Abstract

CONTENTS

1   INTRODUCTION

The area of artificial intelligence and, more specifically, neural networks, is inter-disciplinar since its conception, being born from biological inspirations. Implementation wise, connections with fields like programming, computing and mathematics imposed themselves since the begining.

Attention, however, seems to revolver around the use of advancements in related disciplines to the betterment of neural networks theory in general, be it in the form of convolutional networks, better hyperparameter search, parallel implementations, etc. Generally, neural networks are applyied to some problem, with little concern about transforming the problem in order to make it optimally suited to work in a neural networks approach, outside of necessary adaptations, such as normalization.

This lack is in part justified by the recent progress exhibted in the field, where impressive results are abundant; a word of caution is needed here, however, as this is a field that experimented its fair share of hype cycles, followed by winters. neural networks faced many winters since its introduction. The pattern seems to repeat itself: hardware improvements enable attractive new results, the interest rises, and some technical advancements are made. Although, in its current renaissance, the field is still blooming, there are some that are already pointing to a decrease in the research output.

We feel that this tool still offers a great potential in its current state of development; a potential thtat will only be fully explored once it is realised that workflows and auxiliary theories are needed, to be used in conjunctions with neural networks.

In this sense, we introduce a novel representation for pseudo periodic time series, of which sound representation are our primary interest. Although much of the theory is drawn from conventional areas related to digital signal processing, image processing, etc, the representation was formulated from the beginning having in mind the strengths and limitations of neural networks in general.

In this context, the primary objective of this work is to illustrate how such integration can be done, by developing a new sound representation designed specifically to be used by neural networks in sound synthesis tasks.

Since the beginning of computing, even when mainframes were the only available medium, restricted to organizations with considerable budgets, people found ways and motivation to overcome the technical limitations and develop artistic applications, like video games, in their (and the machine's) spare time.

On a parallel note, also relevant to this thesis, those first machines, like the Electronic Numeric Integrator and Calculator (ENIAC) in 1946, brought with then the promise of the artificial intelligence as their ultimate goal (Donovan, 2010).

Music, on the other hand, is another staple of humanity's creativity. Is only natural that, as soon as this technology became reasonably stable, musical applications started to emerge.

Ambitious as it may seem, it is the goal of this monograph to motivate a change of approach in domains that, despite their modest intersections, share more attributes than is apparent at a superficial glance: Both machine learning and digital musical acoustics are areas that, despite their current relevance, are fresh out of their respective infancies, and a scrutiny of both promptly reveals the difficulties that arise from this state of affairs. A lack of their own terminology, with frequent borrows from other, more established areas, can be readily cited, as the pronounced prominence of few individual contributions.

Those shortcomings have, however, a bright side to them, in that they still expose those areas to reinterpretation, rendering them somewhat open to the influence of outside ideas.

Digital musical acoustics borrows heavily from the digital signal processing terminology, the last having its roots in the analogic world. Many algorithms are conceived in terms of filters, circuits and other similar legacy constructs, a fact that burdens the conceptual, theoretical framework of the area. Besides the introduction of a noveau envelope extraction method, we find that his formulation in terms of (differential) geometric concepts is, perhaps, as great a contribution as the algorithm per se.

2 BIBLIOGRAPHIC APPROACH

With the advance of technology, some classic problems of bibliographic research were mitigated, while new ones were introduced. Fe w would argue that the ability to carry thousands of books in a handheld device wasn't an improvement, with accessibility improvements such as the possibility of hearing text, instead of reading it, distribution, etc revolutionizing the consumption, creation of knoledge. This revolution was not without its drawbacks, however.

A modern problem encountered by the author during the development of this work is the fragmentation of data. In the bygone days of physical research, a paper existed, in the researcher files, in one physical location. All the metadata available, perhaps even notes, where in a single logic object, such as a book or a photocopy.

When access to the original document was limited, such as a piece of literature residing in a library, nevertheless the approach could be the same, the logical object now being a card sumarizing the metadata and relevant content. This approach is, of course, translatable to the digital world, where one could maintain a folder where digital files, such as pdfs, reside, with notes and the relevant metadata embedded in the file, its filename of even in the metadata section.

This approach quickly becomes problematic, however, for scientific writting, specially in the context of bigger productions such as books and, as is the present case, a thesis, for a number of reasons. Digital documents, especially the most common ones, are designed to be as generic as possible, serving variated purposes, as are their metadata systems, if present.

The management of metadata is best acomplished with the used of dedicated software, of which zotero and mendeley are some examples, Being specifically designed to this task, facilities such as metadata retrieval and validation are often available. If on intends to make use of a typesetting system, such as Latex, avoiding bibliographic management software is virtually impossible.

Another problem in bibliography management in the digital era is to maintain syncronized the original files, most commonly available in pdf or epub format, and generally trated as general files by most file brownsing interfaces, and its metadata, that must exhib a more sophisticated and uniform structure. pdf, despite its ubiquity, is a tricky format, being somewhat opaque. Besides, multiple ways exist to encode a similar document. Fortunately, we don't need to extract correct or even meninful text from it,

the only thing we need is a way tu uniquely identify each file, impevious to file changes such as the addition of comments, changes in location in the filesystem or renaming.

bibtex uses keys. as tempting as it might sound to use those keys as identifiers fro the files, that approach presents problems.

we adress the meta problem of organizing the bibliography introducing a piece of bibliographic software based on an novel paradigm.

writing a phd thesis is a peculiar endeavour in the academic world, in the sense that it is a requirement that must be fullfilled only once: while a thriving academic carreer pressuposes a constant output of papers, and possibly books, or book chapters, the closest contact an academic has with the format of a thesis is probably during the development of his masters dissertation; that one endeavour being, all in all, substantially different.

It comes as no surprise, then, that bibliographic software, being in general geared towards the task of aiding in the productions of academic articles, can be subpar for a thesis, where the hability to handle a mutch larger volume of bibliographic material is needed.

3 CONVENTIONS AND DEFINITIONS

The field explored in this work, besides being extensive and fragmented, is largely interdisciplinar, with terminology strong influenced by widespread usage in the community rather than academic tradition. To maintain coherence thougout the text, it is convenient to stablish some definitions, as some of the terms here applyed may deviate from their usage in particular fields.

Effort was made to comply with the most commom nomenclature when terms are encountered frequently in literature.

This work will make extensive use of the term sample, that, even in the context of digital signal processing, can be ambiguous. It can refer to a, usually short, sound file, as when refering to sample based instruments, or it can sometimes mean a unique point of a discrete sound wave, as when used in sample rate. We will adopt the first meaning, and prefer frame for the second; altough the usage is more comum when referring to videos, it is used in terms such as frames per second.

4 THEORY

In this sections we present some topics that will serve as basis for later chapters, from a purely theoretical point of view. This is specially important in this work, since many different, seemingly disconnected areas are brought togheter.

4.1 WAVES

It can be insightful to analyse the behaviour of the crytical points and the zeroes of an arbitrary, periodic wave over time. We will assume, for the sake of simplicity, that each cycle of said wave has a unique maximum and minimum, and crosses the horizontal axis one single time. Those assumption dont compromise the generality of the analysis, as in the case of multiple maxima, minima and/or zeroes, one of thosee can be choosen randomly to represent the global one, as long as consistency between cycles is maintained.

We are going to also assume that a wave with a zero phase has an maximum instantaneous amplitude at time zero, as is the case of the cosine wave, for example. Altough that assumption somewhat departs from the physical intuition, it is useful in making our description consistent with the theory behind the Fourier Transform.

For a continuous wave, if we were to plot the time at which any of those values ocurred as a function of their ordinality, the result would be parallel straight lines, their slope somehow related to the frequency of the underlying wave. We will illustrated the point in the case of discrete waves, as this case will be of more immediate use in this work.

4.1.1 Continuous and discrete waves

First, it is necessary to establish a correspondence between a continuous and a discrete wave, as the latter will be the object of most of this work. To do that we will define a simple continuous sinusoidal wave of the form $S(t) = a\cos\left(\phi + 2\pi\mathcal{F}t\right)$, where $a \in \mathbb{R}^+$ is the amplitude; $\phi \in \mathbb{R}^+, 0 \leq \phi < 2\pi$ is the phase; $\mathcal{F} \in \mathbb{R}^+$ is the absolute frequency of the wave, in Hz and $t \in \mathbb{R}$ is time, in seconds.

The cosine function is used for a twofold reason: its use helps to underline the periodic nature of the wave in consideration and is the parameter of the phase when the FT is used. It is important to keep in mind, however, that this discussion applies any arbitrary function of the form $f(x) = f(x + P)\forall x \in D(f)$, where D is the domain of the function f.

If we were to sample $S(t)$ at regular intervals, that is, to measure the value of the

function at a constant rate of h frames per second (FPS), neglecting errors of measurement, we would have a discrete representation of the continuous wave of the form $W[i] = a\cos\left(\phi + 2\pi f \frac{i}{n}\right)$ where, alongside the quantities introduced before, we would have $i \in \mathbb{N}$ as the frame number; $n \in \mathbb{N}$ as the number of discrete samples and $f \in \mathbb{R}^+$ being the relative frequency of our discrete wave in cycles per n samples.

The relative, local frequency f of the discrete representation is related to the absolute frequency $\mathcal{F}$ via the equation $f = \frac{n}{\text{fps}}\mathcal{F}$ while the absolute time, in the continuous case can be obtained from the frame number i of the discrete signal via $i = t$ fps.

4.1.2 Points of interest of periodic waves

4.2 LEAST MEAN SQUARES

The linear Least Mean Squares (LMS), also known as linear regression (Stephen Boyd, 2018) is a data fitting technique.

We present here the theory for real numbers, noting that it can be easily extended for complex numbers (Lawson and Hanson, 1995), should the necessity arise. That, however, will not be the case in this work.

Given two vectors X and Y, such that:

$$X = [x_0, x_1, \cdots, x_{m-1}]^T, \ x_i \ \in \ \mathbb{R} \ \forall \ i, \ 0 \leq i < m \in \mathbb{N}$$

$$Y = [y_0, y_1, \cdots, y_{m-1}]^T, \ y_i \ \in \ \mathbb{R} \ \forall \ i, \ 0 \leq i < m \in \mathbb{N}$$

We ideally want to find the vector of coefficients $A = [a_0, a_1, \cdots, a_k]^T, \ a_j \ \in \ \mathbb{R} \ \forall \ j, \ 0 \leq j \leq k \in \mathbb{N}$ that satisfies the equality $x_i^0 a_0 + x_i^1 a_1 + \cdots + x_i^{k-1} a_{k-1} + x_i^k a_k = y_i \quad \forall \quad i \in \{0, \cdots, m-1\}$.

Put in that way, what we have is a system of linear equations that can be rewritten in matrix form as seen in equation 1.

$$
\overset{Q_{m\times(k+1)}}{\begin{bmatrix} 1 & x_0 & \cdots & x_0^k \\ 1 & x_1 & \cdots & x_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m-1} & \cdots & x_{m-1} \end{bmatrix}}
\overset{A_{(k+1)\times 1}}{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}}
=
\overset{Y_{m\times 1}}{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \end{bmatrix}}
\tag{1}
$$

More compactly, we have that $QA = Y$. This system will generally be over-determined, only having an exact solution in the case where Y is a linear combination of

the columns in Q (Stephen Boyd, 2018).

Since an exact solution is often beyond reach, we can at least choose A in a way that makes QA and Y close. A way to do that is to minimize the absolute errors (or their square) between those two quantities; $\min_{A} ||QA - Y||_2^2$

Using calculus, for example, on can arrive at the equation $\hat{A} = (Q^T Q)^{-1} Q^T Y$, where $\hat{A}$ is the vector of coefficients that minimizes the squared errors. A detailed derivation is available at Stephen Boyd, 2018.

4.3 CONSTRAINED LEAST MEAN SQUARES

In the constrained version of the LMS, known as the Constrained Least Mean Squares (CLMS) problem we are still interested in finding the coefficient vector $\hat{A}$ that minimizes $||QA - Y||_2^2$, but respecting a restriction on this vector of the general form of $VA = B$.

We can then define the Lagrangian and set its derivatives to zero to obtain, as described in Selesnick (2013), the closed form solution

$$\hat{A} = \left(Q^T Q\right)^{-1} \left(Q^T Y - V^T \left(V \left(Q^T Q\right)^{-1} V^T\right)^{-1} V \left(Q^T Q\right)^{-1} Q^T Y - B\right)$$

4.4 SAVITZKY–GOLAY FILTER

5 CHARACTERIZATION OF A DIGITAL SIGNAL

5.1 ENVELOPE OF A DIGITAL SIGNAL

Despite being an elusive concept, the temporal amplitude envelope of a signal is essential for its complete characterization, being the primary information-carrying medium in spoken voice and telecommunications, for example. Intuitively, the temporal envelope can be understood as a slowly varying function that multiplies the signal, being responsible for its outer shape. Envelope detection techniques have applications in areas like medicine, sound classification and synthesis, seismology and speech recognition. Nevertheless, a general approach to digital envelope detection of signals with rich spectral content doesn't exist, as most methods involve manual intervention, in the form of filter design, smoothing, and other specific design choices, based on prior knowledge of the signals under investigation. To address this problem, we propose an algorithm that uses intrinsic characteristics of a signal to estimate its envelope, eliminating the necessity of

parameter tuning. The approach here described draws inspiration from geometric concepts to estimate the temporal envelope of an arbitrary signal; specifically, a new measure of discrete curvature is used to obtain the average radius of curvature of a discrete wave, that will serve as a threshold to identify the wave's samples that are part of the envelope. The algorithm compares favourably with classic envelope detection techniques based on smoothing, filtering and the Hilbert Transform, besides being physically plausible. We provide visualizations of the envelope extracted via the algorithm for various real-world signals, with very diverse characteristics, such as voice, spoken and sang, and pitched and non-pitched musical instruments, and discuss some approaches to assess the quality of the obtained envelopes. A Python module implementing the algorithm was made available via the Python Package Index (PyPI); interactive visualizations of envelopes for a diverse range of digital waves, as well as the source code for the Python implementation, are available online at Tarjano (2020). Besides the most direct applications of this work to audio classification and synthesis, we foresee impact in compression techniques and machine learning approaches to audio. We briefly discuss some potential paths in this direction. The discrete curvature definition presented could also be extended to three-dimensional settings, to improve shape detection algorithms based on alpha shapes.

Envelope detection, also known as demodulation, is ubiquitous in both analogue and digital signal processing (Caetano and Rodet, 2011). Nevertheless, the literature in this area is fragmented (Lyons, 2017). Besides, most envelope detection techniques are designed to account for very specific settings, like pure sinusoids with moderate noise content, a limitation that excludes most physical signals, as is the case of recorded sound, for example; that limitation arises in part due to the lack of a strict mathematical definition of a temporal envelope (Meng et al., 2013).

In many contexts, however, the temporal amplitude envelope of a signal plays a prominent role: according to Qi et al. (2017), for example, the envelope is at least as important as the fine structure of a sound wave in the context of the intelligibility of Mandarin tones. This is also the case for the English language (Shannon et al., 1995), where even envelopes modulating mostly noise are still capable of conveying meaning.

The envelope also helps to impart emotion and identity to the human voice (Zhu et al., 2018) and music: Concert halls with envelope preserving characteristics are deemed more pleasant (Lokki et al., 2011), for example.

When dealing with broadband signals, approaches tailored to specific applications are prevalent, such as the one presented by W. Yang, Y. Yang, and M. Yang (2014) for

the distributed monitoring of fibre optic or the one formulated by Assef et al. (2018) in the context of medical ultrasound imaging.

The problem of envelope detection, normally associated with the field of digital signal processing, can be made equivalent to the geometric problem of defining the (outer) shape of a set of points in $\mathbb{R}^2$, provided a method for addressing the difference of units in the horizontal, time related, and vertical, amplitude related, axes of a digital signal is available, rendering the original DSP problem readily approachable via geometric techniques.

