

**Ogg/Vorbis**

Een onderzoeksverslag van: Lucien Immink

Rotterdam, 27 september 2002

## **Voorwoord**

Dit document is geschreven in het kader van het onderzoek van Lucien Immink wat hij schreef in het vierde jaar van de opleiding Media Technologie. Van de student wordt verwacht een literair onderzoek te doen naar een relevant onderdeel van de opleiding, dit kan iets zijn wat de student is tegen gekomen tijdens de lessen, maar mag ook iets zijn wat de student interesseert. De totale duur van het onderzoek bedraagt 80 uur, wat gelijk staat aan 2 studiepunten.

Ik heb voor dit specifieke onderzoek gekozen omdat het onderwerp me al enkele jaren meer dan boeit. Thuis gebruik ik Ogg/Vorbis dagelijks om naar de, op mijn pc staande, muziek collectie te luisteren. Nu rees de vraag in mij op hoe het formaat (als het er 1 is) nu precies in elkaar zit. Het gaat mij dan niet om hoe het gemaakt is of hoe het werkt, klinkt.

## Samenvatting

Ogg codecs gebruiken vectoren van 8 van raw, gecomprimeerde data (packets). Raw packets mogen direct gebruikt worden door de transport mechanismes die hun eigen frames en packet-separatie mechanismes hebben (zoals UDP). Voor stream geïntendeerde opslag (zoals files) en transport (TCP streams), gebruiken Vorbis en ander Ogg codecs het Ogg formaat om frames, synchronisatie, synchronisatie na fout, herkenning na spoelen en andere informatie om data terug in packets te krijgen zonder decodering om zo aan de omvang van een packet te komen.

Raw packets zijn gegroepeerd en geëncodeerd in lopende pagina's van gestructureerde data, een *logische bitstream*. Meerdere logische bitstreams kunnen gecombineerd worden tot één *Physische bitstream*. De eerste pagina van een logische Ogg bitstream bestaat uit één, kleine header packet die genoeg informatie geeft om tot het gebruikte codec type te komen (type en inhoud).

De simpelste manier om logische bitstreams samen te voegen is aaneenschakelen (chaining).

Complete logische bitstreams worden achter elkaar in volgorde geplakt.

Logische bitstreams mogen ook samengevoegd worden in parrallel liggende groepen (grouping). Een voorbeeld hiervan is het apart streamen van audio en video streams, die verschillende codecs gebruiken en verschillende logische bitstreams.

Groepen gelijktijdig samengevoegde bitstreams mogen aanelkaar gesloten worden. De onstane physische bitstream houdt zich aan de regels van beide stream varianten.

Vorbis is een generale audio codec bedoeld om een zo breed mogelijk scala van audiobronnen te coderen. Zo kan vorbis overweg met high-end audio systemen (24bit – 192 kHz), maar evengoed met lage kwaliteit audio (8bit – 8kHz). Tevens bied vorbis ondersteuning voor meerdere kanalen audio (mono, stereo, 5.1 enz.) Vorbis kan tot 255 discrete kanalen tegelijk bevatten.

De vorbis Codec gaat van een complexe, met de psycho akoestiek rekening houdende encoder en een simpele decoder. Vorbis is derhalve makkelijker te decoderen dan mp3, maar heeft wel meer geheugen ruimte nodig.

De Vorbis decoder bestaat uit meerder, op zichzelf staande, onderdelen die een speciale stap van het decoderen van de stream voor hun rekening neemt. Elke Vorbis frame is gecodeerd volgens een 'master mode'. Een bitstream kan meerdere modes gebruiken. Het mode mechanisme wordt gebruikt om een frame te coderen. Een mapping bestaat uit een omgeschrijving hoe de kanalen zijn gekoppeld en een lijst van 'deelmappen' die een groepen kanaalvectoren samenvoegen om gegroepeerde encoding en decoding mogelijk te maken. Vorbis encodeerd een speciale 'floor' vector voor elk PCM kanaal. Deze vector is een lage-resolutie representatie vna het audio spectrum voor het gegeven kanaal in het gegeven frame, meestal in samenwerking met een filter. Het spectrale overblijfsel(residue) is de fijne structuur van het audio spectrum zodra de floor curve eruit is gehaald. Codeboeken zijn op zichzelf staande abstracties die entropie decoding en entropie gedecodeerde integere getallen gebruiken als een compensatie in een index van vectoren. De codeboeken worden gebruikt om Floor en Residue waardes te berekenen / verkrijgen.

De Vorbis tekst opmerking header is het 2e header packet wat een Vorbis stream begint. Het is bedoeld om kleine, tekst opmerkingen, niet noodzakelijke metadata. Niet noodzakelijke metadata behoort in een op zich zelf staande logische bitstream (meestal een XML stream) wat extra informatie verschaft over de bitstream. De opmerkingen header is een lijst van 8-bit vectoren; het aantal vectoren is kleiner dan  $2^{32} - 1$  en de lengte van elke vectoren is kleiner of gelijk aan  $2^{32} - 1$  bytes. De opmerkingen vectoren zijn gestructureer zoals een UNIX omgevings variabele.

