THE WABLET: SCANNED SYNTHESIS ON A MULTI-TOUCH INTERFACE

Robert Tubb, Anssi Klapuri and Simon Dixon

The Centre for Digital Music
Queen Mary University of London
London, UK
robert.tubb@eecs.qmul.ac.uk

ABSTRACT

This paper presents research into scanned synthesis on a multi-touch screen device. This synthesis technique involves scanning a wavetable that is dynamically evolving in the manner of a mass-spring network. It is argued that scanned synthesis can provide a good solution to some of the issues in digital musical instrument design, and is particularly well suited to multi-touch screens.

In this implementation, vibrating mass-spring networks with a variety of configurations can be created. These can be manipulated by touching, dragging and altering the orientation of the tablet. Arbitrary scanning paths can be drawn onto the structure.

Several extensions to the original scanned synthesis technique are proposed, most important of which for multi-touch implementations is the freedom of the masses to move in two dimensions.

An analysis of the scanned output in the case of a 1D ideal string model is given, and scanned synthesis is also discussed as being a generalisation of a number of other synthesis methods.

1. INTRODUCTION

Direct control of timbre has been a dream of experimentalists since the very first investigations into electronic sound synthesis. Once the sound wave could be recorded, visualised and manipulated, the possibility arose that by carefully designing waveforms, any imaginable sound texture could be produced. Indeed the very first "artificial sound" was produced by *hand drawing* waveforms onto optical film [1][2]. The unwieldiness of this approach soon became apparent, and the project of completely specifying time-domain signals was largely abandoned. Higher level control of the *spectral* properties of the sound, being much easier and more intuitive to conduct in real-time, became the norm. However, with the increasing sophistication of human-computer interfaces, it may be that direct interactions with the waveform may yet experience a renaissance.

Scanned Synthesis was developed in 1999 at Interval Research Inc. by Bill Verplank, Max Matthews and Rob Shaw [3]. The method consists of reading audio sample values from a dynamically changing wavetable. This wavetable usually consists of a vibrating physical model, evolving at low frequencies that humans can interact with: typically from 0 to 15Hz. These are referred to as "Haptic Frequencies", examples of which might be the travelling waves produced by shaking the end of a taut rope, or ripples on the surface of a pool. The shape of these waves is scanned in a cyclical fashion at the frequency of the desired musical note.

Further work was carried out by Boulanger et al. [4] and Couturier [5], but there seems to have been few recent publications, and the technique remains relatively unexplored compared to many other synthesis methods. In particular there has been no investigation into exploring the technique with recent multi-touch screen devices[1]. There is a clear rationale for attempting this combination: the vibrating model from which the wavetable is obtained is evolving at "haptic" rates, and is thus both visualisable as an animated object on a screen, and manipulable by hand. It seems to make sense then to connect these two capabilities in an immediate and intuitive way, i.e. multiple fingers directly interacting with the structure, creating vibrations and deformations that will have a tangible effect on the sound output.

There have been a number of papers on multi-touch musical interfaces, many choose to focus more on the potential for "social music" on phone size devices [7][8] or large scale multi-touch interfaces [9][10]. These instruments often take the approach of using pre-designed sounds - the interface provides ways to trigger these sounds or arrange melodic patterns. Overall it seems there could be more investigation into novel synthesis algorithms that enable expressive touch control of timbre.

2. DESIGN AND IMPLEMENTATION

Figure 1 shows an overview of the system, which consists of an application running on the iPad, and a Max/MSP controller interface on a separate machine.

Simulating large 2D mass-spring networks can be computationally intensive, however sending the positions of all the masses and springs over wireless to the tablet would also be difficult. As it was also desirable to have a functioning stand-alone application, the physical modelling calculations were carried out on the iPad.

The application was written in C++ making use of the open frameworks library for graphics calls [11].

Control data such as MIDI note pitch and parameter changes affecting the physical model could be sent to the app via OSC (Open Sound Control [12]) messages, thus the Wablet can integrate with other music production software.

2.1. The Mesh

The central part of this design is the "Mesh". This is a two dimensional mass-spring network. The mesh object contains all the spring and masses (called here "Lumps"), and keeps track of the connections between them. Different structures and sizes of Mesh are selectable by the user, and can produce very different behaviours and sounds. Some examples of mesh configurations are shown in figures 2, 3 and 4 .

[1] Although in [6], Couturier and Arfib used a glove style multi-touch system to interact with the scanned string.

