TOPOLOGIZING SOUND SYNTHESIS VIA SHEAVES

Georg Essl

Department of Mathematical Sciences
University of Wisconsin - Milwaukee
Milwaukee, U.S.A.
essl@uwm.edu

ABSTRACT

In recent years, a range of topological methods have emerged for processing digital signals. In this paper we show how the construction of topological filters via sheaves can be used to *topologize* existing sound synthesis methods. I illustrate this process on two classes of synthesis approaches: (1) based on linear-time invariant digital filters and (2) based on oscillators defined on a circle. We use the computationally-friendly approach to modeling topologies via a simplicial complex, and we attach our classical synthesis methods to them via sheaves. In particular, we explore examples of simplicial topologies that mimic sampled lines and loops. Over these spaces we realize concrete examples of simple discrete harmonic oscillators (resonant filters), and simple comb filter based algorithms (such as Karplus-Strong) as well as frequency modulation.

1. INTRODUCTION

Topology is the mathematical description of connectivity. It disregards information such as distances or angles. Thus, topological constructions make weaker assumptions than geometry and provide broader applicability. In recent years, topological ideas have increasingly found their way into applications.

In this paper, we explore how topological spaces can be made an explicit part of the construction of sound synthesis methods based on two building blocks: (1) Simplicial topological spaces and (2) sheaves. Simplicial topological spaces have the advantage of being suitable for computer implementation. While the full theory of sheaves can be daunting [1], sheaves over simplicial topologicies, with some further requirements, become much more straightforward to understand and use. Using these two constructions, we employ so-called *topological filters* as proposed by Robinson [2, 3]. Topological filters are very general in principle, and largely prescribe a decomposition of input, state, and output and their related computation over a topological space, which in our case relates to temporal dynamics. This means that one can envision that most, perhaps all, existing sound synthesis methods can be realized in this framework.

In order to illustrate how existing synthesis methods can be related to a topological space, or *topologized*, we give explicit constructions of two classes of sound synthesis methods in this paper. The first class consists of methods based on linear time-invariant digital filters [4, 5]. The realization of both feedforward (all-zero) filters [2, 3] and general pole-zero filters [6] have been derived theoretically. This paper gives explicit demonstrations of these filters by realizing a classical IIR resonance filter and the Karplus-Strong plucked string algorithm in this formalism. These have been chosen as they are closely related to widely used types of synthesis methods based on filters. Resonant filters are elementary building blocks in modal synthesis [7, 8, 9]. The Karplus-Strong plucked string algorithm is constructed from a comb-filter with a low-order loop filter [10]. It is a simple prototype of a linear time-invariant filter-based physical model that can be enriched to arrive at physical models based on digital waveguides [11]. The second class of synthesis methods realized over topologies are oscillatory synthesis methods. It has recently been shown that numerous oscillatory synthesis methods can be subsumed under the general formulations of maps from the circle to itself [12]. We specifically use frequency modulation [13] as a particularly popular, yet rich oscillatory prototype of this class of synthesis methods.

The relationship of topological space to audio samples, the ultimate outcome of sound synthesis methods, is a rich playing field that involves choices of topological spaces, metric information, and geometric manipulations such as projections. A number of concrete constructions illustrate these concepts in the topologized sound synthesis computation.

The goal of the paper is to have concrete, simple, yet sufficiently rich examples to serve as a template for similar constructions for other existing sound synthesis methods, or how novel methods might be constructed using this approach. Hence, these examples seek to illustrate the general strategy of "topologizing" synthesis algorithms using simplicial topologies and sheaves.

1.1. Related Work

While the development of topology is predominantly in the realm of pure mathematics, topological ideas have increasingly been found useful in applied domains over the last two decades through the emergence of the field of computational topology [14], applied topology [15], and, most recently, the direct application of topological constructions and ideas to data analysis [16] and signal processing [3, 17, 18].

Of particular relevance is recent work generalizing linear-time invariant FIR filters [2, 3] as well as IIR filters [6] to topological filters over *sheaves*. The basic intuition of a sheaf is the ability to connect data locally to a topological space, while retaining consistency of data during traversal of the topological space.

Basic topological notions have long been present in digital filter theory, primarily with respect to so-called "filter topologies" and have been an important facet of comparing realizations of digital filters (see for example [19]). Furthermore, certain topological properties have been used to inform sound synthesis methods. Trautmann published two Waveguide constructions of flat

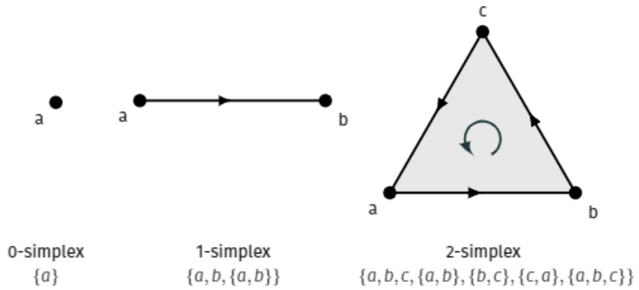

Figure 1: Low-dimensional simplices and their combinatorial description as simplicial sets.

metric Möbius bands as string models via twisted boundary conditions [20]. Topology plays an important role in understanding nonlinear problems as it often allows for the characterization of desirable properties even when a method for precisely solving the problem is unknown. The reason why circle maps are desirable one-dimensional iterative maps for chaotic sound synthesis is that their circular topology guarantees stability of the dynamics [21]. Many classical oscillatory synthesis methods can be reinterpreted as dynamics on a circular topology [12]. Furthermore, studying the topological structure of synthesis algorithms allows us to generalize results beyond specific metric settings as is the case in the work of classifying excitation locations across standard and banded waveguides in one and two dimensions [22]. Topological foundations have also played an important role in computational and mathematical musicology [23]. Munkres [24] provides an accessible introduction to topology.