One direct way of defining the shape of a set of points in $\mathbb{R}^2$ is using the concept of a convex hull: as defined in Berg et al. (2008), the convex hull of a set of points S in two-dimensional space is the smallest convex subset of the plane that contains all the points in S. Convex hulls are unique, in the sense that a set of points define one and only one convex hull.

Generalizing that definition, H. Edelsbrunner, Kirkpatrick, and Seidel (1983) introduced the concept of alpha shapes, that can be seen as a concave hull; a mathematically well-defined generalization of the convex hull of a finite set of points, closely related to the Delaunay triangulation and Voronoi diagrams of those points.

The intuition behind the alpha shapes algorithm is that only points that can be touched by a circle of radius α coming from an infinite distance towards the set of points of interest are part of the frontier, that is, of the concave hull of this set.

Once a radius α is defined, alpha shapes conserve the uniqueness property, in fact tending to the convex hull when $\alpha \to \infty$. Consider, for example, the artificially generated discrete wave in figure 1 below, where the links between consecutive samples were deemphasized to highlight the "set of points" nature of a signal: The convex hull and a concave hull, in the form of an alpha shape with a particular value of α are illustrated. The successive points can be interpreted as a wave, in which case the vertical axis represents its instantaneous amplitude at each frame, the meaning of the horizontal axis. If we disconsider the lines linking each consecute point it is easy to see the image as a cloud of points in a cartesian plane, however.

The alpha shapes approach is used in areas such as detection of features in images (Varytimidis et al., 2016), reconstruction of surfaces from a cloud of points (B. Wu et al., 2015) and spectroscopy (Xu et al., 2019).

This last work, which involves the estimation and removal of the Blaze function (a kind of envelope) of an echelle spectrograph, being particularly illustrative of the potential

synergy between geometric and DSP approaches.

Other steps in the direction of translating geometric algorithms to the context of envelope detection were made by C. Yang, Tiebe, et al. (2015) via an algorithm based on the construction of a skeleton underlying the digital wave of interest, and also via the direct translation of computer vision methods to the task of envelope detection (C. Yang, Grzegorzek, and Lukasik, 2015).

Following this path, we present a general approach to envelope detection, exploiting the intrinsic characteristics of a generic, spectrally complex wave, in order to completely avoid the need for manual intervention or parameter tuning.

The rest of this paper is divided as follows: We start by defining as specifically as possible what a temporal envelope is, especially in the context of this work and providing a mathematical definition, in terms of the Hilbert transform, for the case of a narrowband signal.

We proceed to illustrate the representation of a discrete wave used in this work, largely based on the concept of dividing the wave into its constituent pulses, and how it can simplify the algorithm, reducing the high dimensionality generally present in digital representations of waves.

Next, we review methods for the estimation of the curvature of a discrete wave, after which we present our own; this method will then be used to estimate the equivalent circle, that will be used to define the envelope of the wave, via the procedure described in sequence.

In the results section, we present some illustrations of the envelopes obtained with our method, comparing both accuracy and running time with the most common algorithms encountered in the literature. We also suggest an alternative use of the algorithm to identify the negative and positive frontiers of a wave, that can be seen as partial envelopes considering only the positive and negative pulses of a wave, respectively.

We follow with suggestions about how the approach here presented can be useful beyond envelope detection, exemplifying with some applications, such as the extractions of the average waveform of a digital wave, and concluding with commentary on potential future developments of the method.

5.2 METHODS

After defining what will be considered an envelope in the context of this work, in this section we explore some simplifications that can be applied to a discrete wave,

defining the representation that will be used in the subsequent part of the work.

We next explore the concept of discrete curvature, putting forth our own method, tailored to the definition of the equivalent circle of a wave; we end this section by showing how this circle can be used to identify the envelope of a discrete wave, in a similar approach to that of the alpha shapes theory, but taking advantage of some unique structure present in discrete waves.

5.2.1   Characterization of a temporal envelope

The interest in the theory of envelope detection arose in the analog domain, with the widespread adoption of radio communications, where the problem is typically restricted to simple sinusoids, often with well-defined frequencies (Turner and Sahani, 2011).

In this context, the most mathematically sound definition of an envelope involves the representation of its underlying wave as an analytic signal, as introduced by Gabor in 1946 (Hahn, 2007). In his work, Gabor (1946) applies the then relatively new mathematical machinery of the quantum mechanics to unify the time and frequency-domain representations of a wave, showing how the Hilbert transform could be applied to a real signal in order to obtain an equivalent complex signal, later known as the analytic signal. This analytic signal has the form $A(t) = S(t) + \mathbb{H}(S(t))$ i (He and Dellwo, 2016) where $S(t)$ is the original real signal. $\mathbb{H}(S(t))$, the Hilbert transform of the original signal, becomes the imaginary part of the analytic signal; the envelope of a signal thus represented can be straightforwardly obtained by the computation of the complex modulus of $\mathbb{H}(S(t))$.

Despite its widespread use, and effectiveness in the context of narrowband signals, envelope detection techniques based on the Hilbert transform don't behave well for broadband signals (Dau, 2012), being even physically paradoxical in some cases (Loughlin and Tacer, 1996).

Figure 2 exemplifies the Hilbert envelope, for a pure sinusoid modulated by a polynomial, illustrating the difference in the shape of the envelope in the presence and absence of Gaussian white noise. It is readily noticeable from the figure that, as soon as noise is introduced in the original signal, it is reflected in the envelope.

In general, the problem of envelope detection, or demodulation, can be interpreted as the task of, given a continuous wave $W(t)$, decomposing it in two components $E(t)$ and $C(t)$, such that $W(t) = E(t)C(t)$ (Turner and Sahani, 2011).

$E(t)$ represents the slow varying part of the wave, also known as the (temporal) envelope, modulator component, or amplitude modulation (AM) of W, while $C(t)$ models
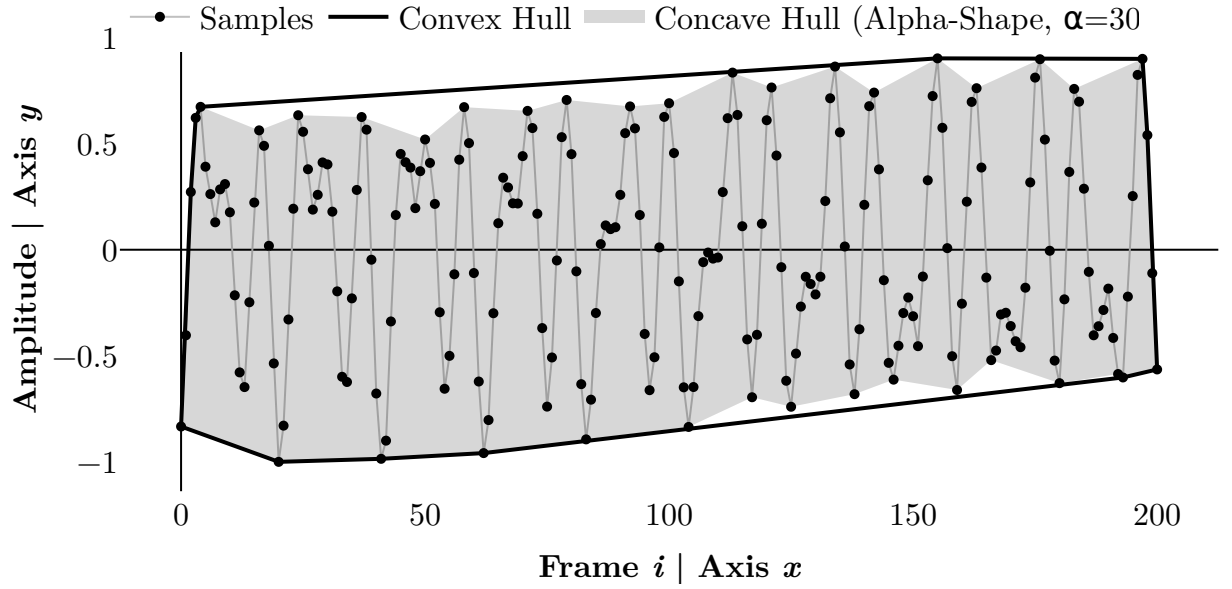
Figure 1: A discrete signal interpreted as a set of points in the Cartesian plane, and the convex and concave hull at $\alpha = 300$ defined by it.
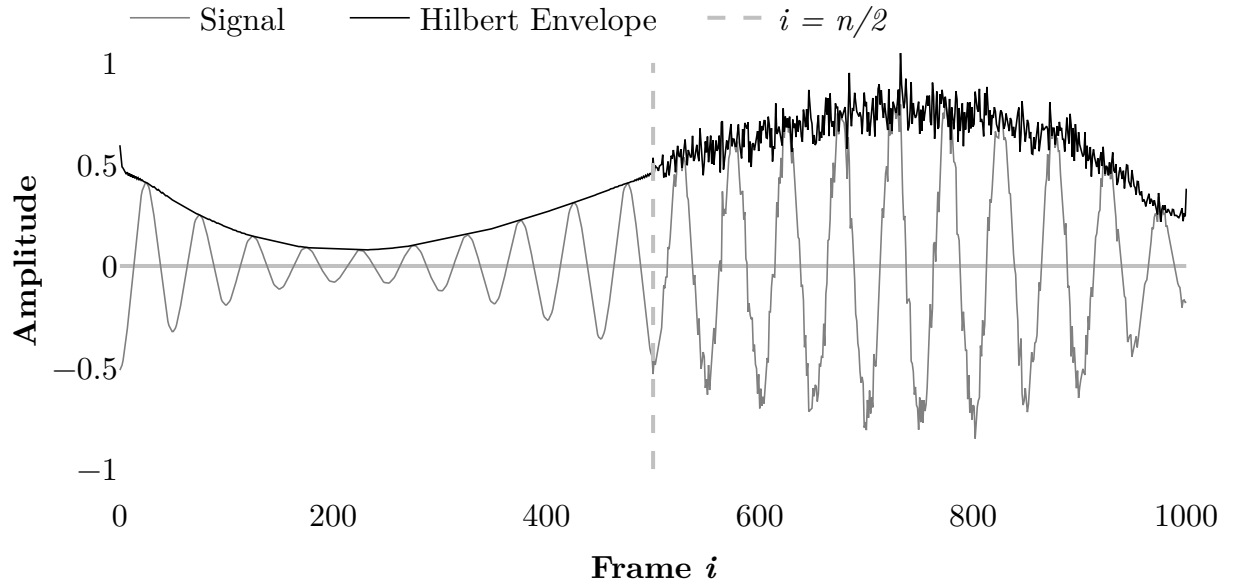

Figure 2: Envelope of a pure sinusoid with a local frequency of 20 cycles modulated by a polynomial of degree 3, as obtained by the Hilbert Transform approach. In the first half, the sinusoid is free of noise, while in the second half Gaussian noise with a standard deviation of 1/10 of the maximum amplitude of the wave was added to the base signal.

its fast varying part, called throughout the literature as its (temporal) fine structure, carrier component, or frequency modulation (FM).

In the case of broadband signals, the problem of envelope detection of an arbitrary wave is ill-posed, in the sense that an infinite number of pairs of $E(t)$ and $C(t)$ can result in the same specific $W(t)$ (Turner and Sahani, 2011; Meng et al., 2013; Loughlin and Tacer, 1996).

We address this problem by assuming $C(t)$ normalized between $\{-1, 1\}$. This assumption doesn't cause any loss of generality since, given an arbitrary $C(t)$, we can obtain its normalization by dividing the function by its absolute global maximum, that is, $\hat{C}(t) = C(t)/\max(|C(t)|)$, provided that $\max(|C(t)|) \neq 0$.

In this work we are concerned with the discrete version of this problem: given a finite digital wave represented by the vector $\mathbf{w} \in \mathbb{R}^n$, instead of the continuous function $W(t)$, obtaining the temporal envelope of $\mathbf{w}$.

The preceding definitions can be translated to this discrete scenario assuming that the discrete quantities arise from observing the continuous ones at regular time intervals. In that case, the equality $i = t$ fps can be used to link both settings, where i is the index of each observation, t stands for the time in seconds, and fps is the frame rate, or the number of observations made in the period of a second. We can thus define:

$$\mathbf{w}, \mathbf{e}, \mathbf{c} \in \mathbb{R}^n \tag{2a}$$

$$\mathbf{w} = (w_0, w_1, \cdots, w_{n-1}) \tag{2b}$$

$$\mathbf{e} = (e_0, e_1, \cdots, e_{n-1}) \tag{2c}$$

$$\mathbf{c} = (c_0, c_1, \cdots, c_{n-1}) \tag{2d}$$

$$\mathbf{w} = \mathbf{e} \odot \mathbf{c} \quad w_i = e_i c_i \ \forall \ i, \ 0 \leq i \leq n - 1 \tag{2e}$$

The $\odot$ operator in 2 stands for the Hadamard product, denoting element-wise multiplication of two vectors. Figure 3 provides an example of the vectors just defined: $\mathbf{w}$ is obtained by modulating the normalized carrier wave $\mathbf{c}$ using the envelope $\mathbf{e}$.

Note from Figure 3 that, at some local maxima, the envelope is well-defined, being in fact equal to the value of the wave w_i, since $c_i = 1$ at those points, rendering $e_i = 1 w_i$.

The carrier wave $\mathbf{c}$ in the figure is periodic, and normalized between $\{-1, 1\}$. Each cycle of the wave, then, reaches the value 1 only at its maximum making w_i equal to the value of the envelope at one point in each cycle.

Image 3, along with the concave hull illustrated in figure 1, helps forming the intuition behind the method developed in this work: the idea is to identify the local extrema that touch the envelope, those being also the extrema of each (pseudo-)cycle.

This is accomplished with the use of a circle with a defined radius, large enough to prevent it from reaching the smallest local extrema when "rolled" around the (geometric) representation of the wave. Only the extrema touched by the circle will be marked as part of the envelope.

In the next section, a compatible geometric representation will be formulated, along with the necessary tools to estimate the average curvature of the wave and, consequently, the appropriate radius of the circle.

5.2.2   Simplified Representation of a Wave

Instrumental to the method here presented is the definition of a pulse: each time the value of $\mathbf{w}$ changes sign, that is, every time the discrete wave $\mathbf{w}$ crosses the horizontal axis, the beginning of a pulse is defined, with the next crossing defining its end. This definition is in line with the one presented in Division and Service (1996), where a pulse is defined as a rapid change in the amplitude of a signal, followed by a fast return to the baseline value; zero, in our case.

From 3 and the discussion in the preceding chapter, we can see that only the maximum point of a positive pulse or the minimum point of a negative one can potentially be equal to the envelope and are, thus, our only points of interest. We can then proceed, for the rest of the method, considering only those points, to great computational economy.

For this reason we define P as the set of the points $\{P_0, P_1, \cdots, P_{m-1}\}$ where $P_j = (i_j, |w_j|)$, that is, $|w_j|$, the absolute value of the extrema of each pulse, becomes the ordinate of each point and i_j, its original index, becomes the abscissa.

This relation is illustrated in figure 4. We call it P to emphasize the connection with the pulses of $\mathbf{w}$.

The points in P are not fit for geometric interpretation yet, since the abscissa and ordinate of their orthogonal coordinate system have different units: in the vertical axis, we have a unit related to the instantaneous intensity of the wave, while in the horizontal axis we have the index at which each extremum occurs, ultimately a time-related unit.

