Granular Synthesis

"All sound is an integration of grains, of elementary sonic particles, of sonic quanta." -Iannis Xenakis, Greek Composer (1971).

Granular Synthesis

- Sound synthesis method that operates on the microsound time scale.
- Based on the same principles as sampling/wavetable synthesis but often includes analog technology as well.
- **Difference** Samples are not used directly to make usual sounds:
  - Split in small pieces of around 1 to 50 ms (milliseconds) in length, **the grains**.
  - Multiple grains may be layered on top of each other all playing at different speed, phase and volume.

Granular Synthesis: Soundscape

Result is no single tone, but a soundscape!

- Often a cloud, that is subject to manipulation
- Unlike any natural sound and also unlike the sounds produced by most other synthesis techniques.
- By varying the waveform, envelope, duration, spatial position, and density of the grains many different sounds can be produced.

Granular Synthesis: Is this musical?

- Usable as music or soundscapes (ambient)
- Usable as Sound effects
- **MUSICAL:** Usable to alter sample speed while preserving the original pitch/tempo information —**pitch/tempo synchronous granular synthesis**
- Usable as Raw material for further processing by other synthesis or DSP effects.
- The range of effects that can be produced include amplitude modulation, time stretching, stereo or multichannel scattering, random reordering, disintegration and morphing.

Granular Synthesis: Background

Strong Physics Background:

- Quantum physics has shown that sound can be atomically reduced to physical particles
- Physical form of sound was first envisioned by the Dutch scientist Isaac Beeckman (1618):
  "Sound travels through the air as globules of sonic data.
- Denis Gabor (1947) proposed the idea of a grain as the quantum of sound and more recently
- Xenakis (1971) first musical use of granular synthesis — a reel to reel tape recorder, a razor blade, sticky tape, and a lot of time.
- Curtis Roads (1988), digital granular synthesis
- Barry Truax (1990) real-time granular synthesis composition Riverrun, Buy the CD!

Granular Synthesis: Implementations

- Software:

  Many implementations nowadays:

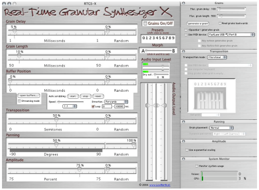
  Programmable: Csound, MATLAB, MAX/MSP routines:

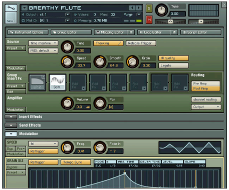
  Standalone: Supercollider, Granulab, RTGS X.

  DAW plug-ins standalone: VSTis etc.

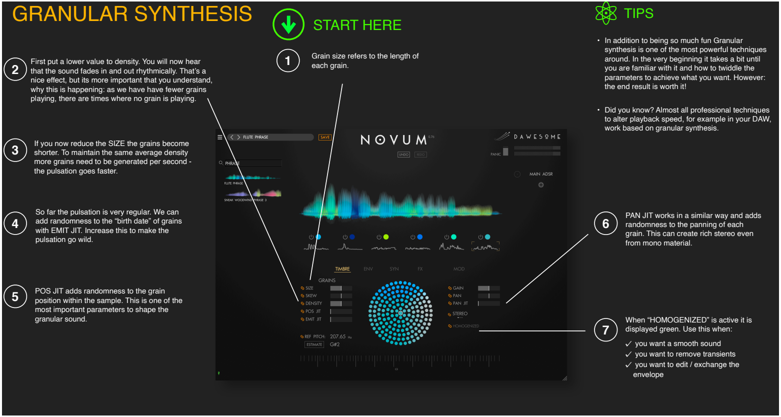
  Modern Music Samplers: Native Instruments' Kontakt, Intakt...., others

- Hardware: Korg Kaos Pad.





Novum Granular Synthesis



GRANULAR SYNTHESIS

START HERE

1 Grain size refers to the length of each grain.

2 First put a lower value to density. You will now hear that the sound fades in and out rhythmically. That's a nice effect, but its more important that you understand, why this is happening: as we have have fewer grains playing, there are times where no grain is playing.

3 If you now reduce the SIZE the grains become shorter. To maintain the same average density more grains need to be generated per second - the pulsation goes faster.

4 So far the pulsation is very regular. We can add randomness to the "birth date" of grains with EMIT JIT. Increase this to make the pulsation go wild.

5 POS JIT adds randomness to the grain position within the sample. This is one of the most important parameters to shape the granular sound.