2. SIMPLICIAL TOPOLOGY

At the most basic level, the study of topology investigates connectivity patterns. General topology can be hard to model explicitly algorithmically. However, there is a powerful yet very computational description of topology that allows us to formulate topological spaces combinatorially, which can be implemented in a straightforward fashion. These topological spaces are called simplicial topologies [14, 15], and use simplices as their basic building blocks. Our exposition here is closely related to [3, 6]. Low-order simplices are depicted in Figure 1. A 0-simplex can be thought of as a topological point. A 1-simplex is a topological line segment terminated by two points (0-simplices). n-simplices can be thought of as modeling different dimensional objects topologically. The next higher dimension would be that of an area. The 2-simplex can be thought of as a topological area bounded by three 1-simplices as its faces. This process can be continued to arbitrary higher dimensions and in each case encodes that a $n+1$-dimensional (hyper)-volume is filled in between surrounding faces of n-dimensional (hyper)-volumes. Simplicial complexes are constructed by gluing together n-simplicies from shared points, edges, areas and so forth. This paper will focus on a subset of simplicial complexes: path constructions realized as line- or loop-complexes.

An extended line containing multiple points can be constructed by gluing a number of 1-simplices together at a shared 0-simplex. If there is no branching, this simplicial complex is called a *line complex*[1]. This construction looks quite analogous to a sampled version of the real line $\mathbb{R}$, except that there is none of the metric structure of $\mathbb{R}$. Throughout this paper, 0-simplices are interpreted as topological sample points and 1-simplices define their respective connectivity and sequential order. In fact, the sampled real line is an example of a line complex, endowed with additional information (such as the metric structure of $\mathbb{R}$). Any irregularly sampled real line is an example of the line complex. Thus, all sampled real lines share the same topological structure. The following two operations allow one to relate n-simplices to each other.

Definition 2.1. A **boundary operation** b of an n-simplex $\mathcal{X}_n$ returns an ordered set of all n-1-simplices that constitute its boundary. A 0-simplex returns the empty set. A **face operation** f of an n-simplex returns an ordered set of all $n+1$-simplices of which it is a boundary. We call two simplices **directly connected** if they are relatable through one face or boundary operation.

These two operations are sufficient to traverse across neighboring simplicies, keeping in mind the simplex already visited and the ones yet to be visited, or having some local notion of orientation.

2.1. Sheaves: Associating Data to a Topology

The traditional setting for signal processing of audio follows this basic model: Time and amplitude are modeled as Cartesian product yielding a Eucledian plane $\mathbb{R}^2$ in which time-parametrized functions of amplitude are considered. Furthermore, time is discretized, yielding a sampled version of the time axis.

The sampled time line can be topologized by replacing it with a line complex with the 0-simplicies corresponding to sampling points. This construction is indeed a topological version of the traditional time line, except that the distance between sample points is no longer considered up front (though it may be reintroduced later). Next the amplitude component as well as computational relations between them are topologized.

Attaching data to the line complex can be achieved by a mechanism called *sheaves*. Sheaf theory [26] was developed in the 1940s for this purpose, though the original applications of the theory fall within the purely mathematical study of topology and geometry. The construction of attaching data to cellular or simplicial structures emerged later [27], with applications emerging only recently [28, 15, 3].

It is rather common to consider data in association with some topological structure, including objects such as colored or labeled graphs. Sheaves are a formal process of associating not only data, but also functions, to a topological space. Within category theory, this notion is called a *functor* [15]. Still, category theory is not required to understand our constructions. Therefore we limit ourselves to studying maps. For treatments using category theory more explicitly, see [3, 15, 28]. We use the following definition definition of a sheaf [6]:

Definition 2.2. A **sheaf** of data $\mathcal{D}$ of a simplicial complex $\mathcal{S}$ each indexed by $i \in \mathcal{I}$ satisfying

1. Each Data $\mathcal{D}_i$ is attached to each n-simplex $\mathcal{S}_i$.
2. Local data $\mathcal{D}_i$ is unique. We call this the **consistency condition**.
3. If two simplices $\mathcal{S}_i, \mathcal{S}_j$ are *directly connected*, then there exists at least one mapping between $\mathcal{D}_i$ and $\mathcal{D}_j$, called the **consistency map**.

[1]More general simplicial complexes which are limited to 0 and 1-simplices are related to signal processing over graphs [25] which has been recently extended to processing over simplicial complexes [17, 18].

Despite already being somewhat tailored to our needs, this is still a rather general definition. Note that any simplicial complex is permissible and there are only some general requirements for the nature of the data and maps between them. The main difference between our definition and other proposals [15, 3] is that our definition fixes consistency locally. We are interested in constructing synthesis methods and digital filters, rather than in analyzing inconsistency in given data (as is done in [29]), hence it is appropriate to impose local consistency on all our constructions. This behavior is implied in traditional digital filters.

