cdot \mathrm{SMR}_{j,b}}, \tag{3.21}$$

where $\mathrm{SMR}_{i,b}$ denotes the SMR of the b-th wavelet band in the i-th channel. Unlike in the bit allocation procedure, the SMR is used in the linear domain here. We again see the necessity for normalization in (3.20), since the SMR is not constrained to a fixed value range.

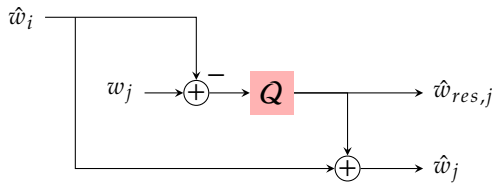


Figure 3.15 Processing of the wavelet coefficients w_j from the ResC j in the residual encoding block of the MVibCode.

3.4.3 Encoding of Root and Residual Channels

After the channels have been clustered, each cluster will contain one RoC and otherwise ResCs. The RoC is the only one that is not predicted from any other channel but only acts as a RefC. Meanwhile, the ResCs have a RefC predicting them, while they can also serve as RefCs for other channels.

Each RoC is encoded in full with the single-channel VC-PWQ. The functionality of this codec was described in Sec. 3.3. The only modification is the mean encoding described in Sec. 3.4.1. Overall, this means that the root encoding module in Fig. 3.13 is an EVUQ with the same bit allocation procedure based on the SMR. Applying the VC-PWQ separately to each RoC means that every RoC will be encoded with the same bit budget N_{bits}. Performing joint bit allocation for all RoCs did not change performance, since these channels have not been clustered, which means that they are perceptually very different to each other. Thus, the separate encoding of RoCs is the most efficient coding method we discovered up to now.

3.4.3.1 Residual Calculation and Quantization

The processing for the ResC wavelet coefficients is shown in Fig. 3.15. First, for channel i predicting channel j, the residual wavelet coefficients of channel j are calculated as

$$w_{res,j} = w_j - \hat{w}_i, \tag{3.22}$$

where w_j are the wavelet coefficients of the j-th channel and $\hat{w}_i$ are the already coded wavelet coefficients from the i-th channel. These $\hat{w}_i$ have either been coded by the VC-PWQ if the i-th channel is a RoC or by the structure in Fig. 3.15 if i represents another ResC.

Then, the obtained $w_{res,j}$ are quantized by the block Q to $\hat{w}_{res,j}$. From these quantized residual coefficients, we can reconstruct the wavelet coefficients $\hat{w}_j$ as

$$\hat{w}_j = \hat{w}_{res,j} + \hat{w}_i. \tag{3.23}$$

These $\hat{w}_j$ can serve as reference for other ResCs. At the decoder $\hat{w}_j$ can be reconstructed from $\hat{w}_{res,j}$ in the same way.

3.4.3.2 Bit Allocation for Residual Quantization

In order to allocate bits to the different wavelet bands in ResCs, we cannot simply reuse the SMR-based method of the VC-PWQ. The reason for this is that the SMR is calculated from the original wavelet coefficients w_j and therefore leads to a bit allocation that is tailored for them. However, the residual wavelet coefficients $w_{res,j}$ have different properties. In particular the lower frequency bands are usually weaker in energy. Therefore, we develop a new bit allocation procedure for ResCs that is based on signal energy.

In order to provide a fast and well-performing method for bit allocation, we use a heuristic. This heuristic directly assigns bits to the wavelet bands b of the i-th channel with the formula

$$bits_{i,b} = \text{round}\left[\frac{N_{bits}}{B} - \log_2\left(\frac{E_i}{E_{res,i}}\right) + \frac{1}{2}\log_2\left(\frac{\sigma_{i,b}^2}{\sqrt[B]{\prod_{n=1}^{B}\sigma_{i,n}^2}}\right)\right], \tag{3.24}$$

Where $\sigma^2_{i,b}$ denotes the variance of the $w_{res,i}$ in the b-th wavelet band and

$$E_i := \sum_{n=0}^{L_{block}-1} w_i^2[n] \tag{3.25}$$

$$E_{res,i} := \sum_{n=0}^{L_{block}-1} w_{res,i}^2[n]. \tag{3.26}$$

Also, if $bits_{i,b}$ turns out to be negative, we set it to 0. The first term in (3.24) allocates all bits from the bit budget equally across all wavelet bands to start with. Then the second term takes into consideration the energy reduction through the prediction. If the prediction reduces the signal energy greatly, then the ratio $\frac{E_i}{E_{res,i}}$ becomes high and thus bits are subtracted for all wavelet bands in the i-th channel. Conversely, if the energy of the residual wavelet coefficients is the same as E_i, then the second term becomes 0. Thus overall, the second term corrects the total bit budget of the entire ResC depending on its energy. Finally, the third term in (3.24) serves the purpose of adapting the number of bits between wavelet bands of the same channel based on their variance. Wavelet bands with high variance receive a bonus in bits, while the ones with small variance get fewer bits assigned to them.

This solution with the heuristic leads to a high efficiency in terms of computational effort, since the bit allocation can be computed in a single step for each channel with no costly loops or optimization algorithms. This is an important contribution towards capability 3 from Sec. 3.1. The bit allocation procedure is adaptive to the quality of the prediction. For good prediction results, the heuristic takes advantage of this to reduce the data rate, which is exactly the desired behavior. However, the heuristic is not designed to take into consideration how the different channels interact but rather treats each channel separately. Thus, an extension to study in the future would be an optimization that allocates bits between channels to optimize signal quality globally.

3.4.4 Parameter Optimization

Different to the VC-PWQ, whose only parameter is the bit budget N_{bits}, the MVibCode now additionally has g_{thr} as parameter. Both parameters will determine the final CR and signal quality. The optimal choice of g_{thr} depends on N_{bits}. So, instead of varying both parameters whenever encoding signals with the MVibCode, we aim to find a mapping from N_{bits} to g_{thr}. Thus, the resulting rate can also be scaled with only one parameter in the MVibCode, therefore satisfying the rate-scalability requirement.

In order to find the appropriate mapping, we conduct an experiment, where we vary both parameters simultaneously. That is, for $L_{block} = 512$ we vary the bit budget from 4 to 120 with 22 different levels. Here, the maximum of 120 bits is chosen as $15 \cdot B$, where B is again the total number of wavelet bands. Then, for each bit budget, we vary g_{thr} from 0 to 4 with steps of 0.25. For every bit budget, we try to find the g_{thr} that is closest to optimal with respect to its mean SNR over CR curve.

We depict the computed close-to-optimal values of g_{thr} over N_{bits} in Fig. 3.16. We see that for lower bit budgets, the close-to-optimal threshold g_{thr} becomes as high as 3.25. This means that for low bitrates, the MVibCode will cluster many channels, even if their prediction does not lead to a reduction in signal energy. This can be explained intuitively by the fact that at the lowes bit budgets large amounts of signal information are discarded in every channel. Thus, the clustering of channels can prove beneficial for highly distorted signals, since it is easier to achieve an energy reduction when performing prediction between them. Overall, it means the codec shifts towards reducing the data rate as much as possible and the relevance of high quality is lessened in this bit budget range.

On the other hand, the g_{thr} reaches 0 for the highest bit budgets. This means that, in this case, channels are almost uniquely compressed by the VC-PWQ. This intuitively comes from the fact that when encoding with such a high bit budget, signal quality is very high. Thus, when computing differences between signal channels, many details remain in the residual signals, which then resemble

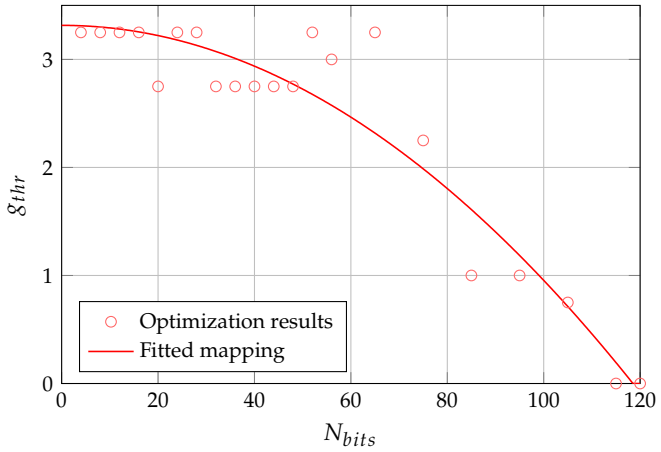


Figure 3.16 Close-to-optimal values of g_{thr} for different values of N_{bits} (dots) and fitted mapping (3.27) (solid line).

noise-like signals. Thus, when aiming to quantize such signals, there is no gain to be expected. This is most probably why it is beneficial to encode each channel individually for high bit budgets.

We now assume a model function for the mapping between N_{bits} and g_{thr} as

$$g_{thr} = -a \cdot \left(\frac{N_{bits}}{B}\right)^2 + b. \tag{3.27}$$

The number of wavelet bands B is used to scale the function of the correct range dictated by the maximum bit budget. This is because the maximum bit budget should always be $15 \cdot B$. The two parameters a and b are determined as

$$a \approx 1.511 \cdot 10^{-2} \tag{3.28}$$

$$b \approx 3.314. \tag{3.29}$$

The resulting curve of the mapping is also shown in Fig. 3.16. We confirm by visual inspection that the mapping leads to the same performance in terms of SNR over CR as the individual close-to-optimal values.

3.4.5 Header Encoding

For the MVibCode, the header is more extensive than for the VC-PWQ. In fact, there is a global header for the entire signal, then a block header for all channels within a block of L_{block} samples and additionally, each channel has another channel-specific header.

The global header contains information on the number of channels. In a sophisticated application setup, this would be part of the handshaking protocol and could therefore be omitted. In this case, we spend 3 bits on coding the number of channels N_{ch}, since we have 8 channels in signals from the CEA reference dataset.

Next, the block header at the beginning of each signal block contains information on the block length, the mean encoding, and the clustering. The block length is coded in the same way as with the VC-PWQ, shown in the second column of Table 3.3.

After that, the mean encoding information is placed in the header. Here, we first start with a signaling bit, which indicates whether the mean encoding was used or not. With this, the mean encoding can be shut off if it is not needed. Then, first the 8 bits for $\overline{s}_{max,q}$ are placed, followed by 8 bits for each $\overline{s}_{i,q}$.

The clustering information is coded in two stages. First, we indicate which channels are members of which cluster. This is done by taking a sequence of N_{ch} bits, which each resemble the numbered

channels from 1 to N_{ch}. Then, the largest cluster is considered and each bit corresponding to a channel that is a member of this cluster is set to 1. Bits from the sequence corresponding to channels that are not in the cluster under consideration are set to zero. For example, if a cluster contains the channels with indices $2, 3, 5$, and 7 out of the 8 channels, the sequence would be 01101010. Then, the bit sequence is reduced to just the bits that were 0 in the first pass and the next cluster is considered. Say, for example, the next cluster contains channels 1 and 8, then the next bit sequence would be 1001. The procedure is then repeated until only singular channels remain that are not part of any cluster. For these remaining channels, a sequence of zeros is produced, whose length is equal to the number of singular channels. Thus, if the sequence starts with only 0, it means that all channels are encoded with the VC-PWQ separately.

In the second stage, the prediction information of each cluster is signaled. For that, we again start with the largest cluster. First, the channel indices are remapped to the range of 1 to $N_{cluster}$, the latter being the number of channels in the cluster under consideration. In the example from before, the indices $2, 3, 5, 7$ would be remapped to $1, 2, 3, 4$. Then, we use $\lceil \log_2(N_{cluster} + 1) \rceil$ bits to code the reference of each channel in the remapped order. Thus, for example if the channel with remapped index 1 is predicted by channel with remapped index 4, we transmit 100 (4 in binary) first. For the RoC of each cluster we transmit the index 0, since it has no reference. If the cluster under consideration contains only one channel, the reference encoding is skipped.

Finally, at the beginning of each channel, the respective header codes the length of the next corresponding bitstream segment as well as the quantization scaling parameters $bits_{max}$ and $\hat{w}_{max}$. This is done in the same way as for the VC-PWQ described in Sec. 3.3.6.

3.4.6 Multi-Channel Compressed Signal Reference Dataset

In order to evaluate the performance of the MVibCode with our quality assessment framework, we generate a reference signal dataset. For that, we take the signals from the CEA reference dataset. These signals are not zero-mean. However, the mean of these signals is irrelevant for display, since the piezoelectric actuators cannot display a constant value. Thus, we subtract the means from all signals and shut the mean encoding in our codec off.

Then, all 25 signals are encoded with bit budgets between 4 and 120 bits with a block length of 512 samples. After coding, we store the obtained bitstreams and determine the data rate and in turn all CRs. These values are then stores as well. Finally, we decode the signals from the bitstream, to obtain the decoded signals and store these as well. Thus, depending on the desired analysis to be conducted, for every RS and bit budget, we have a bitstream, the corresponding CR and the decoded CS.

3.5 Chapter Summary

In this chapter, we presented two compression schemes for data reduction of vibrotactile signals. For the development of the two codecs, we first defined the requirements they should fulfill, namely rate-scalability, perceptual transparency, fast execution, modularity and versatility. So far, to the best of our knowledge, no codec has been presented that is capable of fulfilling all five of these requirements. Subsequently, we analyzed the properties of vibrotactile signals in order to uncover potential challenges for the codec design. Here, we found that vibrotactile signals exhibit a large dynamic range and can have different frequency content, depending on factors such as the recorded material, recording speed and tooltip used. These are aspects that the codec design needs to consider to be able to compress different signals equally well.

After the analyses, we first developed a single-channel vibrotactile codec called vibrotactile codec with perceptual wavelet quantization (VC-PWQ). The main part of the codec features a psychohaptic model that analyzes the frequency content of the incoming signal blocks to compute appropriate control signals for a perceptual steering of a quantizer. The psychohaptic model considers both the human absolute threshold of vibration (ATV) as well as masking thresholds and compares the signal

spectrum to them to determine the signal parts that are perceivable and the ones that are not. As such, the psychohaptic model represents a novel perceptual analysis module that is more extensive than previous approaches. A so-called embedded values uniform quantizer (EVUQ) is introduced to provide a good match for subsequent entropy coding stages and minimize signaling overhead in the header.

The second developed codec is for compression of multi-channel signal data. As such it is named multi-channel vibrotactile codec (MVibCode). Here, we employed a perceptual clustering approach that dynamically groups signal channels. The decision about which channels are to be clustered is based on a newly developed perceptual gain metric. Within a formed cluster, one channel is coded as reference channel and all others are coded predictively as residual channels. These residual channels are coded using an EVUQ with a novel heuristic for the bit allocation that is based on the clustering gain. The performance of both codecs is evaluated in the next chapter.

Chapter 4

Quality Assessment

Since signal information is removed with the quantization in the developed codecs, it is especially crucial to assess the quality of compressed signals (CSs) to make sure we can maintain a high-fidelity human user experience. As such perceptual quality assessment is a key aspect in the signal processing pipeline directly linked to the development of codecs as shown in the figure below. The largest emphasis needs to be given on human factors in this case. That is, we aim to assess the quality of the vibrotactile signals compressed by the codecs from a perceptual standpoint.

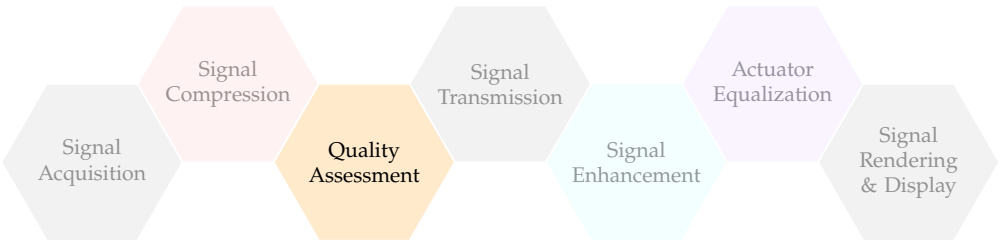
In this chapter we present a comprehensive quality assessment framework based on three main pillars. First, we compute and analyze objective quality metrics for our two developed vibrotactile codecs from Chapter 3. To gain a better understanding and intuition, we showcase exemplary signal waveforms at different levels of compression. We then compare the objective quality performance of the single-channel codec to the state-of-the-art in single-channel codecs. For the multi-channel codec, the comparison is done between its single-channel mode and the true multi-channel processing with enabled clustering. Then, we move on to describe our perceptual quality assessment experiment procedure, with which we can measure perceptual quality scores with human assessors. Finally, we present our development of automated perceptual quality assessment methods, where instead of time consuming experiments, we use computable metrics to grasp perceptual quality. Here, we again use the developed perceptual models. Parts of this chapter have been published in [4], [7], [8].

4.1 Objective Quality Assessment

To assess the objective quality of our CSs, we compute the signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) as described in Sec. 2.4.2.

4.1.1 Overall Coding Performance

At first, we examine the performance of the single-channel vibrotactile codec with perceptual wavelet quantization (VC-PWQ) as described in Sec. 3.3. We take all the signals from the reference dataset described in Sec. 3.3.7 and compute their SNR and PSNR. Then, we compute the average values of both quality metrics over the compression ratio (CR) range from 1 to 40. The average values are computed by binning the CSs by their CR. This means that we have certain intervals of CR and we compute the mean values of SNR, PSNR, and CR of all the CSs that fall within each interval, respectively.



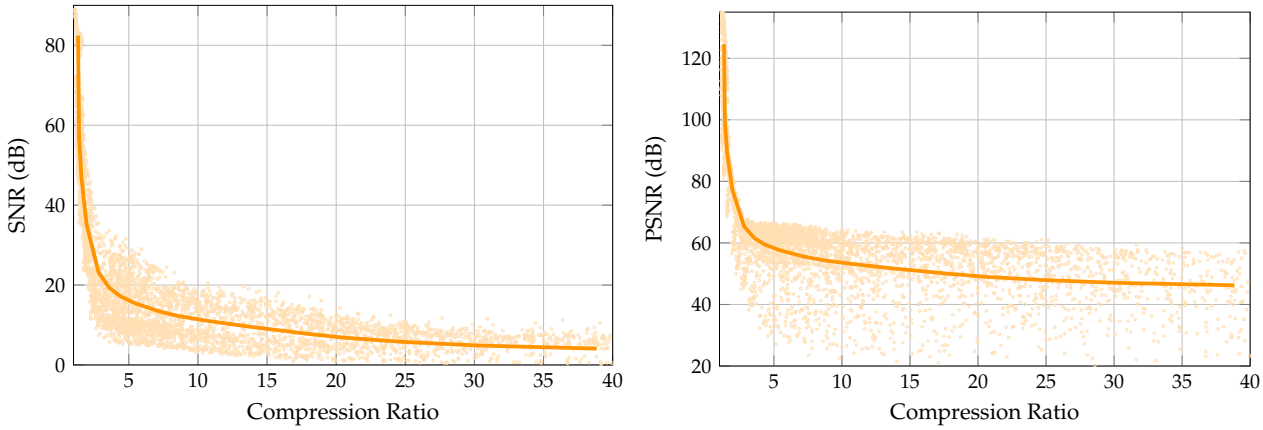


Figure 4.1 SNR and PSNR of signals from the LMT reference dataset compressed with the single-channel VC-PWQ (dots) and average curves (solid lines). Adapted from [7] © IEEE 2021

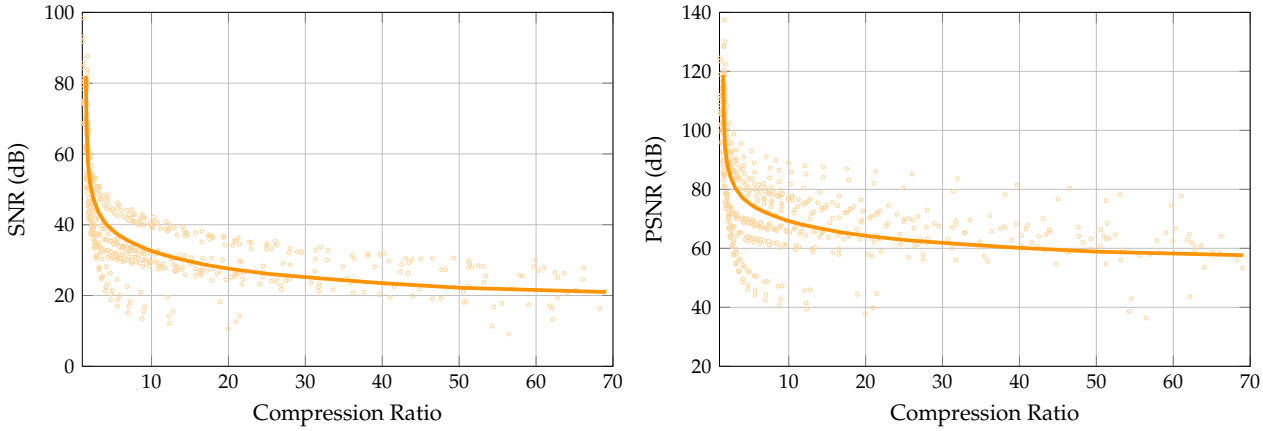


Figure 4.2 SNR and PSNR of signals from the CEA reference dataset compressed with the MVibCode (dots) and average curves (solid lines).

The results are visualized in Fig. 4.1. First, we see the typical behavior of any codec, where the average objective quality decreases with CR in an inversely proportional fashion. With a bit budget of 120, which corresponds to 16 bit/S allocated to each wavelet band, we reach a CR of 1.327. The average SNR is roughly 80 dB and the average PSNR roughly 125 dB at this point. Thus, the codec operates practically losslessly at this point. The CR of 1.327 corresponds to approximately 12.05 bit/S. Therefore, we see that the compression with set partitioning on hierarchical trees (SPIHT) and arithmetic coding (AC) is working quite well in terms of leveraging redundancy.

As the CR increases the objective quality decreases rapidly. For a CR of 10, the average SNR is roughly 11.4 dB and the average PSNR is roughly 53.6 dB. The spread between different signals is significantly higher for the PSNR compared to the SNR. This stems from the vastly different dynamic range of signals as well as the fact that some signals can be compressed better than others.

The rate at which the codec can compress signals is completely scalable through the bit budget. With a bit budget of 8, the codec reaches a CR of almost 40 on average. The highest CR occuring for any signal is 70.

Moving on to multi-channel signals, we examine the performance of the multi-channel vibrotactile codec (MVibCode). Here, we evalute the codec for the signals in the CEA reference dataset. The resulting SNR and PSNR values for each signal as well as average curves are depicted in Fig. 4.2. We again see the clear behavior of the codec to decrease objective quality inversely proportional to the CR. There is again a lossless mode with an average CR of approximately 1.474. However, this

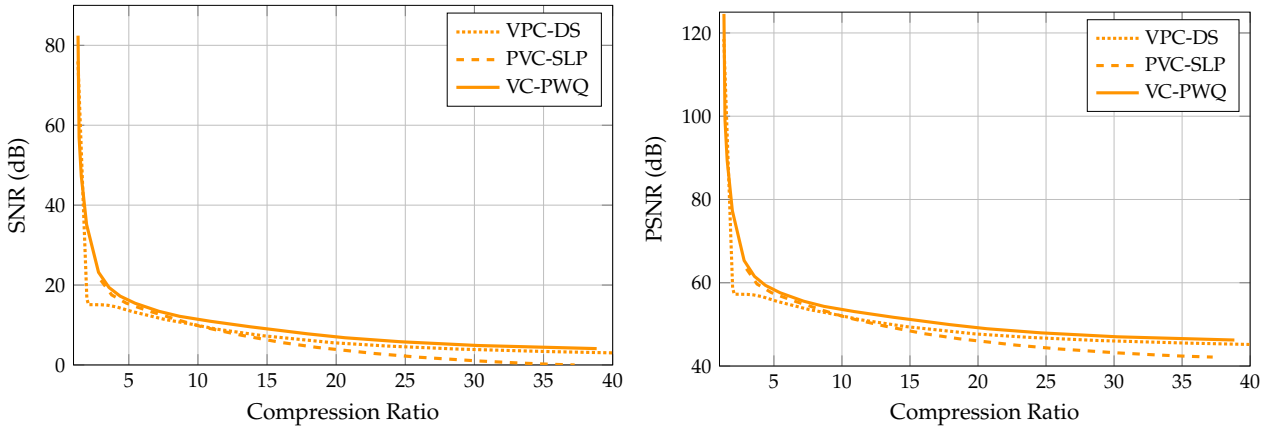


Figure 4.3 Average SNR and PSNR curves of the VC-PWQ compared to state-of-the-art codecs PVC-SLP [43] and VPC-DS [6]. Adapted from [7] © IEEE 2021

increase in CR is attributed largely to the different signal dataset as will also be shown in Sec. 4.1.2. Through the clustering, the codec can easily reach a CR of 70 on average, while the average SNR is around 20 dB and the average PSNR around 57.5 dB. Thus, while the codec compresses much more aggressively than before, objective quality scores still remain comparatively high.

4.1.2 Comparison

Now that we have shown the performance of the codecs from Chapter 3, we aim to put the figures into perspective. For that, we first compare the single-channel VC-PWQ to the state of the art. Specifically, we compare to codecs that are fully rate-scalable and can reach a CR of at least 10. These are the vibrotactile perceptual codec with DWT and SPIHT (VPC-DS) from [6] and the perceptual vibrotactile codec based on sparse linear prediction (PVC-SLP) from [43]. The VPC-DS is an earlier, lower-performance version of the VC-PWQ from this work. Overall, we compute the average SNR and PSNR for all three codecs to compare their performance.

The resulting curves are shown in Fig. 4.3. First, with this objective quality result, we can see that from the two state-of-the-art codecs the PVC-SLP operates better for lower CRs, while the VPC-DS has higher performance in the higher CR range. Our VC-PWQ is able to outperform both codecs for all CRs both in terms of SNR and PSNR. The difference in SNR and PSNR is usually quite small, but even small differences can lead to high dissimilarities in terms of human perception.

