

Hoofdstuk 1 : Introductie tot multimedia

1.1 Wat is multimedia

- verschillende mensen -> verschillende (tegengestelde) meningen
- Alles te maken met **interactiviteit**, beeld en geluid.
- **Integratie** van verschillende applicaties is de toekomst (één multimedia-apparaat dat alles kan)
- **Componenten** : Vb. Teleconferencing, gedistribueerde lesgeven, zoeken in video/ beeldbases, coöperatieve werkomgevingen, audiocues voor deelnemers aan een videovergadering, schaalbare beelden, bewerkbare beelden, tools voor reverse engineering van films, stemherkenning...
- **Onderzoeksvelden** :
 - o **Processing** en **coderen** van multimedia (contentanalyse, beveiliging, compressie,..)
 - o **Systeemondersteuning** en **netwerken** (netwerkprotocollen, internet, besturingssystemen, servers en clients, databases, Quality of service)
 - o **Tools** en **applicaties** (leersystemen, hypermedia, user interfaces)
 - o **Interactie** en **integratie**. (web-everywhere toestellen, leren via multimedia, virtuele omgevingen)
- **Huidige projecten**:
 - o Objecttracken (cameragebaseerd)
 - o 3D motion capturing
 - o Meerdere standpunten (fotorealistische weergave van virtuele acteurs door vastleggen extra dimensies, zoals oppervlakte)
 - o 3D opname: hoog realistische opname van spraakanimaties
 - o Specifieke applicaties voor bijvoorbeeld andersvaliden
 - o Ontwerpen van 'slimme' kledij – sociaal netwerkende kledij
 - o Elektronisch gezondheid van patiënten monitoren (bij hen thuis)
 - o Verbeterde interactie (interfaces tussen echte en virtuele mensen – zoals voor verbeterd vertellen van verhalen)

1.2 Multimedia en Hypermedia

- **Evolutie**: kranten (eerste massacommunicatie met tekst en beeld), bewegende beelden (19^e eeuw), draadloze radio (Italië, eind 19^e eeuw), televisie (20^e eeuw)
- **Vanaf Computer**:
 - o Beschrijving van **Memex** (1945 – hypermedia – aanloop naar WWW)
 - o Jaren '60: eerste **hypertekstsystemen**:
 - Xanadu (1960 – eerste poging tot hypertekst)
 - NLS: On-Line System (1968 – zeer vroeg hypertekstprogramma)
 - FRESS (1969 - zie hierboven)
 - o Project van MIT : **Multiple Multimedia** -> eerste hypermedia videoschijf (Aspen Movie Map – 1976, soort google street view)
 - o **Tim Berners-Lee** stelt **WWW** voor aan het CERN (1985)
 - o **MPEG-1** (1991) – MPEG-2, MPEG-4 (90's)
 - o **PDA** (1991)
 - o **JPEG** -> JPEG2000 (1992)
 - o **MBone** audio multicast (1992) – audio over netwerk beperken in data
 - o **NCSA Mosaic** (1993) - eerste echte internetbrowser
 - o **Netscape** (1994)
 - o **JAVA** (1995)
 - o **DVD** (1996)
 - o **MP3** toestellen (1998)
- **Hypertext** systeem : Niet lineair lezen, met links naar andere plaatsen in de tekst.

- **Hypermedia:** niet per sé beperkt tot tekst, ook beelden, geluid, video
vb. WWW
- -> slaat op een **specifieke multimediatoepassing**
- **Multimedia:** computerinformatie die getoond kan worden door geluid, beelden, video en animatie. (naast tekst en tekeningen)
- Toepassingen: elektronische kranten en tijdschriften, WWW, online referentiewerken, thuisshoppen, interactieve TV, videoconferencing, video-on-demand, interactieve films

1.3 WWW

- 3 doelstellingen:
 1. universele toegang
 2. Effectiviteit
 3. Verantwoordelijkheid
- **Geschiedenis:**
 - **GML** (generalized markup language) – 1960
 - **SGML** (standard) – 1986
 - **HTML** en **HTTP** (Tim Berners-Lee) – 1990
 - **MOSAIC** (eerste populaire browser) – 1993
 - **Netscape** – 1994
 - **XML** – 1998
- **SMIL:** Synchronized Multimedia Integration Language, markup language voor multimediatekst

