

Granulaire Synthese

Probleem

Je hebt een geluid (bijvoorbeeld spraak) van zekere tijdsduur en wilt graag de tijdschaal ervan veranderen, dus langer of korter laten duren.

Op die manier kun je een "geluidenmicroscop" maken die bepaalde eigenschappen van het geluid uitvergroot. Dat geeft soms een interessant, vervreemdend effect.

Op pag 8 van Sound Design wordt verteld dat door het veranderen van de tijdschaal een paar "bijwerkingen" optreden:

1. de toonhoogte van tonen wordt lager of hoger;
2. de klankkleur van het geluid verandert: rek je geluid uit, dan wordt het doffer, donkerder van klank; druk je het in elkaar dan wordt het scherper. Spraakklanken klinken alsof ofwel een reus, ofwel een dwerg aan 't woord is (formantverschuiving). Spraak wordt daarbij ook moeilijker te verstaan.

Een opdracht voor het veranderen van de tijdschaal waarbij je deze effecten goed kunt horen vind je op: http://muziekexact.nl/NLT/baby_raaf.htm.

Hieronder gaan we de tijdschaal zodanig veranderen dat deze twee effecten *niet* zullen optreden.

Idee

(1)

Deel het geluid op in heel korte, elkaar overlappende fragmentjes van gelijke duur. Bijvoorbeeld maak fragmentjes die 10 milliseconden duren en elkaar voor 90% overlappen, dus als volgt:

fragment 1 loopt van 0 tot 10 msec,
fragment 2 loopt van 1 tot 11 msec,
fragment 3 loopt van 2 tot 12 msec,
etc.

(2)

Mix deze fragmentjes overlappend weer bij elkaar, op een leeg spoor (dat we voortaan "doelspoor" noemen), maar wel met *iets grotere onderlinge afstand*, bijvoorbeeld:

fragment 1 loopt van 0 tot 10 msec,
fragment 2 loopt van 3 tot 13 msec,
fragment 3 loopt van 6 tot 16 msec,
etc.

Aangezien de klank en de toonhoogte van zo'n kort fragmentje nauwelijks veranderen, zal zowel de toonhoogte als de klankkleur van het geluid op het doelspoor gelijk zijn aan dat van het oorspronkelijke geluid. Maar niettemin duurt het geluid op het doelspoor *langer*! Daarmee, zo veronderstellen we, hebben we dan het gewenste effect bereikt. Maar werkt 't ook?

Opdracht 1

Geef zelf een antwoord op die vraag, daarbij geholpen door

[Preset_Granulair_Opdracht1.txt](#).

Het antwoord vind je in [Preset_Granulair_Opdracht1_Antwoord.txt](#).

Hann-Venster

Probleem

Het idee blijkt te werken, maar er treden zware vervormingen op: storende tonen en bijgeluiden. Wat is daar de oorzaak van?

Als je een geluid opdeelt in losse fragmentjes, dan zal elk fragmentje beginnen en eindigen met een klik (ook wanneer de uiteinden toevallig een uitwijking nul hebben. Waarom?) Je hoort het heftige, zeer snelle, zeer regelmatige aan- en uit-geklapper van de fragmentjes als een storende toon, terwijl het eigenlijke, beoogde effect van tijdstrek min of meer op de achtergrond verdwijnt.

Idee: in- en uitfaden met een Hann-venster

Een remedie zou kunnen bestaan uit de techniek die in H4.1 werd besproken: we laten de fragmentjes in- en uitfaden, voordat we ze op het doelspoor overlappend mixen. Dan kunnen ze geen klik-toon meer produceren.

Een elegante en veel gebruikte manier om dat te realiseren is gebruik maken van een contour met een volkomen vloeiende flank, die eerst stijgt en dan daalt, waarmee we het fragment achtereenvolgens in- en uitfaden (Sound Design H4.1).

Zo'n contour met een vloeiende flank heet een **Hann-venster** en wordt vaak gebruikt bij de berekening van het spectrum van een signaal. Je komt die term dan ook tegen in de Spectrum Analyzer van WaveWizard.

