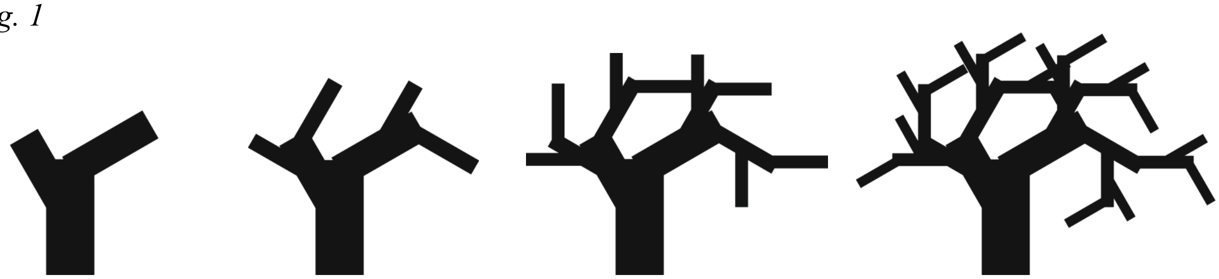
Sonification of Two-dimensional Branching Structures

Adam James Wilson
University of California, San Diego

This paper describes a recursive algorithm designed to sonify two-dimensional branching structures, similar to the fractal structures that can be generated using Lindenmeyer systems. In this experiment, the visual basis for the auditory mapping is a tree. The generator for the tree – the nominal structural building block – consists of a straight-line trunk with a set of shorter straight-line branches attached to it. The construction rules are simple: with each recursion, the branches of the generator are replicated on a progressively smaller scale around each of the branches drawn in the previous recursion. The figure below shows several recursions – including recursion 0, the generator – in the construction of a tree with 30º and -60º branches attached at heights 0.5 and 0.9, respectively, to a trunk of height 1.0. Angles are specified relative to the trunk: the top of the trunk points at 0º and the base points at 180º. With each recursion, identified by its index R, a scaling factor of 0.8^R is applied to the lengths of the new branches, and branches diminish in thickness according to a function $s(\alpha, R)$, whose content will be described later, in which α denotes the thickness of the trunk.

Fig. 1



$R = 0$ $R = 1$ $R = 2$ $R = 3$

A number of qualitative criteria were established in order to guide the formulation of relationships between visual and auditory domain representations of trees belonging to the class of trees exemplified in Fig. 1:

- *Every visual category admitted should be assigned a corresponding category in the auditory domain*. The sonification model described herein operates on line length, line thickness, location of line intersections, and measures of angles produced by line intersections. Categories omitted from this model include formations of negative space and cognitive segmentations of the image based on the recognition of similar structural components comprised of line groups.

- *Visual parameters should map directly to perceivable auditory phenomena*, and should not be converted into higher-level forms that may incidentally influence or indirectly control the production of basic auditory phenomena such as pitch, time, and amplitude. In other words, no mapping should seem trivial or extraneous, employed solely to satisfy the first criterion, and no mapping should incorporate a level of parametric abstraction that cannot be immediately related by a listener to the surface acoustic of the resultant sound object. Simplicity and directness should be maximized, and no mapping should be attempted without carefully considering the appropriateness of its implied visual-to-auditory analogy.

- *Complete mappings should be presented within a window of time not to exceed 30 seconds.* This ensures that all data can easily be held in short term memory, and precludes any notion of working with large-scale formal time structure, in violation of the second criterion.

The most difficult parameter to deal with when translating from the visual to the auditory domain is time, since a picture presents all of its data at once, while music presents its data over a span of time. Time in this case is oriented to the direction of growth: the trunk represents a pitch to be played at time = 0. The node, or root, at which each branch attaches to its parent represents a point in time corresponding to the distance between the base of the trunk and the intersection of the trunk with a perpendicular line drawn between the trunk and the node. The location of this point is actually determined by multiplying the cosine of the angle of the branch's parent relative to the trunk by the distance between the root of the parent branch and the root of the branch in question, and then adding the result to the point in time given by the root of the parent branch.

Fig. 2 on the following page shows the values of all angles and nodes produced over the three levels of recursion depicted graphically in Fig. 1. Consider $n_{1,0}$ in Fig. 2. The subscripts in $n_{1,0}$ indicate, respectively, the recursion index R to which n belongs, and the index of n within the set of nodes produced by recursion R. The value of $n_{1,0}$, 0.85, is given by a function $f(x,R,y,a,p)$, where x is the branch scaling factor, R is the recursion index, y is the branch attachment point for the analogous branch in the generator – in this case, odd values in the second subscript slot of n correspond to $n_{0,1}$ and even values correspond to $n_{0,0}$; a is the angle of the parent branch relative to the trunk, and p is the n value for the parent branch:

$$f(x,R,y,a,p) = x^R y \cos a + p; \text{ therefore, } n_{1,0} = 0.8^1 0.5 \cos 30^\circ + 0.5 = 0.85$$

While branch nodes become time-points, their associated angles become pitches that occur at these time-points. All angles are expressed relative to the trunk. For example, a branch attached at 30º to a parent that is itself attached to the trunk at 30º is said to have an angle of 60º. Furthermore, since the upper pitch boundary of the tree is given by 180º and the lower pitch boundary of the tree is given by -180º, all angles above 180º and below -180º are converted to values between -180º and 180º, inclusive. Accordingly, each angle θ produced by the addition of a new angle of intersection and its parent branch's associated angle is filtered through $f(\theta)$ below prior to translation into pitch.