TIPS

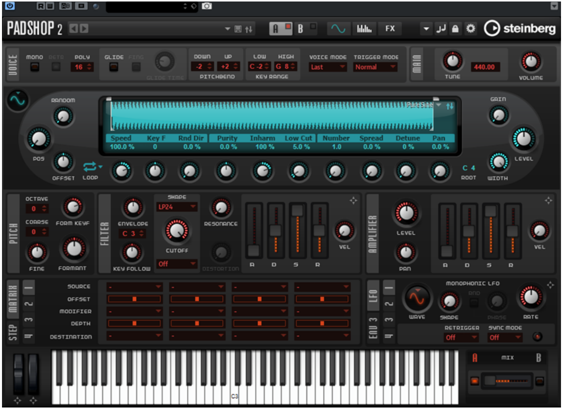
- In addition to being so much fun Granular synthesis is one of the most powerful techniques around. In the very beginning it takes a bit until you are familiar with it and how to twiddle the parameters to achieve what you want. However: the end result is worth it!
- Did you know? Almost all professional techniques to alter playback speed, for example in your DAW, work based on granular synthesis.

6 PAN JIT works in a similar way and adds randomness to the panning of each grain. This can create rich stereo even from mono material.

7 When "HOMOGENIZED" is active it is displayed green. Use this when:

✓ you want a smooth sound
✓ you want to remove transients
✓ you want to edit / exchange the envelope

Cubase Padshop Granular Synthesiser



Speed	Key F	Rnd Dir	Purity	Inharm	Low Cut	Number	Spread	Detune	Pan
100.0 %	0	0.0 %	0.0 %	100 %	5.0 %	1.0	0.0 %	0 %	0.0 %

POLY 16

PITCHBEND DOWN -2 UP +2

KEY RANGE LOW C -2 HIGH G 8

VOICE MODE Last

TRIGGER MODE Normal

TUNE 440.00

ROOT C 4

OCTAVE 0

COARSE 0

SHAPE LP24

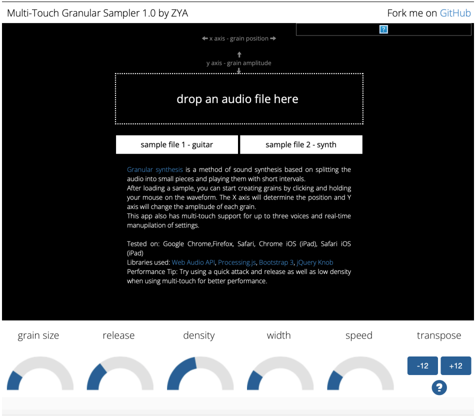
ENVELOPE C 3

RETRIGGER Off

SYNC MODE Off

Web Audio Granular Synthesisers

- https://zya.github.io/granular/



Multi-Touch Granular Sampler 1.0 by ZYA

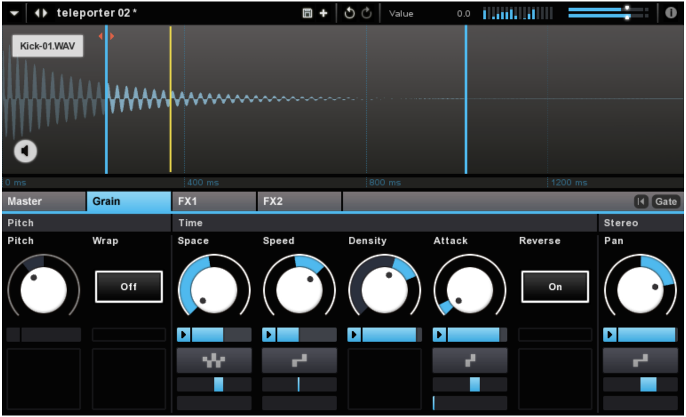
Fork me on GitHub

Granular synthesis is a method of sound synthesis based on splitting the audio into small pieces and playing them with short intervals.
After loading a sample, you can start creating grains by clicking and holding your mouse on the waveform. The X axis will determine the position and Y axis will change the amplitude of each grain.
This app also has multi-touch support for up to three voices and real-time manupilation of settings.

Tested on: Google Chrome,Firefox, Safari, Chrome iOS (iPad), Safari iOS (iPad)
Libraries used: Web Audio API, Processing.js, Bootstrap 3, jQuery Knob
Performance Tip: Try using a quick attack and release as well as low density when using multi-touch for better performance.