Een Hann-venster wordt ook wel eens een *raised cosine* genoemd, een *opgerezen cosinus*:

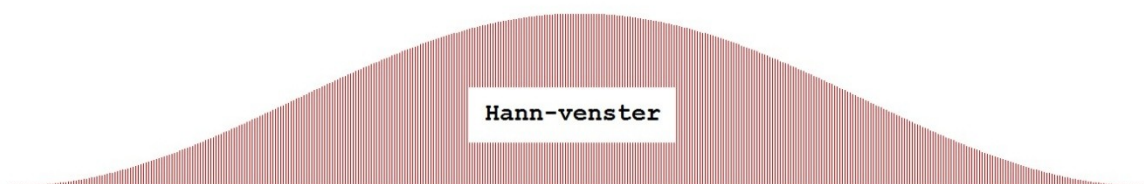
$$\text{Hann}(x) = 1 + \cos(x - \pi)$$

een cosinusfunctie waarbij 1 is opgeteld, zodat je geen negatieve waarden hebt, precies zoals je dat hebt gezien in de ADSR-contourgenerator die evenmin negatieve waarden heeft.

Het Hann-venster dat wij nodig hebben is dan een discrete functie, die de vermenigvuldigingsfactoren bevat. Daarmee kunnen we een fragmentje voorzien van een heel soepele in- en uitfade. Noemen we die discrete functie $F(n)$ dan, dan hebben we:

$$F(n) = 1 + \cos\left(\frac{2\pi}{N}\left(n - \frac{N}{2}\right)\right)$$

Hierin stelt N de lengte (tijdsduur) van de contour voor, uitgedrukt in aantal samples. Bijvoorbeeld als de contour $F(n)$ een duur heeft van 0,01 sec, dan is $N = 0,01F_s = 441$ samples. Zie Figuur.



In WaveWizard-code ziet het Hann-venster $F(n)$ er zo uit:

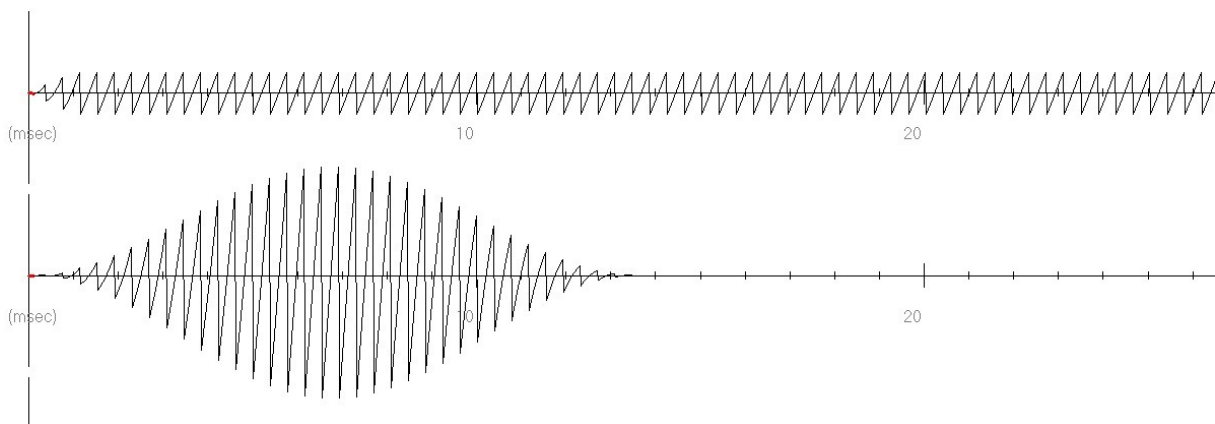
Bewerk signaal

```
n0          0
n1          N-1
Bewerking  F1[n] = 1 + cos(2*pi*(n-N/2)/N)
```

Het Hann-venster wordt opgeslagen in floating pointbuffer $F1[0 \dots N-1]$.

Opdracht 2a

In [Preset_Granulair_Opdracht2a.txt](#) wordt eerst een Hann-venster gemaakt volgens de code hierboven. Dan wordt een zaagtandgolf opgewekt op spoor S1. Tot slot volgt een bewerking waarmee de zaagtand van S1 een fade-in / fade-uit krijgt door dit Hann-venster. Het resultaat wordt opgeslagen op spoor S2. Wat je op het scherm te zien krijgt is de onderstaande figuur.



Vul de ontbrekende code in en klik op Start.

Het antwoord vind je in [Preset_Granulair_Opdracht2a_Antwoord.txt](#).

granulair

Het woord *granulair* heeft dezelfde afleiding als *graan* en betekent *graanvormig*, *korrelvormig*. In de figuur is te zien dat de zaagtandtoon de contour van het Hann-venster overneemt, maar dan natuurlijk zowel bij de positieve als de negatieve uitwijking en krijgt daardoor een korrel-achtige vorm.