For the assessment of multi-channel signals from the CEA reference dataset, we compare the MVibCode with clustering to its single-channel mode. This mode is equivalent to the VC-PWQ because of the backwards-compatible design. The resulting average SNR and PSNR curves are shown in Fig. 4.4. We see that the MVibCode is able to achieve significant gains through the clustering approach. Across all CRs, it achieves on average an approximately 13 dB higher SNR and PSNR. This is a very substantial difference. Such a large difference in objective quality measures is very certain to translate into a perceptual difference.

4.1.3 Impact of Distortions on Signal Waveform

Before transitioning onto the subjective quality evaluation methods, we seek to provide some intuition on the nature of the introduced distortions. For that, we take an exemplary signal from the LMT reference dataset and plot the first 128 samples in Fig. 4.5. The chosen signal is recorded from the material *aluminium grid* with the *3x3 spike* tooltip at *slower* speed.

Then, we visualize the corresponding waveforms of CSs at three different CRs. At CR = 4.96, the signal waveforms are very close to each other. Only slight deviations are visible. Thus, we can intuitively expect that the CS at this CR should have pristine perceptual quality. The SNR of

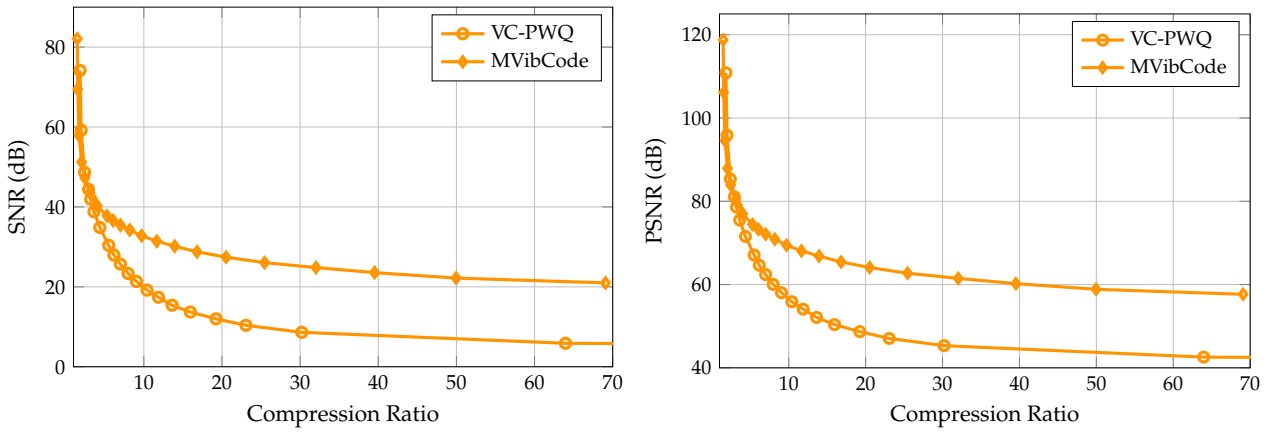


Figure 4.4 Average SNR and PSNR curves of the MVibCode compared to the single-channel codec VC-PWQ.

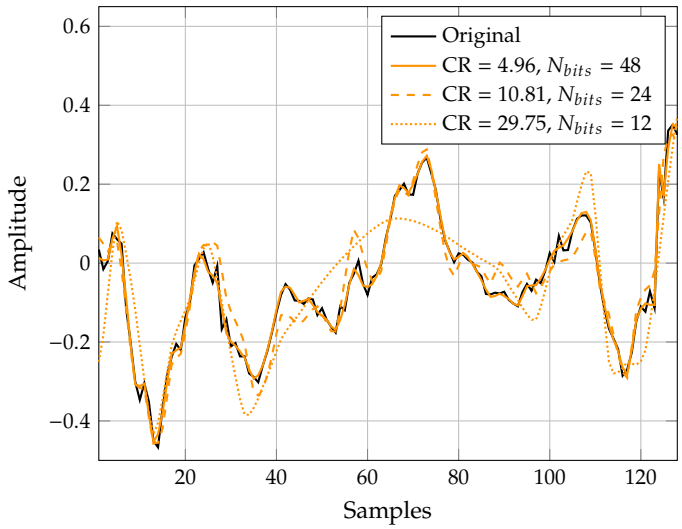


Figure 4.5 First 128 signal samples of an exemplary signal from the LMT reference dataset (black) and respective compressed signal waveforms at three different CRs with their respective bit budgets N_{bits}.

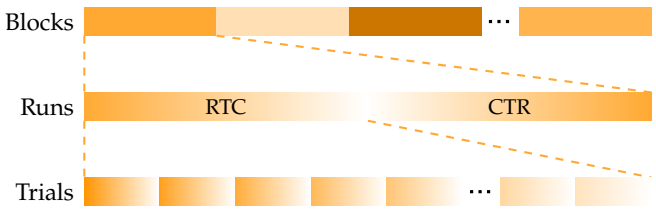


Figure 4.6 Hierarchical structure of the VQA experiment into blocks, runs, and trials. Blocks resemble different test signals. Each block contains two runs (RTC and CTR). Each run contains the same number of trials, resembling CSs at different CRs. Within a block, each rating is done twice, while each test signal is experienced four times. Adapted from [4] © IEEE 2021.

this CS is roughly 20.6 dB. When we roughly double the CR to CR = 10.81, we see that the CS starts deviating from the original more significantly. Subjectively, the compression makes the signal smoother. However, the CS waveform still closely follows the original signal. The SNR is decently high with around 12.4 dB. When moving to a CR = 29.75, we see that the smoothing of the signal is very significant. The essense of the signal waveform is still present, however, it is easily imagineable that there will be perceptually noticeable differences. This is also reflected in the SNR that lies around 6.3 dB.

4.2 Subjective Quality Measurement

As explained, objective quality measures are quite limited in their ability to describe the effects of lossy compression perceptually. In order to assess the perceptual quality of CSs with precision and confidence, we need vibrotactile quality assessment (VQA) experiments with human assessors. For that, we took the multi-stimulus test with hidden reference and anchor (MUSHRA) method from the audio domain as a starting point. To overcome its limitations and mismatches to vibrotactile signals, we adapted many of its aspects to our own VQA experiment, which we describe in detail in the following.

4.2.1 Experimental Design

In contrast to MUSHRA with its unconstrained approach, we design our VQA experiment with a hierarchical structure. In particular, the entire experimental procedure is first separated into blocks. Then each block contains two runs. And finally, each run contains an equal amount of trials. This structure is visualized in Fig. 4.6.

Each block corresponds to one of the different test signals chosen for the experimental procedure. For example, from the 280 signals in the LMT reference dataset, one can choose an arbitrary number of test signals for which to obtain perceptual scores.

In a trial, we compare one CS at some given CR to its corresponding reference signal (RS). For that, the CS and RS are displayed two times each in alternating order. The superordinate type of run then determines the order of CS and RS. This design in visualized in Fig. 4.7. If the run-type is reference-then-compressed (RTC), this means first the RS is displayed, then the CS and then again RS and CS are played back one time each. For the compressed-then-reference (CTR) runs, it is the other way round, i.e., the CS comes first, then RS, CS and RS again. Displaying signals twice with different orders is important to avoid time order effects (TOEs) as described in Sec. 2.4.3. Thus, we eliminate systematic biases from the signal display order. Also, by evaluating each signal at each CR two times, we can get more reliable results as we are able to average the resulting rating score over both runs. Between every displayed signal, we have an interstimulus interval (ISI) of 1 s where there is no signal displayed. This is done to avoid masking and temporal integration effects. After each signal has been displayed two times, the assessor is asked to enter a rating of similarity. This rating is displayed on a scale from 1 to 10 with associated subjective ratings as shown in Fig. 4.8. The scale

Figure 4.7 Trial for evaluating the quality of one compressed signal (CS) compared to the reference signal (RS) for the two different run types reference-then-compressed (RTC) and compressed-then-reference (CTR). Adapted from [4] © IEEE 2021.

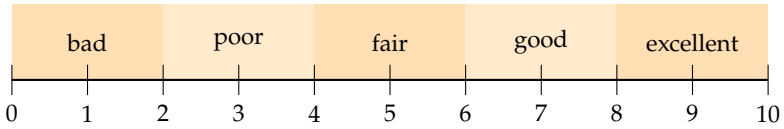


Figure 4.8 Rating scale and corresponding labels for of the VQA experiment.

is chosen to be coarser compared to MUSHRA, because of the lower general level of expertise that assessors have in the vibrotactile domain compared to the audio domain. Still, the scale from 0 to 10 enables a straightforward conversion to percent.

The assessors are guided through each block systematically. In each block, we first perform the RTC and then the CTR run. All trials in a run are randomized, i.e., they are different in order between individual runs. This is to avoid giving cues to the assessors on the quality to be expected.

Altogether, we have as many trials in each run as the number of CSs we aim to test for plus 3. This last addition comes from the inclusion of the hidden reference and anchor signals, much like it is done in MUSHRA. The existence of the hidden reference is known to assessors. The two anchor signals, in particular a low and medium quality anchor, are computed from the respective RS by low-pass filtering. With this, we are able to generate anchors in a controlled way. The low-pass filters used for computation have the parameters given in Table 4.1. We can also compare different codecs in their performance, by including CSs from both codecs at similar CRs, thus mixing codecs on the trial level of the VQA experiment.

4.2.2 Assessor Selection and Post Screening

An important aspect of the design of our method is that we do not require expert assessors. Nonetheless, individual differences in perception between assessors need to be considered [126], [127]. For that, it is essential to collect important demographic data from the assessors, e.g., age, gender, experience in tactile technology, handedness etc. The collection of these data can be done through a questionnaire alongside the experiment.

The experiment has to be set up with methods in the beginning that test the perceivability of signals and the capability of assessors. In order to ensure that all signals are perceivable - in other words, portion of their spectra lies above absolute threshold of vibration (ATV) - we display three sinusoidal signals as part of a pre-test. These signals have 100 ms in length and are at frequencies of 150, 250 and 350 Hz, respectively. We vary the amplitudes of the signals to ensure that the ATV for all assessors

	Medium quality	Low quality
Max. band-pass ripple	±0.1 dB	±0.1 dB
Cut-off frequency f_c	150 Hz	50 Hz
Min. attenuation at $f_c + 25$ Hz	25 dB	25 dB
Min. attenuation at $f_c + 50$ Hz	50 dB	50 dB

Table 4.1 Parameters of the filters used to obtain the anchor signals. Adapted from [4] © IEEE 2021.

is normal and they have no significantly impaired tactile perception. Thus, the pre-test resembles a criterion for prior assessor selection. If assessors are not able to feel the displayed signals, they are not allowed to proceed.

Before conducting the core VQA experiment, the assessors are familiarized with the nature of vibrotactile signals by a so-called training phase. In this training phase, we subsequently display all RSs to the assessors accompanied with two CSs compressed with very high CRs. Then, after having felt all three signals each time, assessors are asked to identify the RS. We set a threshold of 80% of RSs that have to be identified correctly in order for the assessors to be able to proceed to the subsequent evaluation phase.

After the VQA experiment is finished entirely, we need to screen the obtained data for outliers. To detect outliers, we use the median absolute deviation (MAD) method from [128]. The MAD gives an estimate for the absolute deviation from the median of all scores. The advantage of using the median is that it is very insensitive to outliers, which is in sharp contrast to the mean. The MAD is calculated as the median of the absolute deviation from the median of observations. The rejection criterion is chosen as 3 as recommended in [128]. Thus, any value that lies outside of the range of ±3MAD around the median of observations will be considered an outlier. If an assessor produces scores that are classified as outliers more than 50% of the time, his ratings are considered unreliable and removed from the overall dataset.

4.2.3 Experimental Validation and Codec Assessment

By conducting the VQA experiment described up to now, we aim to show its high suitability for the assessment of vibrotactile signals. For that, we record perceptual ratings for the three single-channel codecs VPC-DS [6], PVC-SLP [43] and VC-PWQ.

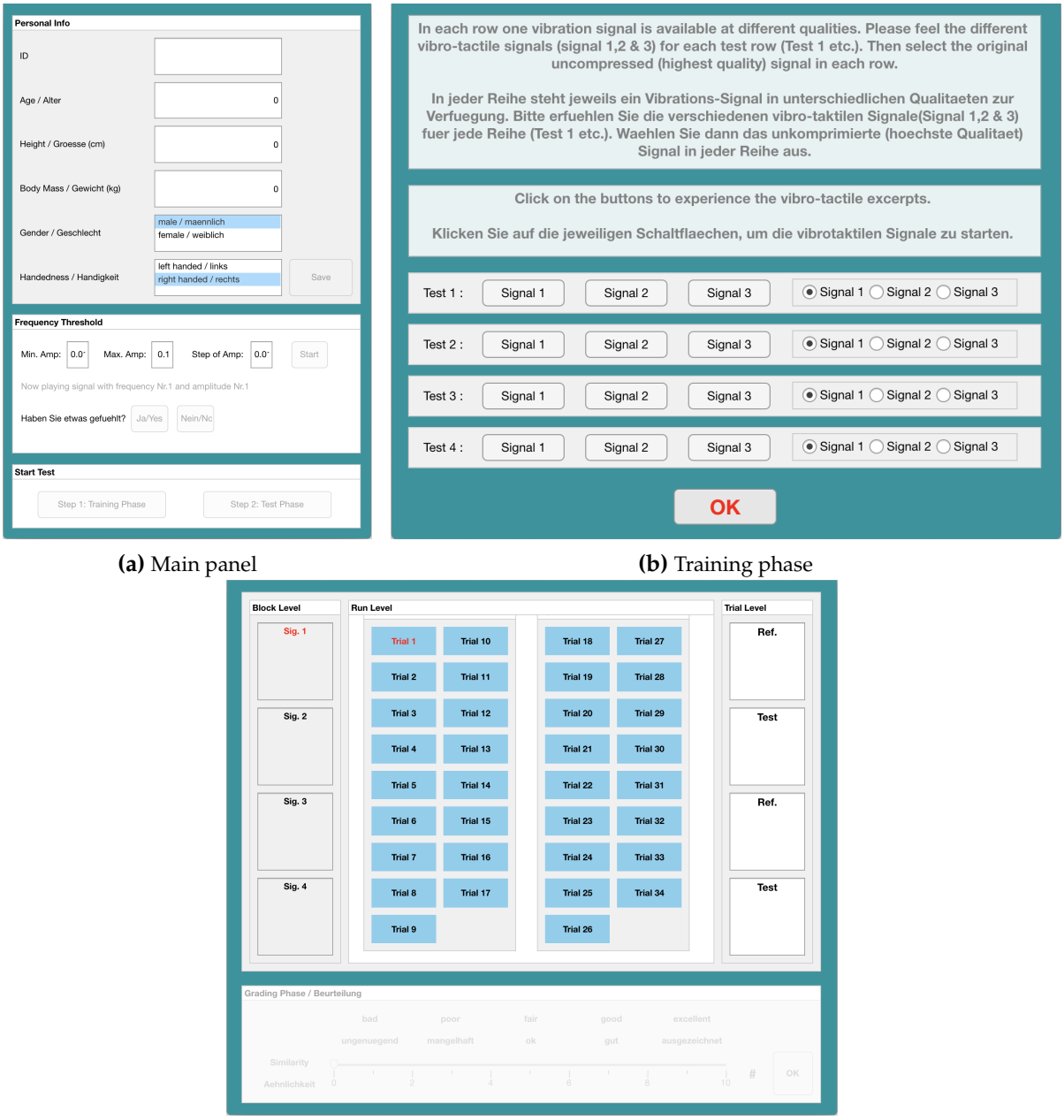
4.2.3.1 Software Implementation

In order to enable the conduction of the VQA experiment, we develop a software tool in MATLAB (MathWorks, USA). The tool was designed to be as intuitive as possible with a 3-step process. First, the users are presented with a questionnaire to enter their demographic information (see Fig. 4.9a). On the same screen, the pre-test is done, where signals are displayed to the assessors. In order to be able to proceed, the assessors have to indicate that they were able to feel the stimuli. Second, the assessors are presented with the panel for the training phase (Fig. 4.9b). Finally, after completing the training with at least 80% correct reference identification, the assessors are directed onto the assessment phase (Fig. 4.9c). Here, the current block, run and trial are shown to the assessors. After each trial, the scale (bottom of the panel) becomes active and the assessors can enter their rating.

4.2.3.2 Test Signal Selection

In order to keep the experiment duration reasonable, we have to restrict assessment to a fairly small subset of the full available signal dataset. For signals that are 1 s long, a trial takes about 15 s. We aim for a duration of maximally around 60 minutes per assessor. Therefore, we can have at most around 240 trials. Since we have 2 runs, that means the product of blocks and trials per run can be at most 120. Therefore, we have a trade-off between the variety of signals we can assess and the amount of CRs we can assess them at.

In order to minimize the amount of blocks, we aim to select signals from the LMT reference database that are representative for the entire set. For that, we first limit ourselves to signals recorded with the *3x1 spike* tooltip, since their amplitude is relatively high and they are quite distinctive according to [5]. In order to find the subset of materials that are representative for the entire set, we calculate signal features from [11]. These features are especially suitable because they are largely invariant to recording speed. Specifically, we compute the features *macroscopic roughness*, *microscopic roughness*, *friction*, and *softness*. Through multidimensional scaling (MDS) we compute the Euclidian distance



(a) Main panel **(b)** Training phase

(c) Assessment phase

Figure 4.9 User interface of the developed software tool for the subjective assessment procedure. Adopted from [4] © IEEE 2021.

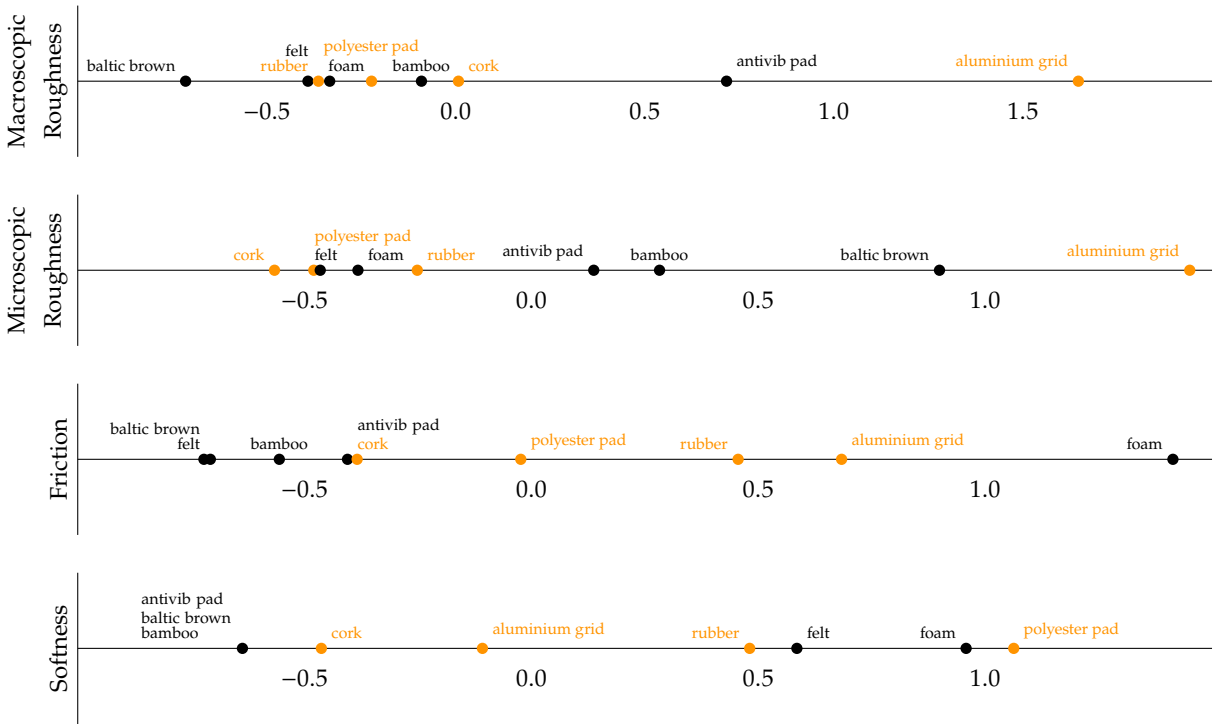


Figure 4.10 Resulting feature space of the surface materials in the signal database from [5].

	fast recording speed	*slower* recording speed
aluminium grid	0.2004	0.0343
rubber	0.0150	0.0030
polyester pad	0.0051	0.0012
cork	0.0042	0.0018

Table 4.2 Signal variance of signals for the chosen materials of interest. Adapted from [4] © IEEE 2021.

matrices of these four features. The materials then span up the feature space as shown in Fig. 4.10. We now choose materials that are able to cover all features sufficiently. We see that we can achieve this by choosing the materials *aluminum grid*, *cork*, *polyester pad*, and *rubber*.

Then, we aim to choose recording speeds that give us a certain variety between signals. For that, we compute the variances of the signals from the four chosen materials at the speeds *fast* and *slower* in Table 4.2. We see that these speeds show a good variety for the signal variance and hence can be chosen for the assessment.

Overall, we now have 8 signals to be assessed in total. This means that we can have at most 15 trials per block. Since 3 trials are reserved for the hidden reference and anchors, we can therefore test for up to 12 different CRs in a single session for this signal choice.

4.2.3.3 Experiment Conduction

The VQA experiment with the described software and signal selection was conducted at the Chair of Lifespan Developmental Neuroscience at the Technical University of Dresden. The details on assessor demographic data and controlled experimental conditions like room or finger temperature are described in [4]. In summary, the two codecs VPC-DS [6] and PVC-SLP [43] were tested in one session, while the VC-PWQ was tested in a separate session. For the VPC-DS and PVC-SLP, in total 20 assessors gave their ratings and 6 different CRs between 5 and 40 were tested. For the VC-PWQ, 10 assessors were gathered and in total 9 CRs between 5 and 45 were evaluated. All participants reported healthy with normal tactile perception. To display the vibrotactile signals, a C-2 tactor [129] was used.

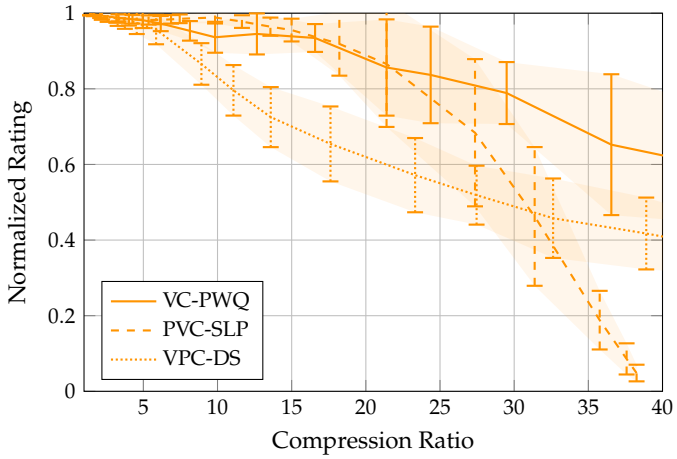


Figure 4.11 Average quality score of the normalized and interpolated subjective quality ratings for the three vibrotactile codecs VC-PWQ, PVC-SLP, and VPC-DS. Adapted from [8] © IEEE 2022.

4.2.3.4 Experimental Results

After the ratings from the assessors have been gathered and stored, we first perform the post screening routine. In the first experiment with 20 assessors, one is classified as an outlier and his ratings are therefore removed from the dataset. For the second experiment with 10 assessors, no outlier was detected.

In order to be able to compare different codecs effectively, we use the hidden reference ratings to normalize the obtained scores. Since the hidden reference is identical to the RS, it should theoretically always be rated with a score of 10. However, that is not the case in practice. Instead, the hidden reference receives a rating between 8.5 and 9 averaged over all assessors for each signal. Additionally, we observe that there is practically no rating of any CS above that of the hidden reference and even if a higher rating occurs, it is by a minuscule amount. Therefore, it is reasonable to assume that if a CS is rated similarly as the hidden reference, it is perceptually indistinguishable from the RS. Therefore, this CS should theoretically be scored at 10 as well. In another sense, through the hidden reference, we can map the raw range of obtained scores to the actual range that perceptual rating should have, i.e., 0 (very bad quality) to 10 (perfect perceptual quality). In order to obtain the normalized scores, we divide all raw scores by the score of the respective hidden reference. This is done for the scores averaged over all assessors, but individually for each signal. The normalized scores are then in the range of 0 to 1. If a normalized score is slightly above 1, it is set to exactly 1.

As described in Sec. 3.3.7, the CSs are available at 17 different CRs each. Since the scores were obtained for only 6 CRs for the VPC-DS and PVC-SLP and 9 CRs for the VC-PWQ, we need to interpolate back onto the original 17 CRs to allow for a fair comparison. This is especially important when aiming to compare the measured scores to computed metrics. To obtain the interpolated scores, we use the `interp1` function in MATLAB. The ratings for each signal are interpolated individually. As interpolation method, `makima` is used. This is because this method is as capable as `spline` methods but has significantly less over- and undershoots at the outermost edges of the interpolated CR range. By inspecting the interpolated ratings visually, we ensure that the interpolation only creates intermediate rating values and does not distort the overall shape of the rating-CR curves.

Now, we compute the average of the normalized, interpolated ratings over all 8 signals for each codec. The resulting average curves are shown in Fig. 4.11. Here, we also depict the standard deviation intervals. We analyze the curves for different ranges of CR:

- **CR from 1 to 5:** The average normalized ratings for all codecs are approximately 1. The standard deviations show minute variations. Thus, in this range of CR, all codecs operate perceptually transparently.