Hoofdstuk 2: Multimedia authoring en tools

2.1 Multimedia authoring

Video overgangen

- Effectieve manier om een verandering van scene aan te duiden.
- Types:
 - o **Cut** : abrupte overgang (eenvoudigst en meest gebruikt)
 - o **Wipe**: vervangen van pixels in een regio van het beeld door pixels van een ander beeld (links-rechts, rechts-links, verticaal, horizontaal, irisopening..)
 - o **Dissolve** (vervagen): Eerste beeld wordt gradueel over de tijd door het tweede vervangen door een geleidelijke mengeling van pixels te gebruiken.
 - **Cross Dissolve**: Elke pixel geleidelijk – lineaire functie
 - = fade-in (A is zwart) en = fade-out (B is zwart)
 - **Dither Dissolve**: Meer en meer pixels abrupt
 - = Whipe (in een specifieke geometrische volgorde)
 - o **Fade-In**: wit/zwart beeld gradueel vervangen door het te tonen beeld.
 - o **Fade-out**: beeld gradueel vervangen door een wit of wart beeld.
- Eigen type: Beeld laten 'wegschuiven'

Design problemen

- **Platform**: veel software lijkt portable, maar baseert zich op run-time modules
- **Videoformaat en resolutie**: NTSC (US), PAL (EU) en SECAM (FR) zijn niet compatibel, dus eerst een conversie nodig om iets af te spelen op een speler die een ander formaat gebruikt.
- **Geheugen en schijfruimte**: Minstens **128 MB RAM** en **20 GB** schijfruimte om accepteerbare performance en opslagruimte te hebben voor multimedia.
- **Levermethodes**:
 - o Niet iedereen heeft al DVD-RW drives
 - o CD-ROMs hebben niet genoeg schijfruimte en zijn te traag
 - o Elektronische levering hangt af van de bandbreedte, streaming is ook een optie.

Automatische authoring

- Een geavanceerde tool om nieuwe multimediapresentaties mee te maken
- Een mechanisme om van bestaande bronnen meer bruikbare multimediadocumenten te maken.
- Hypermedia documenten: het produceren van documenten die bedoeld zijn om **niet lineair** te lezen.
- 3 fases:
 - o **Capturing**: van een tekst, audio digitaliseren of een frame-grabber voor video. Dit is reeds goed ontwikkeld en verloopt automatisch.
 - o **Authoring**: Hoe de data best structureren om verschillende views toe te staan (beter dan één enkele statische view) ? Opslaan als bestaande datastructuren in de informatica (lijsten, grafen, bomen)
 - o **Publicatie**: Het doel van de multimediatools

2.3 VRML (Virtual Reality Modelling Language)

- Platformonafhankelijke taal om te bekijken op het internet
- **Doel**: Mogelijk maken om gekleurde objecten in een 3D omgeving te plaatsen
- **Interpreted** (waardoor relatief traag)
- Maar invloedrijk: eerste taal om een 3D wereld te tonen op het web
- **VRML 1.0 ('95)**: objecten maken + texturen en materialen specificeren
-