Inhoudsopgave

<b>1</b>	<b>INLEIDING</b>	<b>5</b>
<b>2</b>	<b>OGG/VORBIS ?</b>	<b>6</b>
2.1	OGG	6
2.2	LOGISCHE EN PHYSICHE BITSTREAMS	6
2.2.1	<i>Eind tot eind structuur</i>	6
2.2.2	<i>Sequentiele samenvoeging (chaining)</i>	7
2.2.3	<i>gelijktijdige samenvoeging (grouping)</i>	7
2.2.4	<i>Sequentiele en gelijktijdige samenvoeging</i>	7
2.2.5	<i>Samenvoeg voorbeeld</i>	7
2.3	SAMENVATTING	7
<b>3</b>	<b>VORBIS</b>	<b>9</b>
3.1	APPLICATIE	9
3.2	AANNAMES	9
3.3	FORMAAT SPECIFICATIES	9
3.4	SAMENVATTING	9
<b>4</b>	<b>REFERENTIE VORBIS DECODER</b>	<b>10</b>
4.1	DE DECODER	10
4.2	Globale configuratie	10
4.3	MODE	10
4.4	MAPPING	11
4.5	FLOOR	11
4.6	RESIDUE	12
4.7	CODEBOOKS	12
4.8	SAMENVATTING	12
<b>5</b>	<b>VORBIS TEKST</b>	<b>13</b>
5.1	OPMERKINGEN ENCODING	13
5.1.1	<i>Structuur</i>	13
5.1.2	<i>Inhoud vector formaat</i>	13
5.1.3	<i>Veldnamen</i>	14
5.2	SAMENVATTING	14
<b>6</b>	<b>CONCLUSIE</b>	<b>15</b>

---

## 1 Inleiding

Dit onderzoek is verricht voor de opleiding MediaTechnologie van de Hogeschool van Utrecht te Amersfoort. Gesteld is dat elke student een literair onderzoek moet verrichten binnen het vakgebied van de opleiding. Met onderzoek wordt dan bedoeld: het verzamelen, bewerken en analyseren van gegevens om meer te weten te komen. Onderzoek is het combineren van gegevens afkomstig uit verschillende bronnen om tot nieuwe inzichten te komen.

Binnen dit kader is er voor dit specifieke onderzoek gekozen om het volgende te onderzoeken:

*“Hoe zit een Ogg/Vorbis file eruit?”*

Uit deze hoofdvraag zijn 2 deelvragen te destileren namelijk:

- Wat is Ogg en hoe ziet het eruit?
- Wat is Vorbis en hoe ziet het eruit?

Vanuit de literatuur die ik voor dit onderzoek heb gebruikt kwam naar voren dat er nog een belangrijk onderdeel is, namelijk Vorbis Text. Dit is dan ook het derde deel van mijn onderzoek geworden.

## 2 Ogg/Vorbis ?

Ogg en Vorbis zijn twee samenwerkende codecs voor het digitaliseren van geluid. Ogg is een container formaat, zoals avi bijvoorbeeld is voor video en Vorbis is het compressie formaat (mpeg4). In de volgende hoofdstukken worden beide codecs behandeld.

### 2.1 Ogg

Ogg codecs gebruiken vectoren van 8 van raw, gecomprimeerde data (packets). Deze gecomprimeerde packets hebben geen hoge graad van structuur, ze komen over alsof het 'streams' zijn zonder enige vorm van herkenning.

Raw packets mogen direct gebruikt worden door de transport mechanismes die hun eigen frames en packet-seperatie mechanismes hebben (zoals UDP). Voor stream geörienteerde opslag (zoals files) en transport (TCP streams), gebruiken Vorbis en ander Ogg codecs het Ogg formaat om frames, synchronisatie, synchronisatie na fout, herkenning na spoelen en andere informatie om data terug in packets te krijgen zonder decoding om zo aan de omvang van een packet te komen.