- **CR from 5 to 15:** In this range, the quality of the signals compressed by the VPC-DS declines rapidly. This most probably stems from suboptimal codec design choices, like deadzone quantization that discards too much necessary signal information. On the other side, the other two codecs are still rated with a rating close to 1. The VC-PWQ shows slightly higher standard deviation than the PVC-SLP, which is mostly to be attributed to the smaller number of assessors.
- **CR from 15 to 30:** In this range, the quality of signals compressed by the VPC-DS declines further, while for the other two codecs, quality starts to lower steadily. At a CR of approximately 20, the curves of the VC-PWQ and PVC-SLP start to separate. By the time we reach a CR of 30, the VC-PWQ still achieves a mean score of approximately 0.8, while the PVC-SLP and VPC-DS perform with a mean rating of around 0.5.
- **CR above 30:** Beyond a CR of 30, the mean score of the PVC-SLP plummets quickly to almost 0. This is because the CSs from this codec essential become zero signals and the codec cannot compress further. The quality of CSs from the VPC-DS remains relatively steady and reaches 0.4 on average for CR = 40. For the VC-PWQ, we have a steady decline in quality down to around 0.6 as mean score at a CR of 40.

Overall, we observe that the developed VQA experiment method is highly suitable for evaluating the quality of compressed vibrotactile signals. It also makes comparison between different codecs straightforward. The detailed statistical analysis that confirms that the experiment leads to statistically signficant results can be found in [4].

4.3 Automated Subjective Quality Assessment

In order to avoid time-consuming experiments, we develop computable metrics that are able to more accurately predict the measured rating scores. In this section, we first evaluate the codecs VC-PWQ and MVibCode with the state-of-the-art spectral-temporal similarity (ST-SIM) metric. Then, we present two novel approaches for automated VQA.

4.3.1 Codec Evalulation with ST-SIM

We compute the ST-SIM for all CSs coded by the VC-PWQ in the reference dataset described in Sec. 3.3.7. We visualize the median curve of the ST-SIM over CR. In this case, we choose the median instead of the mean since ST-SIM is restricted to lie between 0 and 1 and thus we need to avoid floor and ceiling effects. The results are shown in Fig. 4.12. We see that the shape of the median curve matches the actually measured results much better than the average curves of SNR and PSNR. However, the ST-SIM values turn out to be higher in general than the measured scores.

We then compare the VC-PWQ to the other two codecs VPC-DS [6] and PVC-SLP [43] in terms of ST-SIM. The median ST-SIM curves are shown in Fig. 4.13. We see that in general, the ST-SIM can grasp the relationships between the codecs that were found in the score measurement data. That is, for low CR, the VC-PWQ and PVC-SLP perform very well, while the VPC-DS declines. Then, the curve for the PVC-SLP crosses that of the VPC-DS and goes to zero. However, the curves are generally shifted compared to the measured scores. For example, the crossing point between PVC-SLP and VPC-DS occurs already at a CR of 15 and not around 30, where it was measured to be. Thus, the ST-SIM seems to correlate with the measured scores and can be used to coarsely evaluate codecs in comparison. Nonetheless, it is not suitable as a substitute for the measured scores.

Now, we evaluate the MVibCode perceptually with the ST-SIM. We compute the metric for all CSs from the reference dataset described in Sec. 3.4.6. We again compute the median ST-SIM from the obtained data. Then, we compare the obtained median curve to that achieved by the single-channel VC-PWQ in Fig. 4.14. We see that already for low CRs the two curves start to separate. From a CR of 10 onwards the difference between the curves increases rapidly. The difference between the two

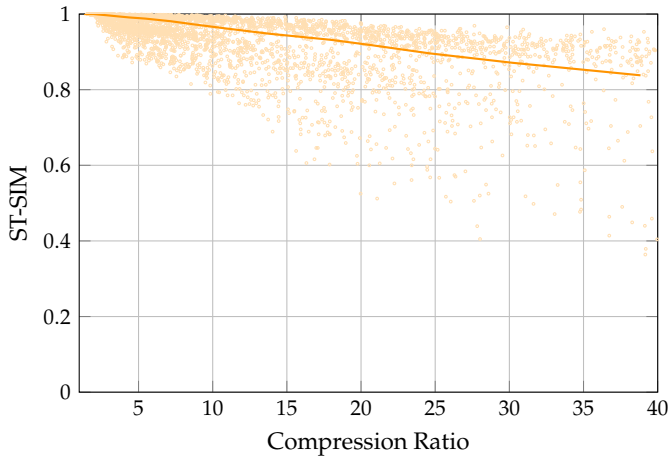


Figure 4.12 ST-SIM of signals from the LMT reference dataset compressed with the single-channel VC-PWQ (dots) and median curve (solid lines).

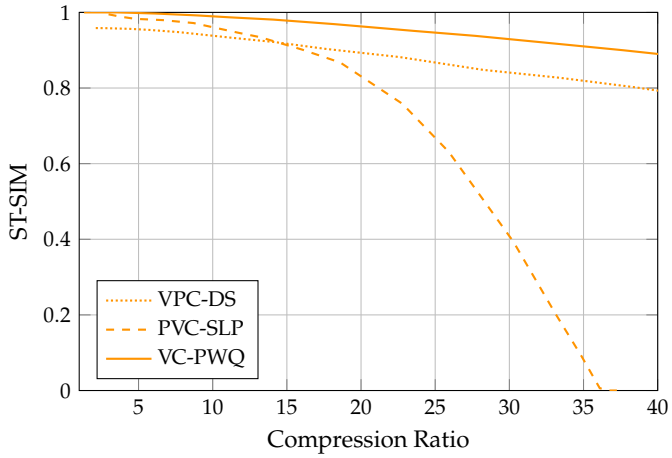


Figure 4.13 Median ST-SIM curves of the VC-PWQ compared to state-of-the-art codecs PVC-SLP [43] and VPC-DS [6]. Adapted from [7] © IEEE 2021

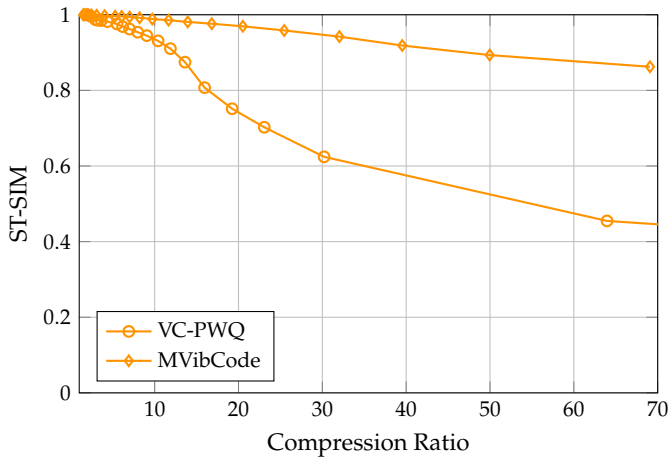


Figure 4.14 Median ST-SIM curve of the MVibCode compared to the single-channel codec VC-PWQ.

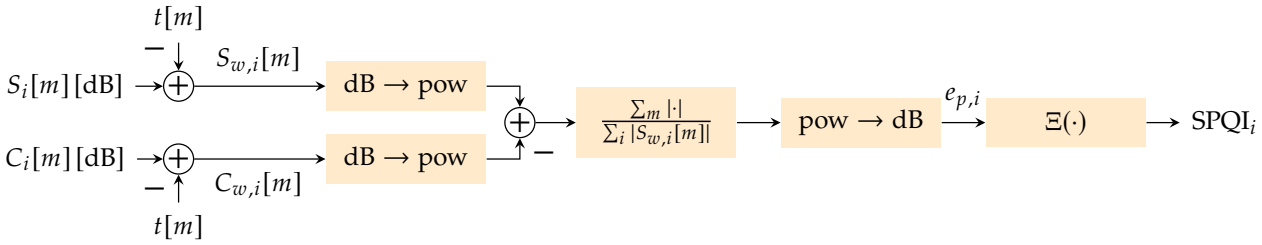


Figure 4.15 Process of computing the SPQI from two signal block spectra $S_i[m]$ and $C_i[m]$. Adapted from [8] © IEEE 2022.

curves can be as high as 0.35 on the scale 0 to 1. This difference is huge and would certainly lead to a substantial difference in VQA experiments. Thus, we can confidently say that the MVibCode operates in a class of its own for the tested multi-channel signals.

4.3.2 Computed Metric Performance Criteria

As we strive to develop novel, better performing perceptual metrics, we first define the criteria of evaluation. Here, in order to evaluate a computed metric, we compute the mean square error (MSE) and Pearson correlation (PC) between the calculated metric scores and the VQA experiment scores. It needs to be emphasized that calculations of these two measures should always be conducted on the entire available signal dataset and not on mean scores. Thus, in our case, for the MSE, we first calculate the squared error between computed scores and measured scores for each signal over the available CRs. Then, we average over CR and then over all signals. For the PC, we perform calculations over the entire matrix of ratings from signals and CRs.

On one hand, the PC is highly suitable for this assessment, because we can gain insights on how much the computed scores correlate with the measured scores. Correlation matters a lot in this context, because in quality assessment, we are often interested in differences in performance between codecs. Thus, if PC between computed and measured scores is high, we can have confidence that if a codec performs better than another codec in terms of a certain metric, then the actually measured scores would show the same.

On the other hand, the MSE is obviously a suitable measure for evaluating metric performance, because it grasps the difference in scores in an objective way. With this information, we can assess, whether a computed metric is able to fully substitute the VQA scores and thus eliminate necessity for time-consuming experiments.

4.3.3 Spectral Perceptual Quality Index

Since the ST-SIM is not able to accurately reflect the measured VQA experiment scores, we aim to develop a novel subjective metric. This metric is called *spectral perceptual quality index (SPQI)*.

4.3.3.1 Metric Design

The SPQI for a CS $c[n]$ with respect to its RS $s[n]$ is computed by first dividing both signals into blocks $c_i[n]$ and $s_i[n]$. These blocks are then first transformed to the spectral domain with a discrete cosine transform (DCT), which yields the real-valued magnitude spectra $C_i[m]$ and $S_i[m]$.

We visualize the computation of the SPQI for one block i in Fig. 4.15. First, we start by taking the spectra $C_i[m]$ and $S_i[m]$ and subtracting the ATV from them. For that, we take the function $t[m]$ as defined in (3.4). Since this operation is performed with all spectra being denoted in dB, it corresponds to a filtering of the signals $c_i[n]$ and $s_i[n]$. With this filtering, we amplify the frequency content of each block that is above threshold and we dampen the parts that are below. This leads to perceptually weighted spectra $S_{w,i}[m]$ and $C_{w,i}[m]$. Then, we remap these spectra from dB to power. After that, the

Metric	VC-PWQ [7]	PVC-SLP [43]	VPC-DS [6]
min MSE SPQI	0.006	0.028	0.005
MSE ST-SIM [67]	0.017	0.009	0.064
max PC SPQI	0.843	0.876	0.960
PC ST-SIM [67]	0.837	0.964	0.921

Table 4.3 Minimal MSE and maximal PC of the SPQI compared to MSE and PC of the ST-SIM for the vibrotactile codecs VC-PWQ, PVC-SLP, and VPC-DS. Best values for each codec are shaded in color. Adapted from [8] © IEEE 2022.

$C_{w,i}[m]$ is subtracted from $S_{w,i}[m]$ in the (linear) power domain. Thus, we get a difference spectrum that is perceptually weighted. Next, we average over all samples of this difference spectrum. To do that, we calculate the sum of the absolute values of the difference spectrum. Then, we divide the result be the sum of the absolute values of $S_{w,i}[m]$. Since the spectra are respresented in terms of power, forming sums of the absolute values is equal to computing the signal energy. Thus, we divide the energy of the perceptually weighted difference by the energy of the original signal. Then, the obtained value is remapped back to dB.

Overall, this results in $e_{p,i}$. This value is a measure for the perceptual error between $c_i[n]$ and $s_i[n]$. This is because it is calculated between perceptually weighted spectra in a way similar to computation of the MSE. The MSE can equivalently be calculated in the spectral domain because of Parseval's Theorem. Thus, $e_{p,i}$, in difference to an objective measure, provides insight into how much of a signal error is actually perceivable.

Now, it can be assumed that humans in their somatosensory processing have a nonlinear mapping of this error to subjective quality. That means, starting from low values of $e_{p,i}$, humans tolerate the perceptual error up to a certain point without determining signals to have bad perceptual quality. However, as $e_{p,i}$ reaches a certain threshold, the perceptually relevant error will become noticeable and quality decreases subjectively for human users. Inspired by this, we aim to map the error $e_{p,i}$ onto a quality score through a nonlinear mapping $\Xi(\cdot)$. In order to obtain the desired behavior, we define

$$\mathrm{SPQI}_i = \Xi(e_{p,i}) := \frac{1}{2}(1 - \tanh(\kappa(e_{p,i} - \tau))). \tag{4.1}$$

Here, τ determines the aforementioned threshold value and κ defines the slope of decline around this threshold. The function $\Xi(\cdot)$ maps $e_{p,i}$ to approximately 1 for $e_{p,i} \ll \tau$ and to approximately 0 for $e_{p,i} \gg \tau$. For $e_{p,i} = \tau$, we have $\Xi(e_{p,i}) = 0.5$. Thus, we see that the mapping has the desired behavior.

Determining correct parameters τ and κ is crucial for an accurate metric calculation. However, to the best of our knowledge, there is no evidence that would empower us to find appropriate values for τ and κ that are founded on human perceptual aspects. Therefore, we find optimal parameter values by using the signal data that we have perceptual scores for from Sec. 4.2. We perform the derivation of these parameters in Sec. 4.3.3.2.

In the end, we obtain the value SPQI_i, which is the SPQI for the i-th block. The SPQI of the entire CS is obtained by averaging the SPQI_i values over all blocks i.

4.3.3.2 Evaluation

To evaluate the ability of the SPQI to produce accurate scoring results, we compute it for the 8 chosen test signals from Sec. 4.2.3.2 at all available 17 CRs for a range of parameters τ and κ. Specifically, we vary the threshold parameter τ in the interval $[-5\,\mathrm{dB}, 0\,\mathrm{dB}]$ with increment of 0.1 dB. The slope parameter κ lies in the interval $[0, 1]$ with an increment of 0.05. Then, we compute the PC and MSE as described in Sec. 4.3.2. Again, we choose $L_{block} = 512$ samples.

First, we calculate the maximally achieved PC and minimally achieved MSE. These values are given in Table 4.3. Here, we compare minimal MSE and maximal PC of the SPQI to values achieved by the ST-SIM. We see that for the VC-PWQ and VPC-DS [6], the SPQI can achieve significantly better results than the ST-SIM. As seen in Fig. 4.13, these two codecs are the ones, where the ST-SIM scores are off

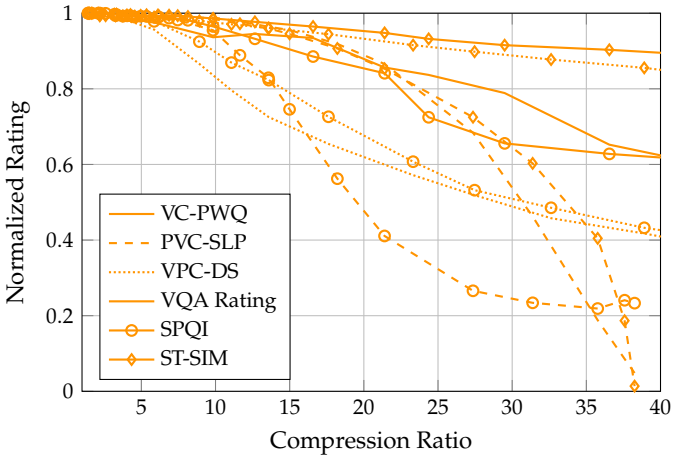


Figure 4.16 Comparison of the SPQI (circle) and ST-SIM (diamond) to the VQA experiment ratings (no marker) for the three vibrotactile codecs VC-PWQ (solid), PVC-SLP (dashed), and VPC-DS (dotted).

from the actually measured scores. Conversely, for the PVC-SLP, even the best choice of parameters in the SPQI cannot lead to a better match towards the measured scores than what the ST-SIM achieves.

In light of these findings, we focus now only on the VC-PWQ and VPC-DS and find close-to-optimal parameters τ and κ. When optimizing with respect to PC, i.e., we aim to find the parameter choice that can maximize this measure, we receive $\tau_{opt,PC} = -3.1\,\text{dB}$ and $\kappa_{opt,PC} = 0.4$. When computing the MSE for the SPQI with these parameters, we receive 0.018 and 0.014 for the VC-PWQ and VPC-DS, respectively. This is very poor performance, as can be quickly seen when comparing to the values in Table 4.3. Therefore, we optimize with respect to MSE, resulting in $\tau_{opt,MSE} = -2.0\,\text{dB}$ and $\kappa_{opt,MSE} = 0.3$. The MSE values turn out to be 0.007 and 0.006 for the VC-PWQ and VPC-DS, respectively. The computed PCs are 0.839 and 0.960 for the VC-PWQ and VPC-DS, respectively. This means that we are able to achieve almost the maximum PC from Table 4.3 by choosing $\tau_{opt,MSE}$ and $\kappa_{opt,MSE}$.

Finall, we show the median curves for the SPQI with the chosen parameters in comparison to the median from the scores from the VQA experiment from Fig. 4.11 and the median score from the ST-SIM in Fig. 4.16. Each curve is computed for all 8 test signals and individually for each of the three codecs. We see that the SPQI is able to predict the measured scores very well for the VC-PWQ and VPC-DS. However, for the PVC-SLP, there is a large discrepancy between measured ratings and SPQI. This is converse to the ST-SIM, which performs well for the PVC-SLP and is quite far off for the VC-PWQ and VPC-DS.

4.3.4 Vibrotactile Multi-Method Assessment Fusion

Since we observe that no metric alone is able to reflect measured scores for all examined codecs, we strive for a combination of metrics. Our approach of combining metrics and fusing them into a more accurate perceptual score is called *vibrotactile multi-method assessment fusion (VibroMAF)*. The approach is inspired by video multi-method assessment fusion (VMAF) from the video domain [130]. This method represents the state of the art for subjective video quality assessment [131].

4.3.4.1 Design

The fusion approach of VibroMAF is shown in Fig. 4.17. By using a support vector machine (SVM), the three metrics normalized signal-to-noise ratio (NSNR), SPQI, and ST-SIM (so-called elementary metrics) are fused into the VibroMAF score. The NSNR resembles the SNR that is normalized with 75 dB. With this, the range of NSNR will be between 0 and 1. Through the metric fusion, we aim to receive a score that leverages the individual strengths of each elementary metric. For that, the

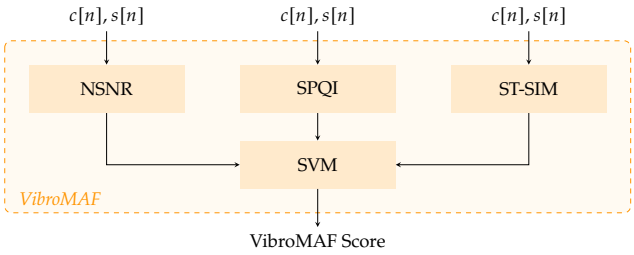


Figure 4.17 Workflow of the proposed VibroMAF. The support vector machine (SVM) regressor determines the weight for each individual metric score calculated from the compressed signal $c[n]$ and original signal $s[n]$. Adapted from [8] © IEEE 2022.

Metric	All Codecs	VC-PWQ [7]	PVC-SLP [43]	VPC-DS [6]
MSE VibroMAF	0.011	0.007	0.019	0.006
MSE SPQI	0.027	0.009	0.067	0.006
MSE ST-SIM [67]	0.037	0.019	0.012	0.080
MSE NSNR	0.440	0.452	0.526	0.341
PC VibroMAF	0.918	0.854	0.901	0.957
PC SPQI	0.800	0.807	0.741	0.982
PC ST-SIM [67]	0.775	0.831	0.945	0.918
PC NSNR	0.453	0.433	0.739	0.536

Table 4.4 MSE and PC computed for VibroMAF, SPQI, ST-SIM, and NSNR for the vibrotactile codecs VC-PWQ, PVC-SLP, and VPC-DS and overall. Best values shaded in color. Adapted from [8] © IEEE 2022.

SVM is trained to map the input to the measured VQA experiment score. Thus, the SVM will learn a weighted average of the three elementary metrics depending on their relationship. The VibroMAF approach is easily extendable with more elementary metrics as they become available in the future.

4.3.4.2 Evaluation

In order to showcase the capabilities of VibroMAF on the available signal data and measured rating scores, we conduct the training of the SVM with subsequent testing. Since the available signal data are small, we aim to provide a proof of concept that is to be fine-tuned as more data become available.

In order to train and test the SVM, we need to split the 8 test signals that we have ratings for into training and test dataset. We choose a split into 6 training signals and 2 testing signals. Since each signal has 17 CRs available, we have a total of 102 datapoints available for training and 34 datapoints for testing. In order to have a test dataset that reflects the overall data as accurately as possible, we again study the features in Fig. 4.10. We see that by choosing the materials *aluminium grid* and *polyester pad* we are able to achieve a somewhat diverse coverage of the feature space. Then, we select the *fast* recording speed for *aluminium grid* and the *slower* recording speed for *polyester pad* to cover both speeds in the test dataset.

The SVM is set up with a radial basis function kernel. For the regularization parameter, we choose 3000 and the epsilon is 0.1. These values were determined empirically, in order to achieve high performance.

After training the SVM, we compute the PC and MSE as described in Sec. 4.3.2 for VibroMAF and the three elementary metrics for each of the three codecs VC-PWQ, PVC-SLP, and VPC-DS as well as overall. The averaged results are shown in Table 4.4. We observe that across all codecs, the VibroMAF clearly is able to outperform all elementary metrics. This means that the fusion approach works as intended to learn the correct mapping from elementary metrics to more accurate fused metric.

Examining the results for each codec, we first see that for the VC-PWQ the VibroMAF is also clearly the best choice. For the PVC-SLP, the ST-SIM is still the best metric, however now closely followed by the VibroMAF. This is most probably because the data that were used for training are from two codecs, where the SPQI performs best and only one codec for which the ST-SIM is best. This means

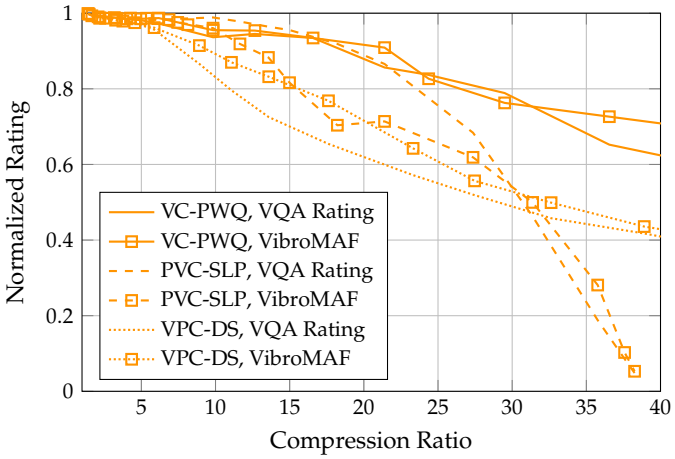


Figure 4.18 VibroMAF score compared to subjective ratings for the vibrotactile codecs VC-PWQ, PVC-SLP, and VPC-DS.

that the mapping in the SVM will naturally give a higher weight to the SPQI. Since the SPQI is not performing well for the PVC-SLP, the results for the SPQI will be suboptimal for this codec. In order to achieve the best performance for all codecs individually with VibroMAF, we need more balanced data. An additional factor is that the MSE for the NSNR is quite high for the PVC-SLP compared to the other codecs. Finally, for the VPC-DS, the VibroMAF is best together with the SPQI in terms of MSE and second-best in terms of PC.

In summary, even though the VibroMAF is not the best-performing metric for each and every codec individually, it leads to the most accurate rating overall. In order to visualize the benefit of VibroMAF in this context, we display the median rating curves for the 2 test signals from the VQA experiment and from VibroMAF in Fig. 4.18. We see that in contrast to the elementary metrics, the VibroMAF is able to deliver a rating that gets decently close to the measured VQA experiment scores for every codec. In order to further improve the fusion approach, we require more signal data with respective ratings for training. Also, the training data need to be balanced in terms of codecs and the respective performance of elementary metrics, to achieve the best results.

4.4 Chapter Summary

In this chapter, we conducted a thorough analysis of the perceptual quality of compressed vibrotactile signals. First, we presented the objective quality results in terms of SNR and PSNR for the vibrotactile codecs presented in Chapter 3. Here, we also compared the codec performance to the state of the art. We found that for single-channel signals, the VC-PWQ presented in Chapter 3 performs best among three examined codecs. The multi-channel vibrotactile codec (MVibCode) showed a strong boost in performance with the clustering, compared to the separate encoding of each channel.

Since objective quality metrics are not able to reflect the human experience, we designed a vibrotactile quality assessment (VQA) experiment method. This experiment procedure can be used to gather perceptual quality ratings with human assessors. The basis for the VQA experiment is the multi-stimulus test with hidden reference and anchor (MUSHRA) from the audio domain since it already contains helpful tools like the hidden reference and anchor signals and guidelines on post-screening of assessors. However, to develop a method that is suitable for vibrotactile signals and non-expert assessors, numerous adaptations were made to the procedure, such as strict timing, a hierarchical structure of different trials, design of the rating scale and adaptation of post-screening routines. By conducting the experimental procedure, we showed that with it we are able to measure perceptual scores accurately. Additionally, we were able to evaluate the presented single-channel codec VC-PWQ perceptually in comparison to the state of the art.

Finally, we developed computable perceptual quality metric that are designed to match the measured perceptual scores. Here, we first designed a novel metric called spectral perceptual quality index (SPQI), which takes the signal spectra of original and compressed signals and weights them perceptually. Then, through the calculation of a perceptual error measure from the weighted spectra, a perceptual score can be calculated. Since the SPQI was not able to reflect the experimentally measured scores for all examined codecs, in a second step we designed a fusion approach called vibrotactile multi-method assessment fusion (VibroMAF). Here, scores from different metrics are fused into one by a support vector machine (SVM). With this approach, we were able to design the only automated quality assessment method for vibrotactile signals to date that is able to predict the experimentally measured scores accurately for all three examined state-of-the-art codecs.