- **VRML 2.0:** mogelijkheid tot maken van interactieve werelden
 - o Animatie en geluid in een interactieve virtuele wereld
 - o Nieuwe objecten
 - o Java en javascript voor interactieve objecten acties gedefinieerd door de gebruiker
- Bestaat uit nodes in een hiërarchie die een scene beschrijven met één of meerdere objecten
- Enkel basisvormen: samenvoegen voor complexe vormen
- **Shape** node als algemene node, **Box, Cylinder, Cone, Sphere** zijn **geometry** nodes die basisobjecten plaatsen in de virtuele wereld
- **Complex** vormen:
 - o **IndexedFaceSet:** verzameling van vlakken om complexe objecten voor te stellen
 - o **Extrusion:** 2D doorsnede van een object (voor bv oppervlakte bloemblaadje)
- **Appearance** node : material node + texture node
 - o **Material** node: eigenschappen van een oppervlak zoals kleur, doorzichtigheid, belichting...
 - o **Texture** nodes: om textures te mappen op een object
 - **ImageTexture:** een externe JPEG of PNG
 - **MovieTextur:** een MPEG video-object
 - **PixelTexture:** een beeld maken om met ImageTexture te gebruiken
- **Text** node
- **Transform** nodes: alle shapes en teksten starten in het midden van de VRML wereld, om ze goed te zetten moeten transform nodes gebruikt worden. (bevat bv. **Translation, Scale, Rotation** nodes)
- **Viewpoint** nodes om het camerastandpunt of het aantal camera's te veranderen
- **Position** node, **orientation** node, **fieldOfView** node voor aanpassing positie, rotatie en kijkhoek
- Drie soorten belichting mogelijk:
 - o **DirectionalLight:** over de hele wereld in een bepaalde richting (zoals zon)
 - o **PointLight:** Licht in alle richtingen vanuit één punt.
 - o **SpotLight:** In een zekere richting vanuit een punt.
- Belichting -> realisme!
- **Background** node om achtergrondkleur/luchtkleur te veranderen
- **Panorama** node om een textuur op de zijanten van de wereld te mappen, gecentreerd rond de gebruiker (dus randen niet te bereiken) – hiervoor wordt een grote omvattende kubus gebruikt.

Hoofdstuk 3: Representaties voor beelden en beelddata

3.1 Datatypes

1-bit beelden

- Bestaande uit pixels of pels (picture elements)
- Elke pixel als 1 bit (die aan of uit staat) -> **Binair beeld**
- **Monochroom beeld**: want bestaat uit slechts 1 kleur (ook nog andere soorten monochrome beelden) – 640 x 480 bits ->/(8*1000) ->38.4 kB

8-bit grijswaarde beelden

- elke pixel heeft een grijswaarde tussen de 0 en de 255, elke pixel in 1 byte
- **Bitmap**: tweedimensionale tabel met pixelwaarden (ook zo in videogegeugen)
- **Beeldresolutie**: het aantal pixels in een digitaal beeld
- Bitmap moet opgeslagen worden in de hardware, dit noemen we de **Frame Buffer** (grafische kaart wordt hiervoor gebruikt), als de resolutie van de grafische kaart niet groot genoeg is voor een bepaald beeld wordt de data tijdelijk bewaard in het RAM-geheugen.
- Kan gezien worden als een set van 1-bit **bit-planes** (bit wordt op 1 gezet als de beeldpixel een niet-nul waarde heeft op of boven dat niveau)
- Grafische kaarten kunnen bitplane data aan videorate refreshen (60 cycles per seconde), maar kunnen de data niet goed bijhouden (zoals RAM dat kan)

Dithering

- Probleem: een dot kan enkel wel of niet geprint worden, geen grijswaarden mogelijk (2-level of 1-bit printers)
- Omzetten van waarden tussen 0 en 255 naar patronen die aangenaam ogen. (meer en meer gevuld naarmate een pixel donkerder wordt)
- Hoofdstrategie houdt in dat een pixel vervagen wordt door een groter patroon (bv. 2x2 of 4x4) – gelijkaardig aan het analoge **halftone printing**, hierbij worden cirkels zwarte inkt die variëren in grote geprint om tintovergangen te simuleren.
- Bijhouden van een dither matrix:
 - o Delen van de pixelwaarde door het aantal elementen
 - o 1 zetten als pixelwaarde hoger is dan het element
- n x n matrix zorgt voor $n^2 + 1$ **intensiteitslevels**
bv. 2*2: 0 dots ingevuld op gelijk welke manier + 1 + 2 +3 + 4 = 5 levels
- Spatiale resolutie (aantal lijnen per inch) omwisselen tegen intensiteitsresolutie (kleurintensiteit). De intensiteit kan verhoogd worden door pixels te vervangen door grotere matrices, maar dan wordt je beeld ook effectief groter, waardoor er minder lijnen per inch zijn en je een blurry effect krijgt.
- Elke pixel vervangen door 4x4 betekent 16x groter beeld!
- Bij **Ordered dithering** gaat men op dezelfde manier te werk, maar dan kijkt men naar de waarde van een pixel zelf en die wordt dan geprint als de waarde hoger is dan de dithermatrixwaarde (voor die pixel), daarna wordt de matrix over 4 pixels horizontaal en 4 verticaal verschoven.

