# Gestural Sounds by Means of Wave Terrain Synthesis

**Anderson Mills and Rodolfo Coelho de Souza**
The University of Texas at Austin

## Abstract

Wave terrain synthesis can be used as an efficient tool for generating sound objects with time varying spectra to create a musical gesture. To accomplish this goal some changes need to be made to Mitsuhashi's original approach to wave terrain synthesis. These changes include less limited terrain functions, complex orbital paths, windowing corrections, and displacement of a pair of orbits defining a stereo sound. An example of a CSound implementation is given.

## 1 INTRODUCTION

The technique of sound generation by means of two variable functions, also known as wave terrain synthesis, is a field that has not yet been explored to its full musical potential. The technique was devised by Mitsuhashi (Mitsuhashi 1982) as an alternative to Chowning's method of FM synthesis (Chowning 1973), considering the goal of hardware implementation, but the technique has also been implemented in software as demonstrated by Borgonovo (Borgonovo and Haus 1986), Nelson (Nelson 1998), and Pinkston (Pinkston 1999). The primary goal of Mitsuhashi's research was to emulate natural sounds or other standard synthesis techniques. Though wave terrain synthesis is a versatile tool that can produce results equivalent to many other synthesis techniques, like FM, AM and wave shaping, it attracted our attention because of the unique results which it can generate.

Roads (Roads 1996) summarizes the technique describing its two basic elements: the choice of a three-dimensional table that can be visualized as a topographic terrain representing amplitudes built by vector product of two-variable functions and a table look up procedure following a chosen path or orbit. These orbits determine the character of the resulting sound object, particularly its development over time and the creation of distinctive shapes outside of the framework of traditionally conceived musical sounds. This implies the generation of gestural sound objects by direct digital synthesis rather than by the *musique concrete* technique of distortion of recorded sounds.

A gesture is defined by Hatten (Hatten 1999) as movement that may be interpreted as significant. As the musical meaning does not point to external references but to relations inside the musical discourse, Smalley (Smalley 1986) emphasizes the importance of spectro-morphological design to control spectral and dynamic shaping of sound objects to create real and imagined motions without the need for spatial movement. The idiomatic procedures of wave terrain synthesis allow and almost imply the generation of many kinds of directional or moving sounds. Therefore, wave terrain synthesis can be considered a gestural synthesis technique by its nature.

## 2 TERRAIN

Because Mitsuhashi uses tables that wrap beginning to end, he proposes many restrictions for the functions that can be used to build the terrain. He recommends that the functions and their first-order partial derivatives in space should be continuous in the area of definition. He also specifies that the values of the functions on the boundaries should be zero and have the same partial derivative in space on the two boundaries. He is concerned that the result should provide smooth waveforms even when a jump occurs from one boundary to another. Borgonovo and Haus followed the path recommended by Mitsuhashi. Their implementation is on a Fairlight platform and experiments with three new functions following the principles suggested by Mitsuhashi as well as his original two.

As Roads points out, these restrictions are necessary if the terrain table is to be used to generate predictable waveforms, particularly when using periodic orbits that extend beyond a single table. When the aim of the technique is to produce more complex output (which happens simply by using time-varying orbits), Pinkston's alternative implementation implies that many of these restrictions might be relaxed and the terrain table may be filled with more complex functions or data. Nevertheless the restriction that the boundary values should have a constant value (not necessarily 0 but also 1 or any other) is convenient for assuring continuity of the waveform when a complex orbit wraps around the table.

For this implementation the terrain is described by the vector product of the function sin(x^4) in both *x* and *y* axis. This function is chosen because it has a varying spectral density that depends upon the value of the independent variable. Any terrain space defined will therefore be “quietest” nearest the origin and “noisiest” farthest from the origin. An orbit moving through these areas allows the variation of spectra in time. The next section will show how appropriate orbits create the gestural movement previously mentioned. Note that because sin(x^4) is a function defined for all x, there is no wrapping of the table as in Mitsuhashi’s implementation.

## 3 ORBITS

The orbits researched by Mitsuhashi and Boronovo were chosen to create a direct relation with the fundamental frequency of the resulting sound. They describe simple trajectories over the terrain table which are a combination of a linear and a periodic term. Though the implementation listed below is capable of reproducing their results, the gestural aspect of wave terrain synthesis is achieved by summing both a slow and fast periodic orbit. A lunar orbit around the sun proves to be a useful model for visualizing this summation. Such an orbit describes a compound path formed by a larger circular or elliptical path, usually in slower, sub-audio rate, and a local smaller and faster circular orbit. We extend this local orbit to rectangular motion to correct for the deviation in the frequency content caused by circular motion in a rectilinear terrain table. In this approach, the small fast orbit is related to the perception of the fundamental frequency while the large slow orbit is related to the evolution of the changes of the harmonic spectra. In our implementation a slow orbit which passes through different areas of the terrain causes the gestural changes we have described.

