Fuzzy Granular Synthesis Controlled by Walsh Functions

Adolfo Maia Jr. [1] [2] **, Eduardo R. Miranda** [1]

[1]Computer Music Research, Faculty of Technology
University of Plymouth, Plymouth, Devon PL4 8AA
United Kingdom.

[2]Departamento de Matemática Aplicada, IMECC and
Núcleo Interdisciplinar de Comunicação Sonora (NICS), UNICAMP,
13.081-970, Campinas, SP, Brazil.

eduardo.miranda@plymouth.ac.uk

adolfo@nics.unicamp.br

Abstract. *In this paper we present a model for granular synthesis in which the internal content (spectrum) of grains are defined as Fuzzy Sets. Markov Chains modulated by Membership Matrices related to the internal structure of the sound grains are used to control the evolution of the sound in time. The final sequencing of grains (sound stream) is controlled by the so called Walsh Functions. Using a number of channels the outputs may have complex polyphonic sound structures. We also provide the mathematical foundations of the model.*

1. Introduction

Granular synthesis [Roads, 1988] is commonly known as a technique that works by generating a rapid succession of tiny sounds, metaphorically referred to as sound grains [Roads, 1996]. Granular synthesis is widely used by musicians to compose electronic or computer music because it can produce a wide range of different sounds, but it also has been used in speech synthesis [Miranda, 2002]. Clearly a discussion about musical aesthetics may arise from these developments. Although such discussion would be a very interesting topic on its own right, we will not deal with these matters in this paper. A good discussion on the aesthetics of microsound can be found in [Thomson, 2004].

Granular synthesis is largely based upon D. Gabor idea of representing a sound using hundreds or thousands of elementary sound particles [Gabor, 1947].

In this work we take C. Roads' definition of sound grain as a point of departure to develop a formal but flexible granular synthesis model. Our model uses stochastic processes, namely Markov Chains with Transition Probability Matrix modulated by Membership Functions of the grains with values in the interval $[0, 1]$, which give fuzzy characteristics to the grains. Thus, we propose a new method for controlling the grains by intertwining Stochastic Processes and Fuzzy Set Theory, where the content of the grains (or internal variables) can change their transition probabilities between states. For the sake of clarity, we have chosen a very simple *State Space* to introduce the model, where each grain is itself a state of a Grain Vector $\mathbf{G}$. Therefore, the membership functions in this case modulate the transition probabilities between states (i.e., grains), changing their ordering position in the time domain. In this paper we present just one of the several possible modes of interaction between internal and external control variables.

Walsh functions were used as tool for code transmission by electromagnetic signals [Beauchamp, 1975], [Hall Jr., 1986]. In adition in the 70s they were used for real sound synthesis [Rozenberg, 1979],[Hutchins Jr., 1973], [Hutchins Jr., 1975],[Insam, 1974]. Our application differs from the last authors since we are most interested to use Walsh functions as a control for sound output of grains streams. As far as we know this is the first model in this direction.

In the next section we present the concept of Fuzzy Grains and their mathematical representation. In section 3 we describe the control of grain streams including halting criteria, a computer implementation, which we named Fuzkov 1.0, and the control of the sound output by Walsh Functions. In section 4 we conclude with some comments and some perspectives for future work. In addition we provided two appendices, namely, the first one is a very short review on Fuzzy Sets and the second one on Walsh Functions.

2. Markov Processes for Fuzzy Grains

Fuzzy Sets [Zadeh, 1965] are able for handling uncertainty, imprecisions or vagueness. Our aim here is to get a kind of Markov Process in which the Transition Matrix could be modified by the internal content of the grains. In order to weight the contribution of each Fourier component we have used a Membership Function for the grains from Fuzzy Sets theory. In Appendix A we present a short review of Fuzzy Sets.

In this work we are mainly interested in the output control of the material generated by Fuzzy Granular Synthesis. For the sake of completicity we explain shortly fuzzy grains and their generation.