Datatypes

- Meest voorkomend: 8 bit en 24 bit
- Sommige hardware/besturingssysteemspecifiek, andere cross-platform
- Applicaties kunnen datatypes herkennen en vertalen (indien niet cross-platform)
- **Compressietechnieken** maken deel uit van de meeste beeldformaten, deze kunnen **lossy** of **lossless** zijn.

24-bit kleurenbeelden

- Elke pixel voorgesteld door 3 bytes, gewoonlijk RGB
- 2^{24} mogelijke kleuren = +- 16 miljoen
- Opslagruimte! (640x480 al 921 kB (640*480*24/8000))
- Meestal opgeslagen als 32 bit met extra informatie voor special effects (**alfa-waarde**)

8-bit kleurenbeelden

- Voor backward compability nodig
- = 256 kleuren
- Gebruiken LUT (Look Up Table) om kleurinformatie te bewaren
- Beeld bewaart geen kleur, maar een index in de LUT met een 3-byte waarde voor de kleur
- Meest gebruikte kleuren kiezen uit een beeld om deze voor te stellen in de LUT, d.m.v. een 3D histogram. (clustering ontdekken)
- 1/3 van de grootte – overhead LUT te verwaarlozen = $256 \times 3 + 1$ bytes = 1 kB

LUT

- Zie hierboven
- Bewaard in de fileheader
- Ook **palette** genoemd
- **Colorpickers** bestaan uit klikbare blokken die gelinkt zijn aan een LUT
- Eenvoudige animatie mogelijk via LUT, namelijk **Color Cycling** of **Palette Animation**.
-
- **Kleurendithering:**
-
- 1 bit per kleurkanaal en de R, G en B dots verdelen over de ruimte (beter als dit niet gedaan wordt, maar goed genoeg voor goedkope printers), beter vooraf mengen.
- Wanneer slechts een beperkt aantal kleuren voorstelbaar zijn, kan gebruik worden gemaakt van het uitmiddelen van de intensiteit van naburige pixels om kleuren na te bootsen.
-
- **Lut opstellen:**
-
- RGB kubus opdelen in gelijke delen voor elke dimensie en het centrum van elk blokje nemen als entry voor de LUT en daarna de ranges herschalen.
- Mensen gevoeliger voor R en G, dus deze herschalen naar 3 bits (0..7), dan voor B, waarvoor we slechts 2 bits gebruiken (0..3)
- Als een kleine verandering in RGB resulteert in een nieuwe code krijgen we een randartefact.
- Beter: **Median-cut algortime**
- Probeert meeste bits te plaatsen waar ze het meest nodig zijn, daar waar kleuren het meest geclusterd zijn.
- R-waarden sorteren en mediaan nemen: kleiner = 0, groter = 1 (mediaan is waar de helft van de pixels groter zijn en de helft kleiner)
- De G-waarden van de pixels met een 0 label sorteren en de mediaan nemen: kleiner = 00, groter = 01. Daarna voor de waarden met een 1 label. Daarna zelfde met B-waarden.
- RGBRGRG is uiteindelijke resultaat
- Dit vormt de indexwaarden, corresponderende kleuren kunnen de centums van de blokjes zijn.
- Kleur wordt 'genomen' als hij bij de grote helft zit => beter onderscheid op basis van het beeld
- Bv. bij rode appels zal het veel langer duren voor een pixel een 1 krijgt voor rood dan anders.
- Nadeel: overhead door berekenen histogrammen/mediaan en opslaan LUT per beeld
- Voordeel: gebruik gemaakt van de gevoeligheid voor R en G, simpel
-

3.2 Fileformaten

- JPEG op dit moment belangrijkst
- 8-bit GIF historisch belangrijk door verbinding met WWW en HTML (eerste herkende beeldformaat op het web)

GIF (Graphics Interchange Format)