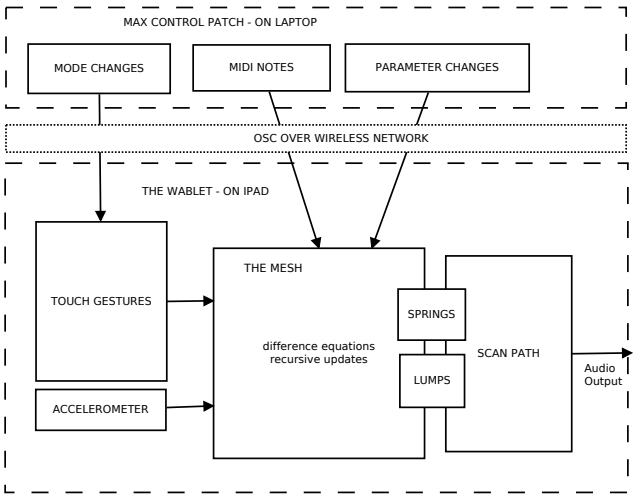

Figure 1: An overview of the design of the Wablet.

The Mesh also deals with user interactions that affect the lumps and springs, such as grabbing, striking and damping.

Lumps are objects representing point particles that have inertia, therefore it is the lumps that take care of the calculations needed for Newton's 2nd law [13], given by

$$\ddot{\mathbf{x}} = \frac{1}{m}\mathbf{f}, \tag{1}$$

with $\ddot{\mathbf{x}}$ being the acceleration vector, m the mass of the lump and $\mathbf{f}$ the force vector. A simple Euler finite difference integration method was used to update the velocity and position at each time step. We make the approximation

$$\dot{\mathbf{x}} \approx \delta_{t+}\mathbf{x} = \frac{1}{h}(\mathbf{x}[n+1] - \mathbf{x}[n]), \tag{2}$$

for the first derivative of the position vector $\mathbf{x}$, and similarly

$$\ddot{\mathbf{x}} \approx \delta_{t+}\mathbf{v} = \frac{1}{h}(\mathbf{v}[n+1] - \mathbf{v}[n]), \tag{3}$$

for the time derivative of $\mathbf{v}$, the velocity vector. Here n is the time index and h is the small time between update frames. Normalising time units such that $h = 1$ and rearranging gives the simple update rules,

$$\mathbf{v}[n+1] \leftarrow \mathbf{v}[n] + \ddot{\mathbf{x}} \tag{4}$$

$$\mathbf{x}[n+1] \leftarrow \mathbf{x}[n] + \mathbf{v}[n+1]. \tag{5}$$

More sophisticated and accurate finite difference schemes were investigated, but in practice there is no real need for numerically accurate physical behaviour as this is not a "realistic" instrument.

A major drawback of this finite difference scheme is the CFL (Courant-Friedrichs-Lewy) condition [14]. If the mass was set too low, or the spring constant too high the Mesh would become unstable and blow up. For very high resolution meshes (desirable for detailed high frequencies) this enforces very slow sound evolution. Thus we end up with a three way trade-off between the resolution of the waveform, the speed of its evolution and processor load.

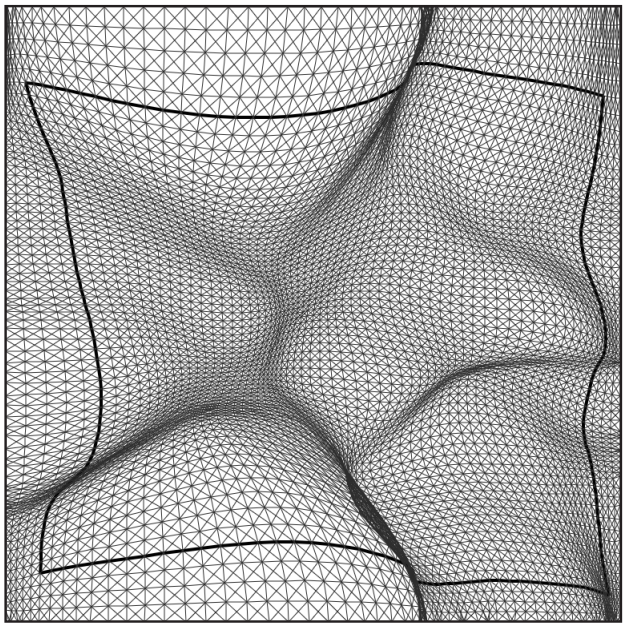

Figure 2: A 70 x 70 "Square Cross Mesh". The scan path is shown in bold. The averaging filter is on, resulting in a smooth form.