### 2.2 Logische en Physische bitstreams

Raw packets zijn gegroepeerd en geëncodeerd in lopende pagina's van gestructureerde data, een *logische bitstream*. Een logische bitstream bestaat uit pagina's, op volgorde, behorend tot één codec instantie. Elke pagina is een op zich zelf staand stuk, dit wil zeggen dat elke pagina los van de rest gedecodeerd kan worden. De grootte van een pagina kan verschillen ten opzichte van elkaar. Meerdere logische bitstreams kunnen gecombineerd worden tot één *Physische bitstream*. Een Physische bitstream bestaat uit meerdere logische bitstreams (boven op elkaar) op basis van een pagina en kan een extra header bevatten aan het begin. Deze extra header kan gebruikt worden om de meerdere logische bitstreams te indentificeren. Physische bitstreams pagina's worden gemaakt door gehele pagina's uit de logische bitstream te combineren tot één geheel. De decoder herbouwt de originele logische bitstreams uit de Physische bitstream door de pagina's uit de Physische bitstream te halen en deze op volgorde aan te bieden aan de verschillende logische bitstream decoder entiteiten. De simpelste Physische bitstream is één, logische bitstream zonder extra header. Deze wordt in de literatuur een 'degenererende stream' genoemd.

#### 2.2.1 Eind tot eind structuur

Een stream bestaat altijd uit gehele pagina's, iets wat makkelijk voor elkaar is te krijgen aangezien pagina's variabele grootte hebben.

Het begin en einde van een stream hebben een header die deze informatie doorgeeft aan de decoder.

De eerste pagina van een logische Ogg bitstream bestaat uit één, kleine header packet die genoeg informatie geeft om tot het gebruikte codec type te komen (type en inhoud). Dit om het indentificeren van de bitstream te vergemakkelijken. Om een bepaald media type te vinden (binnen de Ogg media typen) is slechts een kleine, bekende, hoeveelheid data nodig om deze te kunnen indentificeren.

Ogg Vorbis plaatst bijvoorbeeld de naam, revisie van het Vorbis codec, de sample snelheid en de audio kwaliteit in de header. Hierdoor is het zeer simpel geworden om deze Ogg Vorbis stream te indentificeren.

### 2.2.2 Sequentiele samenvoeging (chaining)

De simpelste manier om logische bitstreams samen te voegen is aaneenschakelen (chaining). Complete logische bitstreams worden achter elkaar in volgorde geplakt. De bitstreams overlappen elkaar niet. De laatste pagina van een gegeven logische bitstream wordt onmiddellijk opgevolgd door de eerste pagina van de volgende bitstream. Chaining is de enige logische  $\rightarrow$  fysieke bitstream conversie toegestaan in Ogg Vorbis.

Elk aaneengesloten logische bitstream moet een uniek serie nummer hebben binnen de Fysieke bitstream.

Ter volledigheid van dit stuk wordt in de volgende paragraaf zeer beknopt ingegaan op gelijktijdige samenvoeging, al is deze niet toegestaan binnen Ogg Vorbis.

### 2.2.3 gelijktijdige samenvoeging (grouping)

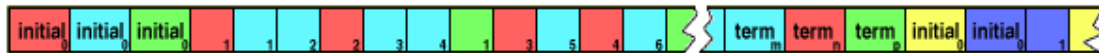
Logische bitstreams mogen ook samengevoegd worden in parrallel liggende groepen (grouping). Een voorbeeld hiervan is het apart streamen van audio en video streams, die verschillende codecs gebruiken en verschillende logische bitstreams. Hele pagina's van verschillende logische bitstreams worden samen gemixed.

### 2.2.4 Sequentiele en gelijktijdige samenvoeging

Groepen gelijktijdig samengevoegde bitstreams mogen aaneelkaar gesloten worden. De onstane fysieke bitstream houdt zich aan de regels van beide stream varianten.

### 2.2.5 Samenvoeg voorbeeld

Figuur 3.1 geeft een voorbeeld van een sequentieel gegroepeerde bitstream.



Figuur 3.1

In dit voorbeeld worden 5 logische bitstreams samengevoegd tot één fysieke bitstream.

Merk hierbij de volgende dingen op:

1. Gegroepeerde bitstreams beginnen allemaal tegelijk. Elke eerste pagina van een logische bitstream moet gegeven worden voor de data. Wanneer gelijktijdige samengevoegde bitstreams sequentieel aan elkaar worden gekoppeld mogen de volgende streams pas beginnen nadat alle streams uit de eerste gelijktijdige samengevoegde streams zijn afgesloten.
2. De pagina's van gelijktijdige samengevoegde bitstreams hoeven niet in een bepaalde volgorde geplaatst te worden. De enige eis die gesteld wordt is dat pagina  $n$  van een bepaalde logische stream pas wordt gegeven nadat pagina  $n-1$  van deze bitstream al is gegeven.