Let us denote Ω the space of all possible oscillators, that is the *frequency* $\times$ *amplitude* space of the ordered pair (ω, a), where the variables ω and a varies in some suitable real intervals. Ω is referred to as a Parameters Space. In this work we define a grain g as a finite set of points $\{(\omega_i(t), a_i(t)), i = 1, 2, \ldots, N\}$ in Ω. The sound in the macro scale, or in more technical words, the time ordering of grains and their subsequent sound output is generated as a Markov Chain. So a grain can be described by its Fourier Partials inside a real interval I. Clearly, this is suitable for producing grains with additive synthesis. Its spectral content can be written, without loss of generality, as

$$G(t) = \sum_{n=1}^{N} a_n sin[2\pi\omega_n t + \delta_n], \tag{1}$$

where a_n, ω_n, δ_n reads for amplitude, frequency and a possible phase, respectively. In granular synthesis a sound can be viewed as a quick stream of grains which, from a geometrical point of view, describes a trajectory in the Ω space. Subsets of points in Ω do not have a natural well ordering for the space part. In our model, the internal content of a grain is coded in matrices. All operations on grains are represented as matrix operations. For the sake of completeness, we present an ordering of the elements of the grains, but this is a highly arbitrary choice. Below we just show the simplest ordering: that one that access the grain content (which is a two dimensional set of points) from the left to the right (that is, from low to high frequencies) and from the bottom to the to top (that is, from low to large amplitudes). This is formally written as follows: let $x_i = (\omega_i, a_i)$ and $x_j = (\omega_j, a_j)$ be two arbitrary points in the Ω space. For $i \neq j$

$$x_i < x_j \Leftrightarrow \begin{cases} \omega_i < \omega_j \\ \omega_i = \omega_j \Rightarrow a_i < a_j \end{cases} \tag{2}$$

With this definition the matrix representation of a grain g_i with r components reads as a $2 \times r$ matrix:

$$g^i = \begin{bmatrix} \omega_1^i & a_1^i \\ \omega_2^i & a_2^i \\ \vdots & \vdots \\ \omega_r^i & a_r^i \end{bmatrix} \quad (3)$$

where the above defined order is implicit.

Now, a fuzzy grain can be represented as a three column matrix

$$G^i = \left[\begin{array}{cc|c} \omega_1^i & a_1^i & \alpha_1^i \\ \omega_2^i & a_2^i & \alpha_2^i \\ \vdots & \vdots & \vdots \\ \omega_r^i & a_r^i & \alpha_r^i \end{array}\right] \quad (4)$$

where we have introduced a third column with the membership frequency and amplitude values of each partial of the grain g^i. Note that g^i is a particular case of G^i for $\alpha_1^i = \alpha_2^i = \ldots = \alpha_r^i = 1$. We can denote shortly this matrix by $G^i = [\omega^i, a^i, \alpha^i]$, where $\omega^i = [\omega_1^i, \omega_2^i, \ldots, \omega_r^i]$, $a^i = [a_1^i, a_2^i, \ldots, a_r^i]$, and $\alpha^i = [\alpha_1^i, \alpha_2^i, \ldots, \alpha_r^i]$ are the frequency, amplitude and membership r-vectors of the grain. Also we define N-vectors of grains $\mathbf{g} = \left[g^1, g^2, \ldots, g^N\right]$ and $\mathbf{G} = \left[G^1, G^2, \ldots, G^N\right]$.

Below we show how the membership functions of fuzzy grains can modify the Markov Transition Matrix and so we get a fuzzy control for the Markov Chain. For a good account of Fuzzy Sets the reader is refereed to [Diamond and Kloeden, 1994]. Let us consider a grain described by its Fourier-like equation (1). Each subset of points in Ω represents a grain with particular Fourier partials, that is, it is a sum of basic sinusoidal frequencies. With the above defined matrices G^i, it is possible to define an unambiguously time evolution of grains through out Markov Chains. This is usually accomplished through a Fuzzy Transition Table, constructed as follows: firstly, suppose that we have a transition matrix for ordinary grains, that is, with no membership vector yet defined. This can be written as follows:

$$\left[\begin{array}{c|cccc} & g^1 & g^2 & \ldots & g^N \\ \hline g^1 & p^{11} & p^{12} & \ldots & p^{1N} \\ g^2 & p^{21} & p^{22} & \ldots & p^{2N} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ g^N & p^{1N} & p^{2N} & \ldots & p^{NN} \end{array}\right]$$

which can be viewed as a function

$$\begin{array}{rccc} p: & \mathbf{g} \times \mathbf{g} & \longrightarrow & [0,1] \\ & (g^i, g^j) & \longmapsto & p\,(g^i, g^j) = p^{ij} \end{array}$$