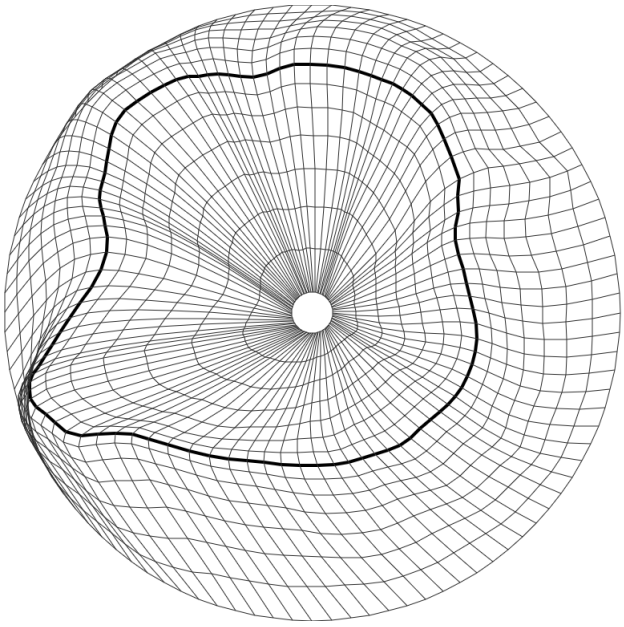

Figure 3: A 90 x 15 "Holed Spider Mesh". The motivation for a polar configuration was to ensure no discontinuous changes in the direction of the scan path.

Lumps also carry out collision detection with the walls, thus keeping the mesh within the confines of the screen. Any lump can be constrained to its original position, implementing Dirichlet boundary conditions. Incoming waves will reflect off these points.

Lumps obtain the net impulsive force by adding the force vec-

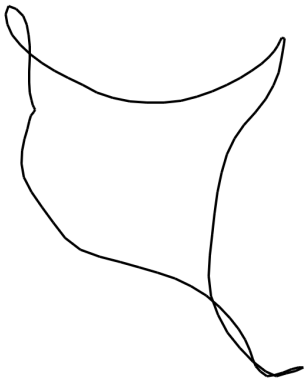

Figure 4: A 100 point Droplet Mesh, a completely unconstrained circular string with "internal pressure". There are no unscanned points.

tors in all connected *Springs*. Springs implement Hooke's Law, given in one dimension by

$$f = kx, \tag{6}$$

with f being the restoring force, k being the spring constant and x the displacement of the end of the spring from its rest position. In two dimensions, using the vector difference between the two end points of the spring, $\mathbf{x}_e = \mathbf{x}_2 - \mathbf{x}_1$, and the rest length l, the 2D force vector $\mathbf{f}$ is calculated thus:

$$\mathbf{f} = -k(l/|\mathbf{x}_e| - 1)\mathbf{x}_e \tag{7}$$

Springs are assumed to have no mass.

A high resolution mesh evolving according to the above difference equations creates a good approximation to the 2D wave equation. That is, a displacement of one of the lumps will radiate outwards as a circular wavefront.

The Mesh is updated for every new graphics frame. Thus when the word "update" or "frame" is used, it can refer to the time steps for either the difference equation, the updating and interpolating of the wavetable, or the screen refresh rate. Typically this ran at 60Hz, but would slow down for large numbers (> 4000) of springs. Thanks to the decoupling of model update and wavetable scan, this slow down does not cause audio drop outs, but it does slow the evolution of the sound.

Most meshes such as the square-cross mesh (figure 2) have 8 springs connected to every lump. Diagonal connections were found to give torsional stiffness to the mesh, which seemed to provide more interesting interaction and faster wave propagation. Therefore the bulk of the processing time was spent in the spring update function. Computational complexity scales linearly ($O(N)$) with the number of springs and lumps hence quadratically ($O(N^2)$) with vertical and horizontal resolution. As this resolution can be set by the user, the Wablet can scale easily from older mobile devices to the fastest desktops. Running a diagnostic profiler tool revealed that graphics rendering occupied the bulk of processor time, despite it consisting of just line drawing. It is likely that speed gains could be made in this department.

There were three kinds of damping applied to the mesh. The first is friction, a simple form of which can be achieved by multiplying the velocity vector in equation 5 by a coefficient slightly less than 1. Friction has the drawback that if we are scanning displacement for the audio sample values, it can "stick" the mesh in a position that is far from the original shape, and thus the amplitude of the output note may still be quite high. This was resolved by providing a different damping function that moves each lump back towards its original rest position by a certain factor; this was christened the "homing filter". In displacement scan mode this has the effect of an exponential decay on the amplitude, as would be desired for a damped instrument.

The third type of damping is an averaging filter. The inspiration for averaging came from the Karplus-Strong algorithm[15]. This applies a simple low-pass filter to the wavetable by setting each sample in the wavetable to equal the average position of all connected mass points. Thus the mesh is progressively smoothed and high frequencies in the vibrating mesh damped more rapidly than low frequencies. It would be interesting to experiment with more types of filters, and perhaps apply 1D filters to the points in the scan path exclusively.

Worth mentioning here specifically is the DropletMesh class (figure 4). The mesh consists of a circular string, but rather than this being displayed as a linear graph (as in most scanned synthesis implementations) this is a free floating circle. The inspiration for this was a droplet or bubble, exhibiting satisfyingly physical properties such as internal pressure, surface tension, acceleration due to gravity, and the ability to realistically bounce off and be squashed against the walls. Behaviour analogous to internal pressure was implemented by applying an outward force perpendicular to the springs, the force being inversely proportional to the area contained within the shape.