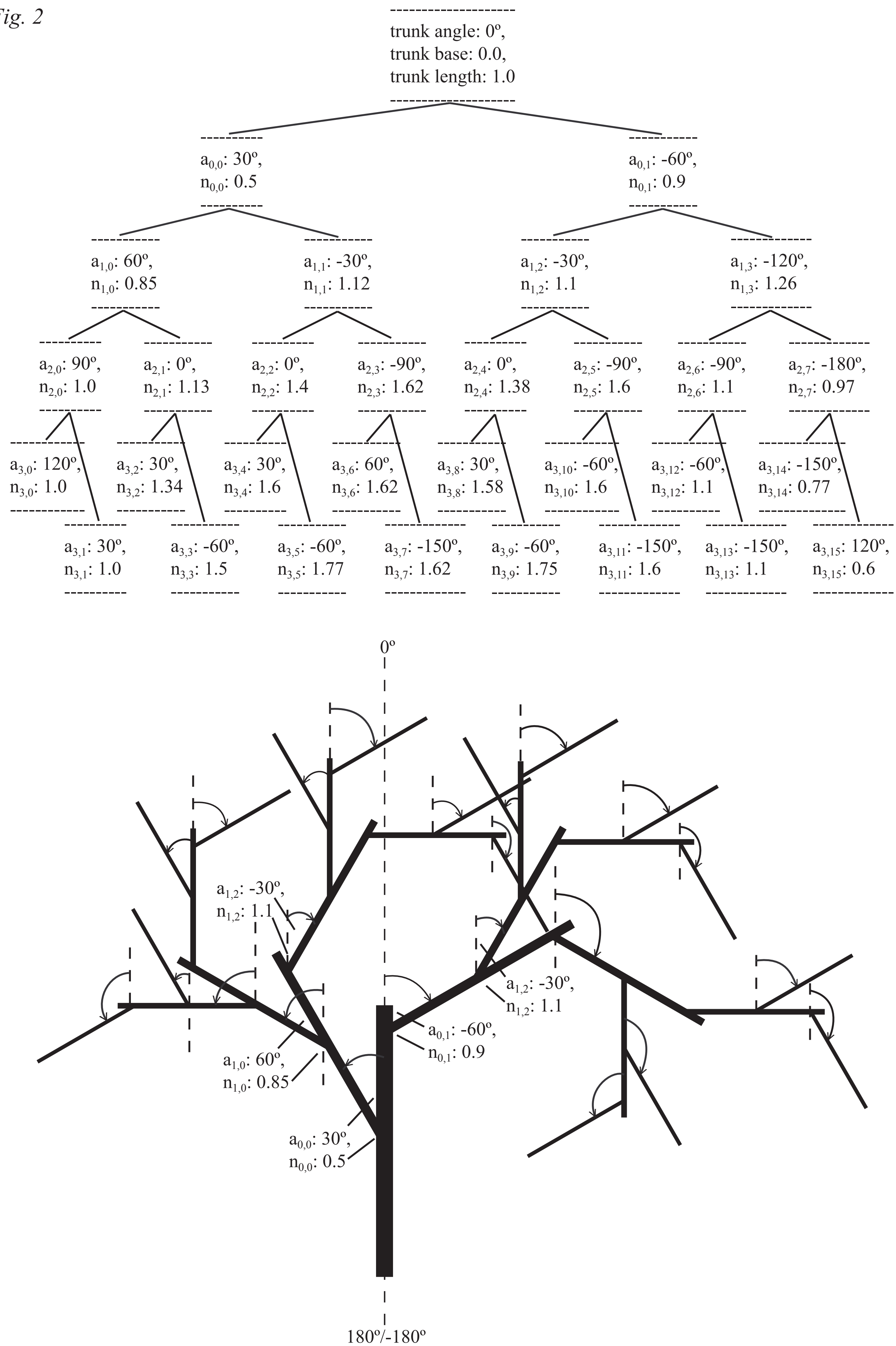
$$f(\theta) = \begin{cases} 0, \textit{ if } |\theta| \bmod 360 = 0; \\ \theta \bmod \dfrac{\theta}{|\theta|}360 - 360\dfrac{\theta}{|\theta|}\left|\left\lfloor \dfrac{180-|\theta|}{180} \right\rfloor\right|, \textit{ if } 0 < |\theta| \bmod 360 < 360. \end{cases}$$

Pitch translation of $f(\theta)$ is achieved with a function $f(\lambda,\beta,\delta,\theta)$, where λ is the "center" frequency assigned to the trunk, β is the maximum coefficient of λ that can be applied to produce a new pitch (equivalent to the upper-bound, or 1/lower-bound), δ is the number of equal divisions of the pitch interval represented by β, and θ is the angle fed to $f(\theta)$:

$$f(\lambda,\beta,\delta,\theta) = \lambda\beta^{\left(\frac{\left\lfloor \delta \frac{f(\theta)}{180} + 0.5 \right\rfloor}{\delta}\right)}$$