3. TOPOLOGIZING SOUND SYNTHESIS VIA SHEAVES

A general template exists shared by all sound synthesis methods topologized in this paper, which can be used to formulate others as well. This construction is precisely the one given by Robinson [3] under the name *topological filter*. The general definition of a *topological filter* via sheaves after Robinson [3, Definition 4.15] is encapsulated in this diagram:

$$\mathcal{S}_i \overset{i}{\leftarrow} \mathcal{S}_s \overset{o}{\rightarrow} \mathcal{S}_o \tag{1}$$

The sheaves $\mathcal{S}_*$ in this construction have the following meaning: $\mathcal{S}_i$ is the input sheaf, $\mathcal{S}_o$ is the output sheaf, and $\mathcal{S}_s$ is the sheaf of the state of the topological filter. These are called sheaves because this structure is attached over simplices of simplicial complex and connected via consistency maps. Tracing out an underlying simplicial line complex results in a general structure [3, 6] as depicted in Figure 2.

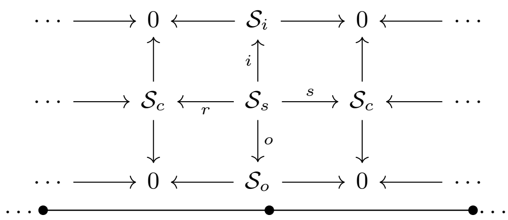

Figure 2: The sheaf filter structure over a line complex. The vertical sheaf structure constitutes a topological filter. It is associated with the 0 and 1-simplex below it. Horizontal maps correspond to the traversal between neighboring simplices via face and boundary maps and the sheaves above it.

This structure captures all essential aspects of sound synthesis over a simplicial topology. It defines an input-output relationship with a temporal dynamic. In the context of sound synthesis, the input captures interactions, excitations, or parametric changes of the synthesis method. The output is typically related to generated digital audio, though it can also serve as an input for further computations.

More specifically, each vertical slice of Figure 2 has the basic form of the general topological filter of equation (1). The state over each 0-simplex $\mathcal{S}_s$ is connected via a 1-simplex. This captures the notion of a transition (or time-step). The consistency sheaf $\mathcal{S}_c$ contains the information that is needed to transition data between the two attached 0 simplices. The top row corresponds to inputs over different simplices of the underlying topological space. The bottom row corresponds to the output at the same simplices. Given that input and output samples are treated as independent, that is, there is no information to consider for their respective consistencies, we have 0 sheaves over 1-simplices for each. In this diagram, we think of time flowing from left to right. According to this interpretation, the maps r and s capture the dynamic evolution of the state as one traverses the simplicial topology, and the maps i and o capture the injection of input into the state, and the computation of output from the state, respectively. For the direction of the arrows, we follow Robinson's convention [3], which generally points from higher dimensional spaces to lower dimensional ones. This makes maps look like traditional (single-valued) functions, but does not necessarily follow the direction of computation in practical realization. For sound synthesis methods considered here, inputs will always be given. Therefore, all inputs are always computed as injections into the state, and never as projections from the state. Furthermore, we will only consider synthesis in a causal direction and therefore the retrieval map r is also always computed as an injection. Hence, the practical computation of maps i and r are always performed against the direction of the arrow in the diagram. Due to injectivity, both maps can populate the state sheaf directly.

4. FILTER-BASED SYNTHESIS METHODS AS SHEAVES

The previous section provides the general outlines of the computations we seek to realize in the sheaf structure. First, we realize linear time-invariant digital filters as topological filters following [2, 6]. This allows us to then give concrete examples of filter-based synthesis methods over various simplicial topologies. General pole-zero IIR filters are needed to model resonant behavior. The full derivation of these filters over sheaves can be found in [6], to which the reader is referred for more details.

4.1. General LTI (Pole-Zero IIR) Filters as Sheaves

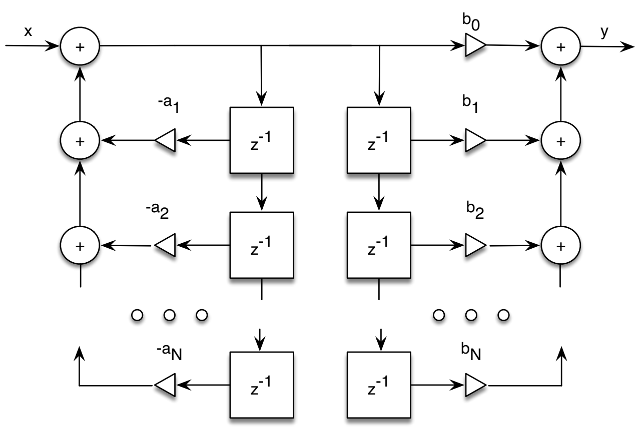

Figure 3: General IIR filter - Split Direct form II depicted as all-pole IIR filter followed by an FIR filter of the same order

The equation for the general LTI filter contains feedforward and feedback contributions as follows:

$$y = \sum_{i=1}^{N} b_i \cdot x_{N-i} + b_0 \cdot x + \sum_{j=1}^{N} -a_j \cdot y_{N-j} \tag{2}$$

There are a number of practical realizations of such filters. However, the Direct Form II [4] (depicted in Figure 3 in split form)

is particularly convenient for our discussion. If the length of the feedback filter on the left and the feedforward filter on the right are the same, the state is in fact identical and can be merged. This is required to arrive at a shared state description for a full pole-zero filter. From equation (2) emerges a state vector of size N. If feedforward or feedback in the construction is lower, we simply pad with filter coefficients set to zero to get matching lengths. The input additionally provides a further dimension. Thus the size of the vector in the state sheaf $\mathcal{S}_s$ is $\mathbb{R}^{N+1}$.
Now we can specify the nature of the data on the sheaf structure of Figure 2 for this case, arriving at Figure 4 consisting of linear maps between vector spaces. It can be shown [6] that the linear maps in this structure for the general IIR filter are as follows:

$$s : (x_0, x_1, ..., x_{N-1}, x) \to (x_1, x_2, ..., x_{N-1}, x + \sum_{j=1}^{N} -a_j \cdot x_{N-j}) \quad (3)$$

$$r : (x_0, x_1, ..., x_{N-1}, x) \to (x_0, x_1, ..., x_{N-1}) \quad (4)$$

$$i : (x_0, x_1, ..., x_{N-1}, x) \to (x) \quad (5)$$

$$o : (x_0, x_1, ..., x_{N-1}, x) \to (b_0 \cdot x + \sum_{i=1}^{N} b_i \cdot x_{N-i}) \quad (6)$$