When a "note on" MIDI message is received by the Max patcher, it generates an OSC message containing note pitch and velocity values. This is sent over the network to the port number of the Wablet App. Depending on the excitation mode, the lumps in the mesh are displaced by a certain amount depending on their position on screen. Excitation is implemented in a similar way to physical modelling synthesis [16], by setting the speed ("hammer") or displacement ("pluck") of the lumps to a value dependent on MIDI velocity and lump position.

2.2. The Scan Path

The process of scanning the mesh to obtain an audio wavetable is handled by the "Scan Path". A scan path consists of a path from lump to lump via the springs.

A novel departure from the original Scanned Synthesis framework by Verplank et al. is that the displacement function possesses two "degrees of freedom" (see figure 5). This is a direct result of the desire to manipulate the structure on a 2D touch screen. 1D interactions felt very limiting when trying to excite the structure. The use of two dimensions also raises the interesting possibility that the two degrees of freedom may be coupled in some way, and also can have different effects on the audio output. For example twisting motions can propagate, and there can be reflections from boundaries at different angles. The 1D scan path only extracts a certain cross-section of the 2D oscillatory features, with 2D gesture waves propagating *through* the scan path. In general, adding dimensionality to a system does not just increase complexity in a linear and predictable way, but new and intricate behaviour can emerge that simply has no parallel in lower dimensions.

There is the question of how the audio amplitude can be read

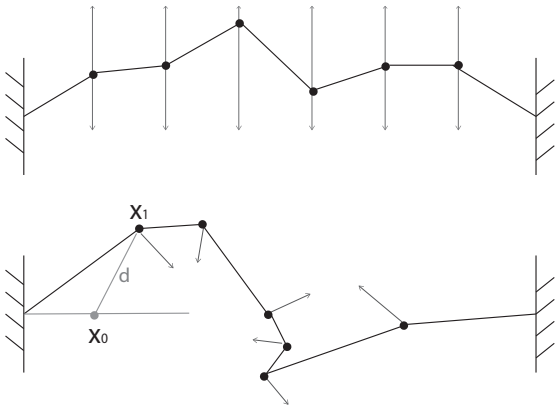

Figure 5: Degrees of freedom: The top diagram shows a string modelled with masses free to move with transverse displacement only. The lower diagram shows what can happen with 2 degrees of freedom. Note the string can double back on itself so scanning the vertical height alone may not make sense, instead the displacement distance (d) from the rest position x_0 is used.

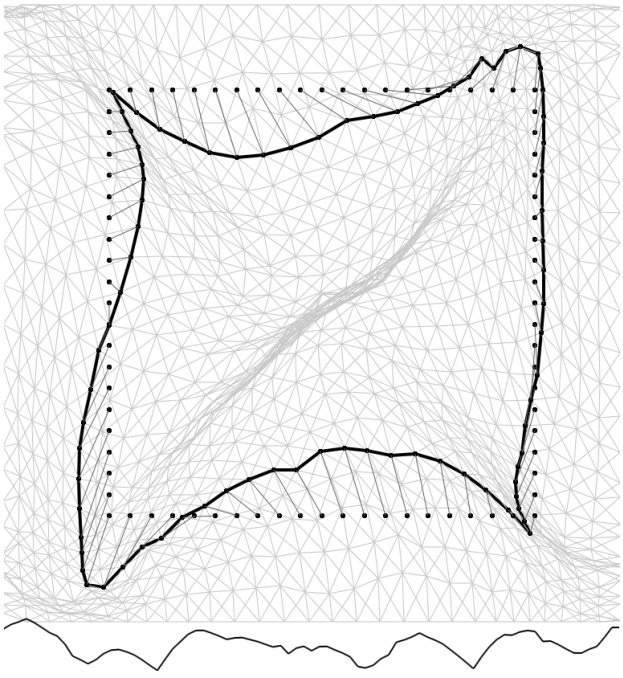

Figure 6: Here the displacements and rest positions for a vibrating square cross mesh have been drawn. A single cycle of the resulting output waveform is shown beneath. The starting position for the scan is in the bottom left corner.

as a single number from the scanning point. A basic read of, say, the Y coordinate was not satisfactory. The sound would be then be dominated by the overall shape of the path, rather than the vibrations in the mesh from gesture input. So instead, the euclidean distance between the current position and the "at rest" position was