Specifically, this coordinate system has a basis vector $\mathbf{b} = \{\mathbf{b}_x, \mathbf{b}_y\}$ with $\mathbf{b}_x = (a, 0), \mathbf{b}_y = (0, i)$ where a is an amplitude unit such as decibel, a measure of the instantaneous sound pressure, volt or even a fraction of a fixed maximum value. i represents the
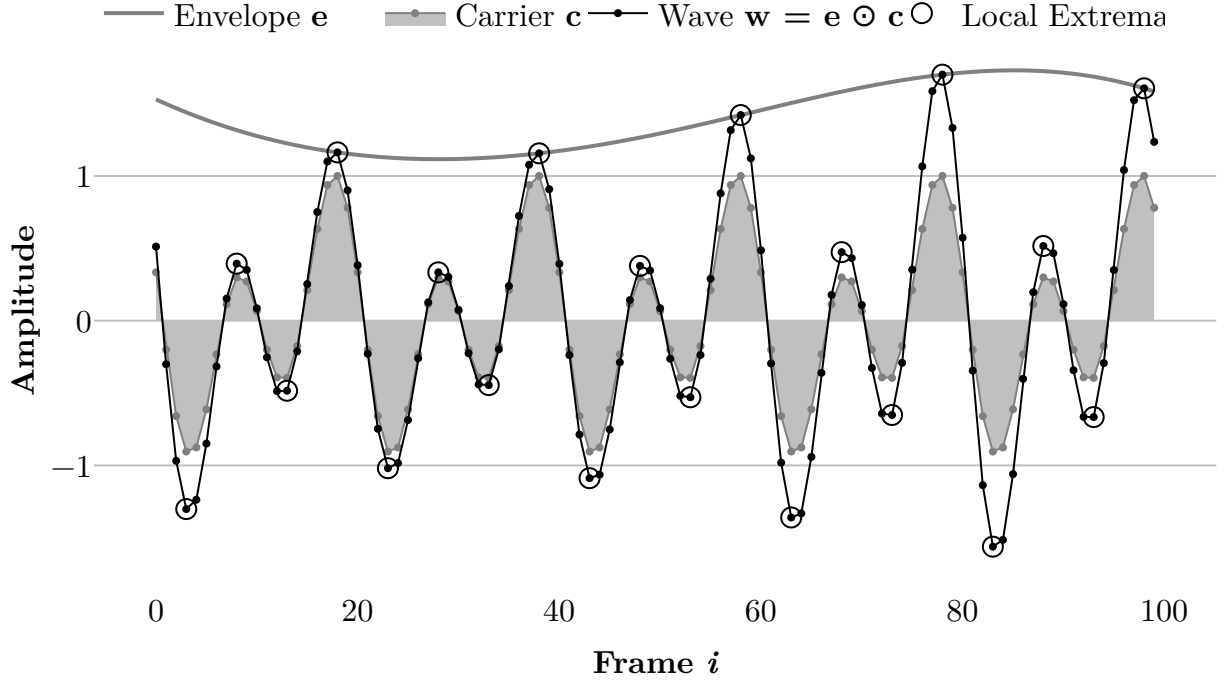
Figure 3: Example of a discrete wave **w** composed by the element-wise multiplication of an envelope **e** and a carrier **c**. The local extrema of **w** are highlighted with a circle.
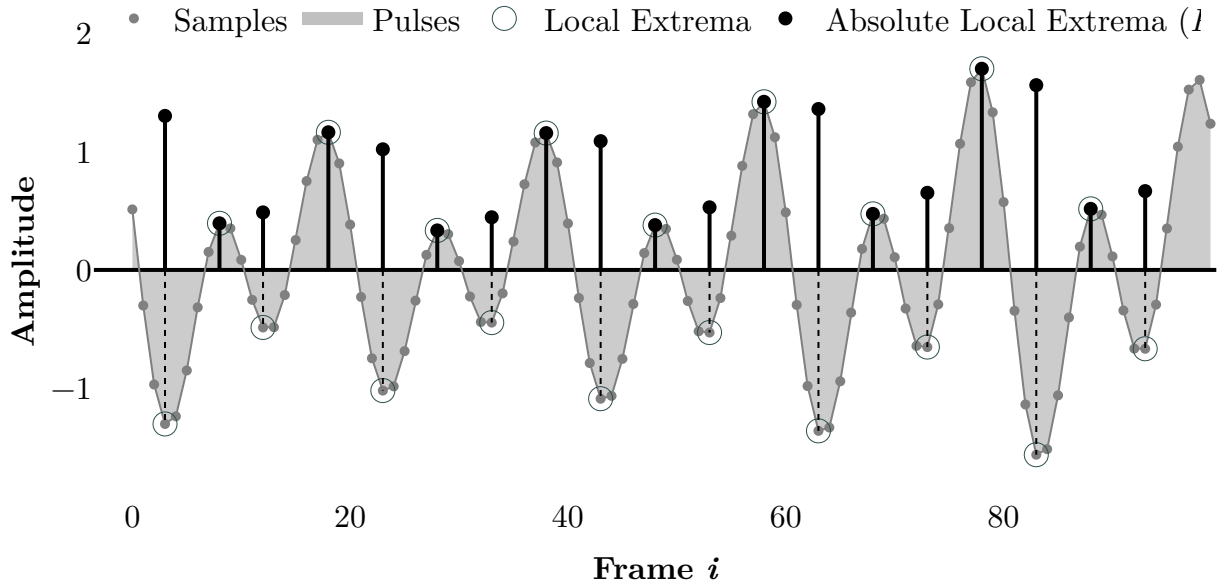

Figure 4: Example of a discrete wave **w** divided into pulses, of which the extrema are highlighted with a circle. P is the set of the points $P_j = (i_j, |w_j|)$ representing the absolute value of those extrema.

index of the sample, and is linked to time by the aforementioned relation $i = t$ fps.

We are interested in finding a coordinate system where a pulse's amplitude $|w_j|$ and length would define, on average, a square.

To achieve that, we divide the basis vector of the original vertical axis by the average of all ordinates, effectively cancelling the unit of the vertical components of the points. We could divide the basis vector of the horizontal axis by the average of the difference between the abscissa of a point and the abscissa of the immediately posterior, likewise eliminating the axis' unit.

The drawback is that we would lose the direct link between P and $\mathbf{w}$ that arises from the fact that the abscissas of the points in P are indices i of $\mathbf{w}$: we choose instead to leave the abscissa intact and multiply the now unitless vertical basis axis by the average of the difference between the abscissa of a point and the abscissa of the immediately posterior. In this way, both axes will have unit i, and the relationship is preserved, making it easier to recover the envelope points later.

In this new coordinate system we have a basis vector $\mathbf{b}' = \{\mathbf{b}'_x, \mathbf{b}'_y\}$ where $\mathbf{b}'_x = (i, 0)$ and $\mathbf{b}'_y = (0, i)$, related to the original basis vector by equation 3:

$$\mathbf{b}'_x = \mathbf{b}_x \tag{3a}$$

$$\mathbf{b}'_y = \frac{\left(\frac{i_{m-1} - i_0}{m-1}\right)}{\left(\frac{\sum_{j=0}^{m-1} |w_j|}{m}\right)} \mathbf{b}_y \tag{3b}$$

The effect of this normalization can be seen in the points shown in figure 5, shown in scale in both coordinate systems. The set V of the vectors from each point of P to the next is also shown in the picture, and it will be used to estimate the average curvature of P. We define this set in equation 4 below.

$$\mathrm{V} = \{v_0, v_1, v_{m-2}\} \quad \text{where} \quad v_k = P_{j+1} - P_j \quad \forall \quad 0 \leq k \leq m - 2 \tag{4}$$

5.2.3   Discrete Curvature Estimation

Translating the explanation of alpha shapes in Herbert Edelsbrunner and Mücke (1994) to our context, we can say that the points that are part of the envelope are those touched by a circle of a given radius r, coming from $y \to \infty$ towards the signal, that is not allowed to contain any point of the signal; Intuitively, one can picture a circle being

rolled above the signal, and marking the points it touches as envelope points.

To infer the appropriate radius r of such circle a measure of discrete curvature is needed. Discrete curvature estimation is an important task in image processing (Fleischmann, Wietzke, and Sommer, 2010) for which no default definition exists.

The two possible general approaches are the derivation of direct methods, that use characteristics of the discrete wave to calculate the curvature, or the use of the curvature of a smooth, continuous curve fitted to the discrete wave (Coeurjolly, Miguet, and Tougne, 2001).

Concerning direct methods Carroll et al. (2014), for example, derive three such definitions based on the approximation of a circle by an inscribed, centered and circumscribed polygon. In the context of three-dimensional meshes, Libor Váša, Kühnert, and Brunnett (2016) evaluate a range of existing estimators from a multivariate point of view.

Those approaches define the discrete curvature in relation to the vertices of a wave (or mesh, in the three-dimensional case). In our particular case, it wouldn't make sense to talk about the curvature of a single pulse, since it is what our points ultimately represent. Instead, we are interested in the change of direction, what the term curvature ultimately means, between two adjacent pulses, as the vectors in the set V suggests.

We thus proceed to define a discrete curvature measure over the edges of a discrete wave. To that end we are going to apply the definition of smooth curvature as the rate of change of the unit tangent to a curve, noting that this is equivalent to that of the osculating circle (L. Váša et al., 2016).

5.2.4 The Equivalent Circle Approach

The rationale is to find, for each vector $v_k \in$ V, the radius r_k of the equivalent circle whose tangent has the same change in direction, in the same horizontal distance, as the vector of interest, as shown in figure 6. The average curvature of P will then be obtained by the average of all radii r_k.

From figure 6 is easy to see that $r_k = v_{k,x} \sin(\theta_k)$ where $v_{k,x}$ is the component of v_k in the horizontal direction; θ_k itself can be obtained using the slope $m_0 = 0$ of a line in the horizontal direction and the slope $m_k = v_{k,y}/v_{k,x}$ defined by v_k via the equality $\theta_k = \arctan\left((m_k - m_0)/(1 + m_k m_0)\right)$.

The radius of the circle that represents the average curvature of V (and P), and will be used to obtain the envelope of the discrete wave **w** can be obtained via equation 5:
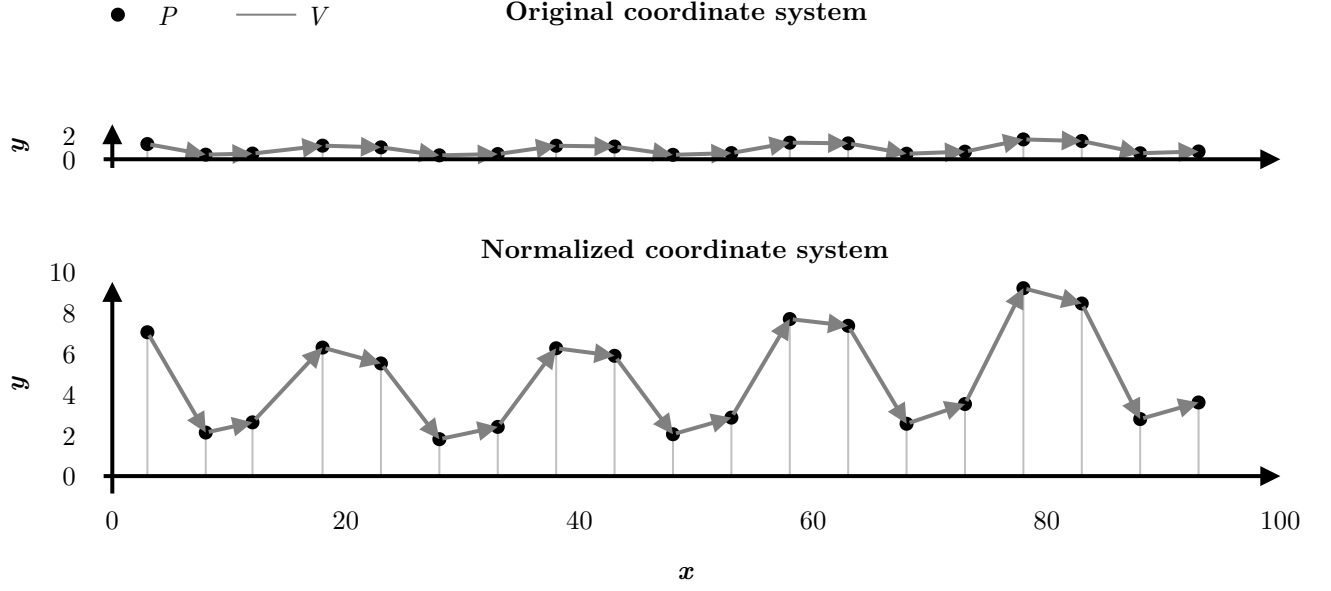
Figure 5: The set of points P and the set V of vectors between two adjacent points of P, in both the original and normalized coordinate systems, shown in scale.
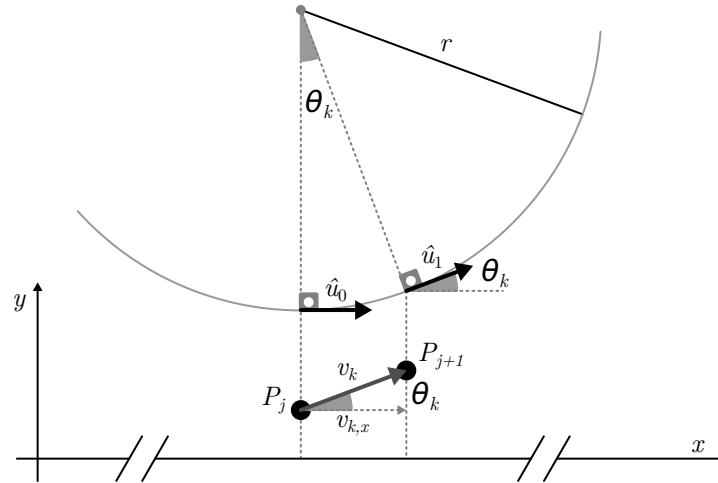

Figure 6: The tangent unit vector of the circle changes from the horizontal direction in $\hat{u}_0$ to an inclination of θ_k in $\hat{u}_1$, θ_k being the angle that the vector v_k makes with the horizontal direction.

$$r = \frac{\sum\limits_{k=0}^{m-2} \left(\frac{v_{k,x}\sqrt{v_{k,x}^2 + v_{k,y}^2}}{v_{k,y}} \right)}{m-2} \tag{5}$$

We need now to construct an algorithm, using the radius just obtained, to identify the points that belong to the envelope. Generally, the algorithm for the alpha shapes approach resorts to the construction of the Delaunay triangulation of the set of points, that is latter filtered to contain only the outer edges. We can adopt a more straightforward method, as our points have a defined structure:

Let $x_o, y_o \in \mathbb{R}, \quad 0 \le y_o < +\infty$ be the coordinates of the center of a circle with radius r, that can be placed anywhere in the upper plane of the coordinate system;

Let $P_a, P_b, P_c \in \mathrm{P}, \quad P_a \ne P_b \ne P_c$ be three different points of the set P;

For all circles of center x_o, y_o and radius r that can be constructed to pass through any two different points of P, that is, $\forall \quad P_b, P_c$ such that $(P_{b,x}-x_o)^2+(P_{b,y}-y_o)^2 = (P_{c,x}-x_o)^2+(P_{c,y}-y_o)^2 = r^2$, P_a will be part of the set E of the points belonging to the envelope if and only if none of those circles contain P_a, or $P_a \in \mathrm{E} \leftrightarrow (P_{b,x}-x_o)^2+(P_{b,y}-y_o)^2 > r^2$. Summarizing this, we have equation 6:

$$P_a \in \mathrm{E} \leftrightarrow (P_{b,x}-x_o)^2 + (P_{b,y}-y_o)^2 > r^2, \tag{6a}$$

$$\forall\, P_b, P_c \mid (P_{b,x}-x_o)^2 + (P_{b,y}-y_o)^2 = (P_{c,x}-x_o)^2 + (P_{c,y}-y_o)^2 = r^2, \tag{6b}$$

$$P_a, P_b, P_c \in \mathrm{P}, \quad P_a \ne P_b \ne P_c \quad \text{and} \quad x_o, y_o \in \mathbb{R}, \quad 0 \le y_o < +\infty \tag{6c}$$

The algorithm follows directly from the definition in 6 after noting that the first and last members of P will always be part of the envelope, since they are reachable by a circle of any radius r. From P_0, our first pivot point, we construct circles passing through $P_1, P_2, \cdots, P_d$ until a circle that doesn't contain any point in P is found; P_d then becomes the new pivot point and is included in E, and this procedure continues until P_{m-1} is reached.

The procedure is formalized in the algorithm 5.2.4. We don't provide similar algorithmic descriptions of the preceding steps since the Python implementation can easily serve as pseudocode.