Chapter 5

Signal Enhancement

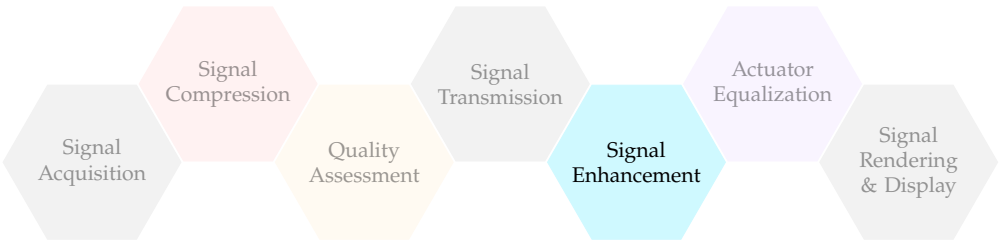
In order to provide a better human user experience when a received and decoded signal waveform is displayed, we enhance the signal quality after transmission and decoding as visualized in the figure below. The methods used are inspired from the image processing domain, where neural networks (NNs) have shown to perform well. In particular, since vibrotactile signals are time sequences, recurrent neural networks (RNNs) are a very promising type of NN here. In addition, we also use residual learning (RL) since it has shown to bring a wide range of benefits.

We aim to showcase the power of NN-based quality enhancement with an example network for single-channel signals. The exemplary NN that we propose acts as a proof of concept, where we show which steps are necessary to adapt NN-based quality enhancement methods to the vibrotactile domain. The concrete best implementation of a NN strongly depends on the available signal data and codec that has been used to compress the signals. Thus, for future extension, e.g., to multi-channel signals and codecs, one can base the method design on the findings in this chapter.

In this philosophy, this chapter focuses shortly on the developed RNN model and puts more emphasis on the derivation principles of the network structure. For that, we start by presenting the developed RNN model with its parameters. Then, we discuss the metrics developed to assess the RNN-based enhancement method. Then, we evaluate the method on single-channel compressed vibrotactile signals. Finally, in an ablation study, we show which aspects are important to consider for the design and parameter tuning of a RNN-based enhancement method specifically for vibrotactile signals. Parts of this chapter have been published in [3].

5.1 Neural Network Structure

We propose to employ the RNN structure with RL that is shown in Fig. 5.1. The RL technique can be seen by the two skip connections that span 4 RNN layers and 1 fully connected (FC) layer each. This FC layer with one neuron is responsible for reducing the dimensionality of the output to 1 after it has been inflated with the larger number of neurons in the RNN layers. Each of these blocks is trained to learn the mapping from compressed signal (CS) to the residual between CS and reference signal (RS). Then, the addition operation at the end of a skip connection adds the CS to the learned estimate of the residual. With this, we retrieve an enhanced version of the CS. The in total $k = 8$ RNN layers use the bidirectional long short-term memory (BiLSTM) and have $n = 75$ neurons each. We are able to use BiLSTM since after decoding every signal block is available all at once. Through the



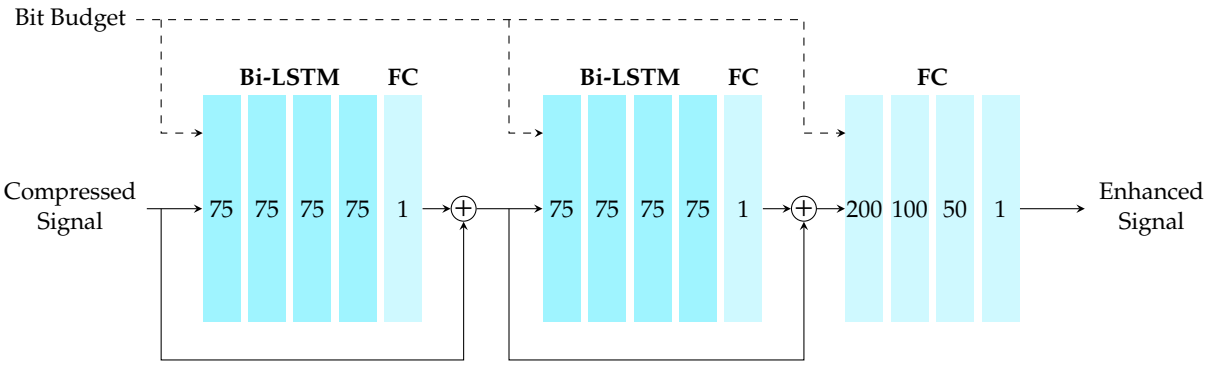


Figure 5.1 Structure of the exemplary neural network for the enhancement of single-channel vibrotactile signals.

bidirectional processing, more correlations can be exploited in the NN. The activation function, i.e., the nonlinearity in the RNN, we choose the standard $f_a(x) = \tanh(x)$. The loss function $L[n]$ is chosen as the mean absolute error (MAE). It was shown in [132] that amplifying the loss function can lead to a significant increase in performance. Therefore, the MAE is multiplied with 1000. As a result, we have for the loss function

$$L[n] = 1000 \cdot |\boldsymbol{y}[n] - \boldsymbol{o}[n]|. \tag{5.1}$$

Subsequent to the BiLSTM RNN, we place 4 additional layers of a FC network. Each of the layers has 200, 100, 50 and 1 neurons, respectively. The FC layers resemble simple feedforward networks. They process the signals entirely in a linear fashion. That means, they practically perform an additional linear regression on top of the nonlinear RNN.

In general, using available side information on the signals is practically always beneficial for enhancement tasks. This is because with additional parameters and descriptors for each signal, the NN is able to learn different parameters for signals with different properties. Overall, we then have a context-sensitive enhancement that can better adapt to different signals rather than a "one network fits all" solution. Such approaches have already been presented for the enhancement of images and videos. In particular, in [133] the quantization table from the JPEG encoder was fed into the NN as side information. For videos, [134], [135] presented enhancement methods that also take into consideration video codec features like quantization parameters, block partitioning map or labeling maps derived from the signal standard deviation.

The bit budget that was used in the vibrotactile codec for compressing each signal is a highly descriptive side information. For a lower bit budget, the distortions in the signal will be more significant and different in nature compared to a signal encoded with a higher bit budget. Therefore, the bit budget that was used to encode a signal is passed into the NN as additional input with each input signal. This can be done, e.g., by adding a weight vector multiplied with the bit budget value in (2.11). The addition of the bit budget as an extra parameter is also shown in Fig. 5.1. This additional input of the bit budget is one of the main contributions of our presented method, tailoring the presented NN approach to specifically target vibrotactile signals and codecs. It is highly probable that providing the bit budget or some other information on the signal data rate is a measure that should universally increase performance, even when developing enhancement methods for multi-channel signals and codecs in the future.

Another important aspect of the NN processing is the pre-processing and preparation of signals. Especially the vast difference in dynamic range can be highly detrimental to the performance of the method. For that, we scale every signal in a way that maximizes performance. Particularly, we first compute the maximum and minimum values of each signal. Then, we compute the difference of these two values, resulting in a measure for the total signal value range. This range is then rescaled to 50 with a corresponding multiplication by an appropriate factor. The scaling factor is stored in order to retrieve the original signal value range after enhancement. It is worth noting that this scaling technique does not produce zero-mean signals, since the mean is not subtracted beforehand.

5.2 Enhancement Performance Measures

In order to evaluate the performance of our enhancement method, we need appropriate, easily computable performance measures. We start with the signal-to-noise ratio (SNR) as main quality metric. Using the SNR and compression ratio (CR), we then calculate three measures that grasp the overall extent of signal quality enhancement and make it possible to compare the network configuration in Sec. 5.1 to other configurations.

The first measure is named CR_{min}. It is calculated as the minimum CR for which the mean difference in SNR between enhanced signal (ES) and CS is positive. This determines the range of CR for which we can expect the RNN to be able to enhance signal quality. Computing an upper limit of CR for which the signals are enhanced is not necessary, since as we will see in Sec. 5.3.3, for any $\mathrm{CR} \geq \mathrm{CR}_{min}$ the mean difference in SNR is positive. If $\mathrm{CR}_{min} = 1$ then signal are enhanced on average for all CRs. The lower CR_{min} is, the better the performance of the NN, since it means that a wider range of CRs can be enhanced.

The second performance measure is denoted as $\Delta\mathrm{SNR}_{max}$. It resembles the maximum occuring mean difference in SNR between ES and CSs. This measure therefore provides us with insight on how much the signals can be improved qualitatively. Of course, the higher $\Delta\mathrm{SNR}_{max}$ the better the performance of the NN is deemed.

Finally, we calculate the portion of signals that are improved from the entire signal dataset, which gives us the so-called enhancement efficiency η_E. In other words, it is calculated by dividing the number of signals for which the difference in SNR between ES and CS is positive by the total number of signals across all available CRs. Again, quite intuitively, the higher η_E is, the better the performance of the enhancement method.

These three measures can be used to keep track of improvements as the RNN structure will be further enhanced for new and different codecs. In the future, it could also be beneficial to use a perceptual metric in addition to SNR, as the perceptual metrics become reliable at predicting perceptual signal quality.

5.3 Experimental Evaluation

5.3.1 Experiment Parameters

The signals from the LMT reference dataset serve as test signals for the NN-based enhancement method. This means we have a total of 280 signals at our disposal for training, validation, and testing of our NN. Especially the balance between training and testing signals is delicate. We need enough signals to train the network, but too few signals for testing hinder us in assessing the generalization performance of the NN properly, which may lead to overfitting. In order to balance these two aspects, we aim for roughly twice as many training signals than testing signals. The separation of signals into sets is conducted by the list of alphabetically sorted signal names. This means, the 280 vibrotactile signals from the dataset are subdivided by taking the first 180 as training signals, the next 20 as validation signals, and the final 80 as testing signals. We used the single-channel codec to compress each signal with 17 different bit budgets. This then leads to a grand total of 3060 training signals, 340 validation signals, and 1360 testing signals.

By dividing the signals not randomly but in the described ordered manner, all fingertip measurements land in the testing signal category. Conversely, the training and validation signal sets contain no fingertip-measured signals. As described in Sec. 3.2.3, the signals measured with the fingertip differ significantly from the rest. Thus, by including them in the testing set, we can test the generalization ability of our NN more effectively, since it has to perform enhancement on partially very different data that are different to what it has been trained with.

The signals are processed block-wise. For one, this is especially beneficial since it allows for the use of a bidirectional recurrent neural network (BiRNN). As block length, we choose 512 samples, since

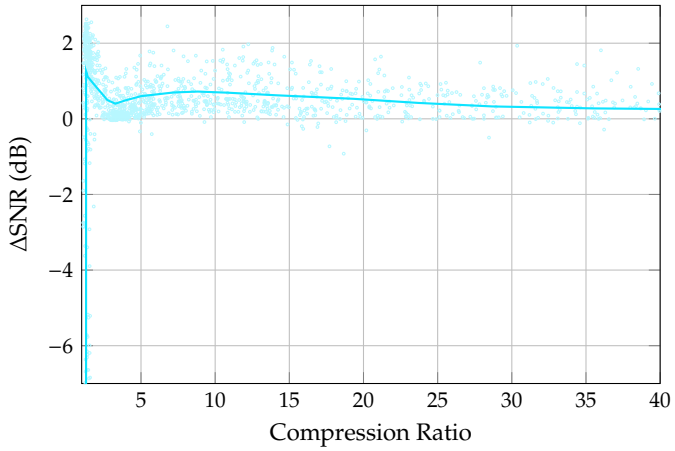


Figure 5.2 Difference in SNR over CR between enhanced signals and compressed signals for all signals in the testing dataset (dots) and mean curve (solid line). Adapted from [3] © IEEE 2021.

the same block length was used in the codec, with which the signals were compressed. We found that the performance of our NN is best for this block length, which most probably comes from this match to the codec block length. This gives us another insight for future method design, which is that it is very probable that matching the codec block length in the enhancement method leads to good performance.

5.3.2 Experimental Procedure

We conduct our experiment on the enhancement capabilities of our NN structure in MATLAB R2020a with GPU acceleration. Specifically, we train the NN using the training and validation sets. We choose the batch size as 20. This means that the entirety of 3060 training signals is processed in 153 batches. Each batch is fed to the RNN, then from the 20 output signals the training loss is calculated and with it a gradient step to optimize the network parameters is performed.

After all the batches have been processed, we have completed an epoch. We train the NN for 30 epochs in total. After each epoch, we use the signals in the validation set to calculate the validation loss. This loss should decrease between subsequent epochs. If that is not the case, that raises awareness that we might be overfitting to the training data.

This insight can aid us in choosing a correct learning rate μ. Specifically, we have to choose μ low enough to avoid overfitting. At the same time, if it is too low, we risk training too slowly and requiring a large number of epochs. Therefore, we analyzed different choices of learning rate and concluded to a mixed scheme with an initial learning rate and reductions of learning rate after a certain number of epochs. Specifically, we choose the initial learning rate as $\mu = 10^{-3}$. After 10 epochs, we multiply the learning rate by a factor of 0.1. These values were chosen empirically because we saw a saturation in validation loss after a few epochs. Thus, by reducing the learning rate, we are able to decrease the validation loss more efficiently.

5.3.3 Experimental Results

After training the NN model as described, we input the test signals and produce the corresponding ESs. For these signals, we then compute the SNR in dB. From these SNR values, we subtract the SNR of the corresponding CSs. The result is denoted as ΔSNR. We plot all the ΔSNR values for each individual test signal as well as the mean curve over CR in Fig. 5.2. First, we can see that an overwhelming majority of signals get enhanced in terms of SNR. We find that $\Delta\text{SNR}_{max} = 1.25\,\text{dB}$. This is quite a significant improvement, when compared to the usual improvements in the image and video domain. These are usually around 1.3 and 1.7 dB for JPEG and 0.2 and 0.5 dB vor HEVC-MSP [136]. For HEVC-MSP, the

improvement is significantly less, because the codec is more sophisticated, leading to higher quality signals to begin with. The NN is not able to enhance signals with the highest quality. These are signals that have been encoded with a bit budget of 120 bits. It is intuitive that these signals cannot be enhanced, because they are very similar to the original signals and thus any processing on them is more likely to increase the error than decrease it. In general, this is not of importance here, because these signals have a quite high data rate and will not be used in practice, since its advantageous to simply use the original signals at CRs below 1.5.

From Fig. 5.2 we can also read that $\mathrm{CR}_{min} = 1.31$. This means that signals are enhanced on average for all but the highest bit budget. For the enhancement efficiency, we calculate $\eta_E = 85.81\,\%$. Excluding the signals encoded with the highest bit budget, for which we never have an improvement, the efficiency increases to $\eta_E \approx 91\,\%$. This result is highly desired, since it means that we can expect a very high rate of signals to be improved by the method. Fundamentally, this tells us that the network generalizes quite well for the testing data.

5.4 Ablation Study

After having presented our exemplary optimized network structure, we conduct an ablation study. In such a study, each parameter is varied across a certain range to see the effects of a singular parameter change. Through this, we can gain insight on how the network behaves under certain conditions and derive rules that help us choose optimal parameters for certain signal data or codecs. With this knowledge, we can then adapt the network structure more easily.

In the following, we present results of our three performance measures from Sec. 5.2 for each individual parameter variation in the form of tables. Where parameters are mutually dependent on each other, we vary them simultaneously.

5.4.1 Number of Layers and Neurons

First, we vary the number of layers k and the number of neurons n. We do this examination for both parameters at once, since they are highly dependent on each other. Usually, increasing one of the two parameters and decreasing the other can be expected to lead to similar performance. Specifically, we choose $k \in \{4, 6, 8, 10, 12\}$ and $n \in \{25, 50, 75, 100\}$. These choices are determined empirically, as outside of this range, performance decreases drastically. The resulting values for the performance measures are given in Table 5.1. Here, we highlight the best values for clarity. For CR_{min} all cells with the best value 1.31 are highlighted. For $\Delta\mathrm{SNR}_{max}$, we highlight all cells with values higher then 1.2 dB. Finally, for η_E, we highlight all values above 85.76%, which is one percentage point short of the best value.

Going through the table one column at a time, we first see that k and n have very little influence on CR_{min}. Out of the 20 combinations of examined parameters, only 4 do not lead to the best value of 1.31. Out of those, 3 combinations are just slightly worse with 1.41 and one is clearly worse. This means that the optimal CR_{min} is a fairly easy to achieve target meaning it is not a very selective criterion. In turn, this means that if the NN is not able to achieve this for a specific choice of parameters, this is a fairly big caveat. Moving on to $\Delta\mathrm{SNR}_{max}$, we see that only two cells are highlighted. Thus, as a criterion, this is very selective. Finally, for η_E, we have four combinations that lead to a close-to-optimal outcome.

Taken together, only one combination has all three measures in the high range, specifically $k = 8$ and $n = 75$. This is the combination of our exemplary NN structure. We choose this combination over $k = 10$ and $n = 25$, despite the slightly lower η_E. This is because the difference in η_E is smaller than 1%, while for $\Delta\mathrm{SNR}_{max}$ the difference is very significant. Therefore, we prefer the much higher amount of improvement over the slight gain in η_E.

k	n	CR_{min}	ΔSNR_{max} (dB)	η_E (%)
4	25	1.41	0.66	84.93
	50	1.31	0.68	84.71
	75	1.31	1.03	85.44
	100	1.41	0.84	81.91
6	25	1.41	0.67	85.07
	50	1.31	1.04	85.22
	75	1.31	0.98	80.81
	100	1.31	0.86	74.46
8	25	1.31	0.85	85.88
	50	1.31	1.03	86.10
	75	1.31	1.25	85.81
	100	1.31	0.98	79.41
10	25	1.31	0.91	86.76
	50	1.31	0.90	84.49
	75	1.31	0.95	82.79
	100	1.31	1.23	82.35
12	25	1.83	0.65	84.71
	50	1.31	0.94	85.44
	75	1.31	1.02	83.53
	100	1.31	0.85	80.66

Table 5.1 Results of performance measures for different number of layers k and neurons n. Adapted from [3] © IEEE 2021.

Bit Budget	CR_{min}	ΔSNR_{max} (dB)	η_E (%)
with	1.31	1.25	85.81
without	4.04	0.46	70.37

Table 5.2 Results of performance measures with and without the inclusion of bit budget as side information in the neural network design. Adapted from [3] © IEEE 2021.

5.4.2 Inclusion of Bit Budget

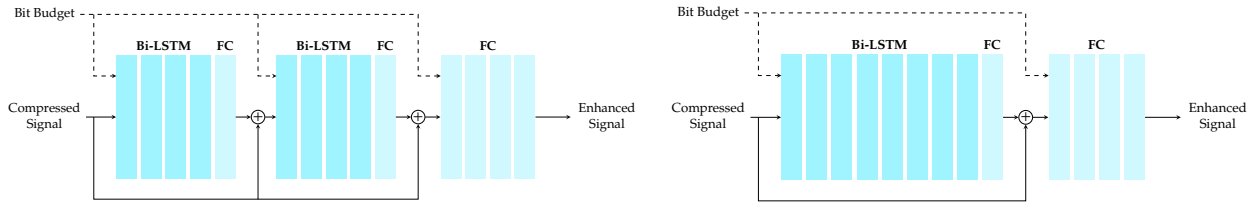
Now, we examine the influence of including the bit budget in our NN structure as a side information parameter. For that we compute the three performance measures with this inclusion and without. The results are shown in Table 5.2, where the best values in each column are highlighted. We can see an extreme difference between the two cases for all three performance measures. Thus, the contribution of including the bit budget as side information gives a very large boost in performance as the NN is able to adapt to differently CSs. With such a large difference, we can expect this to be true for other signals and codecs as well. For codecs that operate differently, i.e., without a bit budget parameter, the inclusion of another rate-determining parameter or the CR should be examined.

5.4.3 RNN Neuron Type

Now, we examine the neuron type and how it influences performance. We have three choices. For unidirectional RNNs, we have the long short-term memory (LSTM) and the gated recurrent unit (GRU). The BiLSTM is then part of a BiRNN where time dependencies in two directions are formed. The resulting values for the three performance measures are given in Table 5.3, where we again highlight the best values for each metric. For CR_{min}, only the BiLSTM is able to achieve the best

RNN Type	CR_{min}	ΔSNR_{max} (dB)	η_E (%)
LSTM	1.41	0.65	87.06
GRU	1.41	0.64	82.65
BiLSTM	1.31	1.25	85.81

Table 5.3 Results of performance measures for different RNN neuron types. Adapted from [3] © IEEE 2021.



(a) Input-wise residual learning configuration.

(b) Single block residual learning configuration.

Figure 5.3 Two alternative configurations of RL shortcut connections to the layer-wise configuration from Fig. 5.1.

Structure	CR_{min}	ΔSNR_{max} (dB)	η_E (%)
Single Block	1.31	1.19	84.49
Layer-wise	1.31	1.25	85.81
Input-wise	1.31	0.90	84.33

Table 5.4 Results of performance measures for three different RL shortcut connection configurations. Adapted from [3] © IEEE 2021.

value. The same hold true for the ΔSNR_{max}. Here, the BiLSTM leads to an improvement thats almost twice as high as the other two choices. Then, for η_E, the LSTM shows the best performance, while the BiLSTM comes in at 1.25% less.

Ultimately, we are faced with a trade-off when aiming to decide for the best choice here. If it is desired to have the highest η_E at the expense of the amount of improvement, we would go for LSTM. However, in this work we deem the very large difference in ΔSNR_{max} more critical while at the same time, the difference in η_E can still be tolerated.

5.4.4 Residual Learning Shortcut Connections

For the NN in Fig. 5.1, there are other options conceivable for the RL shortcut connections. The two most common alternatives to the chosen setup are shown in Fig. 5.3. Here, we keep the number of RNN layers equal for all configurations, namely 8 layers. The setup in Fig. 5.1, contains layer-wise shortcut connections. This means that we have a shortcut connection, spanning 4 RNN layers and then the reconstructed output is the input for another block with yet another shortcut connection. In contrast to that, the setup in Fig. 5.3a always takes the input signal to be added at the end of a shortcut connection. It is thus named, input-wise configuration. Additionally, the alternative in Fig. 5.3b, has only one shortcut connection spanning all the 8 RNN layers. Therefore, we name this as the single block configuration.

For the 3 alternative shortcut configurations, we now compute the performance measures again. The results are shown in Table 5.4, again with highlights on the best values. In short, we observe that the layer-wise configuration is clearly the best choice. However, the single block setup gets fairly close in terms of ΔSNR_{max}. In general, the variations in performance between the three different configurations are not as pronounced. Therefore, when adapting to new signal data or different codecs, this aspect should be one of the first to be examined, because it easily might be the case that the best choice changes.

5.4.5 Inclusion of the Fully Connected Layers

Next, we observe the performance change for when the FC layers after the RNN part are omitted. For that, we again compute the three performance measures from Sec. 5.2 and display the results in Table 5.5. In short, it is clearly favorable to conduct the additionnal linear regression step with the FC layers. This step should therefore be always chosen after the processing of the RNN layers.

FC Layers	CR_{min}	ΔSNR_{max} (dB)	η_E (%)
with	1.31	1.25	85.81
without	1.83	0.72	83.75

Table 5.5 Results of performance measures for NN structure with and without additional FC layers at the end. Adapted from [3] © IEEE 2021.

Pre-Processing	CR_{min}	ΔSNR_{max} (dB)	η_E (%)
None	3.34	0.67	78.46
Fixing range	1.31	1.25	85.81
Normalization	1.31	0.88	83.46
Strict normalization	1.41	0.65	82.13

Table 5.6 Results of performance measures for four different signal pre-processing approaches. Adapted from [3] © IEEE 2021.

5.4.6 Pre-Processing Technique

Now, we aim to examine the influence of different pre-processing techniques on the performance outcome. For clearer reference, the chosen technique is named *fixing range*. In addition to the chosen pre-processing technique, we choose three other approaches for testing. First, we apply no pre-processing at all. This is to evaluate the necessity of any pre-processing in general. The second approach is named *normalization*. Here, we do a very similar approach to *fixing range*, with scaling the signal range to 50 overall. However, in difference to before, we first subtract the mean of each signal from it and then conduct the scaling with an appropriate factor. Thus, the only difference between the two methods is that one produces zero-mean signals, while the other does not. Finally, we choose another alternative approach called *strict normalization*. Here, we first subtract the minimum value of each signal. That means, each signal will then have a minimum of 0 and be completely nonnegative. Then, the range of that signal is scaled to 50 again. Finally, we subtract 25 from the scaled signal. This means that in the end, all signals will have exactly the value range $[-25, 25]$.

The resulting values of the performance measures for the four described approaches are shown in Table 5.6. It is straightforward to see that *fixing range* is the clear winner in terms of performance. No pre-processing is clearly highly unfavorable. Between the three techniques, the performance of *strict normalization* is quite low, whereas for *normalization* it is possible that it could outperform *fixing range* for different signal data or codecs. Overall, we see that preserving information on the mean of signals helps the NN to enhance signals better.

5.4.7 Loss Function and Learning Rate

Finally, we examine the influence of the choice of loss function L and initial learning rate μ. The analysis is done for each parameter itself as well as jointly, since they can have a joint influence on the performance outcome of the enhancement.

First, we start with four different choices of L. On one hand, we have the chosen function of $1000 \cdot \text{MAE}$. We also examine the unscaled version MAE for comparison to see the effect of the amplification of the loss function. As an alternative measure, we examine the mean square error (MSE) as loss function. Here, again we choose both the amplified version as well as the original one, i.e., the two options are $1000 \cdot \text{MSE}$ and MSE.

The results of the three performance measures are shown in Table 5.7. Interestingly, the amplification with 1000 only brings a large benefit for the MAE, while for the MSE there is practically no difference in the outcome. Already the unscaled MAE is performing way better than the MSE. Finally, our choice of $1000 \cdot \text{MAE}$ is clearly the winner in terms of performance. Thus, we take away that a non-squared measure for the loss function will probably be a better choice for other signals and codecs too.