The input map i injects the input into the extended state. The feedforward part of the filter equation 2 is contained in the output map o. The sheaf map s captures the dynamical behaviors in filters. This consists of the discrete shift that both feedforward and feedback filters share and the feedback component of the filter. This provides a convenient interpretation of the s map capturing the complete dynamical behavior of the filter. The map r simply retrieves the computed intermediate state over a 1-simplex to be combined with the next input. Taken together, maps r and s constitute the complete time-stepped dynamic of the filter that went through an intermediate state $\mathbb{R}^N$ which, in our general sheaf filter structure, we called $\mathcal{S}_c$. Sheaf-theoretically, we say that this intermediate state contains the data needed to keep the data between the two time steps *consistent*. This is precisely the data needed to compute the next step. The standard feedforward filter dynamics is recovered when all feedback coefficients are set to 0. Then the map s reduces to a shift. In this form, it is easy to see why feedforward filters are unconditionally stable. It is a unidirectional computation with no further influence on any other components of the sheaf structure (the morphisms between two output sheaves are 0). The state can grow depending on the feedback coefficients. Given that, if all coefficients are 0 it reduces to a shift (which obviously has unit gain), the stability only depends on the feedback coefficients. It can be shown that the sheaf map s is precisely the state space matrix with input [6] hence the stability of the feedback is identical to that of the classical filter with the same coefficients.

4.2. Digital Filters as Resonators

Digital filters are closely related to finite difference recurrence models, which in turn arise as the discretization of differential equations. Through this route it is well-known that a mass-spring system with its damped oscillatory behavior is discretely modeled by low-order feedback systems which in turn become low-order pole-zero IIR filters when interpreted as digital filter structures. Henceforth, we call these types of filters *resonators* [5]. Furthermore, linearity means that each mode of a linear medium can be

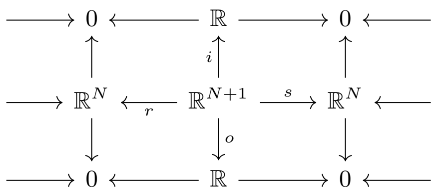

Figure 4: IIR filter in sheaf form using linear maps between vector spaces

modeled independently, and the aggregate behavior can be recovered as a linear combination of individual resonators, which in this context are called modes. This is the core of *modal synthesis* [7]. We use a second order pole-zero equal-gain resonant filter following [8, 9] as the first of our filter-based examples.

It is straightforward to plug the typical equal-gain resonant filter coefficients $a_2 = R^2$ and $a_1 = -2R\cos(2\pi\frac{f}{f_s})$ into equation (3) and the feed forward coefficients $b_0 = 1$, $b_1 = 0$ and $b_2 = -1$ into equation (6) with $N = 2$. R is the distance from the unit circle, capturing the temporal decay. We used 0.99995.The expected spectral outcome of this filter (under classical uniform sampling) is a single narrowband signal that sounds comparable to a sinusoidal oscillator under some exponential damping.

4.3. Digital filters as Physical Models

The ideal 1-D wave equation permits traveling wave solutions. The traveling behavior can be modeled in a straightforward fashion by digital filters using delay lines, with some additional structure. Given that the delay lines are closed onto themselves for finite strings or tubes, these correspond to comb-filter-like digital filters [11]. The plugged string model by Karplus and Strong [10] is a simple example of these types of physical models.

The Karplus-Strong model is a modified comb filter with an averaging in its feedback filter coefficients. We chose $a_N = D \cdot L$ and $a_{N-1} = D \cdot (1 - L)$ with $D = 0.99$ corresponding to an overall damping factor, $L = 0.9$ being a weighted average. $b_0 = 1$ and all other coefficients zero. The length $N = 1 + \frac{f_s}{2f}$ is the loop length roughly tuned to a frequency at f. The spectrum of the Karplus-Strong model is harmonic with partials decaying according to the low-pass filtering in the feedback loop. Hence, high frequency partials decay faster than low frequency ones.

5. OSCILLATORS ON THE CIRCLE AS SHEAVES

The need to use linear maps over vector spaces is not a given in topological filters. A range of examples for feedforward structures have been proposed by Robinson [3]. Here we give an example of a nonlinear topological filter with feedback properties, discussing attaching maps from circles to the circles known as circle maps [21, 12] to sheaves.