[H] Retrieve Envelope **Given** P $= \{P_0, P_1, \cdots, P_{m-1}\}, r$, **circle**, a function that returns the center of a circle of a given radius passing through two points and **distance**, a function that returns the Euclidean distance between two points, **Let** E $= \{P_0\}, \mathrm{id}1 = 0, \mathrm{id}2 = 1$ **Define** $P_c =$ point, empty $=$ boolean $\mathrm{id}2 < m$ $P_c \leftarrow$ **circle**$(r, P_{\mathrm{id}1}, P_{\mathrm{id}2})$ empty

$\leftarrow i = \text{id2} + 1$ m **distance**$(P_c, P_i) < r$ empty $\leftarrow \text{id2} \leftarrow \text{id2} + 1$ **break** empty $\text{E} \leftarrow \text{E}$ $\cup \{P_{\text{id2}}\}$ id1 $\leftarrow$ id2 id2 $\leftarrow$ id2 + 1

The set E obtained via algorithm 5.2.4 is a set of points. The most straightforward way to transform those points into the vector $\mathbf{e} \in \mathbb{R}^n$ is via linear interpolation, the approach used in this work. It is worth noting, however, that many other procedures are available, both for the interpolation and the smooth approximation of points.

5.3 RESULTS AND DISCUSSION

Given the lack of a precise definition of what an envelope should be, we begin this section suggesting an empirical metric, based on the behaviour of the carrier wave $\mathbf{c}$. We also comment on some metrics presented in the literature, and compare our algorithm with traditional demodulation approaches, both in relation to execution time and accuracy; to this last end, a numerical indicator is also introduced.

We also illustrate how the method here presented can be used to isolate not only a single envelope of the wave, but the superior and inferior envelopes as well, that we call frontiers, ending the section with a suggestion on how the proposed algorithm can be useful beyond envelope detection, by identifying the approximate locations of the pseudo-cycles in a quasi-periodic wave.

Due to an assortment of factors, such as the unconventional approach presented in this work and the intrinsic difficulty in visualizing discrete waves given their high dimensionality, the authors deemed helpful to provide a companion website with interactive visualizations (Tarjano, 2020).

An implementation of the method in pure Python, fully functional, can also be found at the repository dedicated to this work as well as a Python module available at the Python Package Index (PyPI) and conveniently installable via the command 'pip install signal-envelope'. Besides, those working under Windows 64-bit can benefit from a specialized dynamic library coded in C++ that is automatically used within this module.

5.3.1 Quality of an envelope

There are no agreed-upon objective metrics to assess the quality of an envelope. Recalling figure 3, however, and the accompanying discussion, an intuitive way is to consider the carrier wave $\mathbf{c}$, and how well it seems to be bounded by the interval $\{-1, 1\}$.

To illustrate how the algorithm performs on such metric, we provide the following figures of original discrete waves $\mathbf{w}$, their extracted envelope $\mathbf{e}$, and the inferred carrier,

obtained after dividing, element-wise, the original wave by its envelope.

From figure 8 it can be seen that the algorithm performs well under the carrier metric. This is especially remarkable for the case of spoken voice, a notorious complex wave with fast changes in pitch and high noise content.

These characteristics are in stark contrast with those of the sang voice in figure 9, where a recording of an alto singer is shown; there, the wave is almost periodic, with a stable waveform.

In an effort to formalize the assessment of the quality of temporal envelopes, Loughlin and Tacer (1996) proposed some conditions necessary, but not sufficient, to ensure the physical plausibility of an envelope. We comment briefly below that the algorithm here presented satisfies the four conditions presented.

The first one states that, if a signal is bounded in magnitude, then its envelope should be, as well. Our algorithm complies with this requisite, as the envelope is composed of selected samples from the original signal itself.

The second statement is that if the signal has a finite frequency range, that frequency range must not be exceeded by the envelope. This is only directly applicable in the continuous case, where this condition was proposed.

Nevertheless, if we picture both the wave and the envelope as samples from continuous functions, is easy to see that this is the case for our algorithm; since the envelope is composed by some samples of the digital wave, in the limit where all the samples of the wave are also part of the envelope, we have that the maximum frequency for both is the same.

In all other cases, the maximum frequency of the envelope will be smaller than that of the original wave. The range of frequencies of the envelope, then, will always lie between zero and the maximum frequency of the original wave.

The third condition states what, for a periodic signal, the envelope should be a straight line with the intercept equal to the amplitude of this signal. Since all the local maxima have the same amplitude in the case of a periodic wave, our envelope will indeed be a straight line, with the same amplitude as the original wave. An example of such behaviour can be observed in figure 10.

The fourth and last condition states that if a wave is multiplied by a constant, the envelope should also be multiplied by the same constant. That is also the case, again because the points that form the envelope are points of the original wave itself.
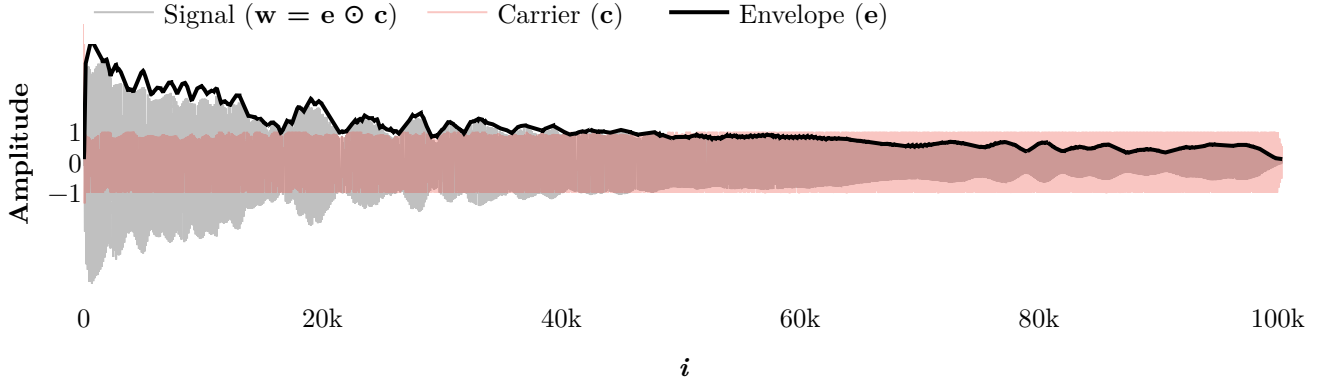
Figure 7: Signal, envelope and carrier for a record of a bend performed on an electric guitar.
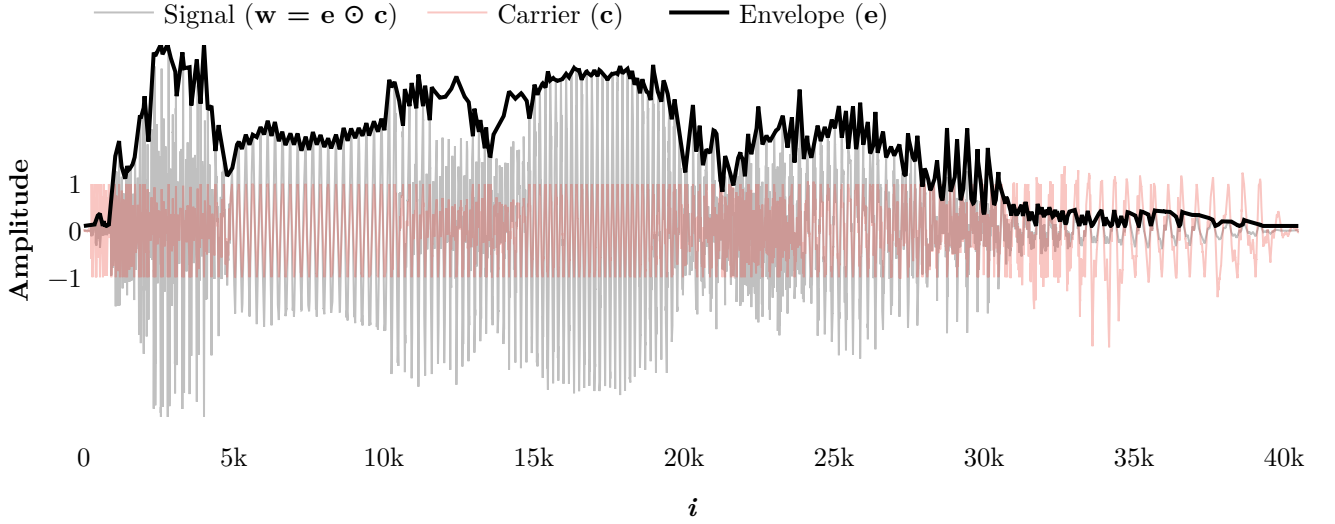

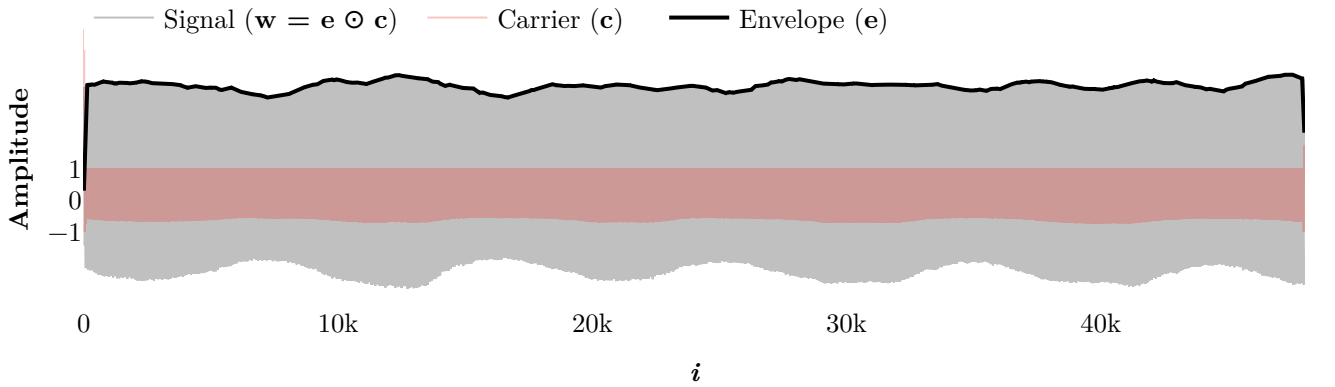Figure 8: Signal, envelope and carrier for a record of male voice uttering the word "amazing".

Figure 9: Signal, envelope and carrier for a record of an alto singer.

5.3.2 Comparison with traditional algorithms

Direct comparison with many of the more recent algorithms is made difficult by the unavailability of digital implementations of such works, many designed to process analog signals (e.g., Assef et al., 2018).

Nevertheless, insight can be gained from a comparison of the results of the method here proposed with some of the most common envelope extraction algorithms. Specifically, we compare the present method with the following approaches:

1. Smoothing: This is a relatively simple procedure that, applied to the absolute values of a discrete signal, can provide an approximation of the variation of its amplitude in time. We use the Savitzky-Golay algorithm (Savitzky and Golay, 1964) with a window width of 3001 samples and a cubic polynomial.

2. Filtering: The absolute value of the wave is filtered using a digital implementation of the Butterworth (Butterworth et al., 1930) low pass filter. We use an order 2 filter with a cut off frequency of 10 Hz.

3. Hilbert Transform: We apply the Hilbert transform to the absolute values of the wave after filtering the original signal with a Butterworth filter with a cut off frequency of 100 Hz, using the absolute value of the analytic signal to obtain the envelope.

In 10 we illustrate the envelope extracted by the conventional algorithms, contrasting it with the result of the method presented in this work, for a simple sinusoid. Note that none of the conventional algorithms complies with the third condition proposed in Loughlin and Tacer (1996) that the envelope of a periodic wave should be a straight line.

For a recording of the key 33 of a grand piano, where parts of the sound are highly percussive and noisy, the Hilbert approach presents considerable oscillations, as can be seen in figure 11.

In figure 12 all envelopes exhibit a similar movement along the samples, but the ones generated by traditional algorithms are undershooting the original wave.

All the traditional methods demanded careful choice of parameters. In the case of the Hilbert transform, pre-processing in the form of filtering was applied. That is in contrast with our method, that organically tunes itself via the automatic choice of the radius of the circle representing the average discrete curvature of the signal.
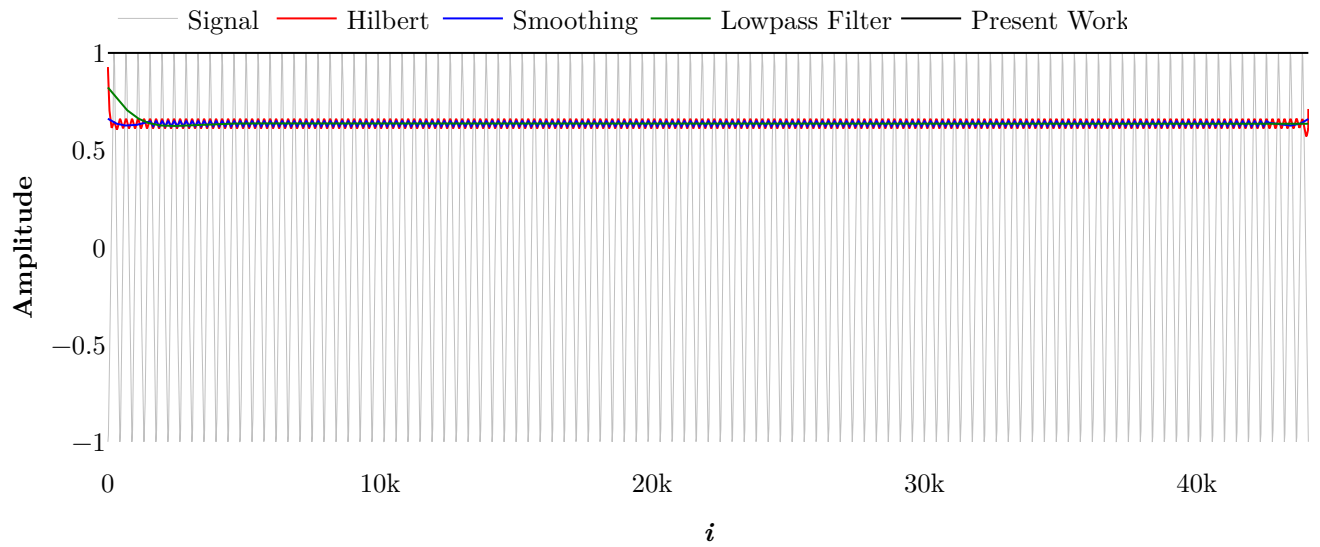
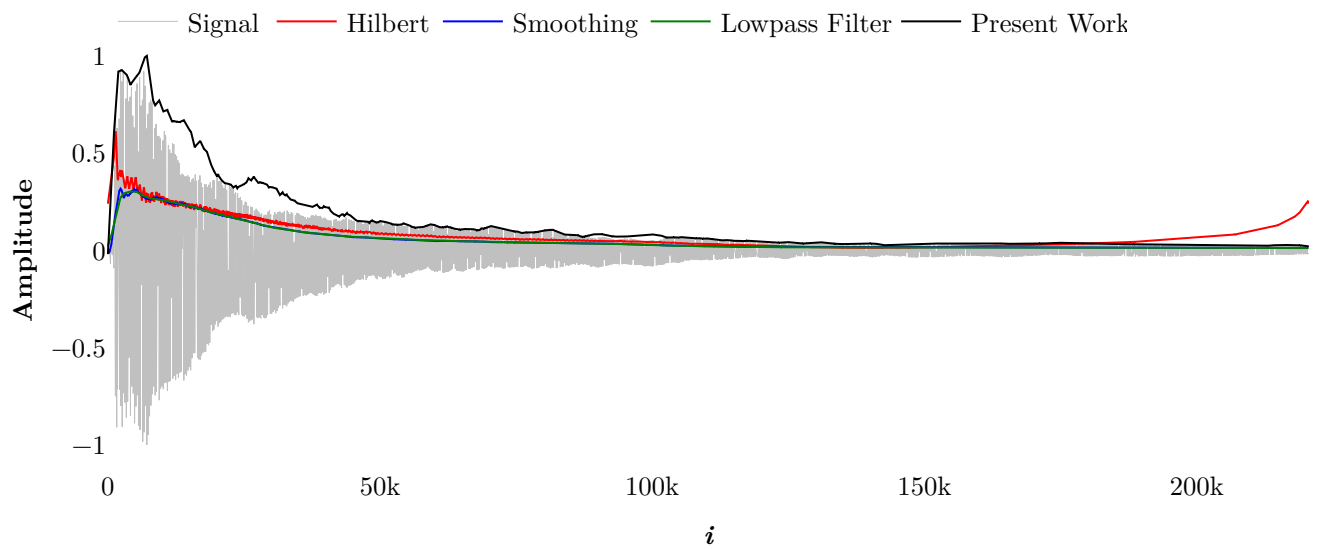Figure 10: Comparison of envelope detection algorithms for a simple sinusoid

Figure 11: Comparison of envelope detection algorithms for the recording of the key 33 of a grand piano

Table 1 presents a comparison of the average errors of the techniques. The average error was calculated by the average of the squared distance of the points in the envelope divided by two and the absolute value of the wave. The rationale is that the values of the envelope divided by two should pass near the average of the absolute values of the wave. This metric is only valid for reasonably smooth envelopes, however.

