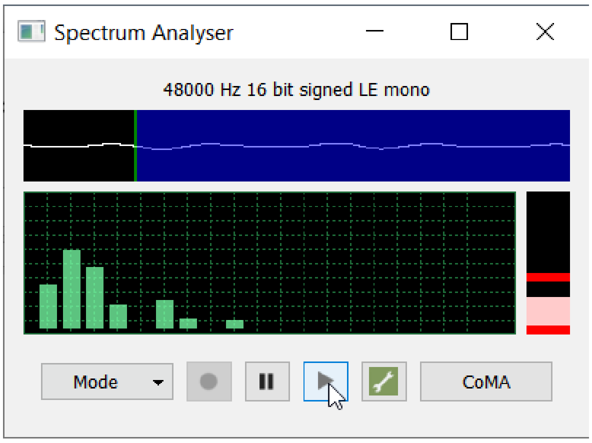
Performance notes for the rhythm visualisation application "Entrain" by Andrew Chadwick 7th February 2021

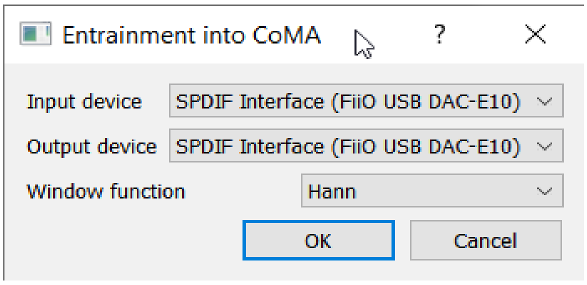
This implementation is incomplete, but starting to come into use within CoMA. Here therefore are a few notes on how it might best be used, the main variables involved, the available views and controls, some limitations, and future possibilities.

1) Main Functions

Entrain was built to give performers who are relying on a remote connection some visual feedback on rhythm. It is built on top of a demonstration spectrum analyser provided by the Qt Company and open-source Qt Project, which captures the amplitude of sound over a wide range of frequencies every 40ms. On top of that, I have built further analysis and a graphical user interface. If you just want to get a feel for what to see in that interface, you may want to jump ahead to section (2)

The original system has the slightest of modifications: an additional CoMA button to open the Entrain dialog. That original system allows playback of a selected .wav format file, generation of a slow frequency sweep as a ramped sine wave, and recording/playback (but not file storage) of approx five minutes of sound, from a source that the user can choose using a toolbox button.





The displays already shown by Spectrum Analyser are (in green) an equaliser-style amplitude bar-chart, linear in frequency, at top a sample of the waveform, together with green line and blue colouring showing how much of the file remains to be played. On the right is a display of current total amplitude with recent upper and lower limits. The toolbox also allows choice of the output device for playback, and a 'Window Function' which applies to minimise error within the Fast Fourier Transform (FFT) algorithm used to calculate the spectrum from the raw audio data: https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf

Entrain, the new facility added for initial use by CoMA, transforms the grouping of frequencies into equal ranges of pitch: 24 bands with separation of approximately a minor third and spanning the range of the piano keyboard. It then interrogates each band using a beat initially set by the user in bpm, and at any tempo, additional subdivisions from 2 (quavers) through 8 (demisemiquavers) and eventually 9 (a compound division). Consider 9 'dinner plates' on a central pivot, spinning in parallel at speeds from 1 to 9, with their patterns synchronising each beat. The incoming pulses at each separate band each land at a position on every plate, where they have their phase calculated (the angle around the circumference of the plate), assuming a steady flow of FFT data. The signals ('weights') around each 'plate' are then balanced out taking account of amplitudes (moments of torque) and reported as a resultant amplitude and phase. Hence a steady input signal should yield zero. A single pulse each beat should show up as once each crotchet, on alternate quavers, once on every third spin of the plate detecting triplets, and so on. A pulse each quaver of exactly equal amplitude should not show up on as a 'crotchet' signal (because it will average out across the 'plate') but only on the quaver 'plate', and on alternate semiquavers. It will also appear on the triplet plates but with a phase that precesses – in the same way you might work out where to place duplets when in triple time, at the start of the first note but then in the middle of the second one and the end of the third one/ the bar line.