5.1. Topological Filters with Circle Maps

Circle Maps are maps from the circle to the circle $\mathbb{S}^1 \to \mathbb{S}^1$. We can think of them as being maps between phases of oscillators [21, 12], hence a general circle map over a normalized phase interval $[0, 1)$ has the form:

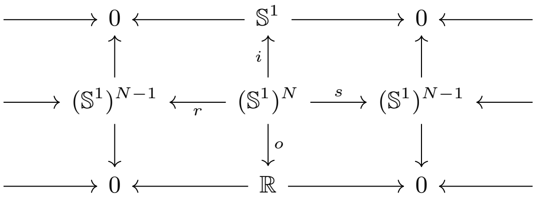

Figure 5: Nonlinear Oscillators in sheaf form using circle maps between circles

$$x_{n+1} = f(x_n) \mod 1 \tag{7}$$

Given that in this equation the past state of the circle position informs future states, this describes a pair of state transition maps s, r. Furthermore, we can modify these maps by introducing an input x from $\mathbb{S}^1$. The general form of a circle map with input with an additional phase is as follows:

$$x_{n+1} = f(x_n, x) \mod 1 \tag{8}$$

Output can be constructed through some projection function (compare [12]):

$$y_n = p(x_n) \tag{9}$$

The *general modulated circle map* that encapsulates a large class of oscillatory synthesis methods while being otherwise not overly general is the following [12]:

$$x_n = x_{n-1} + \Omega + Hf(x_{n-1}, z_{n-1}, \omega_m, x) \mod 1 \tag{10}$$
$$z_n = z_{n-1} + \omega_m \mod 1 \tag{11}$$

We note that this form actually updates two phases on the circle per step. This form indicates that, in principle, more phases can become part of the temporal update. Figure 5 shows the sheaf morphisms with $N-1$ dynamical phases and 1 input for a total state dimension of N.

The parameter $H \in \mathbb{R}$ controls the strength of the effect of the function f, which is traditionally either a nonlinearity or a modulation. The dynamics of z_n corresponds to the oscillation of a possible modulation while x_n is the overall dynamical behavior. A typical projection for oscillators is the orthogonal projection from the circle hence we get $p(\cdot) = sin(2\pi \cdot +\phi)$ to compute the final output.

5.2. Frequency Modulation on Sheaves

In our examples we restrict ourselves to the iterative form of frequency modulation [13, 12]:

$$s : (x_0, z_0, x) \rightarrow (x_0 + \Omega + H\sin(2\pi x_0) + x \mod 1, z_0 + \omega_m \mod 1) \tag{12}$$
$$r : (x_0, z_0, x) \rightarrow (x_0, z_0) \tag{13}$$
$$i : (x_0, z_0, x) \rightarrow (x) \tag{14}$$
$$o : (x_0, z_0, x) \rightarrow (sin(2\pi x_0 + \phi)) \tag{15}$$

In this version, H is usually referred to as the *modulation index*. Ω and ω_m are phase increments corresponding to the frequency of the oscillator f, modulation frequency f_m and sample rate f_s such that $\Omega = f/f_s$ and $\omega_m = f_m/f_s$. For our examples, we use $f = 220$ and $f_m = \frac{5}{2}f$ which produces an inharmonic spectrum. The sheaf maps in this case play an analogous role to the ones for the LTI digital filter discussed in Section 4. The map s computes the temporal dynamics. The map r injects the intermediate state into the next time step to be combined with a new input. The input map i injects data into the state. The output map o computes a projection of the state onto a single sample. Given that all maps from the circle onto the circle are stable (in the sense that they stay on the circle by definition) [12], this topological filter is stable.

6. FROM TOPOLOGY TO AUDIO SAMPLES

So far, we attached existing sound synthesis computations over a simplicial topological space. This is not sufficient information to compute audio samples because, at a minimum, there are no distances associated with the simplicial complex. Therefore, the computation is actually not yet related to any notion of temporal progression and a notion of temporal sampling. Additional information is needed to relate the results of the topological construction to the format required by standard audio digital analogue conversion, which is fixed rate uniformly sampled data.

6.1. Metrization, Embedding, and Projection

A first step is to add information to our simplicial topology that allows a description of some possibly local notion of *distance* (a *metric*). The simplest form of adding metric information is to associate each 1-simplex $\mathcal{X}_1$ and thus, via the boundary map b, two 0-simplicies $\mathcal{X}_0^1$ and $\mathcal{X}_0^2$ with a pair-wise metric $d(\mathcal{X}_1) = d(b(\mathcal{X}_1)) = d(\mathcal{X}_0^1, \mathcal{X}_0^2)$. This notion of a metric is local, only specifies the distance over one 1-simplex, and says nothing about global metric structure.

A process of *embedding* can help to better understand the difference between local and global metric structure. If a simplicial complex is embedded in a space, it takes on its metric structure. A simple visual example of the embedding process is that of a closed rubber band that is dropped on a flat surface and pinned down in a specific configuration of stretching to that surface. The particular configuration and the pinning imposes distances on the rubber band as well as a kind of flat projection of the band. An embedding can provide substantially more metric information than the pairwise metric. One can recover global distances and angles from a Euclidean embedding. Global metric information from embeddings more strongly relates to our geometric intuition. Inversely, we can also use geometric embeddings to construct topological spaces with a given metric structure that appeals to our geometric intuition. We discuss this latter case in Section 7.2.

Finally, the given embedding or metrization may not directly correspond to the dimensionality of an audio sample. If we draw a simplicial loop (consisting of a line complex that closes onto itself) and embed it in the Euclidean plane $\mathbb{R}^2$, there is a mismatch to the dimensionality of the audio sample amplitude, which is in an interval $(-1, 1)$ in $\mathbb{R}$. We call the map from a higher dimensional space to a lower dimensional one a *projection*.