We can see from the table 1 that the method presented here consistently outperforms the traditional approaches for all waves tested, yielding an error 33% smaller, on average, than the other methods. It is also worth noting that the errors of the traditional methods are very similar between them.

The times taken for the algorithms compared here to process each wave are shown in 2. We used specialized implementations available in libraries of the Python programming language. The presented algorithm is in average the second-fastest.

5.3.3 Frontiers

In practice, is not uncommon for a discrete wave, especially in the case of sound, to present somewhat different positive and negative contours; in those cases, the algorithm here presented can be used independently in the positive and negative pulses of the wave to define two "envelopes", that we are going to call superior and inferior frontiers.

Figure 13 illustrates the frontiers of six diverse discrete sound waves, as well as an in-detail view of the highlighted segment for each signal. All waves are records of physical sounds, chosen to represent the applicability of the algorithm in real-world scenarios.

The frontiers are satisfactorily detected in the case of harmonic and inharmonic sounds, and are robust in relation to the number of samples and the frequencies of the waves. That is a peculiarity of our algorithm: the traditional algorithms are unable to generate two distinct envelopes for the same wave.

5.4 APPROXIMATED LOCATION OF PSEUDO-CYCLES

The envelope is shown to add complexity to the spectral representation of a wave (Tarjano and Pereira, 2019), and an accurate description of the envelope, while describing the evolution of the instantaneous amplitude of a signal in time, would also simplify further spectral analysis. Figure 14 shows the frequency-domain power spectrum for the wave and the carrier presented in 3. For the carrier, the power spectrum presents two nonzero values, while the power spectrum for the wave composed of the carrier modulated by the envelope is more complex.
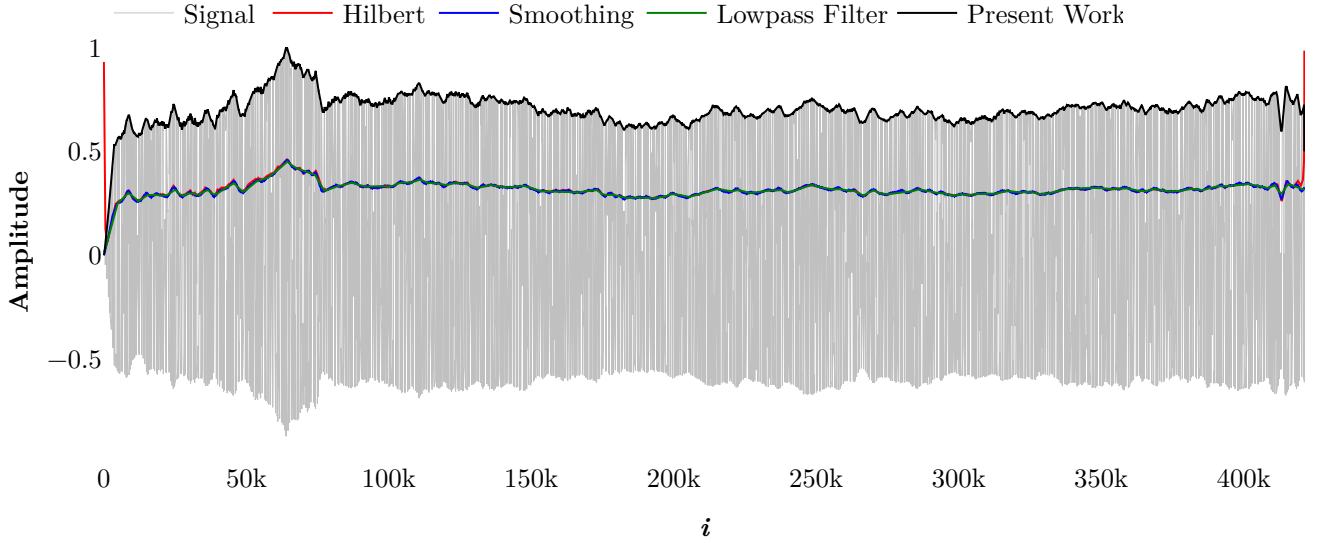
Figure 12: Comparison of envelope detection algorithms for the recording of the vocalization of a soprano singer

	Average error per frame			
	Present Work	Smoothing	Low pass	Hilbert
alto	0.073	0.083	0.083	0.082
bend	0.009	0.015	0.015	0.014
piano	0.005	0.006	0.006	0.005
brass	0.024	0.035	0.035	0.033
soprano	0.042	0.065	0.065	0.065
spoken voice	0.024	0.037	0.037	0.034
tom	0.003	0.005	0.005	0.005
sinusoid	0.113	0.193	0.195	0.191
Average	0.037	0.055	0.055	0.054

Table 1: Comparison of the average errors of the algorithm presented in this work with the most common methods of digital envelope identification. The discrete waves used were normalized between -1 and 1. The average is between the absolute value of those waves and the envelope divided by two.

	Time in seconds			
	Present Work	Smoothing	Low pass	Hilbert
alto	0.008	0.101	0.007	0.011
bend	0.030	0.226	0.012	0.131
piano	0.034	0.374	0.014	0.042
brass	0.032	0.297	0.011	0.045
soprano	0.177	0.735	0.027	0.566
spoken voice	0.014	0.105	0.003	0.038
tom	0.007	0.084	0.018	0.026
sinusoid	0.007	0.091	0.005	0.011
Average	0.039	0.252	0.012	0.109

Table 2: Comparison of the processing time of the algorithm presented in this work with the most common methods of digital envelope identification.
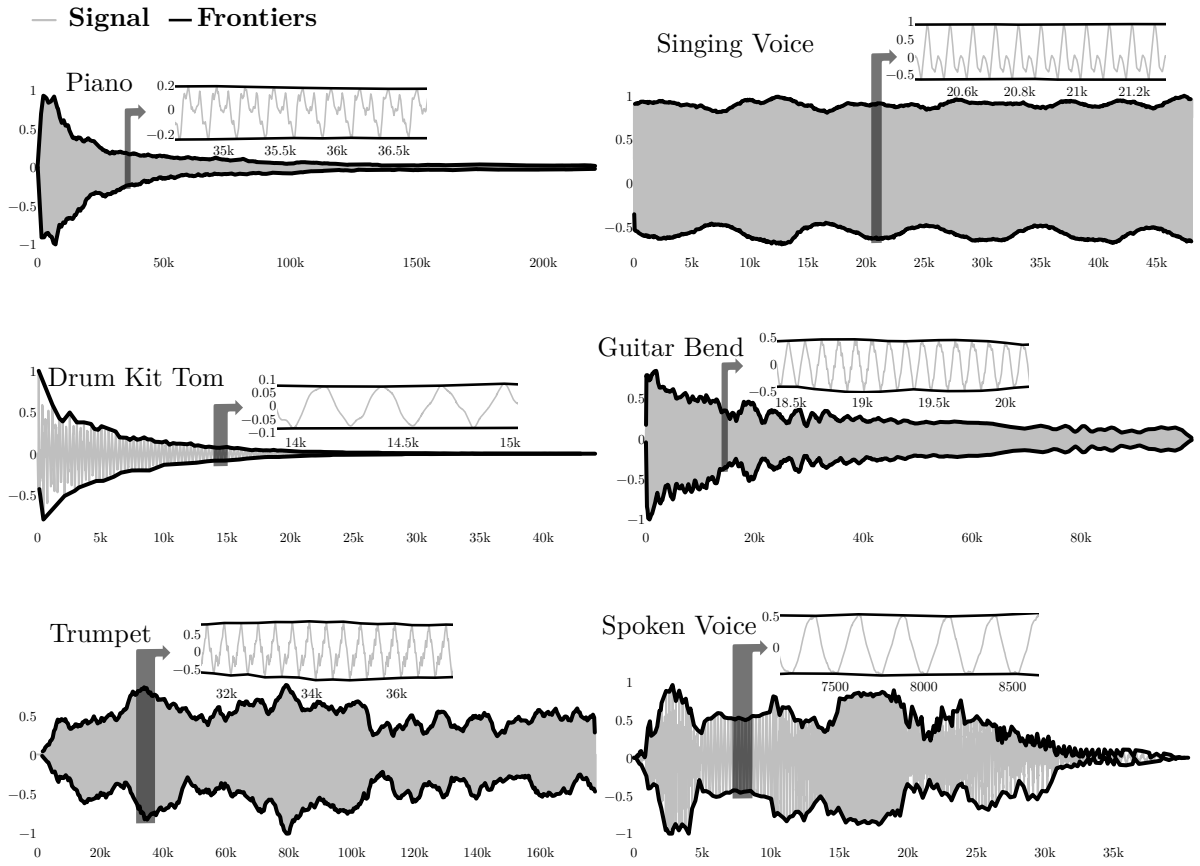
Figure 13: Positive and negative frontiers of six digital waves, as extracted by the algorithm here presented. For each wave, the region highlighted in black is shown in detail besides the whole wave. For each wave, the horizontal axis is the sample number i, while the vertical axis is the normalized amplitude. All 6 waves were recorded at 44100 fps.
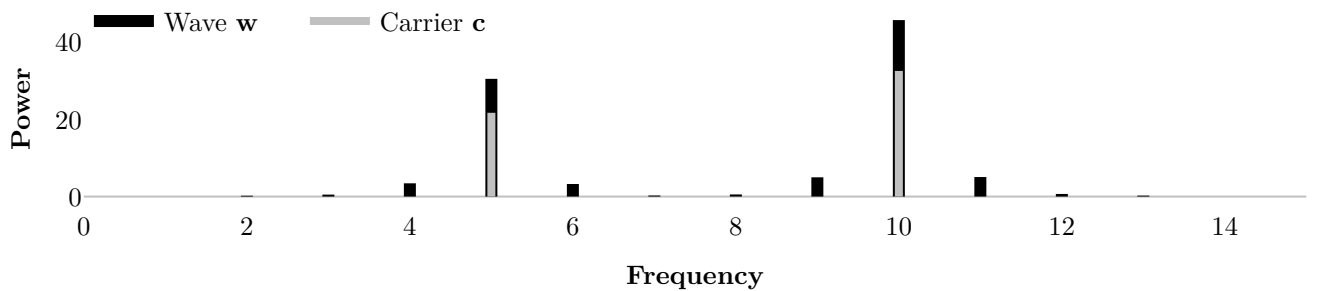

Figure 14: Fourier power spectrum for the wave and carrier shown in 3.

Moreover, the algorithm developed in this work naturally divides a signal into its pseudo-cycles, pinpointing them in the time-domain, thus providing the building blocks for the reconstruction of the fine structure of the wave.

One can see in figure 15 a section of the plot of where the local maxima and minima occurs in the recording of an alto singer show in 9. The average between the two is also shown. With this information, one can split a discrete wave into its pseudo-cycles, as is illustrated in Figure 16: It shows the superposition of this extracted pseudo-cycles, after being normalized by length, and their average. This same average is shown shifted, to exemplify that approximate C^0 and C^1 continuity is achieved.

5.5 CONCLUSION

This work fills a gap identified in the theory of digital signal processing, where the lack of general procedures for the accurate identification of the temporal envelope of arbitrary waves poses an obstacle to the complete description and eventual manipulation of signals.

By presenting a general approach for envelope detection, and eliminating the need for parameter tuning, we expect to facilitate further research at the many areas that rely on envelope detection techniques, and improve already existing algorithm that have envelope detection as an intermediary step.

Illustrating the efficiency of an algorithm inspired by geometric aspects of a discrete signal on a wide range of real-world signals, we also hope to encourage research in this direction.

The present work also highlights some gaps in our current understanding and definition of temporal envelope and, in so doing, endeavours to motivate further research in this topic, that can ultimately give rise to a more mathematically sound envelope theory, especially concerning broadband waves; suggestions in this direction are also presented.

While relevant in its own accord, the procedure here presented approximately isolates the individual pseudo-cycles of a wave, pinpointing them in the time-domain.

Similarly, by decomposing a wave into its envelope and carrier, and simplifying the representation of such carrier in the frequency-domain, a rich set of advancements, especially in the areas of sound compression and synthesis in the time and frequency-domains, can be built upon this initial algorithm.

By illustrating how this information can be used to obtain the average waveform of a discrete wave, we hope to encourage research in this area, ultimately leading to
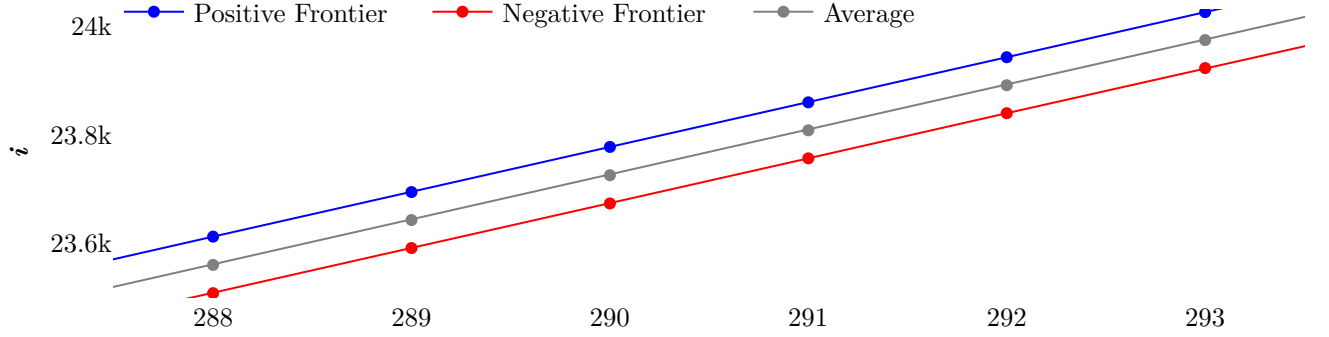
Figure 15: Section of the lines formed by the positions of the positive and negative frontiers of the digital wave illustrated in 9, as extracted by the algorithm presented, and their average.
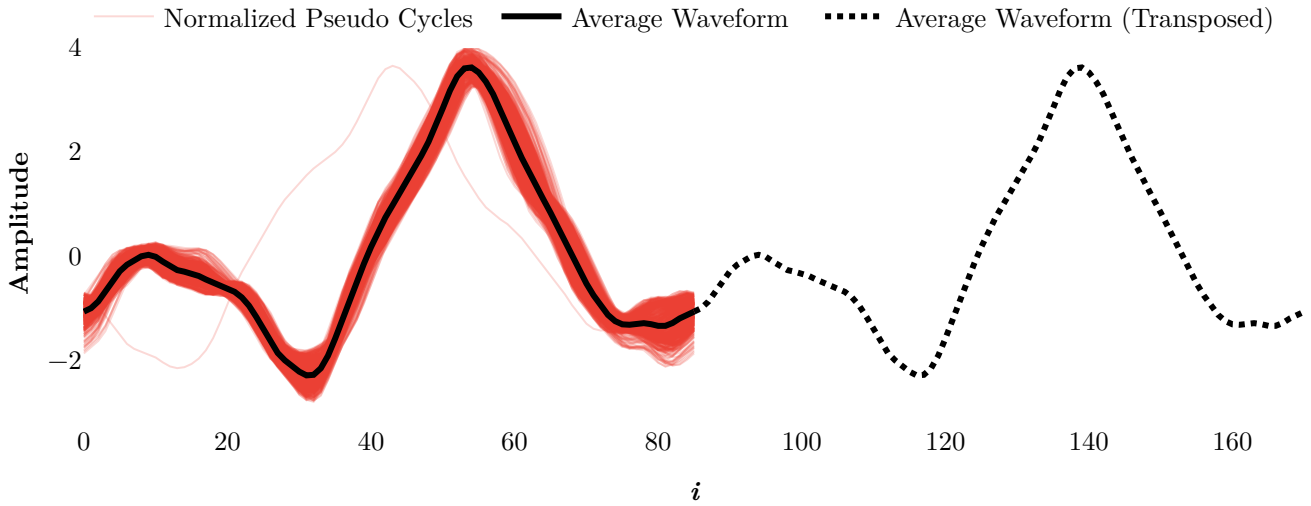

Figure 16: Pseudo-cycles of the wave illustrated in 9, segmented and superposed, and their average. The average is also shown shifted by one average period, to illustrate the average continuity.

alternative representations of discrete waves.