Now, we define a Fuzzy Extended Probability Transition Matrix (or simply Fuzzy Transition Matrix) $Q : \mathbf{G} \times \mathbf{G} \longrightarrow [0,1]$ as

$$Q^{ij} = Q\left(G^i, G^j\right) = \Phi^{ij} * p^{ij} \quad (5)$$

where the symbol $*$ means a matrix operation (e.g., a scalar product, a matrix product or any other well defined operation). The function Φ^{ij} is generated as a finite number of applications of the following basic operations of fuzzy sets: for $i, j = 1, 2, \ldots, N$, we define

1.

$$\phi^{ij} = \max_{1\le k\le r}\left\{\alpha_k^i, \alpha_k^j\right\}, \tag{6}$$

where α^i and α^j are the membership vectors of the grains G^i and G^j respectively.

2.

$$\phi^{ij} = \min_{1\le k\le r}\left\{\alpha_k^i, \alpha_k^j\right\}, \tag{7}$$

where α^i and α^j are the membership vector of the grains G^i and G^j respectively.

3.

$$\alpha_c^i = 1 - \alpha^i. \tag{8}$$

These result in a product like $\Phi^{ij} = \phi_1^{ij}\phi_2^{ij}\ldots\phi_l^{ij}$, where the third operation above can be performed on any product of α_i vectors. These are basic operations on Fuzzy Sets. See reference [Diamond and Kloeden, 1994] for a introduction to Fuzzy Sets and their metrics. Note that since the membership function modulates the probability values p^{ij}, the condition for the probability sum $\sum_{j=1}^{N} Q^{ij} = 1$ can be violated. In order to solve this problem we renormalize the matrix Q^{ij} as follows. Denoting $q_i = \sum_{k=1}^{N} Q_{ik}$ we define the elements of matrix $\mathbf{P}$ as

$$P_{ij} = Q^{ij}/q^i \qquad i,j = 1,2,\ldots,N \tag{9}$$

Now the probability property $\sum_{j=1}^{N} P_{ij} = 1$ is clearly satisfied. The above definition shows that the internal fuzzy content of the grains have a weight (through the function Φ^{ij}) for their transition to a next state of the Markov Chain.

The Fuzzy Transition Matrix (or Table) reads

$$\left[\begin{array}{c|cccc} & G^1 & G^2 & \ldots & G^N \\ \hline G^1 & P^{11} & P^{12} & \ldots & P^{1N} \\ G^2 & P^{21} & P^{22} & \ldots & P^{2N} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ G^N & P^{N1} & P^{N2} & \ldots & P^{NN} \end{array}\right] \tag{10}$$

In this simple model a transition from one state to another corresponds to a jump from a particular grain to another in the grain vector $\mathbf{G}$. In adition the fuzzy content of a grain, that is, its membership vector, can have a significant weight on the probability transition. Since the process is finite, a criterium to halt the process is needed here. This will be discussed in the next section.

The above model is suitable for several kinds of matrix operations on internal as well external variables controlling the grains behaviour in time. There is plenty of room for the definition of a great number of different methods to generate and control the grains. We present our approach on the control of the streams below.

3. Control of Grain Streams

3.1. Halting Criteria

There exist many different ways (algorithms) to control the evolution of the grains in time. We show here one by using the so called Hausdorf Metric which is suitable to measure distance between sets (grains are finite and discrete subsets of Ω). Time evolution can be better controlled using a fuzzy metric that takes into account the degree of membership of the Fourier partials inside each grain. In other words, partials with low membership coefficients contribute little for the Hausdorff distance measure between the grains. Membership vectors define the fuzzy character of the grains, or in a musical jargon, their weighted harmonic content. A metric control is closely related to the notions of approximation and/or the maximal time (or number of steps) available to run a process. Below we indicate three stop criteria we devised to halt a Markov Chain in our model of granular synthesis.

Halting Criteria

1. *Convergent Type*: If the distance between the last generated grain and a fixed grain (target) is smaller than a prefixed arbitrary number ϵ, the process halts.
2. *Cauchy Type*: If the distance between two states is smaller than ϵ the process halts.
3. *Maximal Number of Steps Type (MNS)*: Fix the maximum number of steps for the process to halt.

Any of the above criteria can be used to halt the process. Of course *Maximal Number of Steps Type* is the simplest one, since no metric is required. In our program *Fuzzkov 1.0* we have implemented fully the MNS and partially, the Cauchy type, at the Hausdorff Metric level, but not at the Fuzzy Metric level. We have implemented the Hausdorff Metric as an inequality, so that FuzzKov 1.0 runs in loops until it is satisfied. We obtained good results for both controls of the grains streams working together.