## 2.3 Samenvatting

Ogg codecs gebruiken vectoren van 8 van raw, gecomprimeerde data (packets). Raw packets mogen direct gebruikt worden door de transport mechanismes die hun eigen frames en packet-separatie mechanismes hebben (zoals UDP). Voor stream geörienteerde opslag (zoals files) en transport (TCP

streams), gebruiken Vorbis en ander Ogg codecs het Ogg formaat om frames, synchronisatie, synchronisatie na fout, herkenning na spoelen en andere informatie om data terug in packets te krijgen zonder decodering om zo aan de omvang van een packet te komen. Raw packets zijn gegroepeerd en geëncodeerd in lopende pagina's van gestructureerde data, een *logische bitstream*. Meerdere logische bitstreams kunnen gecombineerd worden tot één *Physische bitstream*. De eerste pagina van een logische Ogg bitstream bestaat uit één, kleine header packet die genoeg informatie geeft om tot het gebruikte codec type te komen (type en inhoud). De simpelste manier om logische bitstreams samen te voegen is aaneenschakelen (chaining). Complete logische bitstreams worden achter elkaar in volgorde geplakt. Logische bitstreams mogen ook samengevoegd worden in parrallel liggende groepen (grouping). Een voorbeeld hiervan is het apart streamen van audio en video streams, die verschillende codecs gebruiken en verschillende logische bitstreams. Groepen gelijktijdig samengevoegde bitstreams mogen aanelkaar gesloten worden. De onstane physische bitstream houdt zich aan de regels van beide stream varianten.



### 3 Vorbis

Voor stream geïntegreerde opslag (zoals files) en transport (TCP streams), gebruiken Vorbis en ander Ogg codecs het Ogg formaat om frames, synchronisatie, synchronisatie na fout, herkenning na spoelen en andere informatie om data terug in packets te krijgen zonder decoding om zo aan de omvang van een packet te komen. Ogg wordt dus alleen gebruikt voor bovengenoemde. Vorbis wordt gebruikt om de audio te comprimeren. Het volgende hoofdstuk gaat dan ook over Vorbis, hoe ziet een Vorbis Stream eruit?

#### 3.1 Applicatie

Vorbis is een generale audio codec bedoeld om een zo breed mogelijk scala van audiobronnen te coderen. Zo kan vorbis overweg met high-end audio systemen (24bit – 192 kHz), maar evengoed met lage kwaliteit audio (8bit – 8kHz). Tevens biedt vorbis ondersteuning voor meerdere kanalen audio (mono, stereo, 5.1 enz.) Vorbis kan tot 255 discrete kanalen tegelijk bevatten.

#### 3.2 Aannames

De vorbis Codec gaat van een complexe, met de psycho akoestiek rekening houdende encoder en een simpele decoder. Vorbis is derhalve makkelijker te decoderen dan mp3, maar heeft wel meer geheugen ruimte nodig.

Vorbis heeft van zichzelf geen frames, synchronisatie of fout protectie, het is slechts een methode om audio te coderen in individuele frames. Deze frames worden gecompriemd tot ruwe, ongeformateerde packets. De decoder kan deze packets accepteren en decoderen. Haalt hier de audio frames weer uit en voegt de audio frames samen. Vorbis is een Variabele Bit Rate (VBR) audio codec. De packets hebben geen minimum of maximum grootte, ook geen voorafgesproken grootte. Mp3 werkt met voorafgesproken grootte in de VBR stream.

Vorbis packets gaan er dus van uit dat er een container formaat is om voor de framing, synchronisatie en fout afhandeling te zorgen. Voorbeelden hiervan zijn Ogg (file) en RTP (multicast). In de rest van dit document wordt er vanuit gegaan dat de Vorbis stream in een Ogg stream gecapsuleerd is, maar dit is niet noodzakelijk.

#### 3.3 Formaat specificaties

Het vorbis formaat is gedefinieerd door de decoder. Een willekeurige encoder waarvan de packets kunnen worden gedecodeerd door de referentie vorbis decoder is een juiste vorbis encoder.

#### 3.4 Samenvatting

Vorbis is een generale audio codec bedoeld om een zo breed mogelijk scala van audiobronnen te coderen. Zo kan vorbis overweg met high-end audio systemen (24bit – 192 kHz), maar evengoed met lage kwaliteit audio (8bit – 8kHz). Tevens biedt vorbis ondersteuning voor meerdere kanalen audio (mono, stereo, 5.1 enz.) Vorbis kan tot 255 discrete kanalen tegelijk bevatten.

De vorbis Codec gaat van een complexe, met de psycho akoestiek rekening houdende encoder en een simpele decoder. Vorbis is derhalve makkelijker te decoderen dan mp3, maar heeft wel meer geheugen ruimte nodig.

Een willekeurige encoder waarvan de packets kunnen worden gedecodeerd door de referentie vorbis decoder is een juiste vorbis encoder.