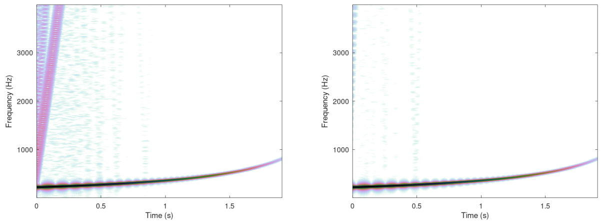

Figure 6: Log-scale spectrogram of a second-order equal-gain resonant filter with resonant frequency at 220Hz subject to an exponential up sweep due to distance shortening of the metric of the line complex. (Left) Without antialiasing. (Right) With 4-time oversampling and a length-4 averaging FIR filter.

7. EXAMPLES

7.1. Resonant Filter over Variable Metrics on a Line Complex

A first simple concrete example illustrates basic properties of sound synthesis over a topological space. This example uses the standard equal-gain reson filter [8] realized as a topological filter (Section 4.2). The filter is implemented over a line complex. We create an upward frequency sweep by exponentially shortening the spacing between sampling in a pairwise metric $d(X_1) = x^n$ computed over 2 seconds at $44100Hz$. We should expect the topological filter to be unconditionally stable by construction as long as the IIR filter design itself is stable. This is achieved in the standard implementation by choosing the radius of the resonance poles to be less than 1. There is a sample position mismatch between the audio samples required by the standard DAC hardware and our non-uniform sampling from the exponential shortening. Our implementation uses the most naive form of dealing with oversampling and undersampling. In the first case, an adaptive length FIR averaging filter computes the average of the samples that fall within one uniform rate audio sample. In the case of undersampling, the previously computed value is returned, so long as the metric of the topological space exceeds the distance between audio samples. Hence, we expect the possibility of aliasing and rectangular waveforms artifacts in the result. This can be seen in Figure 6 (left). To illustrate how to deal with aliasing, we implemented a simple 4-time oversampling scheme with a standard 4-length FIR averaging filter. Sample frequency and numbers of samples computed were normalized to maintain the same sweep after discrimination. While far from optimal, this scheme already substantially reduces audible aliasing in the example (see Figure 6 (right)).

7.2. Embedding a Simplicial Loop Complex

We called a line complex a *loop complex* if it closes onto itself by identifying the first and last 0-simplex of a the line complex. On its own, a loop complex requires no metrization or embedding.

Example 1. Circle in the Plane

A simple example of a loop with metric structure would be the standard embedding of the 0-simplices at regular points of the circle in the plane $\mathbb{R}^2$. For a set of N discretely parametrized points $\forall n \in [0, N) \in \mathbb{N}$:

$$x = R\sin(2\pi n/N) \qquad y = R\cos(2\pi n/N) \qquad (16)$$

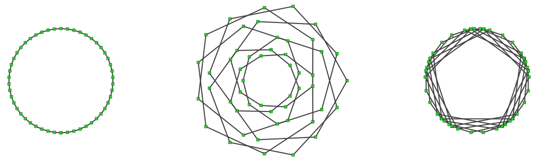

Figure 7: A simplicial loop consisting of 50 0-simplices that is (Left) embedded in the plane as a circle, (Middle) an immersion of a torus knot by projection on the plane, and (Right) an immersion of the FM loop by projection on the plane .

By construction, all pair-wise distances between neighboring 0-simplices have the same distance in this embedding, so we arrive at a uniform sampling. Note that we could keep the same metric circle, and choose another, perhaps irregular set of points on it and we would arrive at a different, perhaps non-uniform, sampling. In fact, any map from the circle to itself can be made into a sampling scheme in this way. However, one can also arrive at variation in pair-wise distances through alternative embeddings of a loop complex in some space.

Example 2. Torus Knot Projected onto the Plane

To illustrate this concept, first consider what is known as the torus knot [30]. The torus knot is topologically a closed loop but is derived from forming a closed path on a torus. In our case, consider the following parametric closed curve in $\mathbb{R}^3$ where we pick a number of discrete points $\forall n \in [0, N) \in \mathbb{N}$:

$$x = R\cos(2\pi kn/N)\cdot(Q+\cos(2\pi ln/N)) \qquad (17)$$
$$y = R\sin(2\pi kn/N)\cdot(Q+\cos(2\pi ln/N)) \qquad (18)$$
$$z = R\sin(2\pi ln/N) \qquad (19)$$

This is a discretely sampled set of points of a winding path on the surface of a torus in $\mathbb{R}^3$ with a big radius R, a ratio Q both in $\mathbb{R}$ and relative prime numbers k and l indicating the winding ratio of the path around the torus. Note that given the uniform sampling of our discrete points, the pair-wise distances are again uniform. Many different embeddings of a simplicial loop can have uniform sampling; this is one example. However, we can generate non-uniform sampling from geometry by considering projections.

More generally than considered before, a *projection* is a map that reduces dimensionality. For example, given two Euclidean spaces $\mathbb{R}^m$ and $\mathbb{R}^n$ where $m > n$ we consider the map $p : \mathbb{R}^m \to \mathbb{R}^n$. The simplicial torus knot with Euclidean embedding in $\mathbb{R}^3$ can be projected down to a plane $\mathbb{R}^2$. The most straightforward projections are those that simply remove one dimension. That is any plane xy, yz, xz makes a projection (as does any rotation in $\mathbb{R}^3$ followed by such a projection). Figure 7 (middle) shows one such projection of the simplicial torus knot from an embedding in $\mathbb{R}^3$ into the plane. Not all pair-wise distances are the same in this planar projection because of the distance that was captured by the projected dimension that has been removed, which shortened the distance in the projection appropriately. For example, taking a z-projection onto the xy-plane, we have the following relationship between pair-wise metrics: $d_3(\Delta x, \Delta y, \Delta z) = \sqrt{(\Delta x^2 + \Delta y^2 + \Delta z^2)}$ and $d_2(\Delta x, \Delta y) = \sqrt{(\Delta x^2 + \Delta y^2)}$. If z vanishes, these two metrics are identical, otherwise any other distance in the projection is diminished. Thus the sampling distance becomes non-uniform according to the projected dimension.