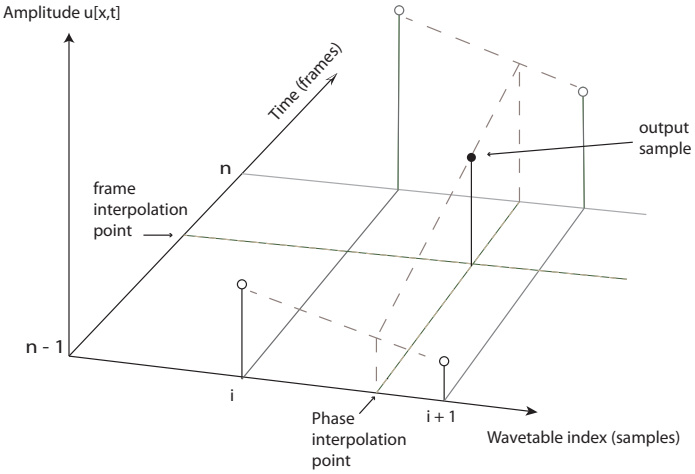

Figure 7: Bilinear interpolation between samples and frames eliminates excessive discontinuities in the output.

used. Figure 6 illustrates the displacement for a square scan path. Another scalar quantity that can be obtained from a particle moving in 2D is the speed. The magnitude of the force in the springs was a further option. As these quantities are always positive, a large DC component is consistently present in the output. This was eliminated by using a 2nd order Butterworth high-pass filter, with a cutoff frequency of 30Hz.

Since the number of points along the scan path will probably not be the same as the number of samples in a single period of the desired pitch, interpolation must be used, in a similar fashion to traditional wavetable synthesis [17] [18]. In addition the transition between successive mesh updates may cause discontinuity in the audio, so two successive frames of the scan path must be stored and also interpolated between. Figure 7 illustrates this calculation.

There are also general "global" parameters that can be adjusted such as the amount of gravity (the influence of the iPad accelerometer on the structure) a speed limit (designed to help with stability but in the end not very useful), the reflectivity of the walls, and the interaction modes discussed in the next section.

2.3. Touch Interactions

There are 6 modes for touch interactions, which can be selected in a menu in the Max patch.

1. Grab. This is the simplest and most intuitive way of exciting the mesh. On placing a finger on the screen, the closest lump is set to follow the touch point as it is dragged.
2. Force Field - creates an inverse square force field around the touch point.
3. Constrain - locks all touched lumps in place.
4. Unconstrain - a touch frees any constrained lumps.
5. Inscribe Scan Path - running a finger along the springs draws a new scan path onto the structure.
6. Spatial Harmonic - excites the entire mesh with a sinusoidal displacement. The number of fingers touching the screen determines which harmonic will be imposed.

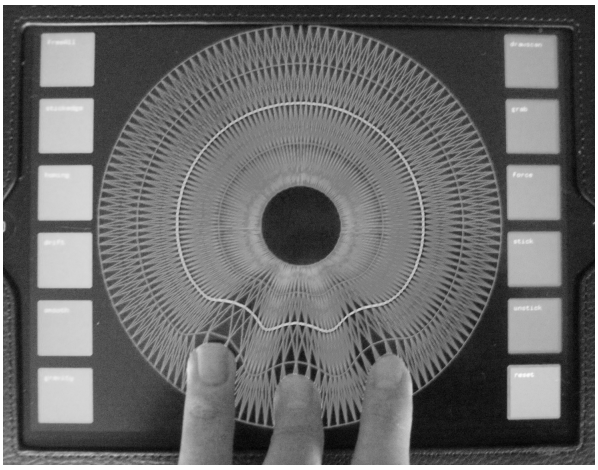

Figure 8: The Wablet in use. Force-field touch mode is in operation. Touching weakly repels the mesh and deforms the scan path.

Another useful touch input is placing five fingers on the screen. This applies a strong homing filter in order to silence the sound and still the vibrations of the Mesh. When a finger is removed the homing filter amount is set back to its previous value.

Incoming MIDI notes can also excite the mesh by displacing or imparting velocity to various lumps.

3. EVALUATION: THE WABLET IN USE

So far, no formal user evaluation has carried out. However, the following observations emerged from the design and development process and informal demonstrations.

3.1. Audiovisual Associations

Playing the Wablet is certainly an intriguing synaesthetic experience. Seeing excitations ripple out from the fingers is visually appealing, even with very basic single pixel lines to draw the mesh. At high resolutions the mesh starts to appear as a continuous fluid or fine fabric (e.g. figure 2).

There are at least four "cross-modal associations" or ways that the visuals are perceived to be linked with the sound:

Energy. The amount of movement in the mesh has a direct parallel in the perceived energy of the sound.

Rhythm. Rhythmic motion in the structure will translate to rhythmic motion in the sound.

Symmetry. Symmetry in the scan path will translate into harmonics in the output. For instance if one is using a circular scan path, and four force fields are used in the corners of the screen, a very strong fourth harmonic is heard, due to the fourfold repetition in the resulting waveform.

Smoothness. Smoothness in space corresponds to smoothness in sound. Low frequencies will correspond to round shapes and resonant smooth tones, whereas high frequencies associate with jagged shapes and chattering, rasping or fizzing tones.