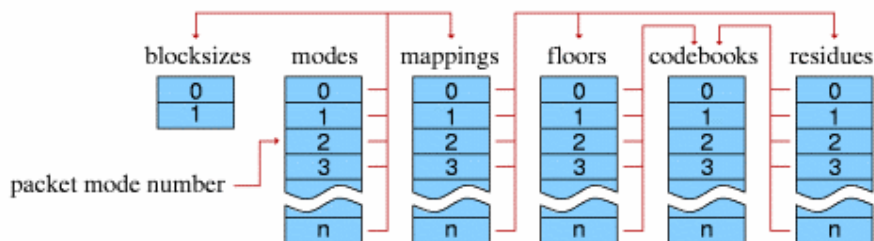
## 4 Referentie Vorbis Decoder

De Vorbis Codec gaat van een complexe, met de psycho akoestiek rekening houdende encoder en een simpele decoder. Vorbis is derhalve makkelijker te decoderen dan mp3, maar heeft wel meer geheugen ruimte nodig.

Een willekeurige encoder waarvan de packets kunnen worden gedecodeerd door de referentie Vorbis decoder is een juiste Vorbis encoder. Vanuit file perspectief is het onbelangrijk of er gekeken wordt naar hoe de file is ontstaan (encoderen) of hoe de file wordt behandeld (decoderen). Vanuit bovenstaande is het een verstandige zet om te kijken naar hoe een Vorbis file wordt behandeld, wordt gedecodeerd.

### 4.1 De decoder

De decoder bestaat uit meerder, op zichzelf staande, onderdelen die een speciale stap van het decoderen van de stream voor hun rekening neemt. Elke verschillende instantie van een component kan worden uitgewisseld met instanties van de zelfde klasse. Figuur 4.1 geeft een voorbeeld volgorde van een decoder. In de volgende paragrafen worden de verschillende onderdelen behandeld.



figuur 4.1

### 4.2 Globale Configuratie

Globale codec configuratie bestaat uit een paar audio gerelateerde onderdelen (sample snelheid, kanalen), Vorbis versie (altijd '0' in Vorbis I), bitrate en een lijst met component instanties. Alle andere configuraties zitten in de context van een specifiek component.

### 4.3 Mode

Elke Vorbis frame is gecodeerd volgens een 'master mode'. Een bitstream kan meerdere modes gebruiken. Het mode mechanisme wordt gebruikt om een frame te coderen. Hierbij wordt van meerdere mogelijkheden uitgegaan, de beste wordt uiteindelijk gekozen. Andere modes kunnen bijvoorbeeld aangeven hoe de frame grootte verandert van frame tot frame. Het mode nummer van een frame kan gezien worden als het configuratie object van de frame. Andere aspecten van de frame hangen af van het mode nummer. Een 'mode' configuratie bestaat uit een frame grootte, het window type (altijd 0, het Vorbis window, in Vorbis 1), het transformeer type (altijd type 0 in Vorbis 1) en het mapping nummer. Het mapping nummer geeft aan welke mapping configuratie moet worden toegepast om de frame te decoderen en te synthetiseren.

## 4.4 Mapping

Een mapping bestaat uit een omschrijving hoe de kanalen zijn gekoppeld en een lijst van 'deelmappen' die een groepen kanaalvectoren samenvoegen om gegroepeerde encoding en decoding mogelijk te maken. Deze deelmaps hebben geen referenties naar externe componenten, ze zijn een intern deel van een map.

Een submap is een configuratie die van toepassing is op een deelset van 'floor' en 'residue' vectoren binnen het mapping object. De submap fungeert als laatste laag, zo dat een specifieke en speciale 'floor' of 'residue' configuratie kan worden toegepast niet alleen op alle vectoren binnen een bepaalde mode maar aan specifieke vectoren in een specifieke mode. Elke submap geeft het juiste 'floor' en 'residue' instantie nummer om de spectrale 'floor' en spectrale 'residue' vectoren te decoderen van de submap.

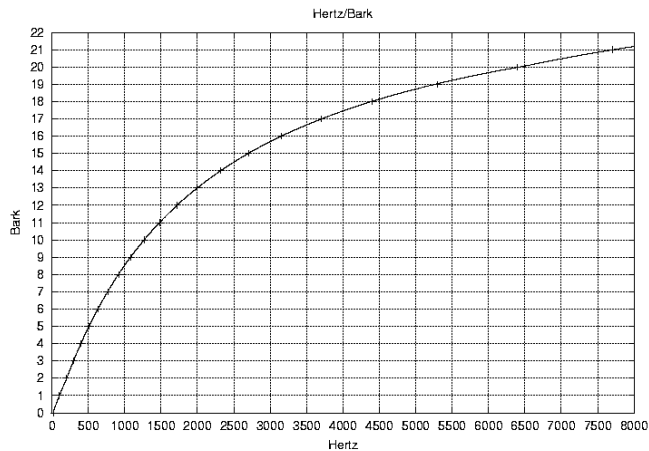
Een voorbeeld kan dit ophelderden:

Ga uit van een Vorbis stream die bestaat uit 6 kanalen, in het standaard 5.1 formaat. Het zesde kanaal, zoals normaal is in 5.1 bestaat uit alleen bas. Daarvoor zou het zonde zijn om het volledige spectrum te gebruiken om dit kanaal te coderen. Het submap mechanisme kan gebruikt worden om een floor en residue encoding toe te passen op kanaal 0 tot en met 4 en alleen de bas door te sturen naar het bas kanaal. Hierdoor wordt ruimte bespaard. In dit voorbeeld behoren kanalen 0 tot en met 4 tot submap 0 (welke aangeeft dat deze het gehele spectrum gebruiken) en kanaal 5 behoort tot submap1, welke alleen bas gebruikt.