Example 3. FM Knot Projected onto the Plane

This idea can be used to construct non-uniform sampling inspired by known synthesis methods. It has been shown that frequency modulation [13] can be understood as an example of a map from a circle to itself [12]. For relative prime ratios of carrier and modulation frequency, these maps close onto themselves in finite time forming a loop complex. This provides a novel way to view frequency modulation as path in $\mathbb{R}^3$. We construct this path in analogy to the torus knot. The torus as well as the torus knot come about as a product of two circles. Recognizing that frequency modulation involves two circles (though not as a product) we can create a path equation in analogy to the ones given for the torus knot in equations (17)–(19) by applying a *lift* onto both circles to the frequency modulation equation. This means that we provide the orthogonal trigonometric function in a second orthonormal dimension. This process yields the following parametric path equation in $\mathbb{R}^3$ for a *frequency modulation knot* (or *FM knot*):

$$x = R\cos(2\pi kn/N + I_m \cos(2\pi ln/N)) \quad (20)$$

$$y = R\sin(2\pi kn/N + I_m \cos(2\pi ln/N)) \quad (21)$$

$$z = R\sin(2\pi ln/N) \quad (22)$$

I_m is called the modulation index in FM theory. If we only consider the z-projection onto the xy-plane, we recover the circle map for frequency modulation in the plane, and a further projection [12] recovers traditional frequency modulation.

Figure 8 shows concrete renderings of the three topologized synthesis methods over two non-uniform loop spaces. In order to improve comparability, all local metric have been normalized by the total loop length. All examples are rendered at 220Hz (assuming uniform sampling) and the loop simplices consist of 88200 0-simplices. This choice leads to one-to-one mapping onto audio samples if the metric is regular. The left column shows the torus knot projected onto the plane and the right column shows the FM knot under the same projection. Both knots use the same relative prime ratios $5 : 7$. None of these results have been oversampled, so aliasing artifacts are visible, perhaps most clearly in the top row showing the result for the resonant IIR filter of Section 4.2. The middle row uses the Karplus-Strong model (Section 4.3) exhibiting clear harmonic spectra under the metric variation. The bottom row renders an inharmonic FM using an iterative circle map (Section 5.2), which shows a densely inharmonic spectra while still maintaining the general spectral variation pattern.

Overall results depicted in Figure 8 show that all methods are stably computed although they are subjected to severe deformation in their sampling due to the underlying geometric construction. This demonstrates that carefully constructing synthesis methods as stable maps on sheaves leads to robust yet highly flexible manipulation mediated by topological and metric information that can now be used as parametric control.

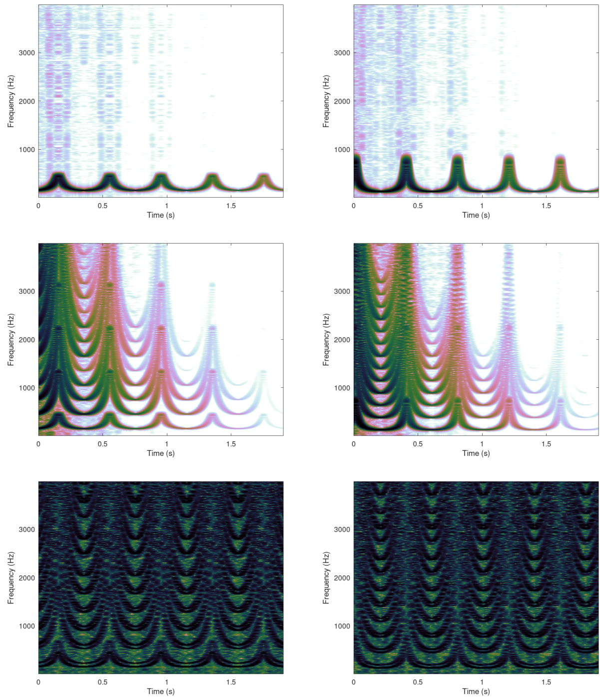

Figure 8: Log-scale spectogram of a second order resonant filter (Top), of a Karplus-Strong filter (Middle) and a circle map realizing frequency modulation at modulation index 0.33 and frequency ratio 2:5 for an inharmonic spectrum (Bottom). Loop-complex metrized by plane projections of knots with 88200 0-simplices. (Left) Torus knot. (Right) FM knot.

8. CONCLUSION

In this paper, we showed how sound synthesis methods can be attached to simplicial topological spaces via sheaves. In particular, we used two broad approaches to sound synthesis: digital linear time-invariant filter-based methods as are widely used in modal and physical modeling synthesis, as well as oscillatory-based methods understood as circle maps. However, these are but two examples of the general process of generalizing sound synthesis methods by associating them with topological space via sheaves. This paper can hopefully serve as a template for this construction for other synthesis methods. A key goal of this work is to show how topology can become an explicit building block and methodology for digital sound synthesis and processing.

Much work remains to be done to fully explore how topological methods can be fruitfully used in digital audio. Discussed in the paper are concepts of aliasing, metrization, and embedding. All of these aspects can be realized and systematized in a context broader than was possible to cover here. The resulting variation in temporal patterns has a relationship to signal processing on irregular samples [31], and event-based signal processing [32]. Depending on the implementation, topological filters can end up having very similar requirements as filtering in these cases (see recent work on band-limited filtering over irregular samples [33, 34] as well as the use of heterogeneous signal processing systems that combine multiple fixed and event-rate processing [35, 36]. To develop these connections in full detail is interesting future work.