3.2. Playing Techniques

The most central aspect of Scanned Synthesis is the ability to "perform timbre". To explore the effect of gestures on timbre, long and low notes were found to be best. For instance, just using a single frequency drone, or inputting a slow repeating bass motif as MIDI, and then concentrating on manipulating the mesh using the touchscreen.

Inputting a harmonic. If the finger is moved in a periodic fashion in "grab" mode, the entire mesh begins to resonate at the mode closest in frequency to this input. This will produce a corresponding harmonic in the audio output.

Separate gestures as separate sound objects. A quick flick of the finger sets up a chattering high frequency component, whilst another slow moving finger can be producing a low modulated throb. They sound like separate sonic events despite there only being one output wave.

A convoluted scan path will create high harmonics in the sound. If the path folds over itself many times, the resulting waveform will have multiple regions that follow more or less the exact same contour and move in unison. The result is a metallic and brittle tone.

Shaking the accelerometer. Shaking the iPad laterally introduces standing waves along the direction of the shake, again this will emphasise a particular harmonic.

3.3. Outstanding Issues

Thus far the evaluation of the Wablet "user experience" seems to bring up three main problems:

Redundancy: An average mesh excited with an average gesture will tend to sound fairly similar to any other, despite looking very different. There is a large parameter space, most of which will generate similar results. It is simply not the case that the visual information directly corresponds to the auditory, even in the 1D case, and users expressed a desire that the sound should change more in response to gestures. This "sameness" of output is probably the main drawback of scanned synthesis in general.

Stability is a major issue. Certain settings will cause the Mesh to blow up and all the variables to overflow. Stability is a constant issue with recursive difference equations.

Time consuming set up: Inscribing new scan paths and tuning parameters can be awkward and time consuming. Spreading the software across two machines has a negative impact on the accessibility, and fails to take advantage of the "instant on" usability of a self contained iPad app.

Unfortunately time constraints did not allow solving these issues satisfactorily, but none of them are intrinsically unresolvable.

One way to tackle the stability problem would be to use implicit methods to handle the numerical integration. Whilst implicit calculations are slower, the safety and flexibility obtained should improve the usability of the software. The author is not aware if this has been tried in other implementations of scanned synthesis, but it is certainly used in physical models [19][16]. Alternatively, it may just be easiest to calculate exactly the frame rate necessary for stability, and use that. This approach would necessitate decoupling the mesh update rate from the graphics display rate.

The setting up of meshes and scan paths could be helped greatly by the ability to save good configurations as presets. This would also help with the first problem, as only those meshes with particularly interesting properties would be saved. Other authors have stressed the use of non-uniform physical constants as being a good source of interesting sounds [4]. This should certainly be investigated. Again, a two dimensional membrane gives a lot of scope for different forms of parameter gradients.

More attention should be brought to finding a way to integrate all the functionality provided by the Max/MSP interface into the iPad app. For example, the position of the initial touch, which was fairly irrelevant to timbre, could determine the pitch of the note. More creative concepts for producing multiple pitches might be imagined, such as having multiple droplets of different sizes (the note pitch proportional to the circumference) popping in and out of existence, and even bouncing off each other.

The overall sound produced by this implementation can be rather harsh, due to too the build up of noise at half the spatial frequency of the mesh. Further work should be carried out to establish whether this is an intrinsic property of the technique, a result of the simple numerical integration technique or a by-product of the linear interpolation of a fairly low resolution mesh - with the discontinuities and aliasing this may entail. A more sophisticated design of the low-pass averaging filter may remedy this, as may smoothing of the grab point.

It remains to be seen if the Wablet works well as an instrument that might be played alongside other instruments, as much of the evolving detail in the sound seems to be masked in a musical context.

4. ANALYSIS

4.1. Relation to Karplus-Strong Algorithm

With very little effort, the algorithm for scanning a 1D string (with 1 degree of freedom) can be set up to implement the Karplus-Strong algorithm. All that was necessary was to

1. Set the frame update rate equal to the pitch of the scan, i.e. read out all the samples consecutively, eliminating the wavetable interpolation step.
2. Set the frame interpolation function to simply round down to the nearest integer.
3. For the difference equation we use the right hand averaging operator.
4. Use circular string type boundary conditions.

As all four steps above are narrowing constraints on a general scanned synthesis framework, Karplus-Strong could be considered a special case of scanned synthesis. It is interesting watching this on screen, as one can see the waveform being smoothed in real time. Displacing the wavetable by touching the screen indeed produces plucked string like tones at the update frequency 60Hz.

4.2. Relations with Other Synthesis Methods

In order to turn scanned synthesis into a basic form of physical modelling synthesis [20], the wavetable update rate can be set to the audio sampling frequency, and the scan frequency to zero. Thus the wavetable vibrates at audo frequencies with output read from a single point. Of course, display and touch interaction is no longer meaningful at this speed, and the number of calculations needed per sample exceeds the capabilities of most devices.