Amplitudes are assigned to pitches based on a user-defined function $s(\alpha,R)$, where α is the trunk thickness (the upper boundary for amplitude), and R is the index of the recursion in

Fig. 2



trunk angle: 0º,
trunk base: 0.0,
trunk length: 1.0
$a_{0,0}$: 30º, $n_{0,0}$: 0.5
$a_{0,1}$: -60º, $n_{0,1}$: 0.9
$a_{1,0}$: 60º, $n_{1,0}$: 0.85
$a_{1,1}$: -30º, $n_{1,1}$: 1.12
$a_{1,2}$: -30º, $n_{1,2}$: 1.1
$a_{1,3}$: -120º, $n_{1,3}$: 1.26
$a_{2,0}$: 90º, $n_{2,0}$: 1.0
$a_{2,1}$: 0º, $n_{2,1}$: 1.13
$a_{2,2}$: 0º, $n_{2,2}$: 1.4
$a_{2,3}$: -90º, $n_{2,3}$: 1.62
$a_{2,4}$: 0º, $n_{2,4}$: 1.38
$a_{2,5}$: -90º, $n_{2,5}$: 1.6
$a_{2,6}$: -90º, $n_{2,6}$: 1.1
$a_{2,7}$: -180º, $n_{2,7}$: 0.97
$a_{3,0}$: 120º, $n_{3,0}$: 1.0
$a_{3,1}$: 30º, $n_{3,1}$: 1.0
$a_{3,2}$: 30º, $n_{3,2}$: 1.34
$a_{3,3}$: -60º, $n_{3,3}$: 1.5
$a_{3,4}$: 30º, $n_{3,4}$: 1.6
$a_{3,5}$: -60º, $n_{3,5}$: 1.77
$a_{3,6}$: 60º, $n_{3,6}$: 1.62
$a_{3,7}$: -150º, $n_{3,7}$: 1.62
$a_{3,8}$: 30º, $n_{3,8}$: 1.58
$a_{3,9}$: -60º, $n_{3,9}$: 1.75
$a_{3,10}$: -60º, $n_{3,10}$: 1.6
$a_{3,11}$: -150º, $n_{3,11}$: 1.6
$a_{3,12}$: -60º, $n_{3,12}$: 1.1
$a_{3,13}$: -150º, $n_{3,13}$: 1.1
$a_{3,14}$: -150º, $n_{3,14}$: 0.77
$a_{3,15}$: 120º, $n_{3,15}$: 0.6
0º
$a_{1,2}$: -30º, $n_{1,2}$: 1.1
$a_{1,2}$: -30º, $n_{1,2}$: 1.1
$a_{0,1}$: -60º, $n_{0,1}$: 0.9
$a_{1,0}$: 60º, $n_{1,0}$: 0.85
$a_{0,0}$: 30º, $n_{0,0}$: 0.5
180º/-180º

which the pitch was produced. In general, amplitude must decrease with each recursion; however, $s(\alpha, R)$, can be defined as a steep or shallow logarithmic decrease, a linear decrease, or can take the form a lookup table of individually chosen amplitudes.

At the end of tree construction, time-points are sorted from least to greatest along with their associated pitch and amplitude values. The resultant tree can be stretched or compressed in time to unclutter or increase the density of a tree structure. Events can also be quantized to a pulse, and intervals between notes can be normalized – all prior to MIDI output of the tree structure.

Below is a call to "make-tree-1," a LISP function that encapsulates the algorithm described in this paper. Each option given in the function call for "distorting" the tree prior to output is explained in the comments to the right.

```
(make-tree-1
    :trunk-frequency 440             ;; Trunk frequency (λ).
    :angles '(30 -40 67 -170)        ;; Initial angles.
    :nodes '(.5  .2 .9 .8)           ;; Initial nodes.
    :node-scalar .99                 ;; Branch scaling factor (x).
    :equal-temperament 17            ;; Divisions of λ (δ).
    :temperament-base 2.1            ;; Maximum coefficient of λ.
    :recursions 3                    ;; Number of recursions.
    :channelize? 16                  ;; Number of MIDI channels to use.
    :debugging? t                    ;; Print debugging info.
    :new-length? 10                  ;; Specify a new total duration in seconds
                                     ;; for the complete tree.
    :new-divisions? 100000           ;; A number of divisions of the tree
                                     ;; duration to quantize onsets to.
    :override-res-test? T            ;; New time resolution set by
                                     ;;"new-divisions?" must equal or
                                     ;; exceed current res. unless
                                     ;; "override-res-test?" == true.
    :quantize-durs-to-avg? 0.5       ;; "Normalize" time between events by a
                                     ;; percentage of the average time
                                     ;; between each adjacent pair of notes.
    :amp-func?  nil                  ;; A user-defined amplitude function
                                     ;; (defaults to imbedded lookup table).
)
```