## 4.5 Floor

Vorbis encodeerd een speciale 'floor' vector voor elk PCM kanaal. Deze vector is een lage-resolutie representatie van het audio spectrum voor het gegeven kanaal in het gegeven frame, meestal in samenwerking met een filter. Het wordt een vloer (floor) genoemd omdat de referentie encoder het altijd gebruikt heeft als basis voor de spectrale resolutie.

Een floor encoding mag uit twee types bestaan, Floor 0 gebruikt een LSP (Line Spectral Frequency) representatie van een dB amplitude schaal en Bark (zie figuur 4.2) frequentie schaal.



Figuur 4.2: Hertz/Bark frequentie model.

Floor 1 stelt een curve voor die voortkomt uit het interpoleren van een dB amplitude schaal en een lineaire frequentie schaal. De twee 'floors' uit te wisselen tijdens encoding of decoding. Floor 1 heeft wel een meer stabiel inter-frame gedrag, en dus de beste keuze voor stereo en hoge bitrate modus. Floor 1 is ook makkelijker te decoderen dan floor 0.

Floor 0 wordt nog niet als overbodig gezien, maar het wordt hedendaags minder gebruikt. Geen hedendaagse encoder maakt gebruik van floor 0 (Xiph.org's beta 4 encoder was de laatste). De gecodeerde/decodeerde getallen die door een floor vector verkregen worden zijn beide compact en maken gebruik van entropie encoding om ruimte te besparen. Hierdoor gebruiken floor configuraties meerdere codeboeken.

#### **4.6 Residue**

Het spectrale overblijfsel(residue) is de fijne structuur van het audio spectrum zodra de floor curve eruit is gehaald. Simpel gezegd is het gecodeerd in de bitstream, gebruik makend van vector quantizatie volgens een van drie specifieke packing/coding algoritmes (genummerd 0 tot en met 2). De packing algoritmes details zijn geconfigureerd door een residue instantie.

#### **4.7 Codebooks**

Codeboeken zijn op zichzelf staande abstracties die entropie decoding en entropie gedecodeerde integrale getallen gebruiken als een compensatie in een index van vectoren. Ze geven een vector met waarden terug.

De entropie coding in een Vorbis 1 codeboek wordt geleverd door een standaard Huffman binaire boom representatie.

De codeboeken worden gebruikt om Floor en Residue waardes te berekenen / verkrijgen.

#### **4.8 Samenvatting**

De decoder bestaat uit meerder, op zichzelf staande, onderdelen die een speciale stap van het decoderen van de stream voor hun rekening neemt. Elke Vorbis frame is gecodeerd volgens een 'master mode'. Een bitstream kan meerdere modes gebruiken. Het mode mechanisme wordt gebruikt om een frame te coderen. Een mapping bestaat uit een omschrijving hoe de kanalen zijn gekoppeld en een lijst van 'deelmappen' die een groepen kanaalvectoren samenvoegen om gegroepeerde encoding en decoding mogelijk te maken. Vorbis encodeerd een speciale 'floor' vector voor elk PCM kanaal. Deze vector is een lage-resolutie representatie vna het audio spectrum voor het gegeven kanaal in het gegeven frame, meestal in samenwerking met een filter. Het spectrale overblijfsel(residue) is de fijne structuur van het audio spectrum zodra de floor curve eruit is gehaald. Codeboeken zijn op zichzelf staande abstracties die entropie decoding en entropie gedecodeerde integrale getallen gebruiken als een compensatie in een index van vectoren. De codeboeken worden gebruikt om Floor en Residue waardes te berekenen / verkrijgen.

## 5 Vorbis Tekst

De Vorbis tekst opmerking header is het 2e header packet wat een Vorbis stream begint. Het is bedoeld om kleine, tekst opmerkingen, niet noodzakelijke metadata. Niet noodzakelijke metadata behoort in een op zich zelf staande logische bitstream (meestal een XML stream) wat extra informatie verschaft over de bitstream.

Het opmerkingen veld kan men vergelijken met het schrijfbaar gedeelte van een CDR. Het is bedoeld om net even die informatie te verschaffen waardoor een Vorbis (CDR in deze metafoor) stream herkent kan worden. Een voorbeeld kan hier verduidelijkend werken:

“Lacuna Coil – Comalies – 02 – Heaven’s a lie”

Hierin is Lacuna Coil de artiest, Comalies het album, 02 het nummer, Heaven’s a lie de titel.