In Entrain, colour coding is consistently used to refer to these plates, ranging from pillar box red for crotchets through gold for quavers, lime green for triplets, leaf green for semiquavers, to purple for demisemiquavers and finally violet for nine pulses to the beat. At higher tempi though, even noise (sampled by the FFT machine once per 40ms) will trigger the fastest plate (at 9 times 160 bpm, this collects an input from the FFT only once in 42ms). So unless the tempo is 80 bpm or ideally less, a violet colour just indicates that there was plenty of sound but a clear rhythm could not be detected

There is also available the sum of amplitudes across all frequency bands, which is often the basis for more primitive rhythm or beat detection.

Finally there is the option to put a very simple comb filter in the signal train which enhances bands that have two overtones above them at the expensive of bands that are less likely to come from pitched wind or string instruments. This is labelled 'magic-E'. To detect high cymbals or tuned percussion, switch this off.

2) A simple guide to the variables involved

Band: what frequency of sound are we listening to. These go up by minor thirds from the bottom A on a piano to the top F#.

Pulse: what is the division of the crotchet that is showing up as the dominant rhythm for a given band?

Phase: is this rhythm accurate, keeping a consistent phase, getting ahead (phase is more positive on each new pulse) – leading – or falling behind (phase is more negative) – lagging? There are two ways of getting 'in phase' if you are lagging the beat: play faster, or have the controller of the bpm setting reduce its tempo. Note that this application does not care if you stay a fixed time ahead of or behind the beat – it only measures the time interval between your repeated notes. That is by design, since we know that networked playing has a lag, but it seems possible nevertheless to reach a very accurate common speed.

Amplitude: how much sound is being detected. There is no volume control, not even on the sound playback, only via the microphone input. 'Entrain' does some logistic transformation of amplitudes to try to reduce the impact of very loud inputs, but if the input signal is too weak, it will not show up in the analysis. The 'total power' band is also backed off. Volumes are indicated as both density of black and width of the pulse mark in the 'by pitch' and 'by pulse' displays, and also using dot size on the Grid view. The Bubbles respond immediately to amplitude of the signals coming in each 40ms through changed opacity.

3) Screens and controls

At the top of the 'Entrain' dialog, there are two large buttons 'Reset' and 'Close'. 'Reset' will put the graphic displays back to their start-up state. 'Close' also closes the window, but if you then click again on the CoMA button on the Spectrum display, Entrain will open up again with a Reset. These and other buttons respond only when released, which makes them easier to operate on a touch screen.

'Magic-E' toggles the comb filter to enhance instruments with first and second harmonics. The little number next to it provides some information on which frequency band was just analysed for a rhythm pattern; not all bands are analysed each beat. The analysis is relatively more frequent for those bands in the central range of frequencies.

At the foot of the display are spinners to adjust tempo (bpm) over the range 40-160, and to change the memory length between one and five beats. Five is recommended and will be the value on start-up. However for shorter memories the information on the band and pulse Views will be spread out sideways, in case you should want to zoom in on the previous beat or two.

The band and pulse Views cannot be opened at start-up until the memory has been filled, which will then make those buttons responsive. Later, when either of those views is re-opened, new data for each beat will flush out the previous memory.

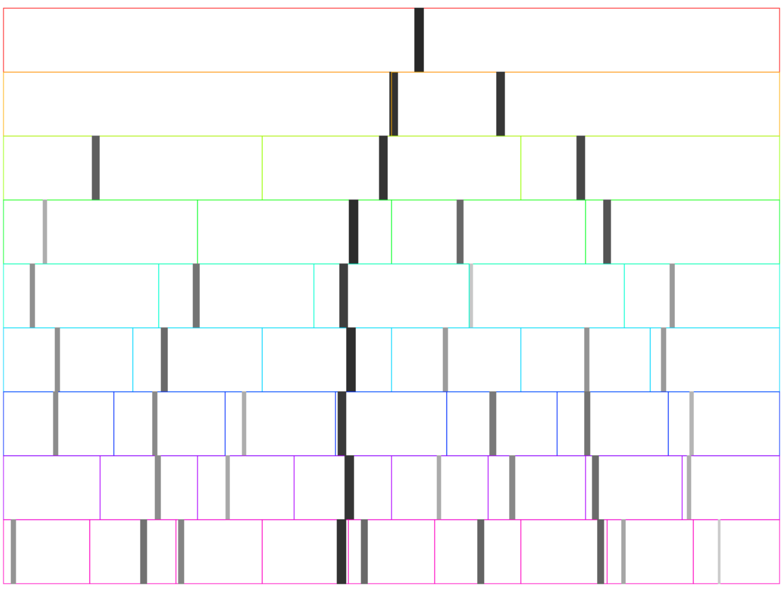
The four main screens are:

- a) Memory by pitch (band View)
- b) Memory by pulse (pulse View)
- c) Intensity Grid
- d) Smooth bubbles

It should be possible to move between these in any order. Each time d) Smooth Bubbles is selected, including clicking on that button when the view is already open, the colours of bubbles are reset, in order to pick up the rhythm from that moment onwards.

a. Band View

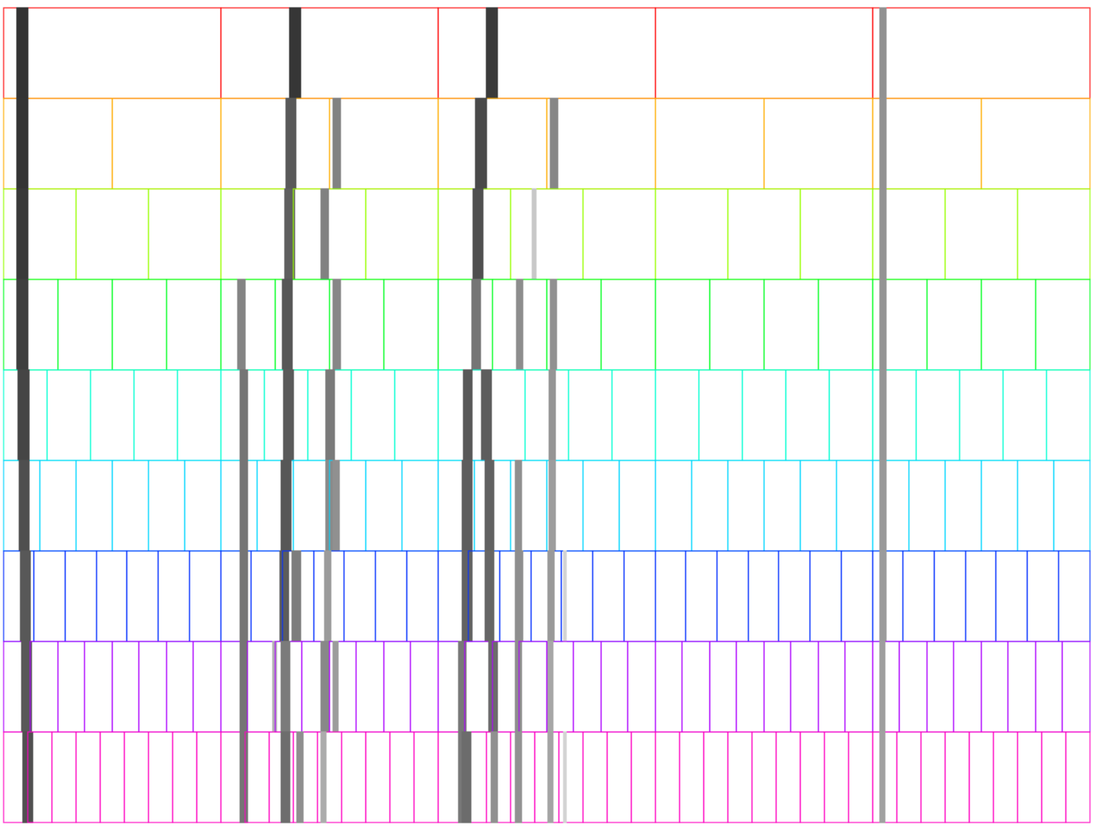
This example displays a memory of just one beat, subdivided from one down to nine sub-beats. It is shown for total power across the spectrum, which appears when the frequency band is set at 84 piano keys up from A (Aup, also familiar as an East Midlands greeting). Time passes from left to right.



There appears to be one main sound during that beat, and several sounds of smaller amplitude. Where the bars are closely aligned vertically, that was the only significant sound during the sub-beat. Moving up the screen, the duration of the sub-beats get longer and so they may average several sounds, in this case making the average reported somewhat later. Averaging of phase is weighted by amplitude, and there is a minimum amplitude threshold on the averaged signal sound for it to be plotted.

In this display the pulse starts at the left margin of each ‘brick’, and so sounds at successive pulses with phase going negative through zero will, at that pulse, appear at the end of the subsequent brick. There is never more than one reported average phase within any one pulse duration (‘brick’).