5.6 FREQUENCY CONTENT

5.7 REPRESENTATIONS

6 PSEUDO CYCLES BASED REPRESENTATION

Sample-based virtual instruments are the de facto standard for the emulation of real-world acoustic instruments. Nevertheless, little attention is given to the encoding of the samples used in those instruments; lossless formats like WAV or proprietary codecs are common, pushing the on-disk size of those libraries to sizes exceeding the gigabytes. This renders the most realistic digital instruments inaccessible in many but the most high-end hardware; controllers, like electronic drum kits and keyboards and even less robust personal computers are often unable to take advantage of those libraries. We propose a lossy compression algorithm tailored specifically to compress pseudo periodic signals, as those comprising these instruments; due to the highly redundant nature of these sounds, improvements in relation to traditional codecs can be obtained. The algorithm operates entirely in the time domain, avoiding domain conversions common in lossy compression algorithms, thus exhibiting low latency in the decoding phase. It operates iteratively, dividing a signal into its pseudo cycles according to an estimate of the average waveform of those pseudo cycles, which is thus updated, and the process is repeated. We provide C++ source code and implementation of the algorithm, that was used to compare its performance against the MP3, AAC, and Opus codecs, finding the proposed method to be at last 13 times faster in decoding tasks, while maintaining comparable compression rate and quality, as measured by MSE and the ViSQOL library. A website and a GitHub repository with complementary material, where results can be heard, are also available.

Domain specific compression algorithms and codecs exist in many areas. Chen et al., 2016, for example, presents a scheme for compressing convolutional neural networks, while Cánovas, Moffat, and Turpin, 2014, besides revising existing methods, proposes two lossy schemes for the compression of quality score data of genomic sequencing. The use of lossy compression for wireless networks of sensors is studied in Zordan et al., 2014.

More recently, the work of Calhoun et al., 2019 explores the benefits of specific lossy compression algorithms in the context of checkpoints of computational simulations that are stored between simulation sessions; New technologies, such as 3D immersive audio, have propelled specific compression algorithms, such as the one presented in Hu

et al., 2021.

Regarding sample-based digital musical instruments, however, no such specialized algorithms exist despite those signals, being highly uniform, offering a potential for compression that is not present in most general sounds and remains largely unexplored.

This overlook arises in part from the marketing strategy of the majority of those products, where the sheer size of the library is advertised as an equivalent of quality and versatility. An overall prejudice against compressed formats in general from the part of the music production community supports this biased view.

At last some of the criticism directed to the MP3 standard, and lossy codecs in general, is meant to establish a symbolic capital in the face of the crescent struggle between established recording labels and alternative channels of music diffusion, such as streaming services O'Grady, 2020.

The adoption of special compressed formats would enable state-of-the-art digital instruments to be implemented in a wider range of hardware, where the storage costs now make the cost prohibitive, such as electronic drum kits, digital pianos, and even cell phones and tablets, democratizing those tools.

A recent review of the advancements in data compression in general can be found in Jayasankar, Thirumal, and Ponnurangam, 2021. Accounts of the research in more specific areas, such as video Laude et al., 2019 and image Rehman, Sharif, and Raza, 2014 were also produced recently.

Concerning lossless audio compression, research seems to be lagging when compared to lossy approaches; Hans and Schafer, 2001 suggested that a limit was about to be reached at the beginning of the 2000s.

The renaissance of neural networks led to interesting results, for example the use of Generative adversarial networks for audio compression presented in Huang, X. Wu, and Qu, 2018.

The ACER codec, as presented in Cunningham and Grout, 2014, offers an unconventional approach to compression, in that it identifies redundant pieces in a signal that are indexed in a dictionary. The results seem to be satisfactory, from a perceptual standpoint, when compared with MP3 and AAC codecs Cunningham and McGregor, 2019.

None of those approaches, however, are designed to take advantage of the redundancies present in harmonic sounds, nor were designed with the low latency requirements imposed by digital musical instruments, especially when operating in real-time; to fill that
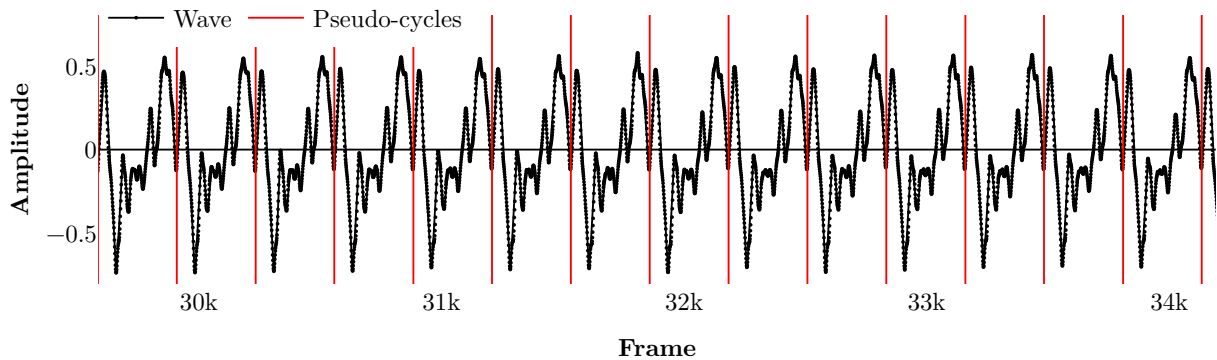
Figure 17: A discrete wave representing the sound of a bowed cello. The red lines represent the frontiers between successive pseudo cycles, as divided using the presented algorithm.

gap, we propose the method presented in this letter.

6.1 METHODOLOGY

A continuous or discrete signal can be interpreted as a succession of consecutive waveforms, each one representing a (pseudo) cycle of the wave. In the case of perfectly periodic signals, those waveforms are identical, and we need only the definition of one waveform to characterize the whole signal. In pseudo periodic waves, on the other hand, there are variations between pseudo cycles, that must be accounted for.

Figure 17 illustrates a possible division in pseudo cycles of a segment of the discrete representation of the sound of a bowed cello. It can be seen that the waveform variation between pseudo cycles is very small, on account of the sound being harmonic.

This rationale of defining a signal as a series of a modified basis function is presented more formally in Sinha and Tewfik, 1993; in that work, however, the bases are wavelets obtained via filter banks applied to the original signal using a pyramid algorithm. A similar approach can be found in Srinivasan and Jamieson, 1998, where a perceptual metric is used. In the presented work this basis is derived directly from the division in pseudo cycles of the underlying signal.

If the wave is sufficiently well-behaved, as is the case with the harmonic sounds used in sample-based digital musical instruments, the bulk of those changes can be attributed to variations in both the amplitude and length of the base waveform; The present algorithm is based on identifying the best basis waveform and registering how it changes over time.

We accomplish that encoding via an iterative algorithm, that divides the signal of interest using a waveform estimate. An updated basis waveform can then be obtained, and the process repeats while improvement in the mean squared error between the initial
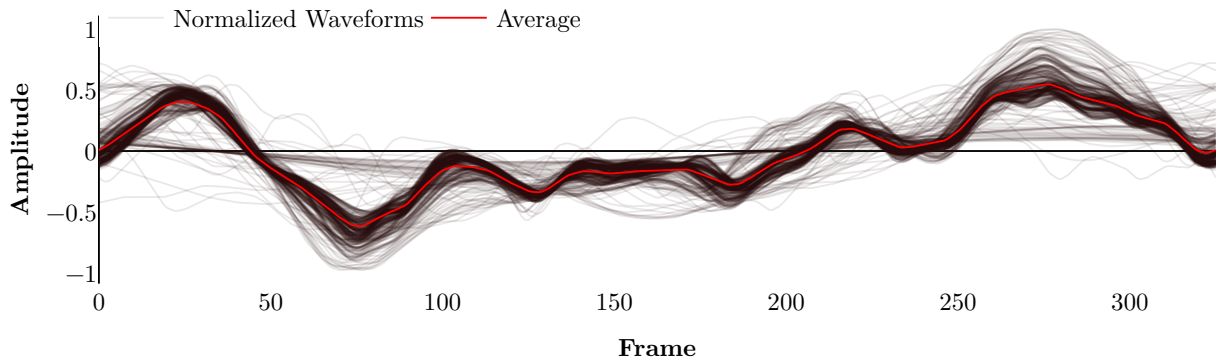
Figure 18: The various pseudo cycles in which the signal is divided are shown superimposed, after being scaled to the same length. The average of those individual waveforms is shown in red.
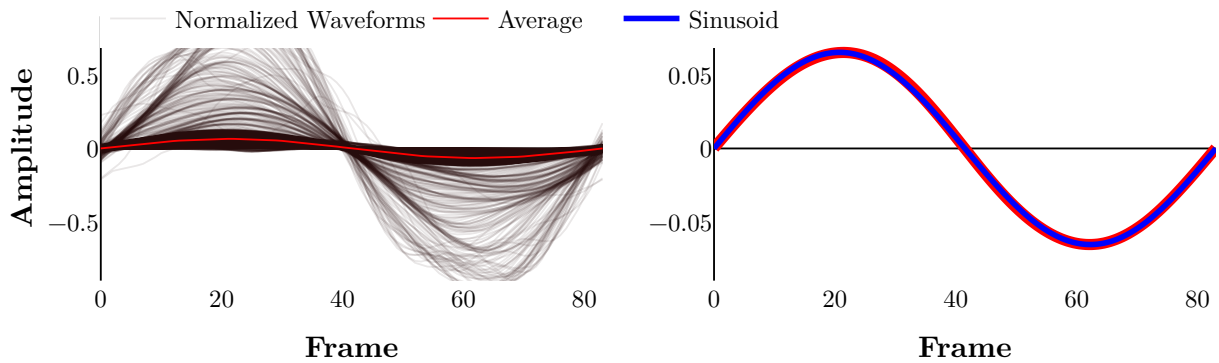

Figure 19: The various pseudo cycles in which the signal is divided are shown superimposed, after being scaled to the same length. The average of those waveforms is shown in red.

uncompressed and the compressed signal is being made.

Figure 18 shows how the base waveform, shown in red, for the cello sample; the base waveform is the average of each individual waveform, after being scaled to the same length. To ensure the continuity of the final wave, the average waveform is rotated between each iteration of the algorithm, so that the absolute vertical distance to zero of its first and last elements is the smallest possible.

For some signals, the average waveform resembles very closely a sinusoid, as can be seen in Fig. 19 that illustrates the average waveform in the case of the sound of a piano key. From a physical perspective, the strings in a piano are excited with an initial impulse left to vibrate undisturbed. After an initial transient period, it is natural that a harmonic mode becomes predominant.

In figure 20 the envelope and the normalized periods of the wave are shown. Using this data to modulate both the amplitude and the length of the average waveform presented in Fig. 18 above, one can approximate the original signal, as will be shown in the
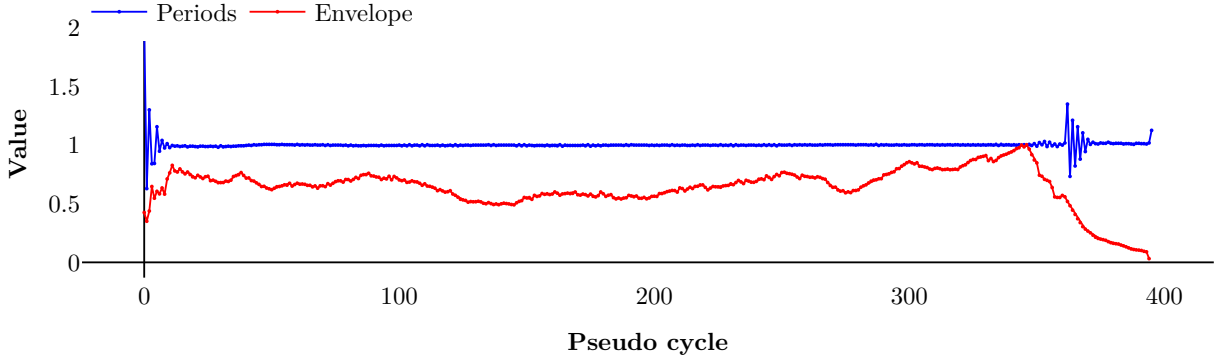
Figure 20: The envelope and the normalized length of each pseudo cycle.

rest of this chapter.

While the raw representation of the original cello signal can be viewed as an array with 129867 individual entries, for the compact representation we have 396 pairs of numbers representing the amplitude and length of each pseudo cycle, and the average waveform represented by an array of size 326. The length of the array storing the encoded representation, in the particular case of the cello sample, would be 1118, or approximately 0.86% of the original, uncompressed representation.

Starting with an initial estimate of the waveform of the signal being processed, one can identify the pseudo cycles of a signal by gradually multiplying, element-wise, different lengths of this estimated waveform with the signal, and observing the highest average value.

For the placement of the initial waveform, one has also to account for the potential phase shift in the signal. This is accomplished by convolving the different lengths of the waveform starting from different positions. For the subsequent placements, the initial position will be already fixed at the end of the preceding pseudo cycle, the length being the only quantity to be estimated.

Figure 21 shows how many times a period with a specific length occurs, for 4 different sound samples illustrating different settings. The first two samples are from instruments, a cello, and a piano, and the last two are voices, from a female soprano and a male bass.