Granite Granular Syntehiser

http://www.newsonicarts.com/html/granite.php



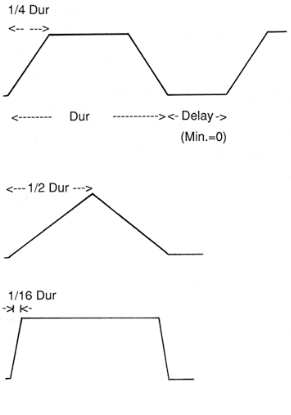
Note: Commercial application but demo available)

Plenty of other example — just search

Granular Synthesis: What is a grain?

A Grain:

- A grain is a **small piece of sonic data**
- Usually have a duration $\approx$ 10 to 50 ms.
- The grain can be broken down into smaller components



GRAIN ENVELOPES

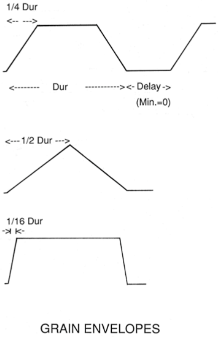
Granular Synthesis: What is a grain?

Grain components:

Envelope: used so no distortion and crunching noises at the **beginning** and **end** of the sample. The shape of the envelope has a **significant** effect on the grain sound.

- For a sampled sound, a short linear attack and decay **prevents clicks** being added to the sound.
- Changing the **slope** of the grain envelope changes the resulting grain **spectrum**,
  E.g. Sharper attacks producing broader bandwidths, just as with very short grain durations.

Contents: The audio: derived from any source: **basic waveforms** or **samples**



Granular Synthesis: Making Sounds

Sounds made by the generation of thousands of short sonic grains:

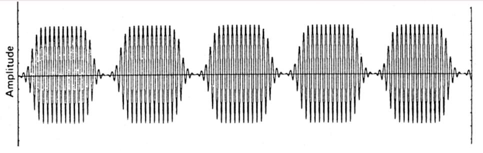
- Combined linearly to form large scale audio events,
- 3 Possible combinations:

  Quasi-synchronous granular synthesis
  Asynchronous granular synthesis
  Pitch/Tempo-synchronous granular synthesis
- The grains' characteristics are also definable and when combined affect the overall sound.

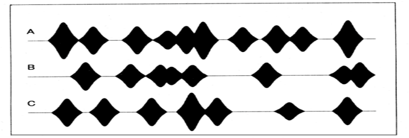
Granular Synthesis: Making Sounds (Cont.)

Quasi-synchronous granular synthesis:

- A grain stream of equal duration grains, produces amplitude modulation (**see later**) with grain durations **less** than 50 ms.



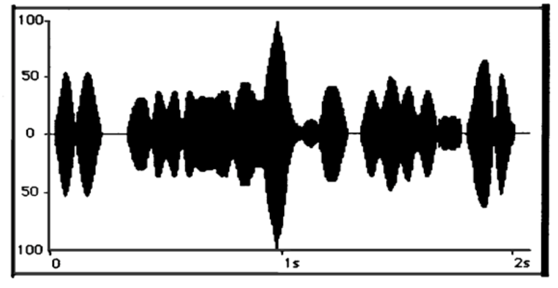
- Grain streams with variable delay time between grains: the sum of which resembles asynchronous granular synthesis.



Granular Synthesis: Making Sounds (Cont.)

Asynchronous granular synthesis:

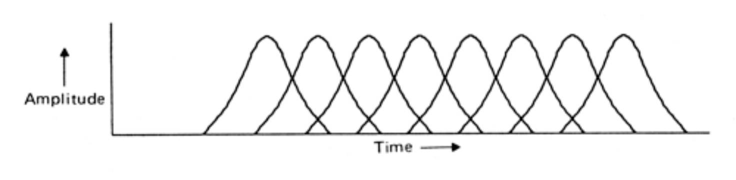
Grains are distributed stochastically with no quasi regularity.



Granular Synthesis: Making Sounds (Cont.)

Pitch/Tempo-synchronous granular synthesis:

- Preserve Pitch/Tempo whilst altering sample playback speed E.g. Intakt, Kontakt.
- Overlapping grain envelopes designed to be **synchronous** with the **frequency** of the grain **waveform**, thereby producing fewer audio artifacts.