If there is sufficient amplitude at a frequency band then the pulse view in that band is likely to be cleaner than the sum over the full spectrum, as in the example below. Here the tempo of the music was similar to the setting of bpm, and a memory of five beats was in use. In beats 1 and 5 (see the end bricks on the top of this ‘rhythm wall’, single short sounds appear in the same place across all pulse rates (confirming accuracy of the algorithm). Between these, in beats 2 and 3, sounds persisted across something around a third to half a beat, so that average phase over the whole pulse is the most obvious information, showing a small phase shift between beats. The onset can be seen from the higher-frequency pulses and looks to fall in a very similar part of the beat. Beat 4 was silent.

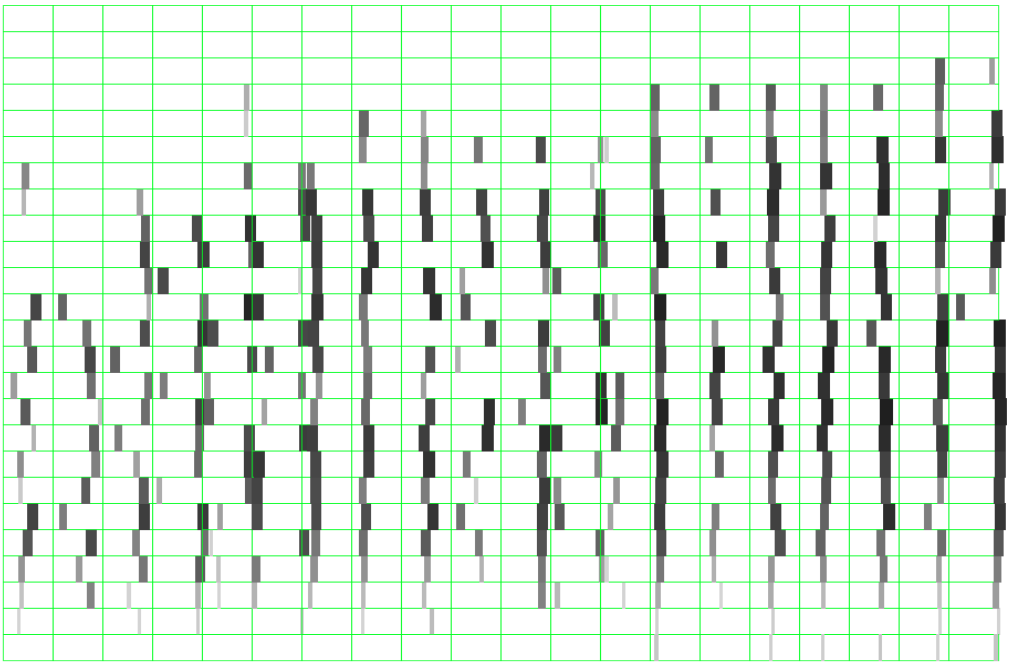


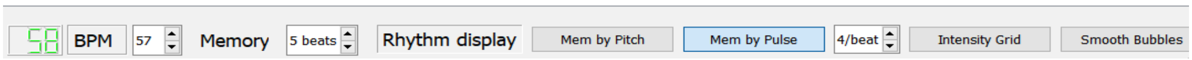
The colour-coding shown here is essentially that of the rainbow. In the following view, conversely, information is graphed for one chosen pulse rate (beat subdivision) but across the full range of frequency bands, the total power being at the foot of the plot.

b. Pulse View

Here the grid is leaf green, as a reminder that we are looking at samples taken over quarter-beats. They span a memory of five beats so there are twenty timeslots here in simple sequence. Towards the later beats, on the right, especially, we are picking up a very clean semiquaver rhythm across most of the spectrum (for a single harp). Here the comb filter was disabled, so it is probably the plucking, rather than sustaining, part of the action that is being timed. Because each successive, vertical set of bars is displaced slightly to the right, the samples are lagging the semiquaver pulse so the true tempo is slightly slower than the 57 bpm being used for this analysis.

Overall, a rising pattern in the pitch of the notes being played is also evident.