L	CR_{min}	$\Delta\mathrm{SNR}_{max}$ (dB)	η_E (%)
MSE	3.34	0.67	68.24
1000·MSE	3.34	0.68	68.24
MAE	1.31	0.95	83.82
1000·MAE	1.31	1.25	85.81

Table 5.7 Results of performance measures for different choices of loss function. Adapted from [3] © IEEE 2021.

μ	CR_{min}	$\Delta\mathrm{SNR}_{max}$ (dB)	η_E (%)
10^{-2}	1.41	0.70	84.41
10^{-3}	1.31	1.25	85.81
10^{-4}	2.80	0.62	85.57

Table 5.8 Results of performance measures for different choices of initial learning rate. Adapted from [3] © IEEE 2021.

Now, for the initial learning rate, we aim to show the general behavior around the chosen value. Thus, we pick 10^{-2} and 10^{-4} as alternatives to 10^{-3}.

The results for the performance measures are given in Table 5.8. The choice of $\mu = 10^{-3}$ gives the best performance. In terms of η_E, $\mu = 10^{-4}$ leads to a very close result, which means that this could lead to a different outcome when confronted with different training data. However, for the other two performance measures, performance is significantly worse.

As a final evaluation, we vary L and μ jointly. The reasoning here is that when scaling the loss function with a factor of 1000 with a certain learning rate, one could expect similar performance for an unscaled loss function with a 1000-fold learning rate instead. Therefore, for both choices of L being MAE and $1000 \cdot \mathrm{MAE}$, we examine the three different learning rates 10^{-6}, 10^{-3} and 1.

The resulting outcome for the three performance measures is shown in Table 5.9. The expectation that, e.g., $1000 \cdot \mathrm{MAE}$ with $\mu = 10^{-3}$ would perform similarly to MAE with $\mu = 1$ is clearly not confirmed. This, the dependency on both variables is not as simple as just an inverse proportionality. For $\mu = 1$, we even have an unstable algorithm and training fails. The chosen values for both parameters clearly give the best performance.

5.5 Chapter Summary

In this chapter, we presented a novel quality enhancement method for compressed vibrotactile signals using recurrent neural networks (RNNs) and residual learning (RL). The basic principle of the method is to have a RNN learn the relationship between compressed and original signals, so then the trained network model is able to reconstruct some of the lost signal information of other compressed signals when they are processed by it. Here, we used RL in order to mitigate the detrimental effect of the high dynamic range differences in vibrotactile signals. Additionally, we designed a signal pre-processing technique that normalizes input signals to a specific range. With these adaptations, the neural network (NN) does not have to interpolate between largely different signal amplitudes, which

μ	L	CR_{min}	$\Delta\mathrm{SNR}_{max}$ (dB)	η_E (%)
10^{-6}	MAE	3.34	0.45	68.46
	1000·MAE	2.80	0.45	68.38
10^{-3}	MAE	1.31	0.95	83.82
	1000·MAE	1.31	1.25	85.81
1	MAE	NaN	NaN	NaN
	1000·MAE	NaN	NaN	NaN

Table 5.9 Results of performance measures for different combinations of loss function and initial learning rate. Adapted from [3] © IEEE 2021.

enhances performance. Additionally, we include side information from the codecs in the form of the bit budgets used for compressing the respective signals. With this contribution, the RNN is able to better adapt to signals of different quality levels and provide better enhancement performance. Overall, the RNN was able to enhance close to 86% of the signals in our testing dataset in their quality. The maximum improvement of the average signal quality was 1.25 dB.

Through an extensive ablation study, we showed important aspects of how the enhancement method was tailored for vibrotactile signals and how it can be adapted for different signal data in the future. In particular, care needs to be taken to choose an appropriate pre-processing technique for the input signals. Also, the interplay of learning rate and loss function needs to be examined in detail when adapting the method for new signals and codecs.

Chapter 6

Actuator Equalization

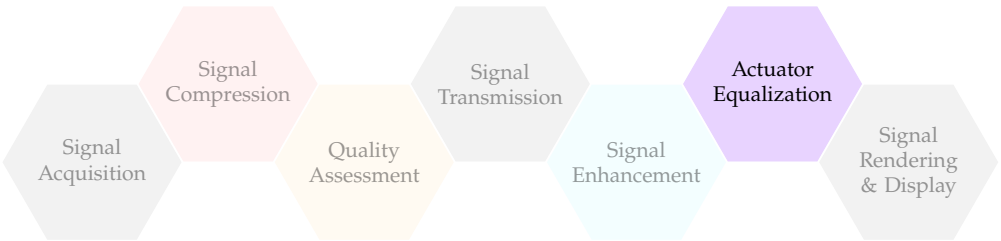
As outlined in Sec. 2.6, when vibrotactile signals are rendered by actuators, distortions are introduced. To counteract this, we employ equalization with adaptive filters, which resembles a signal processing module directly linked to the vibrotactile display as shown in the figure below. The two previously known filter models, the linear filter (LF) and second order Volterra with infinite impulse response filter (SOV-IIRF), are suitable for this task. However, they do not resemble the best match for vibrotactile actuators and exhibit suboptimal performance.

For one, the LF is purely linear but actuators are always nonlinear to some extent. Therefore, the LF will not be able to equalize these nonlinear distortions. A nonlinear filter model is better suitable here. One the other hand, the SOV-IIRF is nonlinear and can even achieve nonlinearities of high order with few parameters. However, for one, it is very prone to instabilities due to the nonlinear feedback. The other key issue is that in this filter model, the input and output are completely uncoupled, i.e., they are never multiplied with each other.

This is in sharp contrast to the behavior of typical actuators. The most widely used actuators, the linear electromagnetic actuator (LEA) work with magnets and coils around them. As a current flows through the coil, the magnet is moved. However, the movement of the magnet, influences the current in the coil. This is especially so, because of mechanical limitations, i.e., the magnet movement will never be perfectly in sync with the current. This introduces distortions produced by a coupling of input and output signals. Due to this behavior of vibrotactile actuators, we believe that an adaptive filter model where input and output are coupled is more powerful for equalization. In the following, we present our approach to equalize these distortions with such a novel adaptive filter model. Parts of this chapter have been published in [2].

6.1 Bilinear Volterra Filter Model

For equalizing vibrotactile actuators, we propose the *bilinear Volterra filter (BVF)* model. The BVF is inspired by the SOV-IIRF. However, it is designed to overcome some of its limitations and provide a better match to vibrotactile actuators in an equalization setup. As such, the recursive second order



term in (2.25) is replaced by a bilinear term, which multiplies delayed versions of the input and output signal. Therefore, the difference equation for the BVF is

$$\begin{aligned} y[n] = &\sum_{i=0}^{N-1} a_i[n]x[n-i] + \sum_{i=0}^{N-1}\sum_{j=i}^{N-1} b_{i,j}[n]x[n-i]x[n-j] \\ &+ \sum_{i=1}^{M} c_i[n]y[n-i] + \sum_{i=0}^{K-1}\sum_{j=1}^{L} d_{i,j}[n]x[n-i]y[n-j], \end{aligned} \tag{6.1}$$

where $a_i[n]$ and $b_{i,j}[n]$ with $i, j = 0, \ldots, N-1$ are the coefficients for the linear and quadratic feedforward part, $c_i[n]$, $i, j = 1 \ldots, M$ for the linear and quadratic feedback part, and $d_{i,j}[k]$, $i = 0, \ldots, K-1, j = 1, \ldots, L$ are the coefficients for the bilinear part.

The inspiration for the BVF comes from the fact that in actuators the input and output signals are usually highly coupled. As such, the quadratic feedback term of the SOV-IIRF is suboptimal in modeling this behavior, because the input and output signals are uncoupled in this model. On the other hand, the BVF is able to still model high order nonlinearities with significantly reduced number of coefficients. Therefore, it is superior to the LF in that regard, since it is able to equalize the nonlinear part of actuators as well. Additionally, due to the multiplication of input and output in the bilinear term, the BVF is completely free of limit cycles as long as the coefficients $c_i[n]$ meet the well known stability criteria for linear filters.

In order to use the BVF in our algorithmic equalization framework, we again reshape (6.1) into a form as in (2.13) with

$$\boldsymbol{\varphi}[n] := \left[\boldsymbol{\varphi}_a^\top[n] \;\; \boldsymbol{\varphi}_b^\top[n] \;\; \boldsymbol{\varphi}_c^\top[n] \;\; \boldsymbol{\varphi}_d^\top[n]\right]^\top \tag{6.2}$$

$$\boldsymbol{\varphi}_a[n] := \left[x[n], x[n-1], \ldots, x[n-N+1]\right]^\top \tag{6.3}$$

$$\begin{aligned} \boldsymbol{\varphi}_b[n] := \big[&x[n]x[n], \ldots, x[n]x[n-N+1], \\ &x[n-1]x[n-1], \ldots, x[n-1]x[n-N+1], \\ &\ldots, x[n-N+1]x[n-N+1]\big]^\top \end{aligned} \tag{6.4}$$

$$\boldsymbol{\varphi}_c[n] := \left[y[n-1], y[n-2], \ldots, y[n-M]\right]^\top \tag{6.5}$$

$$\begin{aligned} \boldsymbol{\varphi}_d[n] := \big[&x[n]y[n-1], \ldots, x[n]y[n-L], \\ &x[n-1]y[n-1], \ldots, x[n-1]y[n-M] \\ &x[n-2]y[n-1], \ldots, x[n-n+1]y[n-M]\big]^\top, \end{aligned} \tag{6.6}$$

and

$$\begin{aligned} \boldsymbol{w}[n] := \big[&a_0[n], \ldots, a_{N-1}[n], b_{0,0}[n], \ldots, b_{0,N-1}[n], \\ &\ldots, b_{N-1,N-1}[n], c_1[n], \ldots, c_M[n], \\ &d_{0,1}[n], \ldots, d_{0,M}[n], d_{1,1}[n], \ldots, d_{M,M}[n]\big]^\top. \end{aligned} \tag{6.7}$$

Thus, we can proceed to examine the performance of the BVF in comparison to the existing adaptive filter models.

6.2 Equalization Performance Measure

In order to evaluate how well an equalization method performs, we use so-called learning curves. These curves are calculated as the mean square error (MSE) between input and output signal. In this calculation in each time step, we take into consideration all available past time steps, i.e., the MSE

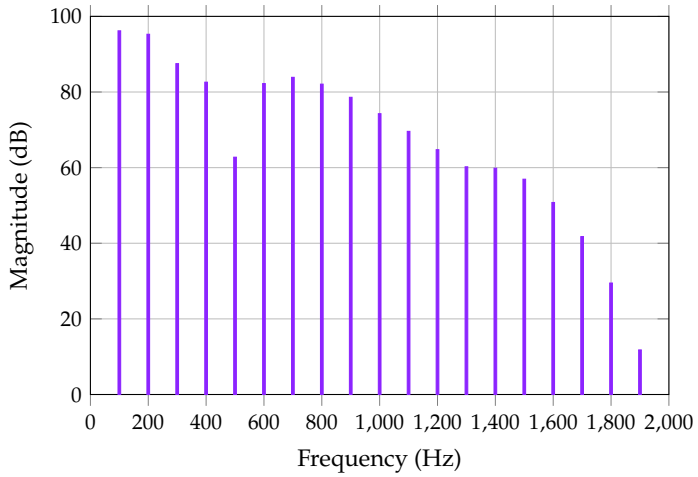


Figure 6.1 Spectrum of the output signal from a random BVF driven by a sinusoidal input signal with $f_0 = 100\,\text{Hz}$. Adapted from [2] © IEEE 2020.

is always calculated from the first signal sample to the current one. The learning curve, denoted as MSE$[n]$ is thus computed by

$$\text{MSE}[n] = \frac{1}{n}\sum_{i=1}^{n} |e[i]|^2. \tag{6.8}$$

The metric of MSE defined in this way is highly suitable for equalization performance assessment. By considering the entire signal history, it emphasizes the role of the adaptation speed rather than the achieved final error measure. Usually, due to the noise in the system, all filter models are capable of reducing the error signal $e[n]$ to similar degree. Thus, if a filter model is able to reach this point quicker, the metric in (6.8) will be lower overall and the fast adaptation speed is rewarded. Especially for the equalization of time-varying vibrotactile actuators, the information on which filter is able to adapt faster is highly valuable. Nonetheless, the total achieved error is not eliminated as a factor, but still plays a role for the metric. Thus, we are able to evaluate the different adaptive filter models across both dimensions of interest (adaptation speed and error) with a single metric.

6.3 Simulative Evaluation

The equalization capabilities of the BVF are first examined in a series of simulation experiments. For that, we first examine the ability of the BVF to model nonlinearities of high order. After that we compare the BVF to the existing approaches concerning its equalization efficiency. To this end, we use well-established benchmark model filters serving as virtual actuators.

6.3.1 Nonlinearity of the BVF

We showcase the ability of the BVF to model nonlinearities with very high order. For that, we generate a BVF with random filter coefficients. To this end, each coefficient is a realization of a uniformly distributed random variable. For $a_i[n]$ and $b_{i,j}[n]$, we have a uniform distribution -1 and 1. For stability reasons, the feedback coefficients are chosen from more narrow intervals, specifically the $c_i[n]$ range from -0.3 to 0.3 and the $d_{i,j}[n]$ from -0.5 to 0.5. The filter tap numbers are chosen to be $N = 10$, $M = L = 3$ and $K = 5$. We assume a sampling frequency of $f_S = 4000\,\text{Hz}$ for the input signal. The input signal is then a pure sine tone with a frequency of $f_0 = 100\,\text{Hz}$, i.e., $d[n] = \sin(2\pi f_0/f_S n)$.

We compute the spectrum of the output signal, filtered by the random BVF. This spectrum is shown in Fig. 6.1. In general, if a filter introduces frequencies into a signal that were not there previously, we have a nonlinear filter. For a nonlinear filter, we observe harmonics, i.e., sinusoidal signal components

at integer multiples of the base frequency from the input signal. In Fig. 6.1, we can clearly see a vast number of harmonics produced by the filtering. The number of harmonics is indicative of the order of the nonlinearity. Therefore, the BVF has been shown to be indeed able of generating nonlinear behavior of very high order.

6.3.2 Benchmark Models

In order to analyze the equalization capabilities of the BVF through simulation, we substitute real actuators with so-called benchmark model filters (BMFs). These BMFs filter the input signal nonlinearly. Here, we choose 3 BMFs that have found wide-spread use in the assessment of adaptive filtering equalization [137]–[139]. Each BMF filters the input signal according to a defined difference equation, followed by additive white Gaussian noise (AWGN).

The first BMF from [137], [138] is defined by

$$x[n] = d^3[n] + \frac{x[n-1]}{1 + x^2[n-1]}, \tag{6.9}$$

where $d[n]$ is the desired signal, i.e., the input signal of the actuator. The input and output of the BMF are uncoupled in this case. The first term of (6.9) produces a nonlinearity of order 3 for the input, which is then extended further by the second term due to its nonlinear feedback contribution.

The second and third BMF from [138], [139] are defined by the difference equation

$$x[n] = \frac{\prod_{i=1}^{3} x[n-i] \cdot d[n-1] \cdot (x[n-3] - b) + c \cdot d[n]}{a + x^2[n-2] + x^2[n-3]}. \tag{6.10}$$

Here, the choice of the parameter a, b, and c is static for the second BMF, while it varies over time for the third BMF. In particular, for the second BMF, we have

$$a = 1; \quad b = 0.6; \quad c = 1. \tag{6.11}$$

Then, for the third BMF, the coefficients are

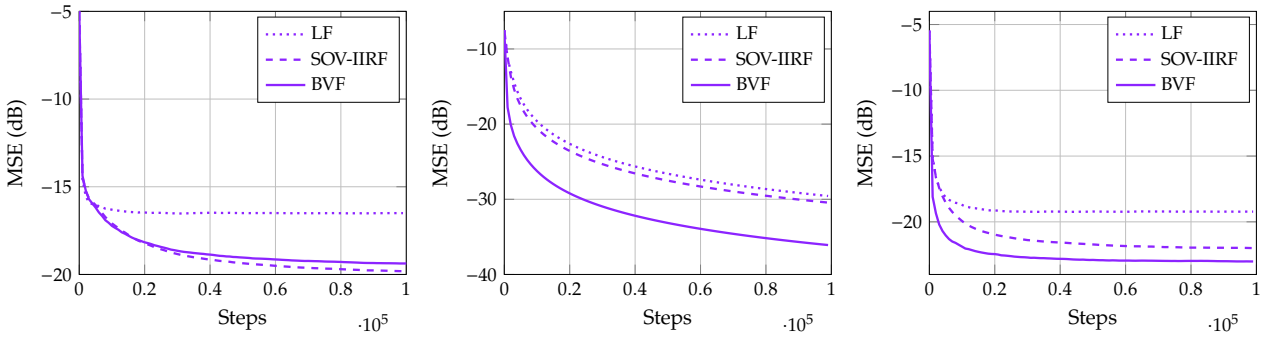
$$\begin{aligned} a[n] &= 1.2 - 0.2 \cos\left(\frac{2\pi n}{T}\right) \\ b[n] &= 1 - 0.4 \sin\left(\frac{2\pi n}{T}\right) \\ c[n] &= 1 + 0.4 \sin\left(\frac{2\pi n}{T}\right). \end{aligned} \tag{6.12}$$

As we see in (6.10), the input and output are coupled now. In the following, each filter model is referred to as BMF 1, 2, and 3, respectively.

6.3.3 Postdistortion Equalization Performance

We conduct our equalization of the virtual actuators simulated by the three BMFs with the postdistortion setup from Fig. 2.5. We choose the filter parameters such that we have equal complexity between all filter models. This means that all three filter models have the same number of coefficients. As such, the tap numbers of the filter models are chosen as $N = 36$ and $M = 35$ for the LF, $N = 8$ and $M = 6$ for the SOV-IIRF and $N = 9$, $M = 5$, $K = 4$, and $L = 3$ for the BVF. Therefore, all filter models have 71 coefficients.

With this choice, we can effectively test how signficant nonlinearities in the actuators are. If the actuators were completely linear, then the LF would probably perform best in equalization, because it has more parameters for linear parts than the other two filter models (71 vs. 14). On the other hand,



(a) BMF 1 **(b)** BMF 2 **(c)** BMF 3

Figure 6.2 Learning curves as defined in (6.8) for three benchmark model filters (BMFs) serving as virtual actuators when equalizing with three different adaptive filter models. Adapted from [2] © IEEE 2020.

for nonlinear actuators, by giving the two nonlinear filter models the same number of coefficients on their nonlinear parts, we effectively test which kind of nonlinearity is best suitable for equalization. If in the actuator, the input and output are highly coupled, then we can expect the BVF to perform best here.

We perform equalization over an input signal chosen as realizations of a uniformly distributed random variable between -1 and 1. With this choice, the input signal contains all possible frequencies. This means that the adaptive filters will adapt and learn the correct equalization mapping for many different frequencies. The length of the input signal is 100000 samples. As initial step size for the normalized least mean squares (NLMS), we choose $\mu = 0.6$. All coefficients of the adaptive filters in $\boldsymbol{w}[n]$ are initialized with zeros.

We visualize the learning curves for the three different BMFs in Fig. 6.2. For BMF 1, we observe that the two nonlinear adaptive filter models outperform the LF by a significant amount. This is not surprising, since the BMF 1 is highly nonlinear. Between the two nonlinear filter models, the SOV-IIRF has a slight advantage over the BVF. This is easily explainable due to the uncoupled input and output in (6.9), which means the SOV-IIRF is a better match. For the BMF 2 and 3, the BVF is the clear winner in terms of performance. Now, due to the coupling of input and output in (6.10), the BVF obviously provides a better match in filter structure. Therefore, we can expect that the BVF will also perform well with real actuators, where input and output are usually highly coupled.

6.3.4 Predistortion Equalization Performance

Now, we switch to using the postdistortion and translation setup from Fig. 2.6. We do so to test the impact of the non-commutative exchange in filter order on the overall equalization performance. For that, we study the learning curves for the BVF only. The learning curves are computed for all three BMFs serving as virtual actuators in the two different equalization setups.

The computed learning curves of the BVF in both setups are shown in Fig. 6.3. We observe that the non-commutative exchange when moving from one setup to the other introduces a slight dent in performance. We observed the same effect for the other two adaptive filter model choices. This effect was to be expected, because of the nonlinearities of filters and actuators and matches many previous findings [107]. Nonetheless, the results show that the postdistortion and translation setup performs well in general and can therefore be used effectively in real actuator hardware setups.

6.4 Experimental Evaluation

Now that we have shown the capability of our novel BVF model, we move on to apply the equalization scheme to real actuators. For a complete evaluation, we conduct a series of experiments. First, we analyze the nonlinear behavior of a typical vibrotactile actuator to exemplify the distortions that are

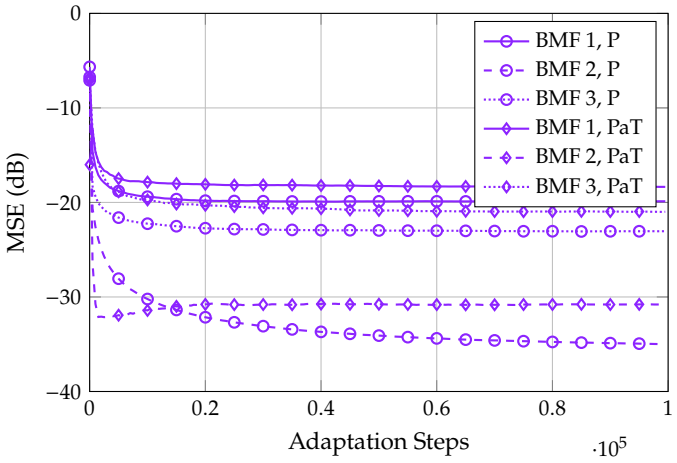


Figure 6.3 Learning curves for equalization with the BVF for three benchmark models for postdistortion setup (P, Fig. 2.5) and for the postdistortion and translation setup (PaT, Fig. 2.6). Adapted from [2] © IEEE 2020.

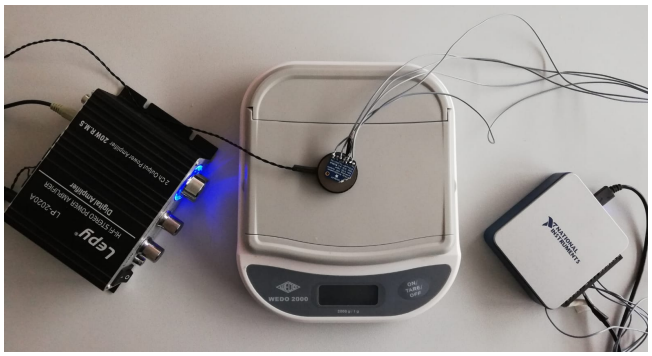


Figure 6.4 Physical setup for evaluating the equalization performance. Adapted from [2] © IEEE 2020.

often introduced. Then, similar to the simulation, we first equalize the actuators offline by recording output signals in full. Here, we use the postdistortion setup for adapting the equalization filter. Finally, we move to the online case, where we use the postdistortion and translation setup to equalize the actuator on the fly.

6.4.1 Actuator Distortions

In order to showcase the actuator distortions, we first define the reference hardware setup used for the evaluation. This reference hardware setup is shown in Fig. 6.4. Overall, we employ a C-2 Tactor [129] as reference actuator. Before the actuator, we place an LEPY2020A amplifier that amplifies the input signals from the audio jack of a computer. On the vibrating area of the actuator, an ADXL335 accelerometer [140] is attached. This accelerometer is held in position by a stylus that presses it onto the actuator from the top (not shown in Fig. 6.4). This stylus is held in place by a tripod. With this setup, we reduce variations in the experiment conditions to a minimum and achieve a high reproducibility between individual test runs. By putting the entire actuator/accelerometer/stylus setup on a scale, we can measure the contact force between accelerometer and actuator. The stylus is placed such that it presses the acceleration on the actuator with a force of approximately 1 N. The accelerometer records acceleration in three spatial dimensions. However, since the accelerometer placement is such that the sensor faces downwards, only the measurement along the z-axis is relevant. Before recording, we capture the steady-state value measured in this channel. This value is then subtracted from all measured acceleration signals, so we obtain zero-mean signals.

When using this setup, we need to consider that the input and output signals resemble different physical quantities. The input of the actuator resembles a displacement signal. However, the ac-

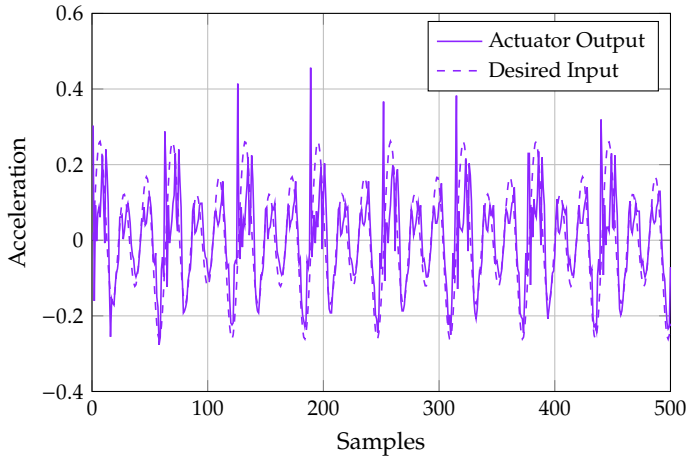


Figure 6.5 Second derivative of the sum of sinusoids input signal (6.13) and corresponding measured actuator output acceleration. Adapted from [2] © IEEE 2020.

celerometer measures the actuator output in terms of acceleration. Thus, we are not able to compare the measured output signal and its filtered signals to the desired signal directly. This means in turn that the setups in Fig. 2.5 and 2.6 cannot be applied directly.