Granular Synthesis MATLAB Example

Simple MATLAB Example: granulation.m

```
[filename,path] = uigetfile({'*.wav;*.waV;','Wav Files'; ...
            '*.*', 'All files (*.*)'}, ...
            'Select a sound file');

if isequal(filename,0) | isequal(path,0)
                cd(savedir);
                return;
end
filenamepath = [path filename];
[x, fs] = audioread(filenamepath);

figure(1)
plot(x);

doit = input('\nPlay Original Wav file? Y/[N:]\n\n', 's');
if doit == 'y',
 sound(x,fs);
end
```

MATLAB Granular Synthesis Example (Cont.)

granulation.m (cont.):

```
Ly=length(x);  y=zeros(Ly,1);              %output signal
timex = Ly/fs;

% Constants
nEv=400; maxL=fs*0.02;  minL=fs*0.01;  Lw=fs*0.01;
% Initializations
L = round((maxL-minL)*rand(nEv,1))+minL;   %grain length
initIn = ceil((Ly-maxL)*rand(nEv,1));      %init grain
initOut= ceil((Ly-maxL)*rand(nEv,1));      %init out grain
a = rand(nEv,1);                           %ampl. grain
endOut=initOut+L-1;
% Do Granular Synthesis
for k=1:nEv,
  grain=grainLn(x,initIn(k),L(k),Lw);
  figure(2)
  plot(grain);
  y(initOut(k):endOut(k))=y(initOut(k):endOut(k))+ grain;
end

% Plot figure and play sound
 .......
```

MATLAB Granular Synthesis Example (Cont.)

grainLn.m

```
function y = grainLn(x,iniz,L,Lw)
% extract a long grain
% x    input signal
% init first sample
% L    grain length (in samples)
% Lw   length fade-in and fade-out (in samples)

if length(x) <= iniz+L , error('length(x) too short.'),  end

y = x(iniz:iniz+L-1);                      % extract segment
w = hanning(2*Lw+1);
y(1:Lw)      = y(1:Lw).*w(1:Lw);           % fade-in
y(L-Lw+1:L) = y(L-Lw+1:L).*w(Lw+2:2*Lw+1); % fade-out
```

More Advanced Granular Synthesis

Above is quite simple and general and can be employed to obtain very different sounds and sound effects.

More control over the sound:

- The above sonds is greatly influenced by the criterion used to choose the instants .
- If these points are regularly spaced in time and the grain waveform does not change too much,
  - the technique can be interpreted as a **filtered pulse train**, i.e. it produces a periodic sound whose spectral envelope is determined by the grain waveform interpreted as impulse response.

PSOLA based Pitch/Tempo-synchronous granular synthesis

The above is an example is the **PSOLA based Pitch/Tempo-synchronous granular** synthesis (**more soon**), where:

- When the distance between two subsequent grains is much greater than L_k, the sound will result in grains separated by interruptions or silences with a specific character.
- When many short grains overlap (i.e. the distance is less than L_k), a sound texture effect is obtained.

See accompanying lab exercise

Short and Overlapping Grains

Short Grains

- The above code, for simplicity of illustration, only uses long grains.
- experiment by mixing or swapping in short grains via the `grainSh.m` function — **See accompanying lab exercise**

Overlapping Grains

It is quite simple to extend the code above to account for overlapping grains:

- To overlap a grain g_k at instant $n_k = \texttt{iniOLA}$ with amplitude a_k, **See accompanying lab exercise**.

```
endOLA = iniOLA+length(grain)-1;
y(iniOLA:endOLA) = y(iniOLA:endOLA) + ak * grain;
```

PSOLA based Pitch/Tempo-synchronous granular synthesis

PSOLA exists as common means of pitch and tempo shifting outside of any synthesis method.

- Historically, predates the phase vocoder but still common approach.
- Historically important to the development of Granular synthesis.
- PSOLA originated for speech processing, paarticularly speech synthesis,
    - It also applicable to musical applications.

PSOLA in action

Not unlike the phase vocoder:

- Used to modify the pitch (scaling) and duration (time stretching) of a speech signal.
- PSOLA works by dividing the speech waveform in small overlapping segments.
    - To change the pitch of the signal, the segments are moved further apart (to decrease the pitch) or closer together (to increase the pitch).
    - To change the duration of the signal, the segments are then repeated multiple times (to increase the duration) or some are eliminated (to decrease the duration).
    - The segments are then combined using the overlap add technique.
- **The difference between PSOLA and the phase vocoder is there is no STFT in PSOLA.**

See Live Scripts for more details and code examples: Ch5_6_Granular_Synthesis.mlx