ISEA2015
Proceedings of the 21st International Symposium on Electronic Art
ISSN: 2451-8611
ISBN: 978-1-910172-00-1

Wave voxel: A multimodal volumetric representation of three dimensional lookup tables for sound synthesis

Anis Haron, George Legrady
Media Arts & Technology Program
University of California Santa Barbara, USA
anisharon@mat.ucsb.edu, legrady@mat.ucsb.edu

Abstract

Our research presents an extension to current implementations of table lookup techniques for sound synthesis. In this paper, we present methods for generating volumetric representations of data as three dimensional lookup tables for sound synthesis.

Keywords

3D lookup tables, wave voxels, three-variable functions.

Introduction

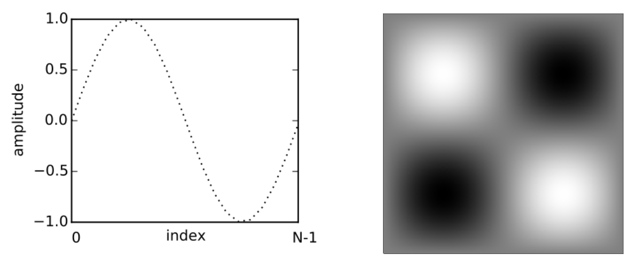
Table lookup techniques are widely used in many sound synthesis applications today as an efficient technique for signal generators (Roads 1996). In this paper, we propose methods to generate volumetric representations of data as three dimensional lookup tables for sound synthesis, based on previous research and experiments in sound synthesis by means of two-variable functions (Mitsuhashi 1982) (Aldo Borgonovo 1986). We introduce the term ***wave voxels***[1] to denote three dimensional lookup tables for sound synthesis.

An overview of 1D and 2D table lookup techniques in sound synthesis.

A one dimensional lookup table of length N is represented graphically in two dimensions as illustrated in figure 1a, where amplitude values (y-axis) changes through time (x-axis). x-axis spans from 0 to $N-1$, while y-axis stores the appropriate amplitude at location n of x-axis. A one dimensional lookup table with amplitude values for one cycle of an arbitrary wave is called a wavetable (Horner 1997) (Roads 1996). Indexing operations for a one dimensional lookup table occurs in one axis. For the purpose sound synthesis, consider a lookup table of length N containing amplitude values for one cycle of a sine wave. To produce a sine wave of frequency $1hz$, we continuously traverse from index 0 to $N-1$ using modulo arithmetic, at a rate of 0 to $N-1$ in 1 second. Traversing twice as fast (0 to $N-1$ in 0.5 seconds) generates a $2hz$ sine wave.

An extension to the wavetable was formally introduced in (Mitsuhashi 1982) as an alternative to frequency modulation (FM) synthesis. This method extends a one dimensional lookup table into two dimensions.

[1] Voxel is a portmanteau for volume and pixel.



(a) a wavetable (b) a wave terrain

Figure 1: An example of a 1D (a) and 2D (b) lookup table for sound synthesis.

Another technique employing two dimensional lookup tables is Scanned Synthesis, introduced by Bill Verplank, Max Mathews and Rob Shaw at Interval Research between 1998 and 2000 (Bill Verplank 1999). Two dimensional lookup tables are graphically represented in three dimensions. In a 2D wavetable of size N_x by N_y, location on x-axis (loc_x) spans from 0 to N_x and location on y-axis (loc_y) spans from 0 to N_y. z-axis stores amplitude values at coordinate (loc_x, loc_y).

A graphical representation of a two dimensional lookup table is possible using a three dimensional mesh surface where the height of a vertex at coordinate (loc_x, loc_y) represents amplitude on z-axis. Alternatively, it could be visualized as a two dimensional plot where a color at coordinate (loc_x, loc_y) represents amplitude on z-axis, as illustrated in figure 1b. A two dimensional lookup table is called a wave terrain in computer music terminology (John Bischoff 1978)(Roads 1996). For two dimensional surfaces, both x and y axes are used for indexing operations. Trajectory of an indexing operation used to read amplitude values in a wave terrain is called an orbit (Aldo Borgonovo 1986).

There are many implementations for generating wave terrains. Y.Mitsuhashi, A.Borgonovo and R.Golds implementation focussed on trigonometric polynomials for terrain generation (Mitsuhashi 1982) (Aldo Borgonovo 1986), A. Di Scipio experimented with functional iterations (Scipio 2002), H.Mikelson uses the Julia set as terrains (Mikelson 1999), D.Overholt fabricated a hardware interface for generation of user defined terrains (Overholt 2002), while R.Dannenberg and T. Neuendorffer uses real-time video images (Roger

B. Dannenberg 2003). Nearly all of the techniques mentioned employs unique indexing operations, each suited to their respective intended purposes.