This procedure leads to a concentration of frequencies within a narrow bandwidth, but with a large bandwidth for the amplitudes. The *halting criterion* here can be taken as the *Cauchy type*. Given an arbitrary (but small) number ϵ, the process stops if $d_H\left(G_i, G_{i+1}\right) \leq \epsilon$, where the distance between two points used for defining the above Hausdorf Distance is given by, for example:

$$d\left(\left(\omega_i, a_i\right), \left(\omega_j, a_j\right)\right) = \max_{1 \leq k \leq r} \left|\omega_i - \omega_j\right|. \tag{11}$$

If we fix a particular grain in the Ω space, such as $\overline{G}$, we can consider the *Convergent halt criterion* , that is the proceess stops if $d_H\left(G_i, \overline{G}\right) \leq \epsilon$.

We can also take the *mean frequency* only of the last m grains and so it reads as

$$\overline{\omega}^{(l)} = \sum_{k=1}^{r} \frac{\omega_k^{l-m} + \omega_k^{l-m+1} + \ldots + \omega_k^{l-1}}{m} \tag{12}$$

and take the r closest frequencies to $\overline{\omega}^{(l)}$ from the set $U_l = \bigcup_{k=l-m}^{l-1} G^k$. Clearly, for $m = l$ we get the previous model.

3.2. Implementation

This section presents *Fuzzkov 1.0*, a prototype implementation of our model. In *Fuzzkov 1.0*, Membership Matrices modulate a Transition Probability Matrix of a Markov Chain, but the internal content of the grains are not changed during the generative process. Thus, *Fuzzkov 1.0* can be thought of as a system for *Coarse Grain Fuzzy Synthesis*. A block diagram of the program is shown in Figure 1. *Fuzzkov 1.0* was implemented in MATLAB. Input information and control parameters are as follows:

1. **Control Parameters for the Markov Process**
   markov_type = type of generation of the Markov Matrix.
   N = number of states (or number of grains).
   n = number of steps of the Markov Process.
   init_vect_type = type of the initial vector (range = 1-3).

2. **Grain Parameters**
   fs = sample frequency.
   dur = grain duration in seconds.
   r = number of points in a grain, where each point is a Fourier Partial.
   grain_type = type of grain (range = 1-3).
3. **Fuzzy Control Parameters**
   memb_type = type of Membership Matrices (range = 1-4).
   alpha_type = type of vector to construct Membership Matrices (range = 1-4).

To begin with, the control parameters for Markov Processes generate a Markov Matrix p. After a manipulation with membership matrices as in Eq. (5) we get a fuzzyfied Markov Matrix as in Eq.(10). The grain parameters control the internal content of grains and their output is a Fourier sum of partials as in Eq.(1). The last group of parameters controls the fuzzy characteristic of the grains as described by Eq. (4) and Eqs. (6)-(8). There are a number of possibilities to generate Membership Matrices. In our implementation we have taken only four possibilities in Fuzzkov 1.0.