The data in blue refers to the initial division of the waves, as obtained by the algorithm presented in Tarjano and Pereira, 2020. Since the algorithm was primarily designed to identify the envelope of a wave, the variation in the lengths of the pseudo cycles is naturally greater than the red data, where the final result after the application of the current method is shown.
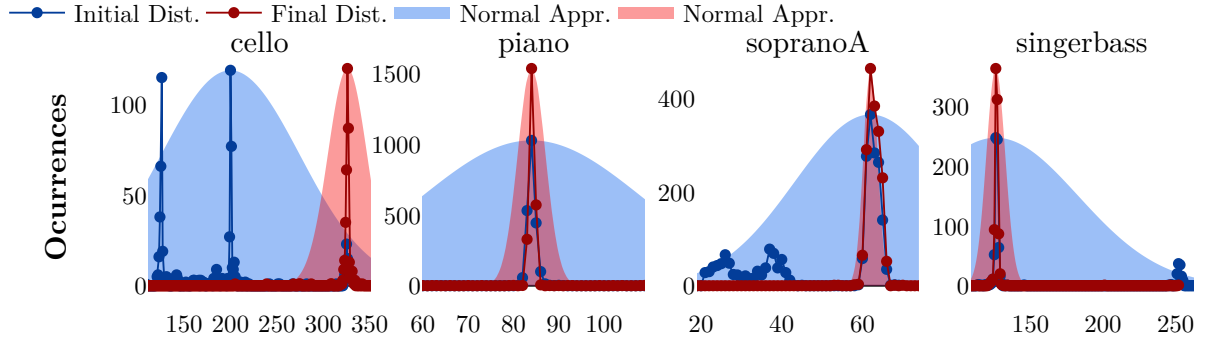
Figure 21: Distribution of the number of occurrences of periods lengths for 4 samples, both for the initial waveform in blue and the final waveform in black.
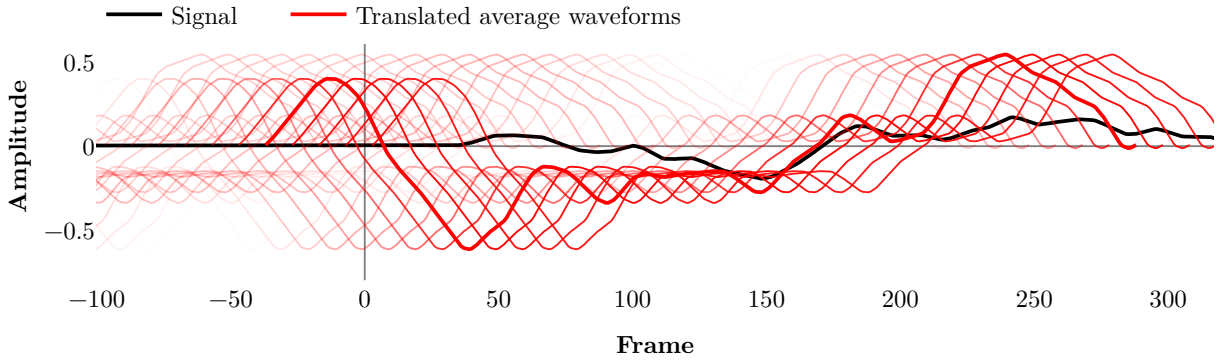

Figure 22: Identification of the first pseudo cycle. The intensity of the color is proportional to the average of the multiplication between the basis waveform and the underlying original signal. Only one waveform length is shown, translated in intervals of 10 frames.

The distribution of the periods' length tends to present a smaller variance, concentrating around the mode, as the algorithm progresses, and better basis waveforms are found. None of these distributions, as per the Shapiro test, are significantly close to a normal distribution, as can be observed comparing the distributions with the equivalent normal distributions plotted in the figure.

The normal distributions, as constructed with mean equal to the mode of the periods distribution and standard deviation equal to the standard variation around the mode tend to have a greater spread. Investigating the range between 3 standard deviations around the mode is a very conservative assumption, then, that we can afford to make without compromising the efficiency of the algorithm.

Fig. 22 illustrates, for the cello wave, the procedure for the identification of the first pseudo cycle of the wave, for a particular period. Although only one example is shown in the picture, the various lengths of the waveform are convolved with the beginning of the original signal, and the optimum chosen.

For the subsequent periods, the procedure is as illustrated in Fig. 23, as taking
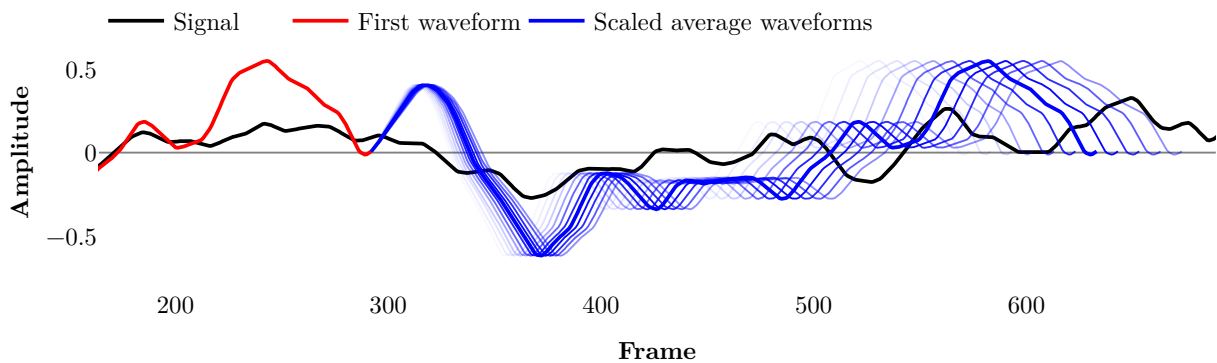
Figure 23: Identification of the second pseudo cycle, by multiplying different lengths of the basis waveform with the underlying original signal. The intensity of the color is proportional to the average of the multiplication.

the original cello wave as an example: only the length of the basis waveform is changed, using cubic interpolation. At the first iteration, the steps illustrated in both Fig. 22 and Fig. 23 are performed with the use of a sinusoid, to ensure the best initial division and faster convergence of the algorithm.

6.2 RESULTS

In this section, we compare the algorithm here presented with the most common lossy codecs in a variety of sound samples chosen to reflect the proposed immediate application of the algorithm, and thus consist of excerpts from notes from various instruments, as well as singing voice, as would be encountered in the libraries of sample-based instruments. Results can be heard on the website prepared for this letter Tarjano, 2021.

The samples from soprano, mezzo-soprano, and bass singers were obtained by processing sounds from the Vocalset libraryWilkins et al., 2018, and are licensed under the Creative Commons Attribution 4.0 International. The male singer sample was based on a sample from the Human Voice Dataset Vocobox, 2015, available under the MIT License.

The double bass, french horn, flute, and saxophone samples were based on samples freely available at the resources page of the Phillarmonia symphony orchestra website *Philharmonia Sound samples* 2021, while the piano, cello, and violin samples were processed from the basis samples offered in the University of Iowa Musical Instrument Samples (MIS) Fritts, 2021, and are also freely available.

All original samples are uncompressed audio encoded in the WAV format, at a standard frame rate of 44100 fps and a bit rate of 705 kbps, mono that were normalized

with long silences manually removed. The format was chosen for its popularity both in general Luo et al., 2014 and in sample-based digital instruments.

For the compressed formats, three lossy codecs were chosen. The first format we compare to is the popular MP3, the third layer of the MPEG-1/2 specification, proposed in 1992 Britanak and Rao, 2018 by the Moving Picture Experts Group.

The AAC format, in addition to being the formal mp3 successor, is used in services like Apple iTunes and YouTube Seichter, Cuccovillo, and Aichroth, 2016. It was developed by a joint effort between industry representatives such as Dolby and Sony and academic representation through Bell Labs and the University of Hannover, for example Britanak and Rao, 2018.

Outside the MPEG committee, the open-source Opus format, standardized by the Internet Engineering Task Force (IETF) in 2012, was chosen. Besides being considered superior to its predecessor Vorbis and the AAC format Gunawan and Kartiwi, 2018, the codec supports both speech and music, has very low delay **2013Valinhigh**, being used in real-time audio applications such as WhatsApp Srivastava et al., 2019.

The Opus codec, along with ACC and MP3, forms the set of the most advanced Laude et al., 2019 and popular Kwan and Luk, 2018 lossy compression formats; Readily available implementations of more experimental algorithms, such as the aforementioned ACER codec Cunningham and Grout, 2014, or the wavelet-based approaches presented in Sinha and Tewfik, 1993 and Srinivasan and Jamieson, 1998 couldn't be found.

All conversions, except for the MP3 encoding done with the Lame library, were made with the FFmpeg library. The PowerShell scripts used are available in the letter's repository Tarjano, 2021. The encoding configurations were chosen in order to generate files with sizes compatible with the present algorithm: for the MP3 format, a bit rate of 8 kb/s and a frame rate of 8000 frames/s was chosen. For the other two formats, no frame rate reduction was needed, and the bit rates were 4 kb/s for the ACC format and 6 kb/s for the Opus format.

As can be seen in the accompanying website, the present work exhibits an average compression rate of 1.09%, slightly larger than the 0.92% obtained with the Opus codec, and better than the AAC (1.19%) and MP3 (1.20%) codecs.

6.3   DECODING TIMES

Encoding times, as can be seen in the accompanying website, are around two times the duration of the original signal, and thus considerably higher than conventional codecs:

Table 3: Original Duration and Decoding Times (milliseconds) for the Present Work and the MP3, ACC, and Opus lossy formats.

Sample	Duration	P.W.	MP3	ACC	Opus
cello	2945	11	105	111	136
doublebass	1388	4	103	201	108
flute	11115	17	110	125	117
frenchhorn	8713	21	211	226	223
mezzosoprano	3259	9	163	147	116
piano	4729	19	728	506	520
saxophone	19825	33	109	115	119
singerbass	2624	8	123	198	197
singermale	1214	3	104	199	106
sopranoA	2598	8	104	120	106
sopranoB	2299	5	115	106	106
violin	2271	6	111	105	108

While part of this discrepancy can be attributed to the immaturity of the implementation when compared to well-established ones, like lame for mp3, with decades of refinement, the majority of the difference can be attributed to the different approach used by the presented algorithm, engaging in a semi exhaustive search for the best basis purely in the time domain.

This approach pays off during the decoding phase, however, since the representation doesn't need to be translated from the frequency domain, for example. In the context of sample-based digital musical instruments, the encoding occurs only once, and takes place in more robust hardware; it is more important to alleviate as much as possible the decoding effort, since, besides enabling the use of more affordable hardware, this task will be performed a far greater number of times.

For digital musical instruments, it is important to maintain latency below 10 milliseconds, since that is the limit above which users start to experiment performance degradation Wessel and Wright, 2002; Jack, Stockman, and McPherson, 2016. Table 3 compares the absolute decoding times, in milliseconds; it can be seen that the proposed algorithm presents the lowest decoding times, with an average of 12 ms against an average of 163.5 ms of the Opus codec, the second-lowest. Besides, as the reconstruction of the sound is made linearly, pseudo cycle by pseudo cycle, entirely in the time domain, it would be possible to alternate between decoding and reproduction tasks, keeping the latency below the acceptable limit for samples of arbitrary length.

Table 4: Ranking of the codecs compared under different quality metrics.

Method	Present Work	MP3	ACC	Opus
MSE	3	1	4	2
ViSQOL audio	4	1	2	3
ViSQOL speech	2	1	4	3

6.4 QUALITY

To assess the overall quality of the proposed algorithm, in addition to the computation of the mean square error in relation to the original signal, we made use of the ViSQOL Chinen et al., 2020 open-source library developed by Google, designed to substitute subjective listening tests when those are unpractical. According to Sloan et al., 2017, ViSQOL is a more accurate alternative to well-known methods, like PEAQ and POLQA.

Table 4 shows the ranking of the compressed formats in those metrics: the MP3 codec performed best in all of them; to reach the compatible size with the other codecs, however, it was the only one that needed to be downsampled.

Although the ViSQOL is designed as a cost-efficient alternative to subjective listening tests, it by any means claims to be a perfect replacement; The reader is strongly encouraged to hear the results in the samples section of the accompanying website Tarjano, 2021.

Despite performing better than the AAC codec in the objective measures, files compressed with the Opus codec present artifacts, in the majority of the samples, that renders them almost unintelligible. To a lesser extent, that is also the case of the AAC codec.

While presenting an efficient algorithm to compress harmonic signals, with quality on par with the MP3, AAC, and Opus codecs and faster decoding times, immediately applicable to digital musical instruments, this work introduces a new framework for representing pseudo periodic waves, that can serve as a basis for future research. Fig. 24 illustrates this representation, in the case of the cello sample.

Starting from the representation in Fig. 24, the surface can be approximated by parametric surfaces, or machine learning approaches, providing more robustness in relation to the average waveform approach, for example.
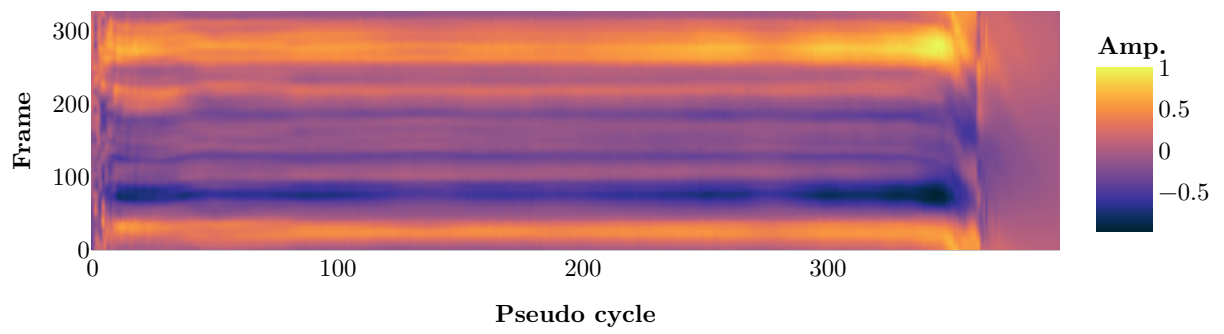
Figure 24: The evolution of the cello sample waveform over time.

REFERENCES

Assef, Amauri Amorin et al. (Jan. 2018). "Modeling and FPGA-based implementation of an efficient and simple envelope detector using a Hilbert Transform FIR filter for ultrasound imaging applications". In: *Research on Biomedical Engineering* 34.1, pp. 87–92. DOI: 10.1590/2446-4740.02417.

Berg, Mark de et al. (2008). *Computational Geometry.* Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-77974-2.

Britanak, Vladimir and K. R. Rao (2018). "Audio Coding Standards, (Proprietary) Audio Compression Algorithms, and Broadcasting/Speech/Data Communication Codecs: Overview of Adopted Filter Banks". In: *Cosine-/Sine-Modulated Filter Banks.* Cham: Springer International Publishing, pp. 13–37. ISBN: 978-3-319-61078-8 978-3-319-61080-1. DOI: 10.1007/978-3-319-61080-1_2. URL: http://link.springer.com/10.1007/978-3-319-61080-1_2 (visited on 03/03/2021).

Butterworth, Stephen et al. (1930). "On the theory of filter amplifiers". In: *Wireless Engineer* 7.6, pp. 536–541.

Caetano, Marcelo and Xavier Rodet (May 2011). "Improved estimation of the amplitude envelope of time-domain signals using true envelope cepstral smoothing". In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. DOI: 10.1109/icassp.2011.5947290.

Calhoun, Jon et al. (Mar. 2019). "Exploring the feasibility of lossy compression for PDE simulations". In: *The International Journal of High Performance Computing Applications* 33.2, pp. 397–410. ISSN: 1094-3420, 1741-2846. DOI: 10.1177/1094342018762036. URL: http://journals.sagepub.com/doi/10.1177/1094342018762036 (visited on 03/03/2021).

Cánovas, Rodrigo, Alistair Moffat, and Andrew Turpin (Aug. 1, 2014). "Lossy compression of quality scores in genomic data". In: *Bioinformatics* 30.15, pp. 2130–2136. ISSN: 1460-2059, 1367-4803. DOI: 10.1093/bioinformatics/btu183. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btu183 (visited on 03/03/2021).

Carroll, Daniel et al. (Oct. 2014). "A Survey of the Differential Geometry of Discrete Curves". In: *The Mathematical Intelligencer* 36.4, pp. 28–35. DOI: 10.1007/s00283-014-9472-2.

Chen, Wenlin et al. (Aug. 13, 2016). "Compressing Convolutional Neural Networks in the Frequency Domain". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco California USA: ACM, pp. 1475–1484. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939839. URL: https://dl.acm.org/doi/10.1145/2939672.2939839 (visited on 03/03/2021).

Chinen, Michael et al. (Apr. 20, 2020). "ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric". In: *arXiv:2004.09584 [cs, eess].* arXiv: 2004.09584. URL: http://arxiv.org/abs/2004.09584 (visited on 03/17/2021).

Coeurjolly, David, Serge Miguet, and Laure Tougne (2001). "Discrete Curvature Based on Osculating Circle Estimation". In: *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, pp. 303–312. DOI: 10.1007/3-540-45129-3_27.

Cunningham, Stuart and Vic Grout (Oct. 2014). "Data reduction of audio by exploiting musical repetition". In: *Multimedia Tools and Applications* 72.3, pp. 2299–2320. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-013-1504-y. URL: http://link.springer.com/10.1007/s11042-013-1504-y (visited on 03/03/2021).