## 4 WINDOWING

Some experiments were done with different small orbit shapes, and it was found that rectangular orbits with linear paths over a known function on each side allow the best control of the harmonic content over time. Unfortunately rectangular orbits lead to a problem due to the change of directions on each corner of the rectangle causing discontinuities in the wave shape's first order partial derivative in time. These cusps on the wave shape have the effect of adding a band of non-harmonic high partials (a parasitic buzz). To avoid this artifact without resorting to filters a windowing function is applied to each side of the rectangular orbit that reduced the amplitude at the corners of the small orbit and eliminated the cusps. This windowing solution controls the problem introducing a form of amplitude modulation.

## 5 SPATIAL LOCALIZATION

Another effect that is implemented in this approach to wave terrain synthesis is spatial movement through the use of displaced slow orbits without the use of specific panning controls. Two different techniques prove significant: slow orbit time displacement and slow orbit terrain-space displacement. The idea is to provide each stereo channel with a slightly displaced orbit from the same terrain table. As our CSound implementation uses phasor functions accessing the terrain table to act as oscillators, a small time difference between the phasors of the two stereo channels can generate significant changes in sound localization as the orbits cross different parts of the terrain. The second approach uses a small spatial offset between the slow orbits. The difference in spectral content between the stereo channels due to spatial displacement is similar to the effect created by time displacement, though as the spatial offset becomes larger, the two stereo channels become dissociated. It is interesting to note that small differences in the visual field create a sense of visual depth and spatial perspective and small differences in the aural field create a sense of aural depth and physical space.

## 6 SUMMARY

We have shown several new aspects to wave terrain synthesis here including less limited terrain functions, more complex orbits (and windowing corrections of undesired artifacts caused by rectangular orbits), and stereo movement created by slightly displaced orbits. Our research shows that there is much more ground to explore in wave terrain synthesis and its potential application to music composition.

## 7 AN EXAMPLE OF CSOUND IMPLEMENTATION OF A WAVE TERRAIN GESTURAL SOUND

```
; terrain.orc

        sr      =       44100
        kr      =       44100
        ksmps   =       1
        nchnls  =       2

        ; for this implementation krate must equal arate

                instr   1

        ; initialization
        idur    =       p3      ;duration
        iamp    =       p4      ;amplitude
        ifsarfn =       p5      ;fast orbit aspect ratio function (x/y)
        ifsfqfn =       p6      ;fast orbit frequency function
        ifsdsfn =       p7      ;fast orbit distance function (center to x side)
        ixlslfn =       p8      ;slow orbit left x function
        iylslfn =       p9      ;slow orbit left y function
        ixrslfn =       p10     ;slow orbit right x function
        iyrslfn =       p11     ;slow orbit right y function
        ixangfn =       p12     ;wave terrain x function
        iyangfn =       p13     ;wave terrain y function
        iwinfn  =       p14     ;window function
        isquafn =       p15     ;square function (must be-1to-1to1to1to-1)

        ; constants
        ixmode  =       1               ;tables go from 0 to 1
        inoff   =       0               ;no offset to the tables
        iwrap   =       1               ;wrap the tables as necessary
        isine   =       1               ;function 1 contains a sine
        iquawv  =       0.25            ;quarter offset
        ihalfwv =       0.5             ;half wave offset
        i3quawv         =       0.75            ;three quarter offset
        ipi     =       3.14159265359   ;pi
        ifour   =       4               ;there are four sides to a rectangle
```

```
          irise     =         0.01                ;rise time
          idecay    =         irise               ;decay time

          ; performance - a ramp at the audio rate that goes from 0 to 1
          aramp     line      0.0, idur, 1.0
          kramp     line      0.0, idur, 1.0

          ; start looking up from tables - slow orbits
          axlslo    tablei    aramp, ixlslfn, ixmode
          aylslo    tablei    aramp, iylslfn, ixmode
          axrslo    tablei    aramp, ixrslfn, ixmode
          ayrslo    tablei    aramp, iyrslfn, ixmode

          ; the fast orbit
          ; the aspect ratio function for the fast orbit
          kasrfas   tablei    kramp, ifsarfn, ixmode
          ; the frequency function for the fast orbit
          kfrqfas   tablei    kramp, ifsfqfn, ixmode
          ; the distance function for the fast orbit
          adstfas   tablei    aramp, ifsdsfn, ixmode

          ; a phasor that changes rate according to aspect ratio
          ; pass from 0to0.25 in kxunit, then 0.25to0.5 in kyunit and repeat to one
          kfasorb   phasor    kfrqfas
          kunit     =         2*kasrfas + 2
          kyunit    =         1/kunit
          kxunit    =         kasrfas/kunit

          if        kfasorb   < kxunit       kgoto undera
          if        kfasorb   < ihalfwv       kgoto underb
          if        kfasorb   < (ihalfwv+kxunit) kgoto underc
                    kgoto     underd
undera:   avarphs=iquawv*kfasorb/kxunit
                    kgoto     done
underb:   avarphs=iquawv*(kfasorb-kxunit)/(ihalfwv-kxunit) + iquawv
                    kgoto     done
underc:   avarphs=iquawv*(kfasorb-ihalfwv)/kxunit + ihalfwv
                    kgoto     done
underd:   avarphs=iquawv*(kfasorb-ihalfwv-kxunit)/(ihalfwv-kxunit) + i3quawv

done:
          ; now the window function (four times each period at kfrqfas)
          awindex =           avarphs*ifour
          awin      tablei    awindex, iwinfn, ixmode, inoff, iwrap
```

```
        ; the x and y fast portion of the indexes into the terrain
        axfas    tablei    avarphs, isquafn, ixmode, iquawv, iwrap
        ; offset by a quarter to make equiv to cos
        axfas    =         axfas*adstfas
        ayfas    tablei    avarphs, isquafn, ixmode, inoff, iwrap
        ayfas    =         ayfas*adstfas/kasrfas

        ; calculate the indexes into the terrain
        alxin    =         axlslo+axfas
        alyin    =         aylslo+ayfas
        arxin    =         axrslo+axfas
        aryin    =         ayrslo+ayfas

        ; pick the point in the terrain
        axangfc  tablei    aramp, ixangfn, ixmode
        ; get the angular factor for x
        ayangfc  tablei    aramp, iyangfn, ixmode
        ; get the angular factor for y
        alxout   =         sin(alxin*alxin*alxin*alxin*axangfc)
        ; sin (x^4*factor)
        alyout   =         sin(alyin*alyin*alyin*alyin*ayangfc)
        arxout   =         sin(arxin*arxin*arxin*arxin*axangfc)
        aryout   =         sin(aryin*aryin*aryin*aryin*ayangfc)

        ; stereo output for this instrument
        ; amplitude gate
        acompgt            linen     iamp, irise, idur, idecay
        alout    =         alxout*alyout*acompgt*awin
        arout    =         arxout*aryout*acompgt*awin

                 outs      alout, arout
                 endin

;=======================================================

; terrain.sco

; functions
; aspect ratio function
;f      start   size    gen     p1
f05     0.0     8193    7       1.0     8193    1.0
; fast orbit radius function
;f      start   size    gen     p1
```

```
f10     0.0     8193    -7      0.08    8193    0.08
f11     0.0     8193    7       0       8193    1.0
; fast orbit frequency function
;f      start   size    gen     p1
f20     0.0     8193    -7      50      8193    50
f22     0.0     8193    -7      200     8193    200
; slow orbit functions
;f      start   size    gen     p1
f32     0.0     8193    -19     1       0.4     315     0.5
f33     0.0     8193    -19     1       0.4     225     0.5
f34     0.0     8193    -19     1.01    0.4     317     0.45
f35     0.0     8193    -19     1.01    0.4     227     0.5
; angular factor functions
;f      start   size    gen     p1
f70     0.0     8193    -7      60.0    8193    60.0
f71     0.0     8193    -7      60.0    8193    30.0
; square function
;f      start   size    gen     p1
f98     0.0     8192    7       -1.0    2048    -1.0    2048    1.0
        2048    1.0     2048 -1.0
; Hamming window function
;f      start   size    gen     p1
f99     0.0     8192    20      1
; tempo
t0      60
; performance
;ins    start   dur     amp     fsarfn  fsfqfn  fsdsfn
i01     0.0     6.0     20000   5       22      10
;xlslfn ylslfn  xrslfn  yrslfn  xangfn  yangfn  winfn   squafn
32      33      34      35      70      70      99      98
;ins    start   dur     amp     fsarfn  fsfqfn  fsdsfn
i01     6.0     12.0    20000   5       20      11
;xlslfn ylslfn  xrslfn  yrslfn  xangfn  yangfn  winfn   squafn
32      33      34      35      71      71      99      98
e
```