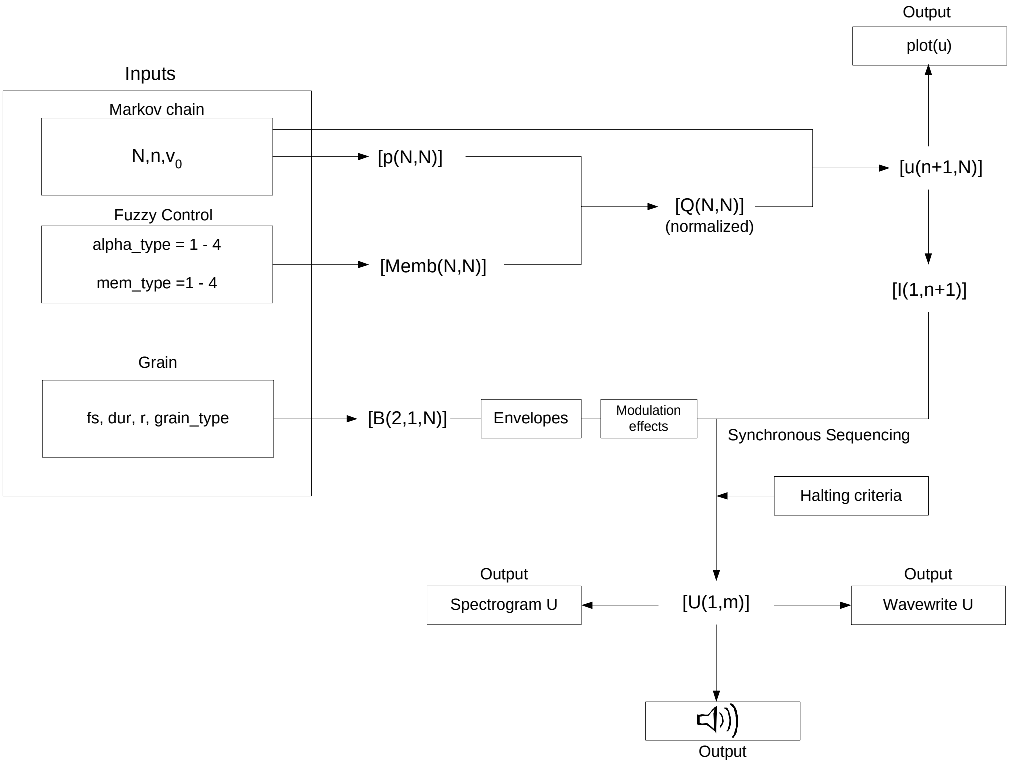

Figure 1: Diagram of FuzzKov 1.0

Sound grains are generated randomly by uniform and gaussian 3D matrices A with dimensions $2 \times r \times N$, which include r normalized frequencies and amplitudes for N grains (Fourier Partials). We have used uniform and Gaussian probability distributions to gerated the sound grains. From this we obtain a Matrix $B(2, 1, N)$ with the sum of Fourier Partials for the N grains. A Markov transition Matrix $p(N, N)$ is generated and modified by a Membership Matrix $Memb(N, N)$. A number of diferent operations are available for this modification. We take a fuzzyfied Markov Matrix $Q(N, N)$, which operates on an array of probability vectors $u(n+1, N)$. Next, a particular filter selects the index of the maximal value of each probability vector $I(1, n + 1)$. Finally, the program reorder the Grain matrix $B(2, 1, N)$ along the index vector $I(1, n + 1)$ and produce the sound. We also generate spectral data for analysis of the results.

3.3. Control of Sound Streams by Walsh Functions

Walsh functions became important for representation of signals through the superposition of members of a set of simple functions which are easy to generate and define [Beauchamp, 1975]. They form an ordered set of rectangular waveforms taking only two amplitudes values $+1$ and -1. Walsh Functions and the Hadamard Matrices which generate them are important tools in several areas such as electrical engineering and code theory. They are suitable to control time sequences and we have used this characteristic to drive our grains streams. We describe shortly, below, how we control the sound output using Walsh functions.

Each value of a Walsh function works as a trigger calling the operation associated with it. For the sake of simplicity, a Walsh function operates on a sound stream just deleting all grains of the stream which correspond to the value -1. Also we made some experiments using the operation of time inversion of the grain when it has the value -1 associated to it. Triggering operations with Walsh functions can produce an endless number of outputs, depending only on the chosen operations. We can also use different Walsh functions associated to different operations and apply them indepedently on the sound streams. In this case the results from the triggering operations can not be easily predicted.

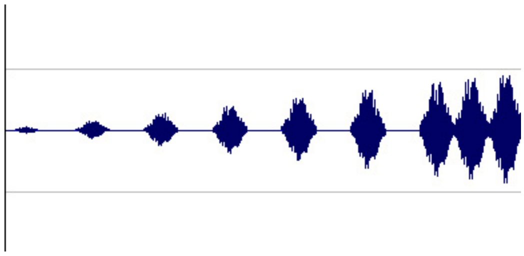

Figure 2: A typical Grains Stream with 15 steps showing the triggering by a Walsh Function. Silence between grains comes from the zero value of the Walsh Function.

An interesting experiment we have realized is to create several independent Walsh controled sound streams (using a channel for each stream) and play them sincronously. Since the Walsh functions have independent actions, it resulted poliphonic streams with a kind of *grain counterpoint*. When combined with some special effects, such as amplitude modulation and others we get some very interesting sounds which could be used in computer music compositions. Walsh functions, as used in our model, are time symmetric. In practice, the symmetrical disposition of values of a Walsh function implies that operations are performed symmetrically in time. This property could very interesting to music composition. However, one should not count on recognizing such a symmetry in the output, since is unlikely for the input to share a similar symmetry. Nevertheless, given the scope of this work, the symmetrical properties of Walsh functions and their possible compositional uses deserve further attention in future work.