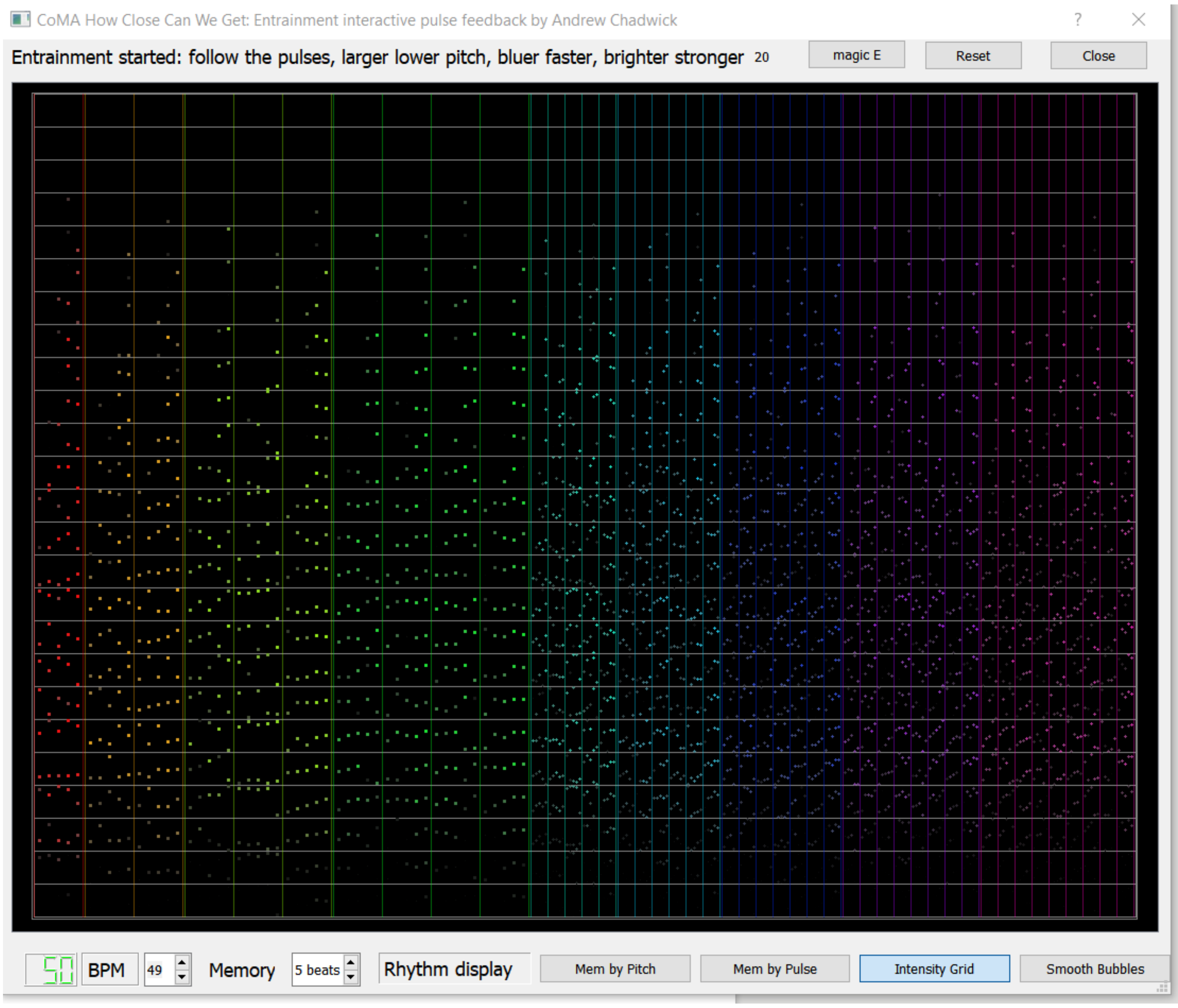
c. Intensity Grid

For the same piece, use of the intensity grid shown overpage allows us to determine that the true tempo is 49 bpm. Here the frequency is higher as you go up the page (power total at the foot) and the nine pulse rates are shown left to right, each holding between the double vertical lines their allotted number of subdivisions of the beat.

In each of the small rectangles representing a subdivision of the beat at a given frequency band, the memory over the last five beats is on display as up to five dots. Quiet sounds have only faint and/or small dots. Zero phase is centred vertically in that rectangle, with advanced phase upwards and lagging phase downwards. The most consistent horizontal, synchronised phase appears in red (crotchets) and in leaf green (semiquavers).