- Bedoeld om over telefoonmodems te verzenden
- Best voor beelden met weinig verschillende kleuren (want slechts 8 bit beschikbaar – 256 kleuren)
- Smpel formaat, maar bevat toch veel algemene concepten
- **Interlacing**: lijnen niet sequentieel doorgestuurd, maar over heel het beeld met enkele lijnen tussen.
- Ondersteuning van simpele **animatie**
- Beeld van een GIF-bestand

GIF signature (6 bytes)	GIF87a (vb.)
Screen descriptor (7)	breedte (pixels, LSB, 2)
	hoogte (pixels, LSB, 2)
	m(1)-cr(3)-0(1)-pixel(3)
	achtergrond
	0 0 0 0 0 0 0
Global color map	Rood intensiteit
	Groen intensiteit
	Blauw intensiteit
Image descriptor (10)	komma
	linkerkant (pixels, LSB,2)
	bovenkant (zelfde)
	breedte (zelfde)
	hoogte (zelfde)
	m-i-0-0-0-pixel(3)
Local color map	
Raster area	
GIF terminator	

- **Screen descriptor:** voor vlaggen
 - m (global color map?), cr +1 (**aangevraagde** color resolution - aantal bits voor kleuren), overbodige 0, pixel +1 (bits/pixel, meestal maar niet altijd zelfde als cr)
 - LUT-index voor **background**
 - Byte met enkel nullen
 - Kleine kleurenresolutie goed voor bv. weergave op schermen met kleine resolutie zoals polshorloges
- **Global color map:** voor als geen lokale map voorzien is (=LUT)
lengte = $2^{(\text{pixel}+1)}$ -> 3x256 mogelijkheden, want gesorteerd van 0 tot 255
- Kan meerdere beelden bevatten, uitbreidbaar want alles genegeerd tussen einde ene beeld en begin van het volgende beeld (aangegeven door komma).
- **Image descriptor** (één voor elk beeld)
 - **Interlacing:** laat toe al een deel van het beeld te tonen voor het volledig geladen is, vergelijkbaar met **progressive mode** in JPEG. (4-pass sequence)
 - start rij 0: om de 8
 - start rij 4: om de 8
 - start rij 2: om de 4
 - start rij 1: om de 2
- De **rasterdata** wordt eerst gecompriemd met LZW (Lempel-Ziv-Welch)
- Mogelijkheid voor **Extension Blocks**
- Gesorteerde LUT zodat als een decoder slechts een beperkt aantal kleuren ter beschikking heeft de belangrijkste genomen worden.

JPEG (Joint Photographic Experts Group)

- Op dit moment de belangrijkste standaard
- Maakt gebruik van specifieke beperkingen van het menselijk zicht voor hoge compressierates
- Kunnen niet goed hoge spatiale frequenties zien (veel veranderingen op kleine x,y), vooral bij kleur. Daarom kan dit uitgemiddeld worden en zelfs deels weggelaten.
- Zelf compressierate instellen (tot 100%, gemiddeld Q=10% met aanneembare kwaliteit)
- Default = 75% -> 5.6% van originele grote (10% komt overeen met 1.5%), GIF comprimeert tot 23% van het origineel.

PNG (Portable Network Graphics)

- bedoeld als vervanging van GIF
- kan tot 48 bits kleurinformatie bijhouden
- Interlacing in 8x8 blokken i.p.v. per lijn

TIFF (Tagged Image File Format)

- Ondersteuning voor tags
 - Type compressie
 - Veel beeldtypes (1-bit, grijswaarden, 8-bit kleur, 24-bit RGB,...)
 - Lossless -> JPEG Tag voor optionele compressie (maar minder controleerbaar, niet echt beter)
- Containerstandaard (geen compressiestandaard)
- Nu worden URI's gebruikt in de plaats van tags (behalve bij o.a. EXIF)

EXIF (Exchange Image File)

- Beeldformaat voor digitale camera's
- Gebruikt JPEG voor compressie
- Veel tags voor hogere kwaliteitsprinten (kleurcorrectie-algoritmes)
 - Flits
 - Exposure
 - Lichtbron
 - Witbalans
 - Scenetype
- Ondersteunt koppeling van audio