4. Conclusion and Perspectives

We have implemented a prototype of our model using Matlab in which Membership Matrices of Fuzzy grains modulate a Transition Probability Matrix of a Markov Chain which is a partial control of the time evolution. In addition sound outputs are controlled by Walsh Functions. This additional control leads the sound output to have a descontinuous sequency of grains. If a number of these outputs are played syncronously we get complex sound structures or, roughly speaking, a kind of *grain's couterpointistic structures*. Our model can be generalized to include other functions besides the Walsh ones. The next step will be to use the so called *Sequency Functions* [Hall Jr., 1986],[Beauchamp, 1975] (an obvious generalization of the Walsh Functions)as triggers, and to incorporate other sound parameters, such as intensity, spacialization, etc. This will be accomplished elsewhere. As mentioned above an interesting aspect to be explored in the control of previous material by Walsh Functions is the fact that they present some symmetries which can also be better explored.

Appendix A: Fuzzy Sets

Fuzzy sets, first proposed by Zadeh [Zadeh, 1965] are able for handling uncertainty, imprecisions or vagueness. It is not a probabilistic approach in a sense that the membership function defined below can have value 1 for several, or even for all elements of the Fuzzy set. Below we present a short sumary of Fuzzy Sets.

Let G be a subset of points of a Euclidian Space $\mathbf{R}^n$. Intuitively G is a Fuzzy set if for each of its elements we associate a membership degree. Formally a Fuzzy Subset G of Ω, is a non empty subset $(x, u(x)), x \in \Omega$ of $\Omega \times [0,1]$ for some function $u : \mathbf{R} \to [0,1]$. This function is named *membership function*. The subset of points in G with non zero membership value is named *support of* G. When the support of G is finite we can consider the membership function as a vector. So, in this way we can use indistinguishably the function u as a Fuzzy Set. Below we show three examples.

1. Let A be a finite subset of $\mathbf{R}$ with m elements.

$$A = \{x_1, x_2, \ldots, x_m\} \tag{13}$$

   and the membership function defined by
   $u(x) = \begin{cases} 1/i, & \text{for } x = x_i, i = 1, 2, \ldots, m; \\ 0 & \text{otherwise} \end{cases}$
   Clearly the support of A is the subset $\{x_1, x_2, \ldots, x_N\}$.
2. Let A be a arbitrary set in $\mathbf{R}$ with membership function given by
   $\chi_A = \begin{cases} 1\text{for} & x \in A \\ 0\text{for} & x \notin A \end{cases}$
   The above example is a extreme case for which the fuzzy set is an ordinary set (also named crisp set), that is, all of its elements have membership value equal 1.
3. Let $\mathbf{B}^n(R)$ the n-dimensional ball with radius R. Define the function
   $u(x) = \begin{cases} 1 - \frac{\|x\|}{R} & \text{for} \quad x \in \mathbf{B}^n(R) \\ 0 & \quad \text{elsewhere} \end{cases}$
   This is an example of a continuous fuzzy set. Observe that the membership is 1 only for the center of the ball and decrease to 0 as x gets closer to the ball boundary.

The Hausdorff Metric

Suppose that the space $\Omega = \mathbf{R}^N$ has a metric $d(x, y)$. Let x be a point in Ω and A a nonempty subset of Ω. We define the distance of the point x to the set A as:

$$\delta(x, A) = inf\,\{d(x, y), y \in A\}\,. \tag{14}$$

The Hausdorff separation of a set B from a set A is defined by

$$\Delta(B, A) = sup\,\{d(y, A), y \in B\} \tag{15}$$

In general, Δ is not symmetric, that is $\Delta(A, B) \neq \Delta(B, A)$. In order to get a symmetric one we define the so called Hausdorff distance by

$$d_H\left(A, B\right) = max\left\{\Delta(A, B), \Delta(B, A)\right\} \tag{16}$$

With this distance function (Ω, d_H) is a Metric Space. Nevertheless this metric do not take into account the fuzzy properties of the sets. Formally, we need to define another metric which will take into account the membership functions. In order to do this we firstly define some important subsets of a given fuzzy set.

Let $u : \mathbf{R}^n \to I = [0, 1]$ a membership function.