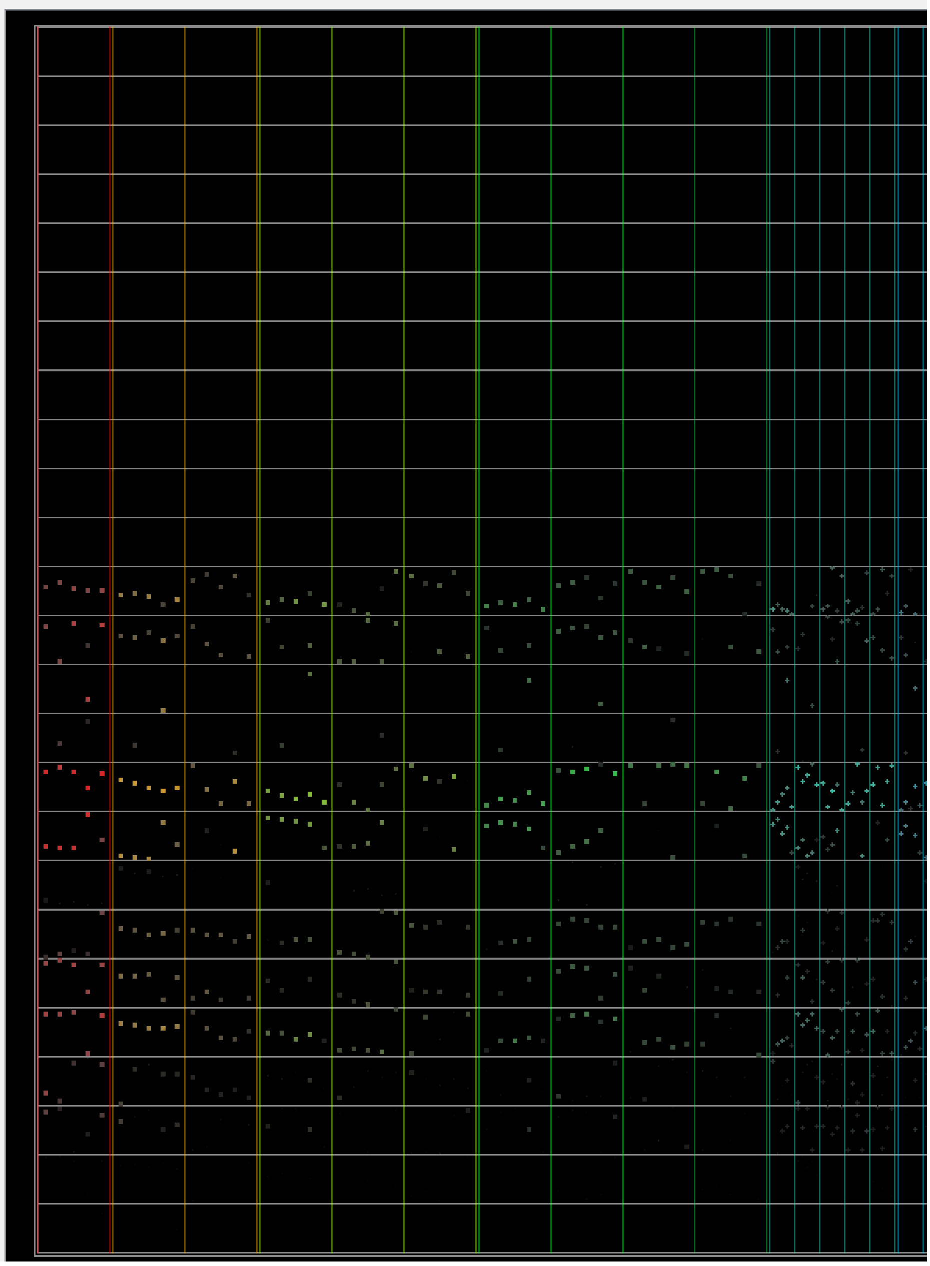
Alternate quavers (in chrome yellow) are also appearing with consistent phase on ‘downbeats’, but in the lime green triplet section, though there is consistency from beat to beat, the phase is different across each of the three triplets, so we are actually seeing semiquavers (which advance in phase when ‘read’ as triplets). Conversely, a five-fold division of the beat (viridian) shows as a progressively lagging phase. It is just about possible to see alternate demisemiquavers (in deep purple) as consistent, as these are reporting the semiquavers.

This display is more intuitive to follow as it advances in real time, each rectangle showing a set of dots moving to the left as each beat passes. There is no advantage to using a memory of less than five beats for this display as the ‘missing beats’ are just shown as gaps. The layout gives more space to the larger subdivisions of the beat (1 to 4) than it does to finer ones (5 to 9), where timing errors are progressively likely to confuse the picture.



The following score was used to explore the limits of detection of cross-rhythms at spread pitches:

Bubble Torture for Virtual Wind Quintet



The comb filter picked up the bassoon better but missed some of the flute part, so was switched off for this extract. The bassoon sound at lowest frequency is only just visible in this screenshot.