### 5.1 Opmerkingen encoding

#### 5.1.1 Structuur

De opmerkingen header is een lijst van 8-bit vectoren; het aantal vectoren is kleiner dan  $2^{32} - 1$  en de lengte van elke vectoren is kleiner of gelijk aan  $2^{32} - 1$  bytes. De vector lengte is geëncodeerd. Naast de vector lijst is er een aparte vector voor de naam van de encoder (ook 8bit, lengte van 32 bit). De encoder van Xiph.org zet bijvoorbeeld de naam: “Xiph.Org libVorbis I 20020713”.

De opmerkingen header wordt op de volgende manier gedecodeerd:

- o [vendor\_length] = wordt gelezen als integer van 32 bit.
- o [vendor\_string] = wordt gelezen als UTF-8 vector met [vendor\_length] octets.
- o [user\_comment\_list\_length] = wordt gelezen als integer van 32 bit
- o itereer [user\_comment\_list\_length] {
  - [length] = wordt gelezen als integer van 32 bit
  - het bijbehorende commentaar = wordt gelezen als UTF-8 vector met [length] octets.}
- 7) [framing\_bit] = wordt gelezen als boolean
- 8) als ([framing\_bit] niet gezet is of niet bestaat) genereer dan een foutmelding.
- 9) klaar.

#### 5.1.2 Inhoud vector formaat

De opmerkingen vectoren zijn gestructureerd zoals een UNIX omgevings variabele. Dat houdt het volgende in:

```
comment[0]="ARTIST=me";
```

```
comment[1]="TITLE=the sound of Vorbis";
```

- Alle case-insensitive velden mogen bestaan uit ASCII code 0x20 tot 0x7D, 0x3D wordt uitgesloten (dit is de '='). ASCII 0x41 tot en met 0x5A (A-Z) wordt als het zelfde gezien als ASCII 0x61 tot en met 0x7A (a-z).
- De veldnaam wordt onmiddellijk gevolgd door ASCII 0x3D ('='). De '=' wordt gebruikt om de veldnaam variabele af te sluiten.
- 0x3D wordt gevolgd door een 8bit UTF-8 geëncodeerde veldinhoud.

### 5.1.3 Veldnamen

Hieronder staat een lijst met standaard veldnamen met een kleine beschrijving waar deze voor gebruikt worden. Geen enkel is verplicht om in te vullen. Een opmerkingen header kan geen, een of alle namen van deze lijst bevatten.

Veldnaam	Omschrijving
TITLE	Track naam
VERSION	Dit veld kan gebruikt worden om verschillende versies van dezelfde track te onderscheiden, vaak gebruikt bij remixes.
ALBUM	De collectie waar deze Vorbis file deel van is
TRACKNUMBER	Het nummer op de CD / collectie
ARTIST	De artiest(en) die het origineel gemaakt hebben
PERFORMER	De artiesten die het nummer hebben uitgevoerd
COPYRIGHT	Het Copyright attribuut
LICENCE	Licentie informatie, dit kan een URL zijn
ORGANIZATION	Organisatie die de nummer geproduceerd hebben.
DESCRIPTION	Een samenvatting van de inhoud
GENRE	Een indicatie tot welke muziekstroom het nummer behoort
DATE	Datum op welke het nummer is opgenomen
LOCATION	De plek waar het nummer is opgenomen
CONTACT	Contact informatie, dit mag een URL of een E-mail adres zijn.
LOCATION	Plek waar het nummer is opgenomen
CONTACT	Contact informatie, dit kan een URL of E-mail zijn.
ISRC	Het ISRC nummer van dit nummer

Het ISRC (International Standard Recording Code) is het internationale indentificatie systeem voor geluidsopnames. Elk ISRC is een unieke en permanente code voor een specifieke opname. Hierdoor is het mogelijk om automatisch een bepaalde opname te indentificeren.

### 5.2 Samenvatting

De Vorbis tekst opmerking header is het 2e header packet wat een Vorbis stream begint. Het is bedoeld om kleine, tekst opmerkingen, niet noodzakelijke metadata. Niet noodzakelijke metadata behoort in een op zich zelf staande logische bitstream (meestal een XML stream) wat extra informatie verschaft over de bitstream. De opmerkingen header is een lijst van 8-bit vectoren; het aantal vectoren is kleiner dan  $2^{32} - 1$  en de lengte van elke vectoren is kleiner of gelijk aan  $2^{32} - 1$  bytes. De opmerkingen vectoren zijn gestructureer zoals een UNIX omgevings variabele.