Definition: For each $\alpha \in [0, 1]$, the α-level set $[u]^\alpha$ of a fuzzy set u is the subset of points $x \in \mathbf{R}^n$ with membership grade $u\left(x\right)$ of a least α, that is

$$[u]^\alpha = \left\{x \in \mathbf{R}^n, u\left(x\right) \geq \alpha\right\}. \tag{17}$$

The support $[u]^0$ of a fuzzy set is then defined as the closure of the union of all its level sets, that is,

$$[u]^0 = \overline{\bigcup_{\alpha\in[0,1]} [u]^\alpha} \tag{18}$$

We consider here only the fuzzy sets which satisfy the property " u maps $\mathbf{R}^n$ onto the real interval $[0, 1]$, or equivalently, $[u]^1 \neq 0$. In addition we consider only membership functions so that $[u]^0$ is a bounded subset of $\mathbf{R}^n$. Below we present some properties of the α-level sets (see [Diamond and Kloeden, 1994] to a detailed presentation and proofs).

1. For all $0 \leq \alpha \leq \beta \leq 1$
$$[u]^\beta \subseteq [u]^\alpha \subseteq [u]^0 \tag{19}$$
2. $[u]^\alpha \neq 0, \forall\alpha \in I$
3. $[u]^\alpha$ is a compact subset of $\mathbf{R}^n$ for all $\alpha \in I$.

Now we are ready to define a fuzzy metric. We define the *supremum metric* d_∞ on $\mathbf{F}$ by

$$d_\infty\left(u, v\right) = sup\left\{d_H\left([u]^\alpha, [v]^\alpha\right), \alpha \in I\right\} \tag{20}$$

for all $u, v \in \mathbf{F}$. It is worth to mention that there exist too many different metrics we can use. The above one was choose due to its simplicity and usefulness. In this work we have used the supremum metric in order to control stream of grains in $\overline{\Omega}$.

Appendix B: Walsh Functions and Hadamard Matrices

Walsh functions became important for representation of signals through the superposition of members of a set of simple functions which are easy to generate and define [Beauchamp, 1975]. They form an ordered set of rectangular waveforms taking only two amplitudes values $+1$ and -1. A simple example of a set of rectangular waveforms are the Rademacher Functions whcich can be defined as

$$RAD(n, t) = sign[sin(2^n\pi t)] \tag{21}$$

where $0 \leq t \leq 1$. Rademacher functions have two arguments n and t such that $RAD(n, t)$ has 2^{n-1} periods of square wave over a normalised time base, or interval $[0, 1]$.

The problem with Rademacher System is that it is not complete in the sense that any signal can be decomposed, like in a Fourier Series, as a sum (perhaps infinite) of

Rademacher Functions. The simplest complete set of rectangular functions (waveforms in the context of this work) is the Walsh Set. From the point of view of signal representation, Walsh functions consist of trains of square pulses (with the allowed states being -1 and 1) such that transitions may only occur at fixed intervals of a unit time step, the initial state is always 1. In general Walsh functions are defined in a Time Base interval T and periodically extended for intervals of lenght $kT, k \in \mathbf{Z}$. They are completely defined by two parameters, its *sequency order* n and its time variable t. It is denoted $WAL\left(n,t\right)$, with $n = 0, 1, 2, \ldots, N-1$, and N is the order of Walsh Functions defined below. Using a normalized time variable t/T the Walsh functions can be defined in the interval $[0, 1]$. In adition, they are symmetrical about the centre and so, when they are defined in the interval $[-1/2, 1/2]$ they are symmetrical. The even functions are collectivelly named CAL and the odd ones SAL wich are in certain sense the counterparts of the cosine and sine trigonometric functions. so, we can write

$$WAL\left(2k,t\right) = CAL\left(k,t\right), k = 1, 2, \ldots, N/2. \quad (22)$$

$$WAL\left(2k-1,t\right) = SAL\left(k,t\right), k = 1, 2, \ldots, N/2. \quad (23)$$

Both Rademacher and Walsh Sets are orthogonal systems in the same way as Fourier Systems of sin and cos functions. Other systems of rectangular functions do exist, such as Haar and Slant Functions. See reference [Beauchamp, 1975] for more information and bibliography on this subject.