The extension of wavetables to wave terrains by adding a second dimension opens up new possibilities. Consider a one dimensional lookup table (wavetable). The indexing operation is only possible in one axis, where a vector may only move forwards or backwards. Adding a second dimension (wave terrain) opens up more degrees of freedom in movement. Besides moving forwards or backwards, a two dimensional space allows a vector to move to the left, or right. A third dimension (wave voxel) allows additional movement along the vertical axis, upwards or downwards, besides the capability of moving forwards, backwards, and to the left and right. Each of these movements translates to a unique sonic characteristic when synthesized.

Wave voxel

A voxel represents a single value on a regular grid in three dimensional space (Huges et al. 2013). Voxel models are volumetric renderings commonly used in medical imaging applications, simulation of scientific data and in video games. A voxel model of size N_x, N_y and N_z contains $N_x \times N_y \times N_z$ individual voxels, while a set of voxels is called a voxel stack. The term wave voxels refers to voxel models catered for sound synthesis applications.

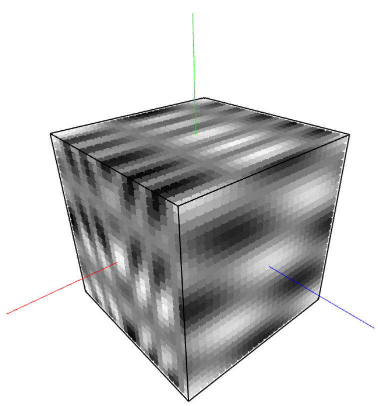


Figure 2: An example of a stack of wave voxels

Voxel stack

The resolution of a voxel stack is its size ($N_x \times N_y \times N_z$). A high resolution stack yields smoother amplitude transition from one voxel to the next, albeit with significantly larger memory requirements. Figure 2 illustrates a stack of wave voxels. Color of each voxel represents amplitude value stored in the three dimensional lookup table at corresponding locations. Amplitudes between 1.0 and -1.0 are mapped to color values 0 (black) to 255 (white).

As an example, the following pseudocode generates a voxel stack using a single cycle of cosine wave on all three axis. For each voxel, $voxel(x, y, z)$ in a voxel stack of size N_x by N_y by N_z,

Algorithm 1 Cosine wave voxel model

for $z = 0$, $z < N_z$, $z++$ **do**

$$val_z = cos(2\pi(z/N_z))$$

for $y = 0$, $y < N_y$, $y++$ **do**

$$val_y = cos(2\pi(y/N_y))$$

for $x = 0$, $x < N_x$, $x++$ **do**

$$val_x = cos(2\pi(x/N_x))$$

$$voxel(x, y, z) = val_x \times val_y \times val_z$$

A voxel stack can be viewed as multiple layers of wave terrains stacked on top of each other. Figure 3 illustrates three slices along z axis of said voxel stack at location $z = 0$, $N_z/2$ and $N_z - 1$.

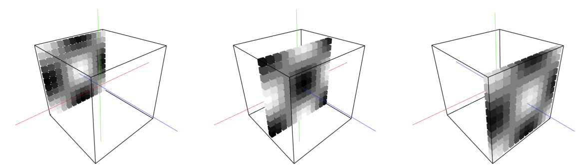


Figure 3: Slices along z axis of a voxel stack [Left to right] voxel(x, y, 0), voxel(x, y, N/2), voxel(x, y, N-1)

Indexing a voxel model is done in three axes. Just as indexing operations for two dimensional surfaces, there are multiple possible approach for orbit definition. Certain approach to indexing might be more suitable for specific purposes. For example, sequential scanning might be more useful for data signification projects, while a more dynamic path might suit artistic applications better.

One way of indexing a stack of wave voxels is by adding a third axis (z) to the orbit equations for wave terrains as presented in (Mitsuhashi 1982) and (Aldo Borgonovo 1986). This third axis would share the same number of variables as both x and y axes, as shown in equations 1, 2 and 3.

$$x = 2f_x t + \phi_x + A sin(2\pi F_x t + \varphi_x) \tag{1}$$

$$y = 2f_y t + \phi_y + B sin(2\pi F_y t + \varphi_y) \tag{2}$$

$$z = 2f_z t + \phi_z + C sin(2\pi F_z t + \varphi_z) \tag{3}$$

In the equations above, t denotes time, f_x, f_y and f_z denotes linear frequency and ϕ_x, ϕ_y and ϕ_z denotes linear phase. While A, B and C denotes amplitude, F_x, F_y and F_z denotes frequency and φ_x, φ_y and φ_z denotes initial phase. It is important to note that equations 1, 2 and 3 presents only **one out of many** possibe ways of indexing a three dimensional lookup table. We chose this particular implementation by Mitsuhashi and Borgonovo as an example because it presents a degree of flexibility and simplicity in defining an orbit trajectory.