## BIBLIOGRAPHY

Borgonovo, Aldo, and Goffredo Haus. (1986). Sound Synthesis by Means of Two-Variable Functions: Experimental Criteria and Results. *Computer Music Journal*, 10 (3), 57-71.

Chowning, John M. (1973). The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of the Audio Engineering Society*, 21(7), 526-534.

Dodge, Charles, and Thomas A. Jerse. (1997) [2nd ed.]. *Computer Music.* New York: Schirmer Books.

Hatten, Robert S. (1999) *Musical Gesture.* Eight Conferences for the Cyber Semiotic Institute Home Page http://www.epas.utoronto.ca/epc/srb/cyber/hat1.html

Mitsuhashi, Yasuhiro. (1982). Audio Signal Synthesis by Functions of Two Variables. *Journal of the Audio Engineering Society,* 30(10), 701-706.

Moore, F. Richard. (1990). *Elements of Computer Music*. Englewood Cliffs, NJ: Prentice-Hall.

Nelson, Jon Christopher. (1998). *Putting the 'Wave' in Wavetable Synthesis.* http://www.music.unt.edu/comp/jnelson.htm

Pinkston, Russell. (1999). *Example of Wave Terrain Synthesis Instrument.* http://www.utexas.edu/cofa/music/ems/mus329m/ClassStuff/terrain.html

Roads, Curtis. (1996). *The Computer Music Tutorial*. Cambridge, Massachusetts: The MIT Press.

Smalley, Denis. (1986). Spectro-Morphology and Structuring Processes. In Simon Emmerson (Ed.): *The Language of Electroacoustic Music*. New York: Harwood Academic Publishers.