Walsh functions can be ordered in a number of ways. One of them is the so called *sequency order*. The sequency k of a Walsh function is defined as half the number of zero crossings in one cycle of the time base. Walsh functions with nonidentical sequencies are orthogonal and the product of two Walsh functions is also a Walsh function. A way to generate Walsh functions is through the so called *Hadamard Matrix* when arranged in the *sequency order*. A Hadamard matrix of order N is a type of square matrix whose entries are only $+1$ and -1 and such that

$$\mathbf{H}\mathbf{H}^T = N\mathbf{I} \quad (24)$$

This equation implies that the rows of $\mathbf{H}$ (or the Walsh functions of order N) are orthogonal. In the so called *normal form* the first row and the first column are formed only by $+1$. The lowest-order Hadamard matrix is the 2 dimensional matrix

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (25)$$

An important theorem on Hadamard Matrices is stated as:

Theorem: If $\mathbf{H_m}$ and $\mathbf{H_n}$ are matrices of orders m and n respectively, then their Direct Product is an $\mathbf{H}$ matrix of order mn.

The proof of this theorem can be founded in [Hall Jr., 1986]. Most of constructions of Hadamard Matrices are based on Direct (Kronecker) Product of two matrices. The definition of Direct Product as as follows. If $\mathbf{A} = (a_{ij})$ is an $m \times m$ matrix and $\mathbf{B} = (b_{rs})$ is an $m \times m$ matrix, then the Direct Product $\mathbf{A} \otimes \mathbf{B}$ is the $mn \times mn$ matrix given by

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & \dots & a_{2m}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{i1}B & a_{i2}B & \dots & a_{im}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mm}B \end{bmatrix} \tag{26}$$

Using this theorem, higher-order matrices, with dimension 2^n are easily obtained by the recursive relationship

$$\mathbf{H}_N = \mathbf{H}_{N/2}\otimes\mathbf{H}_2 \tag{27}$$

where $N = 2^n$. Thus, for example,

$$\mathbf{H}_4 = \mathbf{H}_2\otimes\mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{28}$$

The ordering of Walsh Functions in a Hadamard Matrix is named *natural ordering* and in this case they are also named Hadamard Functions and denoted by $HAD(k,t)$, where $k = 0, 1, \dots, N$. For example, the Hadamard Functions of order 8 are given by

$$\mathbf{H}_8 = \mathbf{H}_4\otimes\mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} HAD(0,t) \\ HAD(1,t) \\ HAD(2,t) \\ HAD(3,t) \\ HAD(4,t) \\ HAD(5,t) \\ HAD(6,t) \\ HAD(7,t) \end{bmatrix} \tag{29}$$

Hadamard remarked that a necessary condition for a Hadamard matrix to exist is that n = 1, 2, or a positive multiple of 4.

A. Acknowledgments

This work was supported by FAPESP, CAPES and FAEPEX/UNICAMP (Brazil). A.M.J is gratefull to the Computer Music Group the University of Plymouth for the hospitality during the making of this work.

References

Beauchamp, K. G. (1975). *Walsh Functions and Their Applications*. Academic Press, London.

Diamond, P. and Kloeden, P. (1994). *Metrics of Fuzzy Sets: Theory and Applications*. World Scientific.

Gabor, D. (1947). Acoustical quanta and the theory of hearing. *Nature*, 159(4044):591–594.

Hall Jr., M. (1986). *Combinatorial Theory*. John Wiley and Sons Inc.

Hutchins Jr., B. (1973). Experimental electronic music devices employing walsh functions. *J. Audio Eng. Soc*, 21:640–645.

Hutchins Jr., B. (1975). Application of a real-time hadamard transform network to sound synthesis. *J. Audio Eng. Soc*, 23:558–562.

Insam, E. (1974). Walsh functions in waveform synthesizers. *J. Audio Eng. Soc*, 22:422–425.

Miranda, E. (2002). Generating source streams for extralinguistic utterances. *Journal of the Audio Engineering Society*, 50(3):165–172.

Roads, C. (1988). Introduction to granular synthesis. *Comp. Mus. J.*, 12(2):11–13.

Roads, C. (1996). *Computer Music Tutorial*. MIT Press, Cambridge, MA.

Rozenberg, M. (1979). Microcomputer-controlled sound processing using walsh functions. *Computer Music Journal*, 3(1):42–47.

Thomson, P. (2004). Atoms and errors: towards a history and aesthetics of microsound. *Organized Sound*, 9(2):207–218.

Zadeh, L. A. (1965). Fuzzy sets. *Informat. Control*, 8:338–353.