A straightforward solution idea would be to integrate the measured acceleration signal two times before processing. However, this produces very unsatisfying results due to noise and accumulation of accelerometer drift. Thus, to solve this, we propose modified equalization setups, where the desired input signal is derivated two times in Sec. 6.4.2 and 6.4.3.

As reference signal for the desired signal in all experiments, we choose

$$d[n] = \frac{1}{C}\sin\left(\frac{100\,\text{Hz}}{f_S}n\right) + \frac{1}{C}\sin\left(\frac{200\,\text{Hz}}{f_S}n\right) + \frac{1}{C}\sin\left(\frac{300\,\text{Hz}}{f_S}n\right), \tag{6.13}$$

where the constant C is chosen such that $d[n] \in [-1, 1]$ and f_S is the sampling frequency. This signal is chosen because it contains multiple distinct frequencies, which tests the capability of the adaptive filters to adapt well for more than one frequency. Additionally, its second derivative can be computed in closed form.

This reference signal is generated with a length of 10000 samples and $f_S = 1000\,\text{Hz}$ and played back by the actuator. The corresponding output is recorded as a whole. Then, we cut out 500 samples of the recorded signal. This signal portion is cut out from the middle of the recorded signal to avoid border effects. We calculate the second derivative of the reference signal for the same portion of samples.

The recorded output signal and the second derivative of $d[n]$ are shown in Fig. 6.5. It is clearly observable that the recorded output signal contains many distortions compared to the input. First, at the instances where the input signal goes from the minimal to the maximal value, the output shows high overshoots. This is most probably due to the hardware of the actuator that makes the vibratory element go higher than it should be when aiming to recreate a steep slope. Second, we observe undershoots, i.e., the output signal falls short of reaching the values of the input. Third, we see that there are high-frequency signal contents introduced. These are most probably due to noise throughout the hardware setup. Finally, we have places where the output signal goes towards zero, while the input is at a peak of one of the sine waves. When the acceleration reaches zero, this corresponds to a constant displacement. Thus, it means that the input signal is clipped for high signal amplitudes. Such clipping of signal waveforms is a very typical nonlinear effect of actuators but also audio speakers. Therefore, we now have clear observation that confirms the nonlinear behavior of vibrotactile actuators.

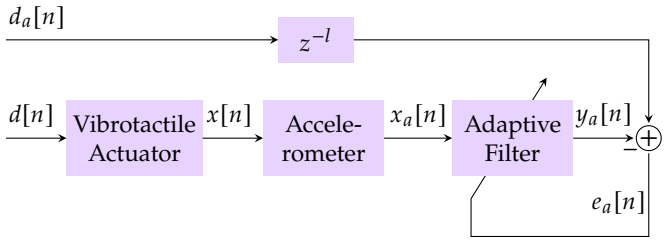


Figure 6.6 Modified postdistortion setup for adaptive offline equalization of the reference hardware setup in Fig. 6.4. Adapted from [2] © IEEE 2020.

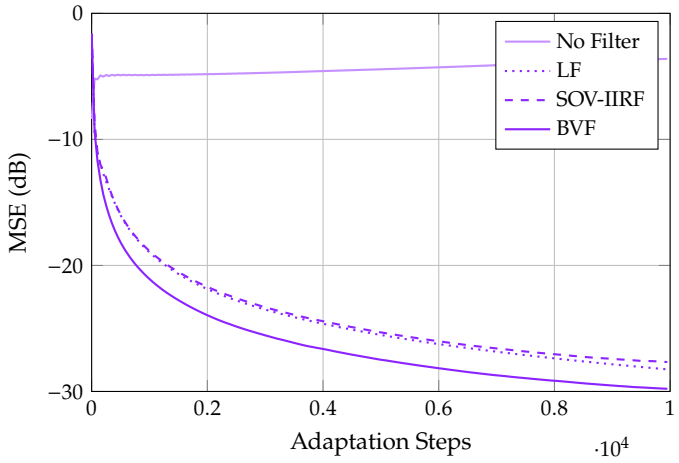


Figure 6.7 Learning curves for no equalization and equalization with three adaptive filter models performed offline using the reference hardware setup in Fig. 6.4 for the input signal from (6.13) using the modified postdistortion setup in Fig. 6.6. Adapted from [2] © IEEE 2020.

6.4.2 Offline Equalization

To perform offline equalization of vibrotactile actuators, we first need to solve the mismatch in signal type. For that, we propose a modified postdistortion setup in Fig. 6.6. Here, we have the desired signal $d[n]$ and its second derivative $d_a[n]$. Since we are performing equalization offline, $d_a[n]$ can be computed from $d[n]$ beforehand as a whole. As the actuator output $x[n]$ is captured by the accelerometer, it is derivated two times, which yields $x_a[n]$. Then, the adaptive filter produces $y_a[n]$, which can be compared to $d_a[n-l]$ as both are acceleration signals.

Now, we can perform offline equalization of the vibrotactile actuator with our three adaptive filter models. For that, we generate $d[n]$ from (6.13) with a length of 5 s at $f_S = 2000$ Hz. This means that $d[n]$ will have 10000 samples in total. For this $d[n]$, the signal $d_a[n]$ can be computed in closed form. All the filter tap numbers are chosen as in Sec. 6.3.3 and for the NLMS step size parameter we choose $\mu = 0.4$.

The learning curves for all filter models are shown in Fig. 6.7. First, observing the learning curve for the case of no equalization, we yet again see the necessity for equalizing actuators. The MSE is at roughly -4 to -5 dB without equalizing. For the three filter models, we see that in general all are suitable to equalize vibrotactile actuators. However, the BVF is clearly the best choice amongst them, achieving the lowest learning curve. This confirms our hypothesis that the BVF with its coupled nonlinearity is a good fit for the actuator distortions, where input and output are also highly coupled. Interestingly, the LF performs better than the SOV-IIRF. This is most probably due to the fact that the uncoupled nonlinearity of the SOV-IIRF does not match the actuator behavior well.

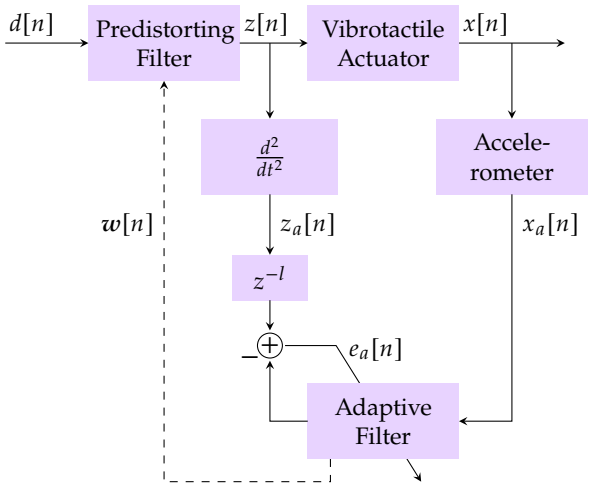


Figure 6.8 Modified postdistortion and translation setup for adaptive online equalization of the reference hardware setup in Fig. 6.4. By the dashed arrow, we denote the copying operation of filter coefficients from the adaptive filter to the predistorting filter in a block-wise manner. Adapted from [2] © IEEE 2020.

6.4.3 Online Equalization

We now move to the online equalization scenario, since this is the most interesting case for vibrotactile actuators. This is because the equalization should lead to physical output signals with higher quality that are available to the human user. The hardware setup for showcasing the online equalization performance remains the same, i.e., the reference hardware setup in Fig. 6.4. Here, the measures taken to achieve high reproducibility are especially important, since we need them to avoid instabilities, where the input signals of the actuators grow uncontrollably.

In order to solve for the mismatch between displacement and acceleration signals again, we propose the modified postdistortion and translation setup in Fig. 6.8. Now, the actuator input signal $z[n]$ is passed through an operator that calculated its second derivative. The derivative can be calculated with the method in [141]. Again, as the accelerometer measures the displacement signal $x[n]$, it outputs the acceleration signal $x_a[n]$, which corresponds to the second derivative as well. Thus, we are now able to compare $x_a[n]$ and $z_a[n]$ to adapt the filter model correctly.

In the setup from Fig. 2.6, the predistorting filter was updated with copied coefficients from the adaptive filter in every time step. However, due to the presence of the hardware setup, we cannot perform this operation in this manner. This is because of the considerable delay that the hardware setup introduces to the system. In our tests, we observed delays of 200 to 300 ms. With that, if we were to wait for every sample to arrive and be processed for the adaptation, we could only achieve a sampling frequency of 3.3 to 5 Hz. On the other hand, if we do not wait but simply copy the coefficients with delay, that means the predistorting filter will be lagging behind always, which may cause instabilities at worst.

Due to this delay, we move to a block-wise updating scheme of the predistorting filter. For that, we first initialize both the predistorting filter and the adaptive filter with $a_0 = 1$ and all other coeffcients equal to zero. This is important, so that at first the signal $d[n]$ is passed on completely unchanged as $z[n]$. Then, first a signal block of certain length L_{ext} is input into the system. The actuator output of this block is recorded and processed to adapt the adaptive filter coefficients. Then, after having processed $L_{short} < L_{ext}$ samples for filter adaptation, we copy the adapted filter coefficients to the predistorting filter. The shorter block length L_{short} is required so the signal can be played uninterruptedly at the actuator output. If we were to wait for the entire block of L_{ext} for copying coefficients, we would experience a gap in the actuator playback. After the first block has been displayed, we can continue with the block length L_{short} for both the adaptation and the actuator playback. Performing the NLMS optimization for the block L_{short} and copying the filter coefficients is very efficient, requiring only a few microseconds. Therefore, we can now operate at sampling frequencies of 3 kHz easily.

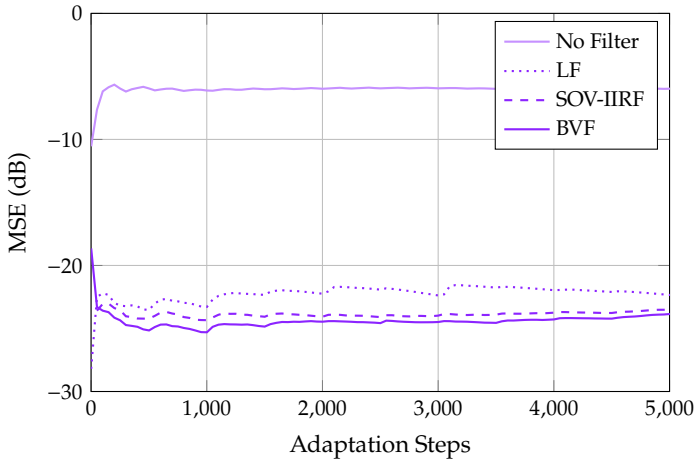


Figure 6.9 Learning curves for no equalization and equalization with three adaptive filter models performed online using the reference hardware setup in Fig. 6.4 for the input signal from (6.13) using the modified postdistortion setup in Fig. 6.8. Adapted from [2] © IEEE 2020.

We perform two experiment for online equalization. For the first, we choose the input signal $d[n]$ from (6.13). We generate this signal at a length of 5 s with a sampling frequency of $f_S = 1000$ Hz. In total, we therefore have 5000 signal samples. The adaptation block length is chosen as $L_{short} = 500$ samples and the longer playback block length for the first block is $L_{ext} = 600$. This means that the predistorting filter will receive updated coefficients every 0.5 s. We choose the filter tap numbers as in Sec. 6.3.3. The NLMS step size parameter is now $\mu = 0.1$.

We show the computed learning curves for this first experiment in Fig. 6.9. First, it is clearly visible that we update the predistorting filter after every 500 samples due to the slight increase in MSE right after these points. However, as the filter adaptation settles in and the coeffcients do not change a lot anymore, this effect vanishes. In general, the learning curves have a similar shape as before. This shows that despite the actuator and filter being nonlinear, the filter order can be exchanged Again, the BVF is the best choice among the three different adaptive filter models.

For the second experiment, we choose $d[n]$ to be one of the real vibrotactile signals from the LMT reference dataset. In particular, we choose the signal from the material *cork*, recorded with the *spike* tooltip at *slower* speed. Since this signal is only 1 s long, we repeat it five times to generate a signal of 5 s length. The signal has a sampling frequency of 2800 Hz, thus we choose $L_{short} = 1400$ to update the predistorting filter every 0.5 s again. All the other parameters are chosen as in the first online experiment.

The learning curves for the second experiment are depicted in Fig. 6.10. First, we see that the MSE has higher fluctuations than before. This most probably comes from the nature of the vibrotactile signal that contains more high-frequency components than the test signal before and also changes greatly over time in contrast to the steady signal from before. Second, the effect of the predistorting filter update is again visible but considerably less significant. Again, this comes from the noisy nature of the input signal. Finally, again the BVF is the best filter model choice. Thus, we have clear indication that the BVF is highly suitable for the equalization of vibrotactile actuators independent of the input signal.

6.5 Chapter Summary

In this chapter, we introduced the concept of adaptive filter equalization for the mitigation of actuator distortions. For that, we first presented a novel adaptive filter model called bilinear Volterra filter (BVF). Through the inclusion of nonlinear feedback with a bilinear term into a model of the well-known second order Volterra filter (SOVF) class, we were able to provide an adaptive filter that

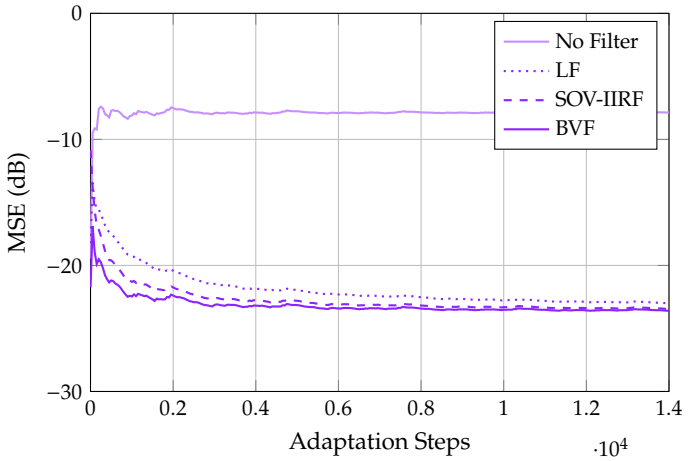


Figure 6.10 Learning curves for no equalization and equalization with three adaptive filter models performed online using the reference hardware setup in Fig. 6.4 for a real vibrotactile signal as input, using the modified postdistortion and translation setup in Fig. 6.8. Adapted from [2] © IEEE 2020.

delivers arbitrary order nonlinear behavior and enhanced stability, compared to previous nonlinear filter models. After presenting the filter model, we developed equalization setups that are specifically tailored for the equalization of vibrotactile actuators. In particular, we accounted for the translation from displacement to acceleration that takes place in vibrotactile actuators by introducing a second order derivative operation into the setup. Additionally, we ensured that the equalized signal is physically available at the actuator output by designing a setup where, the adaptive filter model is copied to the front in a block-wise fashion. Our results showed a consistent improvement in equalization performance over state-of-the-art adaptive filter models for both artificial as well as recorded test input signals.

Chapter 7

Conclusion and Outlook

To conclude this work, we begin by giving an overview of the presented technologies. Then, we discuss remaining limitations and outline potential future challenges. Finally, we sketch pathways to solve these challenges with technologies and methods that need to be investigated and developed.

7.1 Summary

The research area of haptics is gaining momentum, enabling the development of never-before-seen applications more quickly than ever before. This momentum stems from the development of enabler technologies that build the foundation of high-quality haptic experiences. The framework for the high-fidelity processing of vibrotactile signals presented in this work resembles a set of such enabler technologies.

First, we presented a set of two high-performance codecs that are able to compress vibrotactile signals efficiently. The vibrotactile codec with perceptual wavelet quantization (VC-PWQ) was developed to compress single-channel vibrotactile signals, i.e., signals recorded from surface interactions with one point of interaction. Through modern technologies like the discrete wavelet transform (DWT), set partitioning on hierarchical trees (SPIHT), and arithmetic coding (AC), as well as efficiently designed quantization and bit allocation and of course the psychohaptic model that leverages human perceptual limitations, we were able to meet all the capabilities a modern codec should have. This includes being completely rate-scalable, having perceptual transparency (even for fairly aggressive compression), executing quickly, having a modular structure to enable easy improvement, and being versatile to cope sufficiently with many different kinds of signals and application scenarios. Then, we extended the codec framework to the multi-channel vibrotactile codec (MVibCode) that leverages inter-channel correlations in multi-channel vibrotactile signals to achieve even higher compression while maintaining pristine perceptual quality. Central to the approach is a hierarchical clustering (HC) method that dynamically groups channels by their similarity and establishes a prediction hierarchy between them at the same time. This method allows for high flexibility; this is essential for the novel field of vibrotactile multi-channel signal processing, where no established standards exist yet.

Second, we move on to the extensive evaluation of codec performance from a perceptual standpoint. We first presented the objective quality results of the two developed codecs and compared them to the state of the art where possible. Through this, we were able to gain some initial insights into the behavior of the codecs at different rates. In order to evaluate codecs perceptually, we first presented a streamlined and comprehensive human user experiment procedure. In the experimental process, human assessors rate the subjective quality of compressed signal data in terms of similarity to the respective original signal. Through rigorous timing, inclusion of catch trials and assessor selection criteria, and the design of an intuitive graphical user interface (GUI), we are able to deliver a method for obtaining reliable perceptual signal quality ratings. After, we showcase the possibility of computing perceptual ratings from signal data that are close to the measured scores. For that, we both present a subjective quality metric called the spectral perceptual quality index (SPQI) as well as a fusion method based on machine learning that intelligently combines multiple metrics. With this

vibrotactile multi-method assessment fusion (VibroMAF) approach, we are able to achieve a good fit with the measured rating scores. The scheme is designed to be easily enhanceable in the future.

Third, we shifted focus to the signal processing after encoded signal transmission and decoding, where we strive to increase the vibrotactile signal quality again that was reduced during the lossy coding stage. For that, we employ recurrent neural networks (RNNs) with residual learning (RL) to learn a mapping from the compressed signal waveform to an enhanced signal that better resembles the original signal. Here, RL helps to enhance performance for vibrotactile signals that have shown to have a large dynamic range by reducing this dynamic range. By including additional side information on the compression of signals, we were able to increase performance substantially. Our proof of concept can be used as a blueprint for the development of enhancement methods for newly emerging codecs and signal datasets.

Finally, we targeted the reduction of distortions from vibrotactile actuators when signals are recreated for humans to perceive them. For that, we took well-established adaptive filtering methods in terms of equalization setups. We introduced a novel filter model that is able to better reflect the actuator behavior and therefore reduce distortions more efficiently in comparison to previously used filter models. Overall, this means that now when rendering signals for human users, one can create a better, more realistic experience.

7.2 Limitations

Although substantial enhancements over the state of the art were achieved with the presented contributions, it is important to consider the limitations of the presented approaches. Through these, future research directions and possible enhancements are easily identifiable.

For the single-channel vibrotactile codec VC-PWQ, a first limitation concerns the available threshold measurements. As described, no consensus on the exact shape of the absolute threshold of vibration (ATV) exists. Thus, the codec might be performing suboptimally with the current ATV model function. Also, the measurements of masking thresholds are fairly limited in number and frequency range, so the chosen masking threshold model functions might lead to a suboptimal bit allocation as well. For the second limitation of the VC-PWQ, we have that the bit allocation of the codecs is currently steered by the bit budget. This means that bits are allocated to individual wavelet bands until a certain bit budget is reached. In general, when using the DWT, the bitrate contribution of an allocated bit can be very different depending on the wavelet band it was allocated to. For example, allocating one bit to the highest-frequency wavelet band leads to a much higher rate overall than when allocating to a lower-frequency band, due to the different number of coefficients in each band. Thus, overall, we are faced with a large spread in output data rate for equal bit budgets.

Concerning the MVibCode, first due to the low amount of signal data for testing and the lack of suitable display devices, a perceptual evaluation through experiments was not feasible. Therefore, it may be the case that the current codec design, especially the heuristic for the clustering metric threshold, is overfitted to work well with the signals in the CEA reference dataset. Therefore, the generalization capabilities of the codec are somewhat unclear. Also, the designed clustering metric does not yet take into account the spatial arrangement of channels or human biomechanical limitations.

For the vibrotactile quality assessment (VQA) experiment, the most significant limitation is that it does not allow for very fine comparison of compressed signal quality. For one, having assessors distinguish very fine details in signals requires expertise, which is not yet available to sufficient extent in the vibrotactile domain. On the other hand, the multi-stimulus test with hidden reference and anchor (MUSHRA) on which our VQA experiment was based is designed for the evaluation of medium quality audio signals. Thus, it already is designed for rather coarse comparison of signals.

Regarding the perceptual quality metrics, their capability is currently limited the most by the poor availability of suitable signal data for evaluation and parameter tuning. It is probable that the found parameters for the SPQI are not universally optimal and deliver good results for other signal datasets.

Similarly, the VibroMAF may not yet generalize very well, due to the very low number of training and test signals.

For the quality enhancement method with RNNs, it needs to be emphasized that the method presented in this work is very much tailored for the particular signals from the LMT reference dataset compressed with the VC-PWQ. This is, among others, due to factors like the inclusion of the bit budget as side information, the normalization of signals to the range of 50 and the particular number of layers and neurons. The trained network is not straighforwardly applicable to other signals compressed with other codecs but would need to be adapted and optimized first. Therefore, as described before, the presented method acts as a proof of concept.

Finally, concerning the adaptive filtering equalization, it is worth noting that the algorithm does not reduce distortions perceptually. Currently, the normalized least mean squares (NLMS) algorithm seeks to minimize the objective error between distorted and desired signal. This means that only the objective quality of the displayed signal is enhanced by the equalization.

7.3 Next Challenges and Solution Sketches

After we have enabled new applications with the presented methods, new challenges arise. These include both improvements to the presented methods as well as challenges from newly enabled scenarios that were not feasible before.

When it comes to the vibrotactile codecs, several initial improvements are conceivable. First, the psychohaptic model should be further enhanced with more measurements of the ATV and masking thresholds in order to better grasp human perceptual effects. In the MP3 codec, maskers are categorized as tonal or noise-like maskers to compute more accurate masking thresholds. Thus, by conducting more extensive experiments to measure masking thresholds in a more holistic fashion, we can develop a more accurate model function for different kinds of maskers. Also, the bit allocation procedure could be revisited to move away from the SMR-based approach to a more accurate method. Here, MP3 could again serve as inspiration, where in the Layer-3 version, it used noise-shaping methods rather than the more simple approach with the SMR.

The rate scaling of the VC-PWQ should be further investigated, since this can turn out to be suboptimal for some signals when using the bit budget as scaling parameter. Thus, instead of the bit budget, we should investigate switching to a procedure where the total rate of the current signal block is calculated in each step of the bit allocation loop. The stopping criterion would be based on a limit for the calculated rate. This could lead to a more uniform distribution of resulting rates across different signals.

For the MVibCode, the HC clustering metric should be further optimized with additional components that take into account the spatial distance between two channels. Through this, one could enable better encoding of multi-channel signals recorded from irregularly spaced sensors on the human hand. Ultimately, to keep up with the expected fast-paced development of multi-point setups, the codec and clustering method should be enhanced towards a human-body-centered hierarchical approach. This means, in addition to the hierarchy from signal similarity, the codec should also consider hierarchy arising from the structure of the human body and perceptual limitations in this context. As an example, we can establish models where signal channels are first grouped for fingers, then across the hand, then across the arm and then across the entire body. Through such a structure, computational complexity could be decreased and flexibility could be increased as the codec can be designed to allow for semantic information input on which channels from which parts of the human body need to be encoded. For example, if the display device on the decoder side only supports tactile feedback on the hand, but the input signal has been measured for the entire body, the codec can use this information to encode and transmit only the necessary parts.

To this end, hybrid tactile signal transmission approaches are also envisionable, where some parts of vibrotactile signals are recorded and transmitted for display on the receiver side and some others are rendered locally from a signal database. By predicting human behavior, the codec can be enhanced

to dynamically switch between modes and save data rate. This can be especially critical in control scenarios under the presence of delay.

An entirely new application where the developed codecs could play a large role is skill transfer in the context of the tactile internet. By using tactile cues, we could be able to increase the learning performance in all kinds of robotic tasks, e.g., tasks requiring fine motor skills or grasping tasks. In particular, the ability for remote machine learning with tactile signal information in the presence of lossy compression would be an exciting new research direction that is only enabled with the existence of efficient codecs.

Moving on to the quality assessment framework, two future research aspects stand out. First, the developed VQA experiment should be employed to measure more extensive rating scores with more participants and more signals. With more data, we would be able to conduct meaningful investigations into effects arising from age-related differences or experience-related differences in perceptual signal quality. Second, the higher amount of data could be used to enhance the developed automated quality assessment metrics. This holds for both the SPQI as well as the VibroMAF, where using more rating data could be beneficial for a more accurate automatically predicted rating score.

For the decoder-side quality enhancement with neural networks (NNs), there are two clear avenues for improvement. First, the RNN can be designed to work perceptually by using a perceptual quality metric rather than the mean absolute error (MAE). Since our aim is to enhance signals perceptually, this approach could prove advantageous. With the modified loss function, the network could be trained to remove only the distortions that are most relevant in terms of perception. Here, function principles of autoencoders could also serve as inspiration. Second, the employment of convolutional neural networks (CNNs) instead of RNNs for quality enhancement should be investigated. These NNs have been proven to achieve good performance in the image domain, and thus are also a good candidate for the enhancement of vibrotactile signals. This is especially true for multi-channel vibrotactile signals, where CNNs could have an advantage with the joint processing and convolution across multiple channels.

Finally, for the adaptive filtering actuator equalization approach, an improvement worth investigating is to add a perceptual component to the optimization algorithm. By changing the cost function to include a perceptual metric, we can alter the adaptive filter parameters to specifically increase perceptual signal quality.