Cunningham, Stuart and Iain McGregor (July 11, 2019). "Subjective Evaluation of Music Compressed with the ACER Codec Compared to AAC, MP3, and Uncompressed PCM". In: *International Journal of Digital Multimedia Broadcasting* 2019, pp. 1–16. ISSN: 1687-7578, 1687-7586. DOI: 10.1155/2019/8265301. URL: https://www.hindawi.com/journals/ijdmb/2019/8265301/ (visited on 03/03/2021).

Dau, Torsten (2012). *Speech perception and auditory disorders.* Ballerup: Danavox Jubilee Foundation. ISBN: 9788799001330.

Division, National Communications System (U.S.). Technology & Standards and United States. General Services Administration. Information Technology Service (1996). *Telecommunications: Glossary of Telecommunication Terms.* Federal standard. General Services Administration, Information Technology Service. URL: https://books.google.com.br/books?id=RIrPtQEACAAJ.

Donovan, Tristan (2010). *Replay: the history of video games.* East Sussex, England: Yellow Ant. ISBN: 9780956507228.

Edelsbrunner, H., D. Kirkpatrick, and R. Seidel (July 1983). "On the shape of a set of points in the plane". In: *IEEE Transactions on Information Theory* 29.4, pp. 551–559. DOI: 10.1109/tit.1983.1056714.

Edelsbrunner, Herbert and Ernst P. Mücke (Jan. 1994). "Three-dimensional alpha shapes". In: *ACM Transactions on Graphics (TOG)* 13.1, pp. 43–72. DOI: `10.1145/174462.156635`.

Fleischmann, Oliver, Lennart Wietzke, and Gerald Sommer (2010). "A Novel Curvature Estimator for Digital Curves and Images". In: *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, pp. 442–451. DOI: `10.1007/978-3-642-15986-2_45`.

Fritts, Lawrence (2021). *University of Iowa Electronic Music Studios.* URL: `http://theremin.music.uiowa.edu/MIS.html` (visited on 03/08/2021).

Gabor, D. (Nov. 1946). "Theory of communication. Part 1: The analysis of information". In: *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering* 93.26, pp. 429–441. DOI: `10.1049/ji-3-2.1946.0074`.

Gunawan, Teddy Surya and Mira Kartiwi (Apr. 1, 2018). "Performance Evaluation of Multichannel Audio Compression". In: *Indonesian Journal of Electrical Engineering and Computer Science* 10.1, p. 146. ISSN: 2502-4760, 2502-4752. DOI: `10.11591/ijeecs.v10.i1.pp146-153`. URL: `http://ijeecs.iaescore.com/index.php/IJEECS/article/view/10871` (visited on 03/03/2021).

Hahn, Stefan L. (2007). "The History of Applications of Analytic Signals in Electrical and Radio Engineering". In: *EUROCON 2007 - The International Conference on "Computer as a Tool"*. IEEE. DOI: `10.1109/eurcon.2007.4400463`.

Hans, M. and R.W. Schafer (July 2001). "Lossless compression of digital audio". In: *IEEE Signal Processing Magazine* 18.4, pp. 21–32. ISSN: 10535888. DOI: `10.1109/79.939834`. URL: `http://ieeexplore.ieee.org/document/939834/` (visited on 03/03/2021).

He, Lei and Volker Dellwo (Sept. 2016). "A Praat-Based Algorithm to Extract the Amplitude Envelope and Temporal Fine Structure Using the Hilbert Transform". In: *Interspeech 2016*. ISCA. DOI: `10.21437/interspeech.2016-1447`.

Hu, Chenhao et al. (Feb. 19, 2021). "Audio object coding based on N-step residual compensating". In: *Multimedia Tools and Applications.* ISSN: 1380-7501, 1573-7721. DOI: `10.1007/s11042-020-10339-0`. URL: `http://link.springer.com/10.1007/s11042-020-10339-0` (visited on 03/03/2021).

Huang, Qingbo, Xihong Wu, and Tianshu Qu (2018). "Bandwidth extension method based on generative adversarial nets for audio compression". In: *AES 144th Convention, Milan, Italy, 2018 May 23–26*, p. 8.

Jack, Robert H., Tony Stockman, and Andrew McPherson (Oct. 4, 2016). "Effect of latency on performer interaction and subjective quality assessment of a digital musical

instrument". In: *Proceedings of the Audio Mostly 2016*. AM '16: Audio Mostly 2016. Norrköping Sweden: ACM, pp. 116–123. ISBN: 978-1-4503-4822-5. DOI: `10.1145/2986416.2986428`. URL: `https://dl.acm.org/doi/10.1145/2986416.2986428` (visited on 03/16/2021).

Jayasankar, Uthayakumar, Vengattaraman Thirumal, and Dhavachelvan Ponnurangam (Feb. 2021). "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications". In: *Journal of King Saud University - Computer and Information Sciences* 33.2, pp. 119–140. ISSN: 13191578. DOI: `10.1016/j.jksuci.2018.05.006`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1319157818301101` (visited on 03/03/2021).

Kwan, Chiman and Yvonne Luk (Mar. 2018). "Objective Performance Evaluation of Several State-of-the-Art Audio Codecs". In: *2018 Data Compression Conference*. 2018 Data Compression Conference (DCC). Snowbird, UT: IEEE, pp. 418–418. ISBN: 978-1-5386-4883-4. DOI: `10.1109/DCC.2018.00071`. URL: `https://ieeexplore.ieee.org/document/8416635/` (visited on 03/03/2021).

Laude, Thorsten et al. (2019). "A Comprehensive Video Codec Comparison". In: *APSIPA Transactions on Signal and Information Processing* 8, e30. ISSN: 2048-7703. DOI: `10.1017/ATSIP.2019.23`. URL: `https://www.cambridge.org/core/product/identifier/S2048770319000234/type/journal_article` (visited on 03/03/2021).

Lawson, Charles L. and Richard J. Hanson (Jan. 1995). *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics. DOI: `10.1137/1.9781611971217`.

Lokki, Tapio et al. (June 2011). "Engaging concert hall acoustics is made up of temporal envelope preserving reflections". In: *The Journal of the Acoustical Society of America* 129.6, EL223–EL228. DOI: `10.1121/1.3579145`.

Loughlin, Patrick J. and Berkant Tacer (Sept. 1996). "On the amplitude- and frequency-modulation decomposition of signals". In: *The Journal of the Acoustical Society of America* 100.3, pp. 1594–1601. DOI: `10.1121/1.416061`.

Luo, Da et al. (Apr. 2014). "Identifying Compression History of Wave Audio and Its Applications". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 10.3, pp. 1–19. ISSN: 1551-6857, 1551-6865. DOI: `10.1145/2575978`. URL: `https://dl.acm.org/doi/10.1145/2575978` (visited on 03/07/2021).

Lyons, Richard (July 2017). "Digital Envelope Detection: The Good, the Bad, and the Ugly [Tips and Tricks]". In: *IEEE Signal Processing Magazine* 34.4, pp. 183–187. DOI: `10.1109/msp.2017.2690438`.

Meng, Qinglin et al. (Dec. 2013). "An empirical envelope estimation algorithm". In: *2013 6th International Congress on Image and Signal Processing (CISP)*. IEEE. DOI: `10.1109/cisp.2013.6745226`.

O'Grady, Pat (Dec. 9, 2020). "Rethinking criticism about lossy compression: Sound fidelity, large-scale production and audio capital in pop music". In: *Convergence: The International Journal of Research into New Media Technologies*, p. 135485652097645. ISSN: 1354-8565, 1748-7382. DOI: `10.1177/1354856520976454`. URL: `http://journals.sagepub.com/doi/10.1177/1354856520976454` (visited on 03/03/2021).

Philharmonia Sound samples (2021). en-GB. URL: `https://philharmonia.co.uk/resources/sound-samples/` (visited on 03/08/2021).

Qi, Beier et al. (May 2017). "Relative contributions of acoustic temporal fine structure and envelope cues for lexical tone perception in noise". In: *The Journal of the Acoustical Society of America* 141.5, pp. 3022–3029. DOI: `10.1121/1.4982247`.

Rehman, Mehwish, Muhammad Sharif, and Mudassar Raza (Jan. 27, 2014). "Image Compression: A Survey". In: *Research Journal of Applied Sciences, Engineering and Technology* 7.4, pp. 656–672. ISSN: 20407459, 20407467. DOI: `10.19026/rjaset.7.303`. URL: `http://maxwellsci.com/jp/mspabstract.php?jid=RJASET&doi=rjaset.7.303` (visited on 03/03/2021).

Savitzky, Abraham. and M. J. E. Golay (July 1964). "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." In: *Analytical Chemistry* 36.8, pp. 1627–1639. DOI: `10.1021/ac60214a047`.

Seichter, Daniel, Luca Cuccovillo, and Patrick Aichroth (Mar. 2016). "AAC encoding detection and bitrate estimation using a convolutional neural network". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Shanghai: IEEE, pp. 2069–2073. ISBN: 978-1-4799-9988-0. DOI: `10.1109/ICASSP.2016.7472041`. URL: `http://ieeexplore.ieee.org/document/7472041/` (visited on 03/03/2021).

Selesnick, Ivan (Apr. 17, 2013). *Least Squares with Examples in Signal Processing*. OpenStax CNX. URL: `http://cnx.org/contents/5d13ca71-5821-41d6-a40a-d5278652fc11@1`.

Shannon, R. V. et al. (Oct. 1995). "Speech Recognition with Primarily Temporal Cues". In: *Science* 270.5234, pp. 303–304. DOI: `10.1126/science.270.5234.303`.

Sinha, D. and A.H. Tewfik (Dec. 1993). "Low bit rate transparent audio compression using adapted wavelets". In: *IEEE Transactions on Signal Processing* 41.12, pp. 3463–3479. ISSN: 1053587X. DOI: 10.1109/78.258086. URL: http://ieeexplore.ieee.org/document/258086/ (visited on 03/03/2021).

Sloan, Colm et al. (Dec. 2017). "Objective Assessment of Perceptual Audio Quality Using ViSQOLAudio". In: *IEEE Transactions on Broadcasting* 63.4, pp. 693–705. ISSN: 0018-9316, 1557-9611. DOI: 10.1109/TBC.2017.2704421. URL: http://ieeexplore.ieee.org/document/7940042/ (visited on 03/17/2021).

Srinivasan, P. and L.H. Jamieson (Apr. 1998). "High-quality audio compression using an adaptive wavelet packet decomposition and psychoacoustic modeling". In: *IEEE Transactions on Signal Processing* 46.4, pp. 1085–1093. ISSN: 1053587X. DOI: 10.1109/78.668558. URL: http://ieeexplore.ieee.org/document/668558/ (visited on 03/03/2021).

Srivastava, Saurabh Ranjan et al. (Mar. 25, 2019). "Smartphone Triggered Security Challenges - Issues, Case Studies and Prevention". In: *Cyber Security in Parallel and Distributed Computing.* Ed. by DacNhuong Le et al. Hoboken, NJ, USA: John Wiley & Sons, Inc., pp. 187–206. ISBN: 978-1-119-48833-0 978-1-119-48805-7. DOI: 10.1002/9781119488330.ch12. URL: http://doi.wiley.com/10.1002/9781119488330.ch12 (visited on 03/08/2021).

Stephen Boyd, Lieven Vandenberghe (June 7, 2018). *Introduction to Applied Linear Algebra.* Cambridge University Pr. 500 pp. ISBN: 1316518965. URL: https://www.ebook.de/de/product/32105301/stephen_boyd_lieven_vandenberghe_introduction_to_applied_linear_algebra.html.

Tarjano, Carlos (Mar. 2020). *Envelope Visualizations.* URL: https://envelope.netlify.app/ (visited on 12/03/2020).

— (2021). *Compression GitHub repository.* DOI: 10.5281/ZENODO.4618033. URL: https://github.com/tesserato/compression.

Tarjano, Carlos and Valdecy Pereira (2019). "Neuro-Spectral Audio Synthesis: Exploiting Characteristics of the Discrete Fourier Transform in the Real-Time Simulation of Musical Instruments Using Parallel Neural Networks". In: *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series.* Springer International Publishing, pp. 362–375. DOI: 10.1007/978-3-030-30490-4_30.

— (2020). *Digital Envelope Estimation Via Geometric Properties of an Arbitrary Real Signal.* arXiv: 2009.02860 [cs.SD].

Turner, Richard E. and Maneesh Sahani (Nov. 2011). "Demodulation as Probabilistic Inference". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.8, pp. 2398–2411. DOI: 10.1109/tasl.2011.2135852.

Varytimidis, Christos et al. (Feb. 2016). "Alpha-shapes for local feature detection". In: *Pattern Recognition* 50, pp. 56–73. DOI: 10.1016/j.patcog.2015.08.016.

Váša, L. et al. (Aug. 2016). "Mesh Statistics for Robust Curvature Estimation". In: *Computer Graphics Forum* 35.5, pp. 271–280. DOI: 10.1111/cgf.12982.

Váša, Libor, Tom Kühnert, and Guido Brunnett (June 2016). "Multivariate analysis of curvature estimators". In: *Computer-Aided Design and Applications* 14.1, pp. 58–69. DOI: 10.1080/16864360.2016.1199756.

Vocobox (2015). *Human Voice Dataset.* original-date: 2014-12-27T21:21:24Z. URL: https://github.com/vocobox/human-voice-dataset (visited on 03/08/2021).

Wessel, David and Matthew Wright (Sept. 2002). "Problems and Prospects for Intimate Musical Control of Computers". In: *Computer Music Journal* 26.3, pp. 11–22. ISSN: 0148-9267, 1531-5169. DOI: 10.1162/014892602320582945. URL: https://direct.mit.edu/comj/article/26/3/11-22/94758 (visited on 03/16/2021).

Wilkins, Julia et al. (Mar. 8, 2018). "Vocalset: A Singing Voice Dataset". In: *19th International Society for Music Information Retrieval Conference, Paris,France, 2018.* type: dataset. Zenodo. DOI: 10.5281/ZENODO.1203819. URL: https://zenodo.org/record/1203819 (visited on 03/03/2021).

Wu, Bin et al. (Dec. 2015). "Automated extraction of ground surface along urban roads from mobile laser scanning point clouds". In: *Remote Sensing Letters* 7.2, pp. 170–179. DOI: 10.1080/2150704x.2015.1117156.

Xu, Xin et al. (May 2019). "Modeling the Echelle Spectra Continuum with Alpha Shapes and Local Regression Fitting". In: *The Astronomical Journal* 157.6, p. 243. DOI: 10.3847/1538-3881/ab1b47.

Yang, Cong, Marcin Grzegorzek, and Ewa Lukasik (Sept. 2015). "Representing the evolving temporal envelope of musical instruments sounds using Computer Vision methods". In: *2015 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE. DOI: 10.1109/spa.2015.7365137.

Yang, Cong, Oliver Tiebe, et al. (Nov. 2015). "Skeleton-based audio envelope shape analysis". In: *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE. DOI: 10.1109/acpr.2015.7486556.

Yang, Wei, Yuanhong Yang, and Mingwei Yang (Apr. 2014). "Fast digital envelope detector based on generalized harmonic wavelet transform for BOTDR performance improvement". In: *Measurement Science and Technology* 25.6, p. 065103. DOI: `10.1088/0957-0233/25/6/065103`.

Zhu, Zhi et al. (May 2018). "Contributions of temporal cue on the perception of speaker individuality and vocal emotion for noise-vocoded speech". In: *Acoustical Science and Technology* 39.3, pp. 234–242. DOI: `10.1250/ast.39.234`.

Zordan, Davide et al. (Nov. 7, 2014). "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking". In: *ACM Transactions on Sensor Networks* 11.1, pp. 1–34. ISSN: 1550-4859, 1550-4867. DOI: `10.1145/2629660`. URL: `https://dl.acm.org/doi/10.1145/2629660` (visited on 03/03/2021).

GLOSSARY

CLMS Constrained Least Mean Squares. 10

ENIAC Electronic Numeric Integrator and Calculator. 4

FPS frames per second. 9

LMS Least Mean Squares. 9, 10