De mooie eigenschap van zulke "geluidskorrels" is dat je ze op elk willekeurig tijdstip kunt mixen met de trilling die al op een spoor staat, *zonder* dat er kliks optreden!!

Luister maar eens naar de klikvrije "zaagkorrel" op spoor S2! Merk op dat je in zo'n kort geluidje de toonhoogte prima kunt waarnemen. Dus hoeveel zaagkorrels je ook door elkaar mixt, de toonhoogte blijft constant. Natuurlijk kan wel de amplitude toenemen en de maximum waarde van samples (zie Sound Design pag. 54) overschrijden. Vandaar dat je in Opdracht 1 al vast de variabele *Versterking* tegen kwam.

Opdracht 2b

Voeg nu de code voor het Hann-venster toe aan [Preset_Granulair_Opdracht1.txt](#) en zorg ervoor dat de fragmentjes een Hann-venster-contour krijgen voordat ze worden gemixt bij S2. Het antwoord vind je in [Preset_Granulair_Opdracht2b_Antwoord.txt](#).

Pas het effect toe op weerbericht.wav en beschrijf het klinkende effect.

Geluid uit de spuitbus

Probleem

Door de fragmentjes te "verpakken" in Hann-vensters ontstaan graanvormige of granulaire geluidskorrels met een heel geleidelijke fade-in en fade-out, waardoor het inmixen van de fragmenten geen kliks meer veroorzaakt. Daarom klinkt een mix van gegraneerde korrels veel gladder en minder hakkerig dan ongegraneerde (schrijf code waarin je de ongegraneerde en gegraneerde versies op S2 en S3 zet, zodat je ze gemakkelijk kunt vergelijken). Maar niettemin hoor je ook in de granulaire mix storende tonen die in het originele geluid niet voorkwamen. Hoe kunnen die, ondanks de granulering, ontstaan?

granulaire synthese is "spuitbus-synthese"

Ons gehoor is extreem knap in het detecteren van regelmaat en patronen. Als een patroon vaker dan ongeveer 20 keer per seconde optreedt, nemen we het waar als een toon. We hebben fragmentjes gemaakt van een geluid en daarbij hebben we het "masker" steeds over dezelfde korte tijdsafstand verschoven. Op het doelspoor hebben we de gegraneerde fragmentjes ook weer op regelmatige tijdsafstanden van elkaar geplaatst. Die regelmaat neemt je oor waar als een toon.

Als we die toon willen vermijden, moeten we voorkomen dat die regelmaat ontstaat door ervoor te zorgen dat de geluidskorrels nooit precies op onderling gelijke afstanden kunnen komen te liggen. Dat kunnen we bereiken door aan die verschuivingstijd een klein *random* tijdsinterval toe te voegen.

Het effect ervan kun je vergelijken met een verfspuitbus. Daar spuiten heel kleine druppeltjes uit die op random plaatsen terecht komen en dan in elkaar overvloeien. Zouden ze op precies gelijke afstanden komen te liggen, dan zou je dat, ondanks dat samenvloeien, toch nog kunnen zien (ook je oog is heel gevoelig voor patronen). Dat geluidskorrels goed met elkaar kunnen vervloeien is een eigenschap die ze danken aan het Hann-venster. En het spuitbus-effect kunnen we ook toepassen op geluid door aan die onderlinge tijdsafstanden tussen de korrels een random tijdje toe te voegen.

Pas dan kun je echt spreken van *granulaire synthese* als een soort *spuitbus-synthese*. Granulaire synthese wordt in computermuziek en sound design veel toegepast, omdat je er dikke, wollige "klanktapijten" mee kunt maken. Noem het maar de geluidsequivalent voor graffiti.

Hoe voegen we nu een random element toe aan de tijdstippen waarop de korrels terecht moeten komen? We kijken nog eens naar de code van Opdracht 2b, hieronder.

```
FOR k = 0 TO AantalFragmenten
  Print AantalFragmenten - k
  t = k*Strek*Verplaatsing
  q = k*Verplaatsing
  Bewerk signaal
    n0          0
    n1          DuurFragment-1
  Bewerking    S3[n+t] = S3[n+t] + Versterking*F1[n]*S1[n+q]
NEXT k
```