Appendix

Appendix A

Discrete Wavelet Transform on Vibrotactile Signals

A.1 Wavelet Theory Principles

The wavelet transform operates by using so-called wavelet and scaling functions. These two functions get scaled and shifted in order to form a basis for $\mathbb{R}$. The wavelet function, also called mother wavelet $\psi(x)$ and the binary transformed wavelets $\psi_{j,k}(x)$ are related by [142]

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), \quad x \in \mathbb{R} \quad j,k \in \mathbb{Z}. \tag{A.1}$$

The scaling parameter j squeezes and stretches the mother wavelet by a factor of two and the translation parameter k shifts the function by an integer.

The scaling function is usually denoted as $\phi(x)$. Let the scaled and translated version of $\phi(x)$ be [142]

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k), \quad j,k \in \mathbb{Z}. \tag{A.2}$$

The construction of a wavelet orthonormal basis for the Hilbert space $\mathbb{R}$ relies on the concept of multiresolution analysis (MRA) introduced in [143]. By using this concept one can define the spaces

$$W_j = s\tilde{p}an\{\psi_{j,k}(x)\}_{k\in\mathbb{Z}} \tag{A.3}$$

and

$$V_0 = s\tilde{p}an\{\phi_{0,k}(x)\}_{k\in\mathbb{Z}}. \tag{A.4}$$

Using the union of those spaces creates the space $L^2(\mathbb{R})$

$$L^2(\mathbb{R}) = V_0 \oplus (\bigoplus_{j\in\mathbb{N}_0} W_j) \tag{A.5}$$

A function $f(x) \in L^2(\mathbb{R})$ can finally be represented by a wavelet series expansion [144]

$$f(x) = \underbrace{\sum_{k\in\mathbb{Z}} \langle f, \phi_{0,k}\rangle \phi_{0,k}(x)}_{\in V_0} + \underbrace{\sum_{j=0}^{\infty}\sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}(x)}_{\in W_j}.^{1} \tag{A.6}$$

Instead of using continuous functions one can adapt the method for discrete signals. In this case, the wavelet and scaling function must be replaced by wavelet filters and scaling filters. From [145] we know that

$$a_j[k] = \sum_n h_\phi[n-2k]a_{j+1}[n] \quad j \in \{\mathbb{N}_0 \mid 0 \le j < J\} \tag{A.7}$$

$$d_j[k] = \sum_n g_\psi[n-2k]a_{j+1}[n] \quad j \in \{\mathbb{N}_0 \mid 0 \le j < J\}, \tag{A.8}$$

[1] $\langle f(x), g(x)\rangle = \int_{-\infty}^{\infty} f(x)\tilde{g(x)}dx \quad f,g \in L^2(\mathbb{R})$

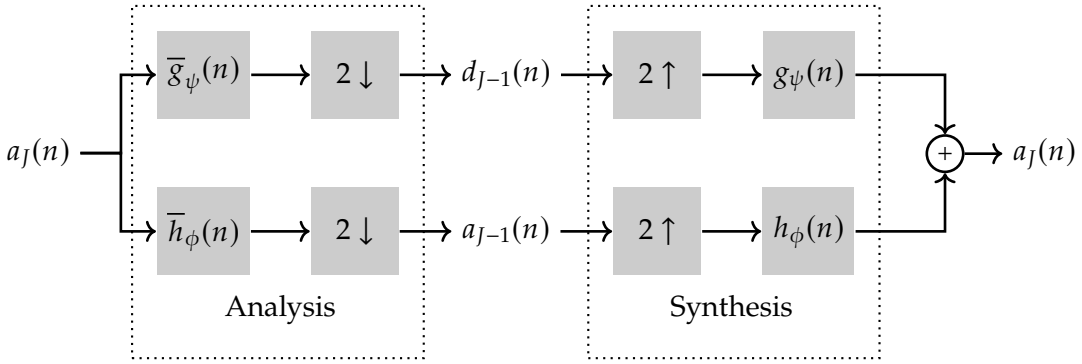


Figure A.1 Two-channel filter bank for DWT.

where $a_{j-1}[k]$ is the approximation of $f(x)$ at the resolution j in the space V_{j-1}. and $d_{j-1}[k]$ are the details or the error between the resolutions in the space W_{j-1}. Iterating through $a_j[k]$ will result in lower resolution spaces until the space W_0 and V_0.

If $\phi(x)$ generates a MRA, it is possible to invert the decomposition using [145]

$$a_{j+1}[k] = \sum_n a_j[n]h_\phi[k-2n] + \sum_n d_j[n]g_\psi[k-2n], \quad j \in \{\mathbb{N}_0 \mid 0 \leq j < J\}. \tag{A.9}$$

In this way it is possible to implement the wavelet transform for discrete functions. These equations can also be expressed through convolution with filters by

$$(A.7) \Leftrightarrow a_j[k] = \tilde{h}_\phi * a_{j+1}[2k] \quad [2] \tag{A.10}$$

$$(A.8) \Leftrightarrow d_j[k] = \tilde{g}_\psi * a_{j+1}[2k] \tag{A.11}$$

$$(A.9) \Leftrightarrow a_{j+1}[k] = h_\phi * \check{a}_j[k] + g_\psi * \check{d}_j[k] \quad [3] \tag{A.12}$$

It is possible to implement (A.10), (A.11) and (A.12) as a filter bank as depicted in Figure A.1. Applying the discrete wavelet transform (DWT) is called analysis, whereas the reconstruction is called synthesis.

Applying the DWT again on the approximation coefficients results in a multilevel DWT, where each recursion step is called one level. Because h_ϕ approximates the function at the resolution j it can be characterized as a low-pass filter. Filter g_ψ generates the details for the space W_j and is therefore a high-pass filter. If the filter bank is able to restore the original signal from the decomposition, it is called a perfect reconstruction filter bank. When designing filters for the DWT, we always aim to have perfect reconstruction.

A.1.1 Wavelet Properties

Wavelets and their corresponding filters have different properties which determine their compression capabilities for specific signals. In this section, a brief overview is given.

A.1.1.1 Orthogonality

Generally, one can distinguish between two types of wavelets. The first one being orthogonal wavelets created through the MRA. However, there is a more generalized form of the MRA [145]. Using this generalized form, it is possible to construct so-called biorthogonal wavelets that have several beneficial properties (see Sec. A.3.4 and A.3.5).

[2] Mirroring: $\tilde{h}[k] = h(-k)$

[3] Downsampling: $\check{a}[k] = \{a(k/2) \text{ if } k \text{ is even}, 0 \text{ if } k \text{ is odd}\} \quad k \in \mathbb{Z}$

However, there is one key disadvantage of biorthogonal filters. In general, they don't preserve energy during the transform. Quantizing the coefficients from the DWT and reconstructing them could magnify the quantization error if they are not orthogonal, which could make them numerically unstable. Furthermore, the sum of high- and low-pass channel energy could increase and hence the entropy would become larger. These issues can be minimized by choosing biorthogonal filters that are as close as possible to orthogonality.

A.1.1.2 Vanishing Moments

The number of vanishing moments (VMs) of a wavelet function is a key property when determining the compression capabilities of the DWT. The highest degree of a polynomial at which the function $\psi(x)$ is still orthogonal, is referred to as VMs. This is expressed by [144]

$$\int_{-\infty}^{\infty} x^{\ell}\psi(x)dx = 0 \quad \text{for} \quad \ell = 0, 1, \ldots, N-1, \tag{A.13}$$

where N is the amount of VMs. The discrete version can be written as

$$\sum_{n=-\infty}^{\infty} n^{\ell} g_{\psi}[n] = 0. \tag{A.14}$$

This means that a polynomial of degree $N-1$ will result in detail coefficients being zero, which undoubtedly is beneficial for compression. Increasing the number of VMs will capture the energy of the signal in the space V_j or low sub-band for smooth signals.

A.1.1.3 Support / Filter Length

One could assume that taking more VMs would result in better compression. However, when increasing the number of VMs the support of the wavelets gets larger, which implies a higher filter length.[4] Thus, discontinuities in the signal can create large undesired coefficients that appear multiple times if the wavelet is large in size. Small size filters will therefore produce fewer large coefficients. Hence, the consideration between VMs and support has to be made depending on the smoothness of a signal [146].

A.1.1.4 Group Delay

Digital filters usually suffer from phase distortion, i.e., they introduce frequency dependent delay. This can be undesired for encoding. To overcome this, linear phase filters can be used. These filters have to be symmetric or antisymmetric[5]. Only biorthogonal wavelets fulfill this condition for compactly supported wavelets.

A.2 Wavelets Families

There are infinitely many wavelet functions that satisfy the MRA condition.[6] However, there are wavelets with specific properties that make them unique. In this work, we examine three families of wavelets.

[4] In the best case, the support size is just $2N-1$ for Daubechies wavelets.
[5] $x[n] = \{0 \text{ if } N \text{ is even and } n = N/2, -x(N-n) \text{ else}\} \quad n, N \in \mathbb{N}$
[6] More precisely, the Quadrature Mirror Filter condition that derives from the MRA. More detail in [145].

A.2.1 Daubechies Wavelets

Introduced in [147], these Daubechies (DB) wavelets named after their inventor are primarily characterized by their VMs. The naming convention is Daubechies wavelet 1 (DB1) for one VM, DB2 for 2 VMs and so on. They are defined for all positive integers. DB1 is equal to the Haar wavelet. They are minimal in size for a given number of VMs. Daubechies wavelets have a support size of $2N-1$, i.e., a filter length of $2N$ for N VMs. They become smoother with increasing number of VMs.

A.2.2 Least Asymmetric Wavelets and Symlets

In order to reduce the phase distortion of DB wavelets, Least Asymmetric wavelets can be used [148]. For more than three VMs DB wavelets can be modified without changing the main properties like orthogonality, magnitude of the frequency response, filter length, and smoothness. This degree of freedom was used to design least asymmetric wavelets that are as close as possible to a linear phase filter. They are also called Symlets and follow the same nomenclature as DB wavelets.[7]

A.2.3 Biorthogonal Wavelets

There are different types of biorthogonal wavelets. We focus on the first ones that were found by A. Cohen and I. Daubechies [149]. They again are defined by their VMs. However, they can have a different number of VMs for the scaling and the wavelet function. Therefore the naming convention is "Bior" followed by the number of VMs of the scaling function and the wavelet function, respectively. For example, a Bior2.4 has two moments in the scaling function and four in the wavelet function. It is not possible to construct biorthogonal wavelets for any arbitrary number of VMs [149].

Bior4.4 and Bior2.2 are often called CDF-9/7 and LeGall-5/3 filters and are frequently used in image compression [20]. The Bior4.4 and Bior6.8 wavelet are almost orthogonal, i.e., they preserve signal energy very well [150].

A.2.4 Non-expansive DWT

The DWT is an expansive transform. This means for finite length signals, more output coefficients are obtained than input signal samples. For a level 1 DWT, input signal length N and even filter length L, the number of wavelet coefficients is $N+L-2$. [8] If and only if the filter length L is two the DWT is non-expansive, which would limit us to the Haar wavelet.

To avoid expansion, we can make use of the fact that the convolution of a periodic signal[9] with any filter will result in a periodic outcome. In order to produce periodic coefficients after the analysis, the finite length input signal has to be extended periodically before transform. The period P is the signal length N. After downsampling, the coefficients need to be windowed to length $N/2$. Both channels combined result in N coefficients and thus the DWT is no longer expansive.

However, periodic extension has the major drawback of creating large discontinuities if the first and last signal value are different. This leads to encoding of high frequency components not corresponding any useful information.

To overcome this, we can extend the input signal symmetrically. If and only if a filter is symmetric itself, the outcome of the convolution will be symmetric. In our case, it is the wavelet and scaling filter that have to be symmetric. Depending on the symmetry of the filter, the extension is either whole-point or half-point symmetric.

For symmetric extension we are limited to biorthogonal wavelets. Thus, it has to be considered whether orthogonality or the capability of symmetric extension is more desired. Further details on the implementation of symmetric and periodic extension for DWT can be found in [151]–[153].

[7] Sym1, Sym2, Sym3, ...
[8] For odd filter length $N+L-1$
[9] $f[n] = f[n+P] \quad \forall n \in \mathbb{Z} \quad P \in \mathbb{N}$

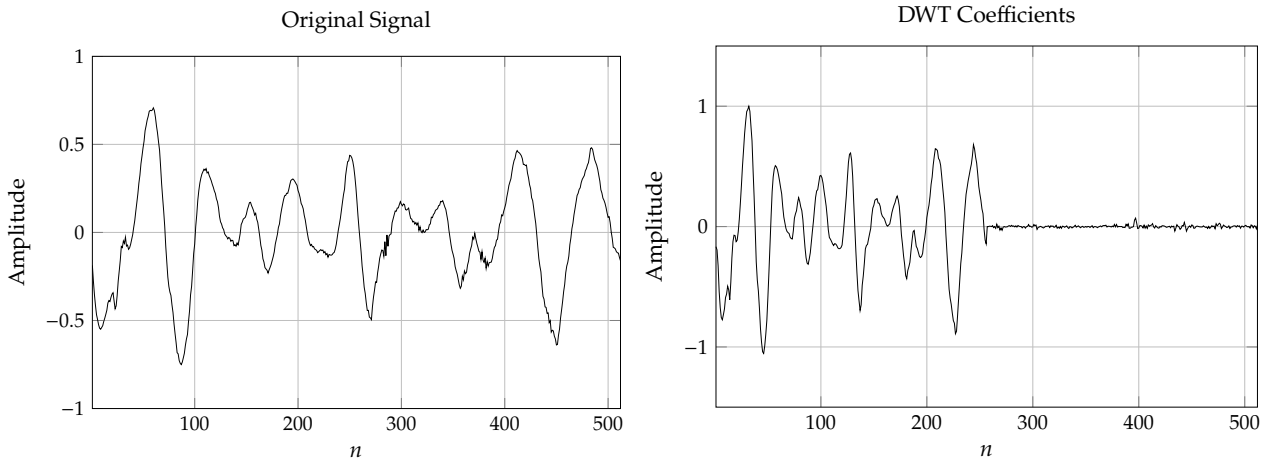


Figure A.2 DB6 one level DWT with periodic extension.

A.3 DWT on Vibrotactile Signals

We evaluate the compression capabilities of the DWT on vibrotactile signals. As reference signals we use the 280 vibrotactile signals from the LMT reference dataset.

A.3.1 Energy Distribution

In order to be able to compress signals effectively, most wavelet coefficients should be close to zero. This can be achieved by taking a scaling and wavelet filter that shift most of the energy into the low-pass band. If all the signal energy would be contained within the low-pass subband, only half of the coefficients need to be stored. Thus, comparing the energy in each wavelet band gives us a good estimation of how well a certain wavelet performs.

Fig. A.2 shows the original signal and wavelet coefficients of a 1-level DWT for a signal block of length 512 using DB6 filters with periodic extension. On average the low-pass coefficients have much higher energy than the high-pass coefficients. This is the desired outcome, since the detail coefficients are very small and can probably be omitted. In the following, this energy distribution is evaluated for varying parameters like VMs, filter length, phase, extension method, block length and DWT.

A.3.2 Test Setup

Each of the 280 vibrotactile test signals has 2800 samples. Each signal is split up into blocks with either 32, 64, 128, 256, 512 or 1024 samples. We will consider 512 as a standard block length for our tests. The average result over all signals and blocks is calculated. The tests are conducted with a self-implemented DWT algorithm in MATLAB.

A.3.3 Vanishing Moments

First, we examine the influence of VMs. In general, the more VMs a wavelet has, the smaller the detail coefficients are. This highly depends on the input signals though.

We compare DB wavelets with different moments in a level one DWT on a block length of 512. Due to the lack of symmetry in the wavelet and scaling filter, a periodic extension is used. The signal energy from the low-pass and high-pass channel is then computed by

$$E = \sum_{n \in I_\gamma} x[n]^2, \quad \gamma \in \{l, h\}, \tag{A.15}$$

Table A.1 Energy distribution between high- (H) and low-pass (L) bands for different VMs on DB wavelets.

Name	VMs	Energy L	Energy H	Energy Total
db1	1	89.68%	10.32%	100%
db2	2	91.03%	8.97%	100%
db3	3	91.50%	8.50%	100%
db4	4	91.87%	8.13%	100%
db5	5	91.97%	8.03%	100%
db6	6	92.06%	7.94%	100%
db7	7	92.22%	7.78%	100%
db8	8	92.21%	7.79%	100%
db9	9	92.27%	7.73%	100%
db10	10	92.35%	7.65%	100%
db11	11	92.31%	7.69%	100%
db12	12	92.38%	7.62%	100%
db13	13	92.41%	7.59%	100%
db14	14	92.37%	7.63%	100%
db15	15	92.45%	7.55%	100%
db16	16	92.44%	7.56%	100%
db17	17	92.42%	7.58%	100%
db18	18	92.49%	7.51%	100%
db19	19	92.45%	7.55%	100%
db20	20	92.47%	7.53%	100%

where I_γ is the index set of the low-pass or high-pass coefficients, respectively. The energy of each channel is divided by the total input signal energy. The mean values over all blocks are given in Table A.1.

We see that for all DB wavelets the DWT is able to locate most of the energy in the low-pass coefficients. Only about 10% of the signal energy remains in the details. The low-pass energy increases with increasing number of VMs. We can say that for 10 VMs we run into saturation with no significant further improvement.

We observe a slight fluctuation in the energy distribution. This can mostly be explained by the large variations between different signals and the periodic extension introducing arbitrary discontinuities.

We also investigate the influence of the block length. For that we again average the energy distribution over all test signals. The results are depicted in Fig. A.3. The longer the blocks are the more energy goes into the low-pass coefficients. Again we run into saturation at 512 samples with no significant improvement from there on.

A.3.4 Phase

Next, we examine the effect of asymmetric filters on the DWT. We compute the DWT of length 256 of an exemplary signal using DB wavelets and Symlets. The resulting low-pass coefficients are depicted in Fig. A.4. It is clearly visible, that for DB wavelets we have a significant shift. This can have bad implications for coding. Symlets exhibit almost no shift at all. The average energy distribution ist almost identical between Symlets and DB wavelets.

A.3.5 Symmetric and Periodic Extension

We examine the effect of the extension method on the energy distribution. For this we use biorthogonal wavelets. The symmetry type used, is the one that results in perfect reconstruction. We compute the DWT with periodic symmetric extension and compare the average energy distribution. For that we compute the difference between the average low-pass energy percentages for different block lengths. The results are shown in Fig. A.5.

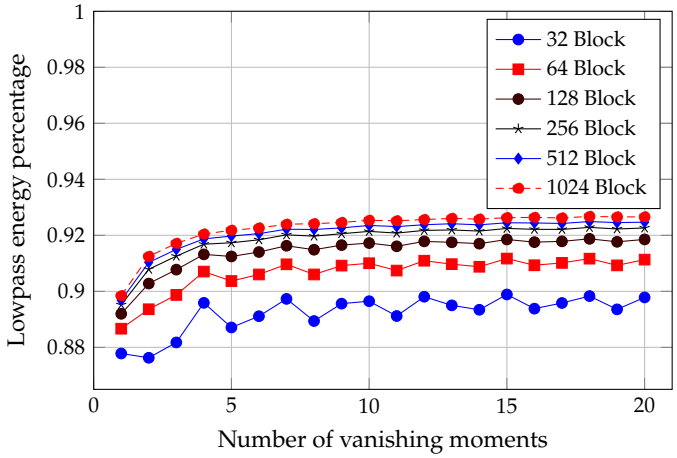


Figure A.3 Percentage of low-pass band energy to input signal energy for varying block length and VMs on a level one DWT with DB wavelets.

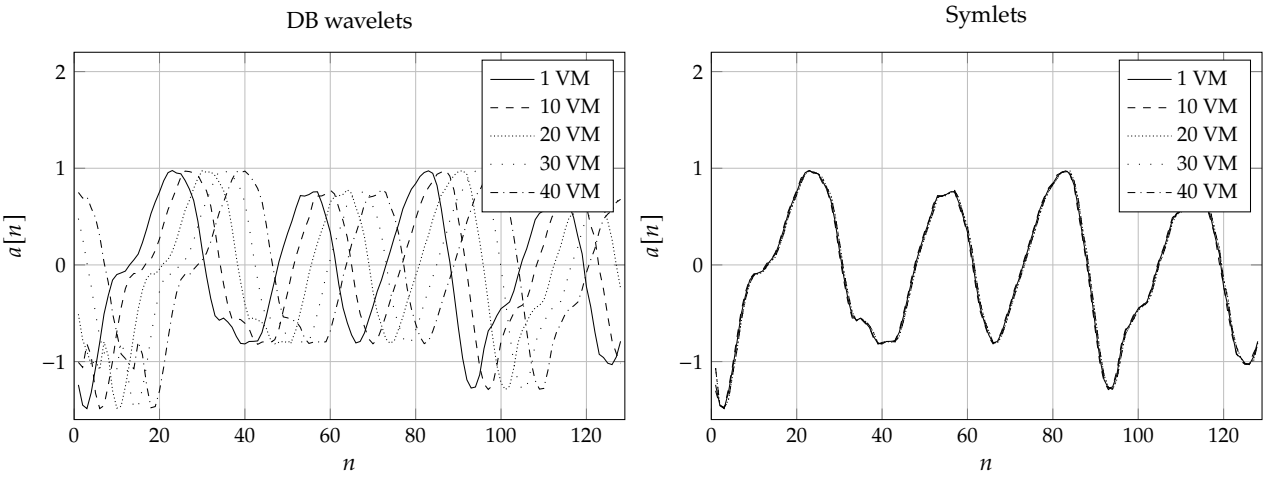


Figure A.4 DWT approximation coefficients for different VMs.

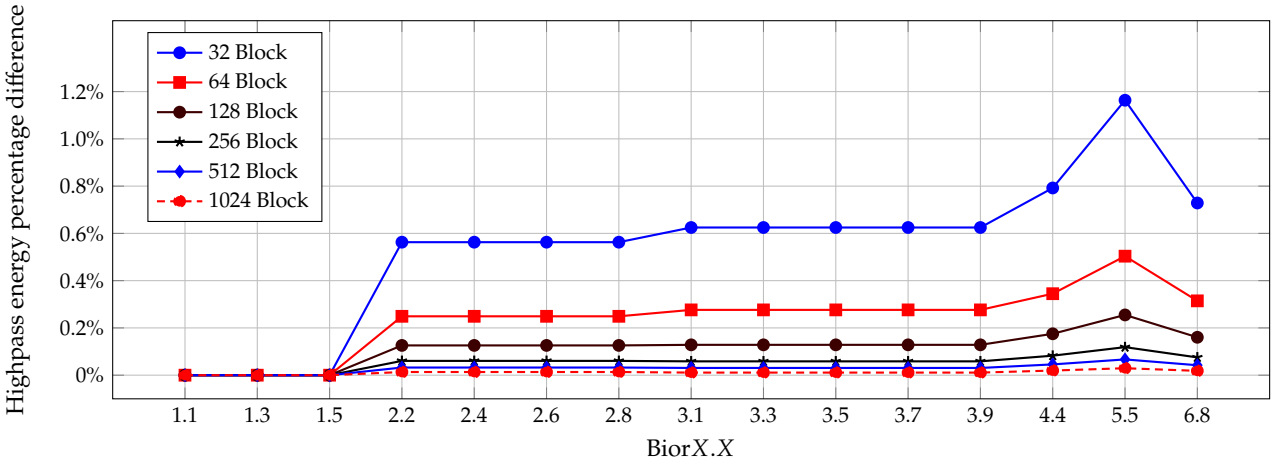


Figure A.5 Difference in approximation energy percentage between symmetric and periodic extension.

It is clearly visible that symmetric extension is always superior to periodic extension. The effect becomes more significant for smaller block lengths. Overall the magnitude of the effect is relatively low, so we believe it is less significant than other wavelet properties.

List of Figures

List of Tables

Bibliography

Publications by the Author

Journal Publications

[1] M. Strese, R. Hassen, A. Noll, and E. Steinbach, "A tactile computer mouse for the display of surface material properties", *IEEE Transactions on Haptics*, vol. 12, no. 1, pp. 18–33, 2018.

[2] A. Noll, C.-D. Curiac, B. Gülecyüz, and E. Steinbach, "Adaptive equalization of vibrotactile actuators", *IEEE Transactions on Haptics*, vol. 14, no. 2, pp. 371–383, 2020.

[3] A. Noll, A. Gürbüz, B. Gülecyüz, K. Cui, and E. Steinbach, "Quality enhancement of compressed vibrotactile signals using recurrent neural networks and residual learning", *IEEE Transactions on Haptics*, vol. 14, no. 2, pp. 316–321, 2021.

[4] E. Muschter, A. Noll, J. Zhao, *et al.*, "Perceptual quality assessment of compressed vibrotactile signals through comparative judgment", *IEEE Transactions on Haptics*, vol. 14, no. 2, pp. 291–296, 2021.

Conference Publications

[5] J. Kirsch, A. Noll, M. Strese, Q. Liu, and E. Steinbach, "A low-cost acquisition, display, and evaluation setup for tactile codec development", in *2018 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, IEEE, 2018, pp. 1–6.

[6] A. Noll, B. Gülecyüz, A. Hofmann, and E. Steinbach, "A rate-scalable perceptual wavelet-based vibrotactile codec", in *2020 IEEE Haptics Symposium (HAPTICS)*, IEEE, 2020, pp. 854–859.

[7] A. Noll, L. Nockenberg, B. Gülecyüz, and E. Steinbach, "Vc-pwq: Vibrotactile signal compression based on perceptual wavelet quantization", in *2021 IEEE World Haptics Conference (WHC)*, IEEE, 2021, pp. 427–432.

[8] A. Noll, M. Hofbauer, E. Muschter, S.-C. Li, and E. Steinbach, "Automated quality assessment for compressed vibrotactile signals using multi-method assessment fusion", in *2022 IEEE Haptics Symposium (HAPTICS)*, IEEE, 2022.