It is also possible to set up a 2D wavetable, and specify any 2D scan orbit and vary the *path* dynamically. This is similar to wave terrain synthesis [17] [21] [22].

So scanned synthesis appears to fit into the taxonomy of synthesis methods at an interesting midpoint between the extremely abstract (e.g. wave terrain) and the very realistic (e.g. physical modelling).

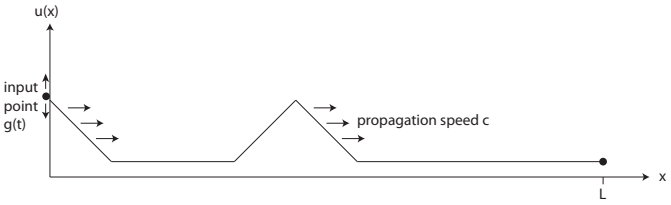

Figure 9: Propagation of a gesture along an ideal string.

4.3. The Sound of a Scanned Ideal String

For simplicity, the analysis of the output of the scanned synthesis was restricted to a 1D ideal string obeying the wave equation.

Consider single manipulable input point on one end of a string of length L, as in figure 9.

If the user moves the input point up and down at a frequency f_g, this gesture is transmitted along the string at the wave propagation speed c.

Let the string be scanned in the direction of increasing x from left to right at an oscillator frequency f_s. Due to the propagation of the wave from right to left, the sound heard will actually have a longer period. The output frequency f_{out} turns out to be related to the scanning frequency f_s, propagation speed c and the gesture frequency f_g by

$$f_{out} = f_g(f_s L/c - 1). \tag{8}$$

Note that 1 is small compared to f_s on the right hand side, thus we get an output of approximately $f_g f_s L/c$ (what we would expect from "freezing" the wavetable) minus a small detuning of f_g. A left travelling wave would be tuned *up* by the same amount, so for a general input the sound produced will be whatever frequencies were present in the user input, multiplied up to audio frequencies and tuned slightly up and down.

Alternatively, suppose an initial "pluck" displacement takes the form: $u(x,0) = \sin(2\pi n x/L)$ i.e. we have initialised the shape of the string with its nth harmonic. The output of this when scanned will be

$$u(t) = \cos(2\pi n f_0 t)\sin(2\pi n f_s t),$$

where f_0 is the fundamental of the string model. This formula tells us that, in terms of what we hear as audio, we have a pitch of f_s modulated at a much lower haptic frequency of nf_0. This is just the well known phenomenon of "beating". This is caused by the detuned left and right components interfering, and we can re-express this again as a sum and difference of two closely spaced frequencies:

$$u(x,t) = \frac{1}{2}\{\sin(\frac{2\pi n}{L}(x - ct)) + \sin(\frac{2\pi n}{L}(x + ct))\}$$

So, ultimately, an arbitrary string vibration will comprise some Fourier series of left and right moving sinusoids. So across the spectrum there will be different modulations depending on the difference between the amplitudes of left and right moving parts, and the different phases of modulation depending on their relative phase. But always, the rate of the beating depends on n.

This constant modulation of each harmonic is one of the dynamic phenomena that gives scanned synthesis its distinctive evolving sound. Further work is needed to analyse the more complicated

2D case, and the interesting effects of boundary conditions, damping and dispersion.

5. CONCLUSION

In summary, scanned synthesis occupies quite a unique place in the pantheon of sound generation algorithms. It can offer real-time interaction with timbre in a visually beautiful and physically intuitive way. It largely bypasses the problem of arbitrary and obscure mappings between physical gestures and synthesis parameters, due to the self contained unity of the interface and the algorithm. To a certain extent the visualisation, the interface, and the method of sound synthesis are one single dynamic system. Many digital instruments lack a visual presence, but this technique lends itself to creating large scale dynamic visuals that will increase audience engagement.

The Wablet offers the most extensive graphical visualisation and touch control of scanned synthesis to date. It is the first implementation on a multi-touch tablet device, and the first to use two-dimensional displacements. Combined, these two new features extend the technique in a valuable way. The instrument has so far provided a very good balance of instant rewards and depths to explore. However, basing any digital musical instrument on recursive difference equations brings many issues. The relationship between adjustable model properties and the resulting behaviour can be unwieldy, and sometimes catastrophic. Therefore some care has to be taken by the instrument designer in providing a useful subset of these parameters. An in depth user evaluation exercise may be a useful next step.

In terms of future research, scanned synthesis clearly provides a whole universe of possibilities for experimentation. Using multiple dynamic scan paths for polyphonic output would be a useful extension, alternatively multiple scanned objects could be present on screen, and they could interact with each other. Given enough processing power the simulation could be extended to 3D: sculptural models in virtual reality, made from custom "substances". Depth sensing devices could be used to scan the surroundings or a moving person, and transform them into sound. Multi-user instruments may also be possible.