Above this, three distinct pitch bands can be distinguished. In each the tempo is constant at 120bpm as scored, as shown by the horizontal line across the five dots within each rectangle.

To find which rhythm is giving a response in each frequency region, we need to look at consistency of phase between the subdivisions of the beat. The horn part above bassoon is picking out quavers well. Above that, oboe is playing semiquavers and the quintuplet rhythm at the top is just about visible.

The clarinet part was perhaps not strongly enough articulated to pick up the rhythm. Further data processing would be needed to detect this.

d. Smooth Bubbles

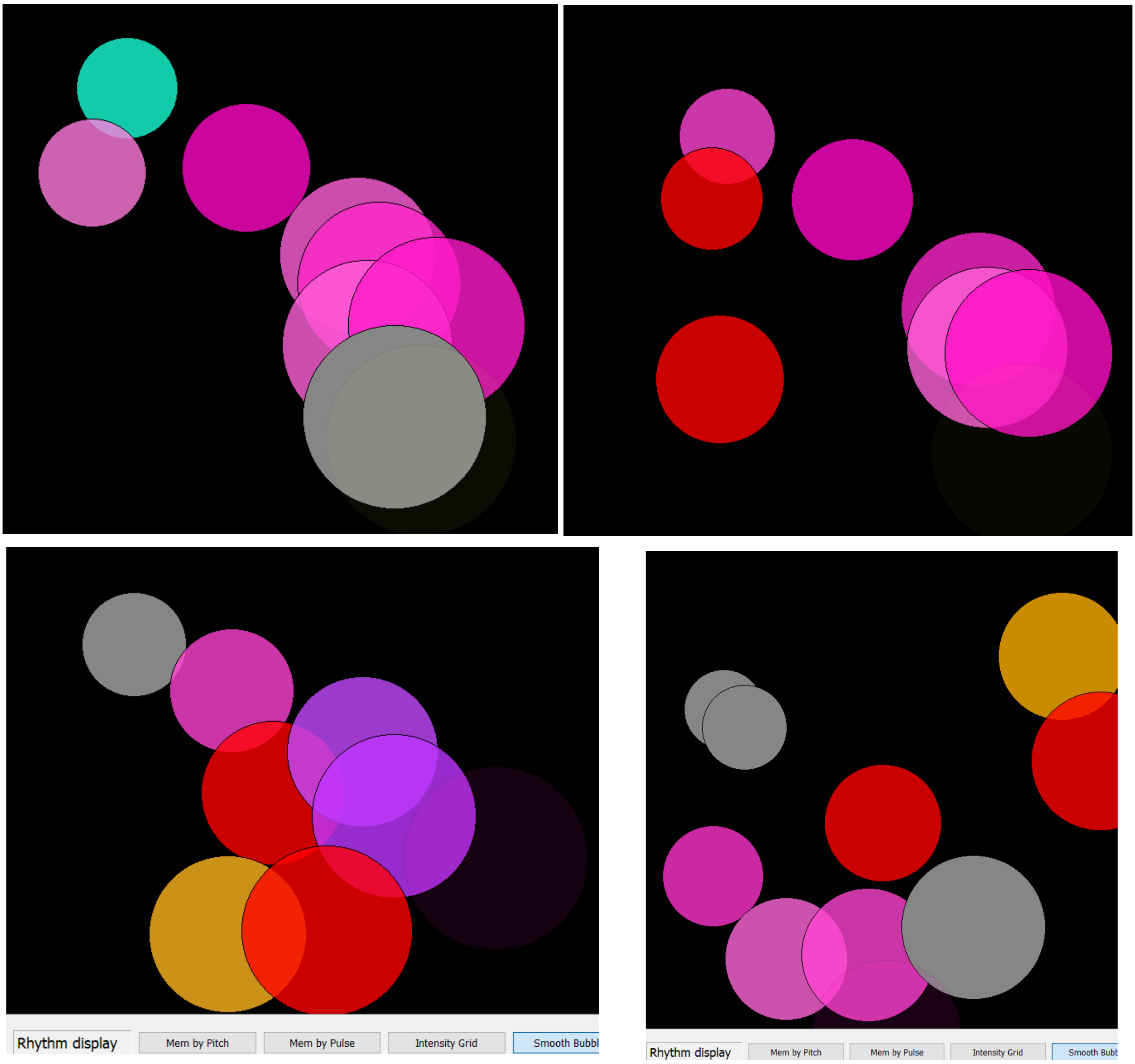
The rightmost button, “Smooth Bubbles”) shows the display designed for general use. Here the bubble size shows pitch (smaller higher; the largest is the total power, always at the front and by default shown a shade of yellow or brown). Bubbles usually start painted in a dull grey. When a rhythm is clearly detected at a pitch, that bubble will light up with the appropriate rainbow hue. Then if rhythm is no longer detected, but the bubble is being activated by sound at that pitch, the ‘value’ of the colour will gradually reduce; e.g. yellow fading to brown and ultimately back to grey. A strong new rhythm will immediately, on detection, override the hue shown for the previous one.