Book Chapters

[9] E. Steinbach, S.-C. Li, B. Güleçyüz, *et al.*, "Chapter 5 - haptic codecs for the tactile internet", in *Tactile Internet*, F. H. Fitzek, S.-C. Li, S. Speidel, T. Strufe, M. Simsek, and M. Reisslein, Eds., Academic Press, 2021, pp. 103–129, ISBN: 978-0-12-821343-8. DOI: https://doi.org/10.1016/B978-0-12-821343-8.00016-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128213438000162.

General Publications

[10] E. Steinbach, M. Strese, M. Eid, *et al.*, "Haptic codecs for the tactile internet", *Proceedings of the IEEE*, vol. 107, no. 2, pp. 447–470, 2018.

[11] M. Strese, L. Brudermueller, J. Kirsch, and E. Steinbach, "Haptic material analysis and classification inspired by human exploratory procedures", *IEEE Transactions on Haptics*, vol. 13, no. 2, pp. 404–424, 2019.

[12] S. Okamoto, H. Nagano, and Y. Yamada, "Psychophysical dimensions of tactile perception of textures", *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 81–93, 2012.

[13] M. Condoluci, T. Mahmoodi, E. Steinbach, and M. Dohler, "Soft resource reservation for low-delayed teleoperation over mobile networks", *IEEE Access*, vol. 5, pp. 10 445–10 455, 2017.

[14] M. Dohler, T. Mahmoodi, M. A. Lema, *et al.*, "Internet of skills, where robotics meets ai, 5g and the tactile internet", in *2017 European Conference on Networks and Communications (EuCNC)*, IEEE, 2017, pp. 1–5.

[15] K. J. Kuchenbecker, J. Romano, and W. McMahan, "Haptography: Capturing and recreating the rich feel of real surfaces", in *Robotics Research*, Springer, 2011, pp. 245–260.

[16] S. Choi and K. J. Kuchenbecker, "Vibrotactile display: Perception, technology, and applications", *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2093–2104, 2013.

[17] J. M. Romano, T. Yoshioka, and K. J. Kuchenbecker, "Automatic filter design for synthesis of haptic textures from recorded acceleration data", in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 1815–1821.

[18] M. Strese, Y. Boeck, and E. Steinbach, "Content-based surface material retrieval", in *2017 IEEE World Haptics Conference (WHC)*, IEEE, 2017, pp. 352–357.

[19] L. R. Manfredi, H. P. Saal, K. J. Brown, *et al.*, "Natural scenes in tactile texture", *Journal of neurophysiology*, vol. 111, no. 9, pp. 1792–1802, 2014.

[20] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002, vol. 642.

[21] L. R. Manfredi, A. T. Baker, D. O. Elias, *et al.*, "The effect of surface wave propagation on neural responses to vibration in primate glabrous skin", *PloS one*, vol. 7, no. 2, e31203, 2012.

[22] R. V. Grigorii, M. A. Peshkin, and J. E. Colgate, "High-bandwidth tribometry as a means of recording natural textures", in *2017 IEEE World Haptics Conference (WHC)*, IEEE, 2017, pp. 629–634.

[23] M Royer, J. Holmen, M. Wurm, O. Aadland, and M Glenn, "Zno on si integrated acoustic sensor", *Sensors and Actuators*, vol. 4, pp. 357–362, 1983.

[24] Y. Shao, H. Hu, and Y. Visell, "A wearable tactile sensor array for large area remote vibration sensing in the hand", *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6612–6623, 2020.

[25] M. Wiertlewski, J. Lozada, and V. Hayward, "The spatial spectrum of tangential skin displacement can encode tactual texture", *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 461–472, 2011.

[26] Y. Tanaka, T. Yoshida, and A. Sano, "Practical utility of a wearable skin vibration sensor using a pvdf film", in *2017 IEEE World Haptics Conference (WHC)*, IEEE, 2017, pp. 623–628.

[27] A. B. Dhiab and C. Hudin, "Confinement of vibrotactile stimuli in narrow plates", in *2019 IEEE World Haptics Conference (WHC)*, IEEE, 2019, pp. 431–436.

[28] M. Morioka and M. J. Griffin, "Thresholds for the perception of hand-transmitted vibration: Dependence on contact area and contact location", *Somatosensory & motor research*, vol. 22, no. 4, pp. 281–297, 2005.

[29] K. O. Johnson, "The roles and functions of cutaneous mechanoreceptors", *Current opinion in neurobiology*, vol. 11, no. 4, pp. 455–461, 2001.

[30] P. Haggard and L. de Boer, "Oral somatosensory awareness", *Neuroscience & Biobehavioral Reviews*, vol. 47, pp. 469–484, 2014.

[31] P. Haggard, M. Taylor-Clarke, and S. Kennett, "Tactile perception, cortical representation and the bodily self", *Current Biology*, vol. 13, no. 5, R170–R173, 2003.

[32] R. Feldtkeller, E. Zwicker, and E. Zwicker, "Das ohr als nachrichtenempfänger: Monographien der elektrischen nachrichtentechnik", 1956.

[33] R. Chaudhari, C. Schuwerk, M. Danaei, and E. Steinbach, "Perceptual and bitrate-scalable coding of haptic surface texture signals", *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 462–473, 2014.

[34] D. Pan, "A tutorial on mpeg/audio compression", *IEEE Multimedia*, vol. 2, no. 2, pp. 60–74, 1995.

[35] S. J. Bolanowski Jr, G. A. Gescheider, R. T. Verrillo, and C. M. Checkosky, "Four channels mediate the mechanical aspects of touch", *The Journal of the Acoustical society of America*, vol. 84, no. 5, pp. 1680–1694, 1988.

[36] R. T. Verrillo, G. A. Gescheider, B. G. Calman, and C. L. Van Doren, "Vibrotactile masking: Effects of one- and two-site stimulation", *Perception & Psychophysics*, vol. 33, no. 4, pp. 379–387, 1983.

[37] R. T. Verrillo, "Effect of contactor area on the vibrotactile threshold", *The Journal of the Acoustical Society of America*, vol. 35, no. 12, pp. 1962–1966, 1963.

[38] T. Miwa, "Evaluation methods for vibration effect part 3. measurements of threshold and equal sensation contours on hand for vertical and horizontal sinusoidal vibrations", *Industrial health*, vol. 5, no. 3-4, pp. 213–220, 1967.

[39] D. Reynolds, K. Standlee, and E. Angevine, "Hand-arm vibration, part iii: Subjective response characteristics of individuals to hand-induced vibration", *Journal of sound and vibration*, vol. 51, no. 2, pp. 267–282, 1977.

[40] N. Harada and M. J. Griffin, "Factors influencing vibration sense thresholds used to assess occupational exposures to hand transmitted vibration.", *Occupational and Environmental Medicine*, vol. 48, no. 3, pp. 185–192, 1991.

[41] A. Brisben, S. Hsiao, and K. Johnson, "Detection of vibration transmitted through an object grasped in the hand", *Journal of neurophysiology*, vol. 81, no. 4, pp. 1548–1558, 1999.

[42] G. A. Gescheider, S. J. Bolanowski, J. V. Pope, and R. T. Verrillo, "A four-channel analysis of the tactile sensitivity of the fingertip: Frequency selectivity, spatial summation, and temporal summation", *Somatosensory & motor research*, vol. 19, no. 2, pp. 114–124, 2002.

[43] R. Hassen, B. Gülecyüz, and E. G. Steinbach, "Pvc-slp: Perceptual vibrotactile-signal compression based-on sparse linear prediction", *IEEE Transactions on Multimedia*, 2020.

[44] J. C. Craig and P. M. Evans, "Vibrotactile masking and the persistence of tactual features", *Perception & psychophysics*, vol. 42, no. 4, pp. 309–317, 1987.

[45] *Apple music lossless support document*, https://support.apple.com/en-us/HT212183, Accessed: 2022-03-17.

[46] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE Transactions on circuits and systems for video technology*, vol. 6, no. 3, pp. 243–250, 1996.

[47] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[48] S. Okamoto and Y. Yamada, "Perceptual properties of vibrotactile material texture: Effects of amplitude changes and stimuli beneath detection thresholds", in *2010 IEEE/SICE International Symposium on System Integration*, IEEE, 2010, pp. 384–389.

[49] S. Okamoto and Y. Yamada, "Lossy data compression of vibrotactile material-like textures", *IEEE transactions on haptics*, vol. 6, no. 1, pp. 69–80, 2012.

[50] R. Chaudhari, B. Çizmeci, K. J. Kuchenbecker, S. Choi, and E. Steinbach, "Low bitrate source-filter model based compression of vibrotactile texture signals in haptic teleoperation", in *Proceedings of the 20th ACM international conference on Multimedia*, ACM, 2012, pp. 409–418.

[51] J. Breebaart, S. Disch, C. Faller, *et al.*, "Mpeg spatial audio coding/mpeg surround: Overview and current status", in *Audio Engineering Society Convention 119*, Audio Engineering Society, 2005.

[52] W.-S. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding", in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2012, pp. 813–816.

[53] G. Fracastoro, D. Thanou, and P. Frossard, "Graph transform learning for image compression", in *2016 Picture Coding Symposium (PCS)*, IEEE, 2016, pp. 1–5.

[54] C.-P. Chen, C.-S. Chen, K.-L. Chung, H.-I. Lu, and G. Y. Tang, "Image set compression through minimal-cost prediction structure", in *2004 International Conference on Image Processing, 2004. ICIP'04.*, IEEE, vol. 2, 2004, pp. 1289–1292.

[55] B Series, "Method for the subjective assessment of intermediate quality level of audio systems", *International Telecommunication Union Radiocommunication Assembly*, 2014.

[56] R. Schubert, S. Haufe, F. Blankenburg, A. Villringer, and G. Curio, "Now you'll feel it, now you won't: Eeg rhythms predict the effectiveness of perceptual masking", *Journal of cognitive neuroscience*, vol. 21, no. 12, pp. 2407–2419, 2009.

[57] V. A. Shah, M. Casadio, R. A. Scheidt, and L. A. Mrotek, "Spatial and temporal influences on discrimination of vibrotactile stimuli on the arm", *Experimental brain research*, vol. 237, no. 8, pp. 2075–2086, 2019.

[58] N. Weisz, A. Wühle, G. Monittola, *et al.*, "Prestimulus oscillatory power and connectivity patterns predispose conscious somatosensory perception", *Proceedings of the National Academy of Sciences*, vol. 111, no. 4, E417–E425, 2014.

[59] F. van Ede, F. de Lange, O. Jensen, and E. Maris, "Orienting attention to an upcoming tactile event involves a spatially and temporally specific modulation of sensorimotor alpha-and beta-band oscillations", *Journal of Neuroscience*, vol. 31, no. 6, pp. 2016–2024, 2011.

[60] E. M. Mc Govern, J. S. Butler, I. Beiser, *et al.*, "A comparison of stimulus presentation methods in temporal discrimination testing", *Physiological Measurement*, vol. 38, no. 2, N57, 2017.

[61] R. J. Sinclair and H. Burton, "Discrimination of vibrotactile frequencies in a delayed pair comparison task", *Perception & psychophysics*, vol. 58, no. 5, pp. 680–692, 1996.

[62] R. T. Verrillo, "Effects of aging on the suprathreshold responses to vibration", *Perception & Psychophysics*, vol. 32, no. 1, pp. 61–68, 1982.

[63] S. Ludwig, J. Herding, and F. Blankenburg, "Oscillatory eeg signatures of postponed somatosensory decisions", *Human brain mapping*, vol. 39, no. 9, pp. 3611–3624, 2018.

[64] N. Schinkel-Bielefeld, N. Lotze, and F. Nagel, "Audio quality evaluation by experienced and inexperienced listeners", in *Proceedings of Meetings on Acoustics ICA2013*, Acoustical Society of America, vol. 19, 2013, p. 060 016.

[65] P. B. Baltes and U. Lindenberger, "Emergence of a powerful connection between sensory and cognitive functions across the adult life span: A new window to the study of cognitive aging?", *Psychology and aging*, vol. 12, no. 1, p. 12, 1997.

[66] S.-C. Li, M. Jordanova, and U. Lindenberger, "From good senses to good sense: A link between tactile information processing and intelligence", *Intelligence*, vol. 26, no. 2, pp. 99–122, 1998.

[67] R. Hassen and E. Steinbach, "Subjective evaluation of the spectral temporal similarity (st-sim) measure for vibrotactile quality assessment", *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 25–31, 2020. DOI: 10.1109/TOH.2019.2962446.

[68] L. A. Jones and N. B. Sarter, "Tactile displays: Guidance for their design and application", *Human factors*, vol. 50, no. 1, pp. 90–111, 2008.

[69] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 576–584.

[70] K. Cui and E. Steinbach, "Decoder side image quality enhancement exploiting inter-channel correlation in a 3-stage cnn: Submission to clic 2018", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2571–2574.

[71] A. Biswas and D. Jia, "Audio codec enhancement with generative adversarial networks", in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 356–360.

[72] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[73] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[74] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm", *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[75] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches", *arXiv preprint arXiv:1409.1259*, 2014.

[76] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", *arXiv preprint arXiv:1412.3555*, 2014.

[77] G. Chrupała, A. Kádár, and A. Alishahi, "Learning language through pictures", *arXiv preprint arXiv:1506.03694*, 2015.

[78] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures", *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[79] R. Taniguchi, K. Hoshiba, K. Itoyama, K. Nishida, and K. Nakadai, "Signal restoration based on bi-directional lstm with spectral filtering for robot audition", in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 2018, pp. 955–960.

[80] Y. Gu, Z.-H. Ling, and L.-R. Dai, "Speech bandwidth extension using bottleneck features and deep recurrent neural networks.", in *Interspeech*, 2016, pp. 297–301.

[81] J. Deng, B. Schuller, F. Eyben, *et al.*, "Exploiting time-frequency patterns with lstm-rnns for low-bitrate audio restoration", *Neural Computing and Applications*, vol. 32, no. 4, pp. 1095–1107, 2020.

[82] S. Nah, S. Son, and K. M. Lee, "Recurrent neural networks with intra-frame iterations for video deblurring", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8102–8111.

[83] Z. Zhong, Y. Gao, Y. Zheng, and B. Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring", in *European Conference on Computer Vision*, Springer, 2020, pp. 191–207.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[85] Z. Zhao, H. Liu, and T. Fingscheidt, "Convolutional neural networks to enhance coded speech", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 663–678, 2018.

[86] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising", *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[87] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4799–4807.

[88] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480–2495, 2020.

[89] W. McMahan and K. J. Kuchenbecker, "Dynamic modeling and control of voice-coil actuators for high-fidelity display of haptic vibrations", in *2014 IEEE Haptics Symposium (HAPTICS)*, IEEE, 2014, pp. 115–122.

[90] H. Culbertson, J. M. Walker, and A. M. Okamura, "Modeling and design of asymmetric vibrations to induce ungrounded pulling sensation through asymmetric skin displacement", in *2016 IEEE Haptics Symposium (HAPTICS)*, IEEE, 2016, pp. 27–33.

[91] T. Tanabe, H. Yano, and H. Iwata, "Evaluation of the perceptual characteristics of a force induced by asymmetric vibrations", *IEEE Transactions on Haptics*, vol. 11, no. 2, pp. 220–231, 2017.

[92] H.-V. Quang and M. Harders, "Improved control methods for vibrotactile rendering", in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, Springer, 2016, pp. 217–228.

[93] A. T. Bukkapatnam, P. Depalle, and M. M. Wanderley, "Defining a vibrotactile toolkit for digital musical instruments: Characterizing voice coil actuators, effects of loading, and equalization of the frequency response", *Journal on Multimodal User Interfaces*, vol. 14, no. 3, pp. 285–301, 2020.

[94] S. Haykin, *Adaptive filter theory.* Pearson Education, India, 2005.

[95] P. S. R. Diniz, *Adaptive filtering.* Springer, New York, NY, 1997.

[96] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive volterra filters", in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 2, 1999, pp. 877–880.

[97] J. Cunningham, Y. Zheng, T. Subramanian, and M. Almekkawy, "Regularization methods for solving third-order volterra filter with improved convergence speed: In-vivo application", in *15th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2018, pp. 1187–1190.

[98] A. H. Sayed, *Adaptive filters.* John Wiley & Sons, Hoboken, NJ, 2011.

[99] H. W. Kang, Y. S. Cho, and D. H. Youn, "On compensating nonlinear distortions of an ofdm system using an efficient adaptive predistorter", *IEEE Transactions on Communications*, vol. 47, no. 4, pp. 522–526, 1999.

[100] A. Zhu and T. J. Brazil, "An adaptive volterra predistorter for the linearization of rf high power amplifiers", in *2002 IEEE MTT-S International Microwave Symposium Digest (Cat. No. 02CH37278)*, IEEE, vol. 1, 2002, pp. 461–464.

[101] P Gilabert, G. Montoro, and E Bertran, "On the wiener and hammerstein models for power amplifier predistortion", in *2005 Asia-Pacific Microwave Conference Proceedings*, IEEE, vol. 2, 2005, 4–pp.

[102] R. Marsalek, P. Jardin, and G. Baudoin, "From post-distortion to pre-distortion for power amplifiers linearization", *IEEE Communications Letters*, vol. 7, no. 7, pp. 308–310, 2003.

[103] L. Ding, G. T. Zhou, D. R. Morgan, *et al.*, "A robust digital baseband predistorter constructed using memory polynomials", *IEEE Transactions on communications*, vol. 52, no. 1, pp. 159–165, 2004.

[104] K.-J. Cho, W.-J. Kim, J.-H. Kim, and S. P. Stapleton, "Linearity optimization of a high power doherty amplifier based on post-distortion compensation", *IEEE Microwave and Wireless Components Letters*, vol. 15, no. 11, pp. 748–750, 2005.

[105] W.-J. Kim, K.-J. Cho, S. P. Stapleton, and J.-H. Kim, "Piecewise pre-equalized linearization of the wireless transmitter with a doherty amplifier", *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 9, pp. 3469–3478, 2006.

[106] P. Gilabert, G Montoro, and A Cesari, "A recursive digital predistorter for linearizing rf power amplifiers with memory effects", in *2006 Asia-Pacific Microwave Conference*, IEEE, 2006, pp. 1040–1043.

[107] P. L. Gilabert Pinal, *Multi look-up table digital predistortion for RF power amplifier linearization*. Universitat Politecnica de Catalunya, 2008.

[108] R Bitmead and B. Anderson, "Performance of adaptive estimation algorithms in dependent random environments", *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 788–794, 1980.

[109] P. L. Feintuch, "An adaptive recursive lms filter", *Proceedings of the IEEE*, vol. 64, no. 11, pp. 1622–1624, 1976.

[110] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Nonlinear spline adaptive filtering.", *Elsevier. Signal Processing.*, vol. 93, no. 4, pp. 772–783, 2013.

[111] W. Liu, J. C. Principe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction.* John Wiley & Sons, Inc., 2011.

[112] F. C. Pinheiro and C. G. Lopes, "Newton-like nonlinear adaptive filters via simple multilinear functionals.", in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, IEEE, Budapest, Hungary, Aug. 2016, pp. 1603–1607.

[113] V. J. Mathews, "Adaptive polynomial filters.", *IEEE Signal Processing Magazine*, vol. 8, no. 3, pp. 10–26, 1991.

[114] A. Guérin, G. Faucon, and R. Le Bouquin-Jeannès, "Nonlinear acoustic echo cancellation based on volterra filters", *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 672–683, 2003.

[115] E Roy, R. W. Stewart, and T. S. Durrani, "Theory and applications of adaptive second order iir volterra filters", in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, IEEE, vol. 3, 1996, pp. 1597–1600.

[116] T. Koh and E Powers, "Second-order volterra filtering and its application to nonlinear system identification", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 6, pp. 1445–1455, 1985.

[117] V. J. Mathews and G. Sicuranza, *Polynomial signal processing*. John Wiley & Sons, Inc., 2000.

[118] E. L. O. Batista, O. J. Tobias, and R. Seara, "A sparse-interpolated scheme for implementing adaptive Volterra filters.", *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2022–2035, 2010.

[119] M. Zeller, L. A. Azpicueta-Ruiz, J. Arenas-Garcia, and W. Kellermann, "Adaptive Volterra filters with evolutionary quadratic kernels using a combination scheme for memory control.", *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1449–1464, 2011.

[120] E. L. O. Batista and R. Seara, "On the performance of adaptive pruned Volterra filters.", *Elsevier. Signal Processing*, vol. 93, no. 7, pp. 1909–1920, 2013.

[121] E. L. O. Batista, O. J. Tobias, and R. Seara, "Stochastic model of the LMS volterra filter.", in *Proceedings of the 15th European Signal Processing Conference*, IEEE, Poznań, Poland, Sep. 2007, pp. 1721–1725.

[122] M. Lutman, "What is the risk of noise-induced hearing loss at 80, 85, 90 db (a) and above?", *Occupational medicine*, vol. 50, no. 4, pp. 274–275, 2000.

[123] K. Liu, E. Belyaev, and J. Guo, "Vlsi architecture of arithmetic coder used in spiht", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, Apr. 2012. DOI: 10.1109/TVLSI.2011.2109068.

[124] A Signoroni, M Arrigoni, F Lazzaroni, and R Leonardi, "Improving spiht-based compression of volumetric medical data", in *Picture Coding Symposium*, 2001, pp. 187–190.

[125] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[126] N. Deshpande, E. J. Metter, S. Ling, R. Conwit, and L. Ferrucci, "Physiological correlates of age-related decline in vibrotactile sensitivity", *Neurobiology of Aging*, vol. 29, no. 5, pp. 765–773, 2008.

[127] Y.-H. Lin, S.-C. Hsieh, C.-C. Chao, Y.-C. Chang, and S.-T. Hsieh, "Influence of aging on thermal and vibratory thresholds of quantitative sensory testing", *Journal of the Peripheral Nervous System*, vol. 10, no. 3, pp. 269–281, 2005.

[128] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median", *Journal of experimental social psychology*, vol. 49, no. 4, pp. 764–766, 2013.

[129] *C-2 tactor product data sheet atac tactor*, Electronic Article, 2016. [Online]. Available: https://www.eaiinfo.com.

[130] N. T. Blog, *Toward A Practical Perceptual Video Quality Metric*, en, Jun. 2016. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652 (visited on 08/31/2021).

[131] N. T. Blog, *VMAF: The Journey Continues*, en, Oct. 2018. [Online]. Available: https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12 (visited on 08/31/2021).

[132] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks", *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017. DOI: 10.1109/TCI.2016.2644865.

[133] J. Li, Y. Wang, H. Xie, and K.-K. Ma, "Learning a single model with a wide range of quality factors for jpeg image artifacts removal", *IEEE Transactions on Image Processing*, vol. 29, pp. 8842–8854, 2020.

[134] M.-Z. Wang, S. Wan, H. Gong, and M.-Y. Ma, "Attention-based dual-scale cnn in-loop filter for versatile video coding", *IEEE Access*, vol. 7, pp. 145 214–145 226, 2019.

[135] B. Zheng, Y. Chen, X. Tian, F. Zhou, and X. Liu, "Implicit dual-domain convolutional network for robust color image compression artifact reduction", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3982–3994, 2019.

[136] Q. Xing, M. Xu, T. Li, and Z. Guan, "Early exit or not: Resource-efficient blind quality enhancement for compressed images", in *European Conference on Computer Vision*, Springer, 2020, pp. 275–292.

[137] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[138] R. H. Abiyev, O. Kaynak, T. Alshanableh, and F. Mamedov, "A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization", *Applied Soft Computing*, vol. 11, no. 1, pp. 1396–1406, 2011.

[139] Y. Takahashi, "Adaptive predictive control of nonlinear time-varying systems using neural networks", in *IEEE International Conference on Neural Networks*, IEEE, 1993, pp. 1464–1468.

[140] *Analog Devices ADXL335 Accelerometer*, https://www.analog.com/en/products/adxl335.html, Accessed: 2019-08-28.

[141] S. G. Johnson, "Notes on fft-based differentiation", *MIT Applied Mathematics, Tech. Rep.*, 2011.

[142] R. E. W. Rafael C. Gonzalez, *Digital Image Processing, 4th Edition*, 4th ed. Pearson, 2018, ISBN: 978-0-1333-5672-4.

[143] S. G. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$", *Transactions of the American Mathematical Society*, vol. 315, no. 1, pp. 69–87, 1989, ISSN: 00029947. [Online]. Available: http://www.jstor.org/stable/2001373.

[144] O. Christensen, *Functions, Spaces, and Expansions: Mathematical Tools in Physics and Engineering*, 1st ed. Birkhäuser Verlag, 2010, ISBN: 978-0-8176-4979-1.

[145] D. F. Walnut, *An Introduction to Wavelet Analysis*, 1st ed. Springer Science+Business Media, LLC, 2014, ISBN: 978-1-4612-6567-2.

[146] S. G. Mallat, *A wavelet tour of signal processing*. San Diego, Calif.: Academic Press, 2006, ISBN: 9780124666061.

[147] I. Daubechies, "Orthonormal bases of compactly supported wavelets", *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, Oct. 1988, ISSN: 1097-0312.

[148] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992, ISBN: 0-89871-274-2.

[149] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets", *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485–560, Jun. 1992, ISSN: 1097-0312.

[150] M. Lightstone, E. Majani, and S. K. Mitra, "Low bit-rate design considerations for wavelet-based image coding", *Multidimensional Systems and Signal Processing*, vol. 8, no. 1, pp. 111–128, Jan. 1997, ISSN: 1573-0824.

[151] D Sundararajan, *Discrete Wavelet Transform: A Signal Processing Approach*. Chennai, India: John Wiley and Sons Singapore Pte. Ltd., 2015, pp. 1–323, ISBN: 9781119046066.

[152] B. Usevitch, "A tutorial on modern lossy wavelet image compression: Foundations of jpeg 2000", *Signal Processing Magazine, IEEE*, vol. 18, pp. 22 –35, Oct. 2001.

[153] C. J. I. Services, *Wsq gray-scale fingerprint image compression specification*, https://www.fbibiospecs.cjis.gov/Document/Get?fileName=WSQ_Gray-scale_Specification_Version_3_1_Final.pdf, Dec. 1997.