Het doelspoor is S3. In de instructie **Bewerk signaal** is te zien dat de samples van een korrel (fragment dus) geplaatst worden op tijdstip $[n+t]$. Bij de eerste sample van de korrel is n gelijk aan 0. Dus de variabele t is als enige verantwoordelijk voor de tijdspositie van de k -de korrel en t is enkele regels daarboven gedefinieerd als $k*Strek*Verplaatsing$. Het tijdsinterval tussen elke twee korrels is dus gelijk aan

Strek*Verplaatsing. Dat *constante* getal is de oorzaak van de regelmaat die we als toon waarnemen. Willen we die regelmaat breken, dan moeten we er een random getal aan toevoegen dat groter of gelijk is aan 0 en kleiner dan **Strek*Verplaatsing**. (De WaveWizardfunctie `random(a;b)` wordt besproken in H6.2 op pag 107-108.) Daarmee passen we de definitie van **t** als volgt aan:

```
t = k*Strek*Verplaatsing + random(0; Streck*Verplaatsing)
```

Opdracht 3

Voer de bovenstaande nieuwe definitie van **t** in, klik op Start. Beschrijf het resultaat.

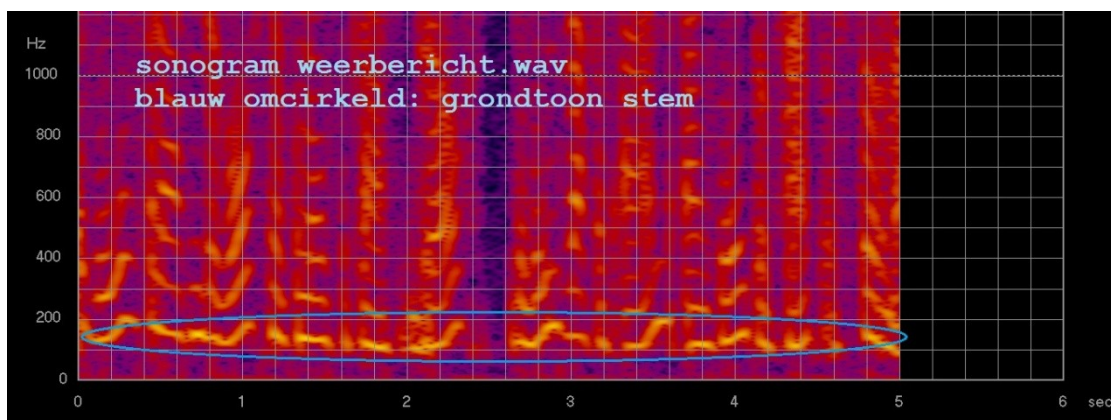
tot slot

Hiermee hebben we Granulaire Synthese gecreëerd volgens de regels van de kunst, maar waarschijnlijk ben je niet tevreden over het eindresultaat. De nieuwslezer klinkt alsof hij een ingrijpende stembandoperatie heeft ondergaan; z'n stem is onherkenbaar veranderd en produceert 'deep-throat'-geluiden die je doorgaans slechts hoort in het goedkoopste soort Hollywood horror slasher movies.

De oorzaak daarvan is onze keuze voor de lengte van het Hann-venster. In de code staat:

```
DuurFragment = 0,01*Fs
```

Elke korrel duurt dus 0,01 seconde. Dat dit veel te kort is, kunnen we vaststellen als we een sonogram maken van de originele stem, waarvan hieronder het belangrijkste deel is weergegeven.



In het blauw omcirkelde gebied zie je de grondtoon van de stem van de nieuwslezer afgebeeld, die voortdurend fluctueert ("prosodie", "zangerigheid") tussen 100 en 200 Hz. Er past dus vaak maar één stembandperiode in een korrel. Dat veroorzaakt dat schorre, boertige stemgeluid. Kiezen we bijvoorbeeld `DuurFragment = 0,02*Fs` dan zijn dat altijd minimaal twee periodes. Dan verdwijnt dat deep-throat effect en beginnen we de echte toonhoogte van de nieuwslezer goed te horen.

Met dit aanpassen van de instellingen van de Granulaire Synthesizer begint het echte sound design werk. "Tweaken" noemen ze dat in de audiostudio: net zo lang aan de knopjes draaien tot het geluid helemaal naar je zin is. Bij elk geluid dat je granuleert kijk je eerst weer naar de instellingen. Daarbij maak je graag gebruik van spectrum analyzers zoals het sonogram.

De code van de uiteindelijke granulering van weerbericht en sonogram vind je in [Preset Granulair Opdracht3 antwoord](#).