Ten years ago Boulanger, Smaragdis and Ffitch concluded:

> "To our delight and frustration, scanned synthesis has opened many ideas that are hard to pursue all at once. We hope that in this paper we might have provided some excitement to get interested researchers working on some of these ideas."[4]

This project has inspired a similar sentiment. Hopefully the current proliferation of interaction devices will renew interest in exploring this method.

Accompanying material for this project, including sound examples, video, and the full masters thesis can be found online at `http://www.rootnot.co.uk/wablet.html`.

6. ACKNOWLEDGEMENTS

Supported by the European Commission, FP7 (Seventh Framework Programme), ICT-2011.1.5 Networked Media and Search Systems, grant agreement No 287711, project: Roadmap for Music Information Research.

7. REFERENCES

[1] Thomas Levin, "Tones from out of nowhere: Rudolph pfenninger and the archaeology of synthetic sound," *Grey Room MIT press*, 2003.

[2] Emily D. Robertson, "It looks like sound! : Drawing a history of animated music in the early twentieth century," M.S. thesis, University of Maryland, 2010.

[3] Bill Verplank, Max Mathews, and Rob Shaw, "Scanned synthesis," *The Journal of the Acoustical Society of America*, vol. 109, no. 5, pp. 2400–2400, 2001.

[4] Richard Boulanger, Paris Smaragdis, and John Ffitch, "Scanned synthesis: An introduction and demonstration of a new synthesis and signal processing technique," in *ICMC 2000, Berlin, Germany*, 2000.

[5] Couturier Jean-Michel, "A scanned synthesis virtual instrument," in *Proceedings of the 2002 conference on New interfaces for musical expression*, 2002, NIME '02, pp. 176–178.

[6] Jean-Michel Couturier and Daniel Arfib, "Pointing fingers: using multiple direct interactions with visual objects to perform music," in *Proceedings of the 2003 conference on New interfaces for musical expression*, Singapore, Singapore, 2003, NIME '03, pp. 184–187, National University of Singapore.

[7] N. Bryan-Kinns and P. G. T. Healey, "Daisyphone: support for remote music collaboration," in *Proceedings of the 2004 conference on New interfaces for musical expression*, 2004, NIME '04, pp. 27–30.

[8] Gunter Geiger, "Using the touch screen as a controller for portable computer music instruments," in *Proceedings of the 2006 conference on New interfaces for musical expression*, Paris, France, 2006, NIME '06, pp. 61–64.

[9] Philip L. Davidson and Jefferson Y. Han, "Synthesis and control on large scale multi-touch sensing displays," in *Proceedings of the 2006 conference on New interfaces for musical expression*, Paris, France, 2006, NIME '06, pp. 216–219, IRCAM; Centre Pompidou.

[10] Jordan Hochenbaum, Owen Vallis, Dimitri Diakopoulo, Jim Murphy, and Ajay Kapur, "Designing expressive musical interfaces for tabletop surfaces," in *Proceedings of the 2010 International Conference on New Interfaces for Musical Expression. June 2010. Sydney, Australia*, 2010.

[11] Joshua Noble and Noble Joshua, *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*, O'Reilly Media, Inc., 1st edition, 2009.

[12] Matthew Wright and Adrian Freed, "Open sound control: A new protocol for communicating with sound synthesizers," in *ICMC 1997, Thessaloniki, Greece*, 1997.

[13] Sir Newton, Isaac, *Philosophiae Naturalis Principia Mathematica*, Cambridge University Press, 1972.

[14] R. Courant, K. Friedrichs, and H. Lewy, "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, vol. 100, pp. 32–74, 1928.

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

[16] Stefan Bilbao, *Numerical Sound Synthesis*, John Wiley and Sons, Ltd, 2009.

[17] Curtis Roads, *The Computer Music Tutorial*, The MIT press, 1996.

[18] Richard Boulanger, Victor Lazzarini, and Gabriel Maldonado, *The Audio Programming Book*, the MIT Press, 2011.

[19] Balázs Bank, Federico Avanzini, Gianpaolo Borin, Giovanni De Poli, Federico Fontana, and Davide Rocchesso, "Physically informed signal processing methods for piano sound synthesis: a research overview," *EURASIP J. Appl. Signal Process.*, vol. 2003, pp. 941–952, January 2003.

[20] Annie Luciani Claude Cadoz and Jean Loup Florens, "Cordis-anima: A modelling and simulation system for sound and image synthesis: The general formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, 1993.

[21] Y Mitsuhashi, "Audio synthesis by functions of two variables," *Journal of the Audio Engineering Society*, 1982.

[22] Stuart James, "Developing a flexible and expressive realtime polyphonic wave terrain synthesis instrument based on a visual and multidimensional methodology," M.S. thesis, Edith Cowan University, Australia, 2005.