Multimodality of voxel stack contents

Voxel models for sound synthesis could be constructed using different types of sources. In this section, we present four identified examples of sources along with the proposed application.

Trigonometric In voxel stacks shown in figure 2 and 3, the contents of both stacks were generated mathematically using trigonometric and windowing (tapering) functions.

As shown in algorithm 1, individual voxels in a stack are the product of sample-wise values from signals on each axis. With trigonometric functions, this method is akin to combining both frequency modulation (FM) and amplitude modulation (AM) synthesis. This approach would be suitable for computer music applications.

Audio samples Each axis of a voxel stack could also store a short segment of recorded audio. A voxel model of size N_x, N_y, $N_z = 512$ could store 512 samples of audio at each axis.

These 512 samples could be continuously extracted from a longer audio recording. Samples on each axis could originate from different locations of the same audio file, or from different sources altogether.

Each axis could also traverse an audio file at different rates and directions. This method can be thought of as three dimensional sample granulation, another example of an approach suitable for computer music applications.

Video A live camera feed could be used as the source for generation of voxel stack contents. The resolution of these sequence of images should be scaled to the desired voxel model size. A voxel stack of size N_x, N_y, $N_z = 512$ could store 512 frames of 512px by 512px image sequence.

A voxel stack could be created by layering each frame of the resized image sequence. For example a video stream of size 640px by 420px needs to be interpolated and decimated appropriately in order to construct a 512px by 512px image. A streaming rate of 30fps and a stack depth of 512 voxels would store around 17 seconds of video.

Alternatively, a multiple camera configuration could be used to computationally reconstruct a volumetric scene (Slabaugh et al. 2001). This approach might be more suitable in a more controlled environment, perhaps in an interactive audiovisual installation and/or interactive performance setup.

Recorded & simulated data Recorded data such as MRI scans and simulated data such as smoke and fluid simulations could also be used as voxel stack contents. Audification of such data could be useful for research in auditory displays.

Conclusion and future work

We presented methods for generating volumetric representations of data as three dimensional lookup table for sound synthesis. We addressed the multimodality of this technique, suggesting suitable sources for creation of voxel models for use in computer music applications, as an element in artistic installation projects and data sonification projects.

Another interesting prospect to explore is the creation of hybrid voxel models. A stack of voxels could be created using an audio sample on one axis while using trigonometric functions on the other two axes.

For future work, we will also be exploring different indexing techniques for use with wave voxel models. We are interested in understanding how different methods of indexing operation behaves harmonically. We hope to study and understand spectral characteristics of different indexing techniques in order to distinguish and match the appropriate possible techniques with an intended application.

References

Aldo Borgonovo, G. H. 1986. Sound synthesis by means of two-variable functions: Experimental criteria and results. *Computer Music Journal* 10(3):57–71.

Bill Verplank, M. M. 1999. Scanned synthesis. *Interval Research Corporation*.

Horner, A. 1997. A comparison of wavetable and fm parameter spaces. *Computer Music Journal* 21(4):55–85.

Huges, J.; Dam, A.; McGuire, M.; Sklar, D.; Foley, J.; Feiner, S. K.; and Akeley, K. 2013. *Computer Graphics: Principles and Practice*. Addison-Wesley Professional, 3 edition.

John Bischoff, Rich Gold, J. H. 1978. Music for an interactive network of microcomputers. *Computer Music Journal* 2(3):24–29.

Mikelson, H. 1999. Sound generation with the julia set. *CSound Magazine* Summer.

Mitsuhashi, Y. 1982. Audio signal synthesis by functions of two variables. *Journal of the Audio Engineering Society* 30(19):701–706.

Overholt, D. 2002. New musical mappings for the matrix interface. *International Computer Music Conference* 486–488.

Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, Massachusetts: MIT Press.

Roger B. Dannenberg, T. N. 2003. Sound synthesis from real-time video images. In *International Computer Music Conference*, 385–388. International Computer Music Association.

Scipio, A. D. 2002. The synthesis of environmental sound textures by iterated nonlinear functions, and its ecological relevance to perceptual modeling. *Journal of New Music Research* 2(32):109–117.

Slabaugh, G.; Culbertson, B.; Malzbender, T.; and Schafer, R. 2001. A survey of methods for volumetric scene reconstruction from photographs. *The International Workshop on Volume Graphics*.