Bubbles also move in position, in one direction for finer subdivisions of the beat, and in the other for coarser ones, most strongly for the (crotchet) beat itself, so their distance from the diagonal represents the accumulated history of pitch detection. Violet bubbles in one direction are seeing mostly noise, whereas red ones in the other are consistently responding to the beat. Yellow ones are ‘hearing’ quavers and follow the red ones across the screen at a lower rate.

Bubbles are slowed as they reach the sides of the screen and if enough of them reach an edge, the direction of travel is reversed. To reset positions, press ‘Reset’. To restore grey colours without moving the bubbles, click once more on on ‘Smooth Bubbles’. If an audio file is played more than once, the bubbles will start each time from the position they just reached, so you can exaggerate the profile of rhythm detection or just go for an aesthetically pleasing ‘Kandinsky’ colour field!

The bubble opacity responds quickly to the raw average amplitude in their pitch band each 40ms. In the quietest passages hardly any bubbles will show. For aesthetic and machine performance reasons the rhythm colouring changes are made only once each beat, at the most frequent, and in practice at a slower rate at the top and bottom of the pitch range, which we will use less often.

A quartet of snapshots of the Bubble View response to the above wind quintet test file:



The lower two pictures have magic-E switched on which then clearly picks up the crotchet and quaver rhythms from the lower instruments. In the fourth of these images, the whole wind test piece is on its second playing, so the position of the slower pulses has reversed towards the top right. The different sizes of bubble are starting to reorder.

4) Performance limits

In summary, in its current form, the application works under favourable circumstances. There is no real control over relative volumes, so if used in a live or Zoom session, the director will need to adjust instrument balance. The magic-E control has a strong effect and can help rebalance between low and high instruments, or accentuate percussion or tuned instruments.

To help detect a rhythm, some silence between the notes is clearly very helpful.

Sextolets or faster rhythms could, even in theory, only be detectable towards the low end of the tempo range.

There is a maximum running time for recording, set by the Spectrum underlying tool of something around five minutes, possibly depending on memory capacity and the amount of data collected. There is no way of saving this recorded audio to disk file. Once the recording has stopped, or has been stopped by the user, playback from should be possible and Entrain will then respond to the captured audio as it plays back through the computer sound output system, if the correct output channel has been selected using the Spectrum tool button. USB connections for microphone and/or speaker output need to have been made *before* starting up Spectrum, or there will be a message that there is no sound.

5) Future possibilities

The only active plan is to set up a parallel processing thread and then look to detect phase consistency or trends. This should make the rhythm detection more discerning and allow “Entrain” to entrain: to follow moderate fluctuations in tempo, once a clear rhythm has been seen.

The little LED indicator at bottom left of the screen currently shows a ‘dummy’ result of the detected tempo. Once a consistent moderate phase shift has been established, the tempo displayed there will change and the tempo used for analysis of rhythm in the music will entrain to that new value. This then becomes a phase-locked loop. A tempo shift of 10% over five beats of memory ought to be easily detectable if any of the rhythm detectors are consistently triggered. Larger shifts can sometimes not be distinguished from syncopation or a cross-rhythm such as a hemiola.

A ‘nice to have’ feature that would require modification to the code of the Spectrum tool, would be to remember (in machine registry, if to persist between sessions), and prompt, the ‘favourite’ folder of the user for stored sound files.

If balance problems on the input, or inaudibility of some instruments, prove an impediment, then it may be worth experimenting with pre-processing of the input using an equaliser and/or compressor. Both of these are available via free tools such as Voicemeeter, which substitutes for the Windows mixer, but it is not yet known if this can coexist with the Spectrum toolbox.

As this new test system relies heavily on code provided free by the Qt Company and Project, and also a third party FFT algorithm, it could be problematic for both technical and licensing reasons to release it generally as an installable application. In the meantime I am happy to try creating ‘bubble videos’ from sound files sent to me and will welcome comment and suggestions for enhancement.

Andrew Chadwick 7th Feb 2021