## 6 Conclusie

Ogg codecs gebruiken vectoren van 8 van raw, gecomprimeerde data (packets). Raw packets mogen direct gebruikt worden door de transport mechanismes die hun eigen frames en packet-separatie mechanismes hebben (zoals UDP). Voor stream geïntendeerde opslag (zoals files) en transport (TCP streams), gebruiken Vorbis en ander Ogg codecs het Ogg formaat om frames, synchronisatie, synchronisatie na fout, herkenning na spoelen en andere informatie om data terug in packets te krijgen zonder decodering om zo aan de omvang van een packet te komen.

Raw packets zijn gegroepeerd en geëncodeerd in lopende pagina's van gestructureerde data, een *logische bitstream*. Meerdere logische bitstreams kunnen gecombineerd worden tot één *Physische bitstream*. De eerste pagina van een logische Ogg bitstream bestaat uit één, kleine header packet die genoeg informatie geeft om tot het gebruikte codec type te komen (type en inhoud).

De simpelste manier om logische bitstreams samen te voegen is aaneenschakelen (chaining).

Complete logische bitstreams worden achter elkaar in volgorde geplakt.

Logische bitstreams mogen ook samengevoegd worden in parrallel liggende groepen (grouping). Een voorbeeld hiervan is het apart streamen van audio en video streams, die verschillende codecs gebruiken en verschillende logische bitstreams.

Groepen gelijktijdig samengevoegde bitstreams mogen aanelkaar gesloten worden. De onstane physische bitstream houdt zich aan de regels van beide stream varianten.

Vorbis is een generale audio codec bedoeld om een zo breed mogelijk scala van audiobronnen te coderen. Zo kan vorbis overweg met high-end audio systemen (24bit – 192 kHz), maar evengoed met lage kwaliteit audio (8bit – 8kHz). Tevens bied vorbis ondersteuning voor meerdere kanalen audio (mono, stereo, 5.1 enz.) Vorbis kan tot 255 discrete kanalen tegelijk bevatten.

De vorbis Codec gaat van een complexe, met de psycho akoestiek rekening houdende encoder en een simpele decoder. Vorbis is derhalve makkelijker te decoderen dan mp3, maar heeft wel meer geheugen ruimte nodig.

De Vorbis decoder bestaat uit meerder, op zichzelf staande, onderdelen die een speciale stap van het decoderen van de stream voor hun rekening neemt. Elke Vorbis frame is gecodeerd volgens een 'master mode'. Een bitstream kan meerdere modes gebruiken. Het mode mechanisme wordt gebruikt om een frame te coderen. Een mapping bestaat uit een omgeschrijving hoe de kanalen zijn gekoppeld en een lijst van 'deelmappen' die een groepen kanaalvectoren samenvoegen om gegroepeerde encoding en decoding mogelijk te maken. Vorbis encodeerd een speciale 'floor' vector voor elk PCM kanaal. Deze vector is een lage-resolutie representatie vna het audio spectrum voor het gegeven kanaal in het gegeven frame, meestal in samenwerking met een filter. Het spectrale overblijfsel(residue) is de fijne structuur van het audio spectrum zodra de floor curve eruit is gehaald. Codeboeken zijn op zichzelf staande abstracties die entropie decoding en entropie gedecodeerde integere getallen gebruiken als een compensatie in een index van vectoren. De codeboeken worden gebruikt om Floor en Residue waardes te berekenen / verkrijgen.

De Vorbis tekst opmerking header is het 2e header packet wat een Vorbis stream begint. Het is bedoeld om kleine, tekst opmerkingen, niet noodzakelijke metadata. Niet noodzakelijke metadata behoort in een op zich zelf staande logische bitstream (meestal een XML stream) wat extra informatie verschaft over de bitstream. De opmerkingen header is een lijst van 8-bit vectoren; het aantal vectoren is kleiner dan  $2^{32} - 1$  en de lengte van elke vectoren is kleiner of gelijk aan  $2^{32} - 1$  bytes. De opmerkingen vectoren zijn gestructureer zoals een UNIX omgevings variabele.

### Appendix A: Bronnenlijst

- Pro Audio Visie, September, jaargang 14: Algemeen informatie Ogg/Vorbis
- C'T magazine September 2002: Compressie formaten vergeleken
- <http://www.vorbis.com/intro.html>: Algemene informatie Ogg/Vorbis
- <http://www.openp2p.com/pub/a/p2p/2000/10/13/vorbis.html>: Kan Vorbis mp3 vervangen
- <http://ff123.net/>: Luistertest Vorbis, MP3, AAC en andere; veel informatie
- <http://www.codecreview.com/>: Links naar sites over codecs
- <http://www.xiph.org>: Homepage makers Ogg/Vorbis; zeer uitgebreide informatie over Ogg, Vorbis en Vorbis Tekst
- <http://www.infoanarchy.org/story/2002/9/8/23472/23921>: Samenvatting luistertest C'T magazine.