9. ACKNOWLEDGMENTS

This work was made possible by a fellowship of the John Simon Guggenheim Memorial Foundation. Many thanks to Perry Cook and Bob Adams for helpful discussions and references. I appreciated comments by four anonymous reviewers, which helped improve the manuscript.

10. REFERENCES

[1] Glen E. Bredon, *Sheaf Theory, Graduate Texts in Mathematics*, Springer New York, 1997.

[2] Michael Robinson, "Understanding networks and their behaviors using sheaf theory," in *Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2013, pp. 911–914.

[3] Michael Robinson, *Topological signal processing*, Springer, 2014.

[4] Julius O. Smith, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, 2007.

[5] Ken Steiglitz, *A digital signal processing primer, with applications to digital audio and computer music*, Addison Wesley Longman Publishing Co., Inc., 1997.

[6] Georg Essl, "Topological IIR Filters Over Simplicial Topologies via Sheaves," *IEEE Signal Processing Letters*, vol. 27, pp. 1215–1219, 2020.

[7] Jean-Marie Adrien, *The Missing Link: Modal Synthesis*, p. 269–298, MIT Press, Cambridge, MA, USA, 1991.

[8] Julius O. Smith and James B. Angell, "A constant-gain digital resonator tuned by a single coefficient," *Computer Music Journal*, vol. 6, no. 4, pp. 36–40, 1982.

[9] Ken Steiglitz, "A note on constant-gain digital resonators," *Computer Music Journal*, vol. 18, no. 4, pp. 8–10, 1994.

[10] Kevin Karplus and Alex Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.

[11] Julius O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992.

[12] Georg Essl, "Iterative phase functions on the circle and their projections: Connecting circle maps, waveshaping, and phase modulation," in *Perception, Representations, Image, Sound, Music*, Richard Kronland-Martinet, Sølvi Ystad, and Mitsuko Aramaki, Eds., Cham, 2021, pp. 681–698, Springer International Publishing.

[13] John M. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the audio engineering society*, vol. 21, no. 7, pp. 526–534, 1973.

[14] Herbert Edelsbrunner and John Harer, *Computational topology: an introduction*, American Mathematical Society, 2010.

[15] Robert Ghrist, *Elementary applied topology*, Createspace, 2014.

[16] Gunnar Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[17] Sergio Barbarossa and Stefania Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.

[18] Feng Ji, Giacomo Kahn, and Wee Peng Tay, "Signal processing on simplicial complexes," 2020.

[19] Stephen E. Belter and Steven C. Bass, "Computer-aided analysis and design of digital filters with arbitrary topology," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 10, pp. 810–819, 1975.

[20] Steven Trautmann, "A Physical String Model with a Twist," in *Proceedings of the International Computer Music Conference (ICMC)*, 1995.

[21] Georg Essl, "Circle maps as a simple oscillators for complex behavior: I. Basics," in *Proceedings of the International Computer Music Conference (ICMC)*, New Orleans, November 2006.

[22] Georg Essl, "Aspects of the Topology of Interactions on Loop Dynamics in One and Two Dimensions," in *LNCS 3310 Proceedings of the International Symposium on Computer Music Modeling and Retrieval 2004*, pp. 220–231. Springer Verlag, Esbjerg, Denmark, 2005.

[23] Guerino Mazzola, Stefan. Göller, and Stefan. Müller, *The topos of music : geometric logic of concepts, theory, and performance*, Birkhäuser, Basel; Boston, 2002.

[24] James R. Munkres, *Topology*, Prentice Hall, Incorporated, 2000.

[25] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[26] John W. Gray, "Fragments of the history of sheaf theory," in *Applications of sheaves*, pp. 1–79. Springer, 1979.

[27] Allen D. Shepard, *A Cellular Description of the Derived Category of a Stratified Space*, Ph.D. thesis, Brown University, 1986.

[28] Justin M. Curry, *Sheaves, Cosheaves and Applications*, Ph.D. thesis, The University of Pennsylvania, 2014.

[29] Michael Robinson, "Universal factorizations of quasiperiodic functions," in *Sampling Theory and Applications (SampTA), 2015 International Conference on*. IEEE, 2015, pp. 588–592.

[30] Andrzej Stasiak, Vsevolod Katritch, and Louis H. Kauffman, Eds., *Ideal Knots*, World Scientific, 1998.

[31] Farokh Marvasti, *Nonuniform sampling: theory and practice*, Springer Science & Business Media, 2012.

[32] Marek Miskowicz, *Event-based control and signal processing*, CRC press, 2018.

[33] Laurent Fesquet and Brigitte Bidégaray-Fesquet, "IIR digital filtering of non-uniformly sampled signals via state representation," *Signal Processing*, vol. 90, no. 10, pp. 2811–2821, 2010.

[34] Adrian E. G. Huber and Shih-Chii Liu, "Filtering of nonuniformly sampled bandlimited functions," *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 1036–1040, 2019.

[35] Georg Essl, "Ursound-live patching of audio and multimedia using a multi-rate normed single-stream data-flow engine.," in *Proceedings of the International Computer Music Conference (ICMC)*, 2010.

[36] Georg Essl, "Playing with time: manipulation of time and rate in a multi-rate signal processing pipeline.," in *Proceedings of the International Computer Music Conference (ICMC)*, 2012.