Hoofdstuk 4 : Kleur in beeld en video

4.1 Kleurwetenschap

Licht en spectra

- Licht is een elektromagnetische golf, kleur wordt gekarakteriseerd door zijn golflengte
 - o Lasers: 1 golflengte
 - o Meeste lichtbronnen: grote hoeveelheid golflengtes (daglicht bijna alle golflengtes)
 - o Mensen kunnen slechts een beperkt interval zien (400 – 700 nm)
 - o Kort = blauw, lang = rood
- **Spectrophotometer**: een toestel om zichtbaar licht te meten (door middel van lichtbreking worden de golflengtes opgesplitst)
- **Dispersie**: regenboogeffect
- **Spectral Power Distribution (SPD): Spectrum**, toont de relatieve hoeveelheid lichtenergie (elektromagnetisch signaal) per golflengte (λ voor golflengte) $E(\lambda)$
- Curve uit metingen is eigenlijk binnen intervallen van ongeveer 5-10 nm, dus eigenlijk niet continu.

Menselijk zicht

- Oog werkt als een lens die een beeld plaatst op het netvlies (ondersteboven gespiegeld).
- Het netvlies bestaat uit een reeks staafjes (rods) en drie soorten kegeltjes (cones).
- De staafjes dienen vooral onder slechte lichtomstandigheden.
- Drie soorten kegeltjes het meest gevoelig voor R, G en B (elke soort voor één kleur)
- Verderop weinig geweten, waarschijnlijk maken de hersenen gebruik van verschillen en combinaties van kleuren (in een hoog lichtniveau achromatisch kanaal).

Spectrale gevoeligheid van het oog

- Meest gevoelig voor licht in het midden van het zichtbare spectrum
 - Gevoeligheid voor blauw is gevoelig lager! (bijna niet zichtbaar op een grafiek), waarschijnlijk door zijn late toevoeging aan de evolutie
 - **Luminous-efficiency** functie: som van gevoeligheden per kleur $V(\lambda)$
 - BGR op een grafiek (volgorde van kleuren)
 - Gewoonlijk worden de functies R,G,B voorgesteld als :
- $$q(\lambda) = (q_R(\lambda), q_G(\lambda), q_B(\lambda))^T$$
- Gevoeligheid van de staafjes lijkt op luminous-efficiency, maar iets meer naar R toe. Gevoelig voor een groot aantal golflengtes, maar produceren enkel een signaal voor de waarneming van zwart-wit.
 - Verdeling van kegels is 40:20:1 (RGB) (\Rightarrow achromatisch kanaal = $2R + G + B/20$)
 - De respons in elk kleurenkanaal van het oog is proportioneel met het aantal neuronen die 'schieten', elke golflengte in het spectrum veroorzaakt een bepaald aantal afvurende neuronen. De respons hierop is afhankelijk van de gevoeligheid.
 - Totale respons op elk van de drie kleuren is de som van alle golflengtes uit het spectrum vermenigvuldigd met de respons erop: (G en B analoog)

$$R = \int E(\lambda)q_R(\lambda)d\lambda$$

- werkt ENKEL voor objecten die **zelf licht geven!**

Beeldvorming

- Oppervlakken reflecteren verschillende hoeveelheden licht op verschillende golflengtes
- Donkere oppervlakken reflecteren minder energie dan lichte oppervlakken
- Reflectiefunctie : $S(\lambda)$
- Licht wordt uitgestoten ($E(\lambda)$), licht wordt gereflecteerd ($S(\lambda)$) en gefilterd door het menselijk oog -> de kegeltjes ($C(\lambda)$)
- $C(\lambda)$ is het **kleursignaal**, het product van $E(\lambda)$ en $S(\lambda)$
- totale respons, rekening houdend met reflectie:

$$R = \int E(\lambda)S(\lambda)q_R(\lambda)d\lambda$$

Camerasystemen

- Gemaakt als het oog, produceert drie signalen per pixel (komt overeen met locatie op de retina)
- Analoge signalen omgezet naar digitale, gedumpt in integers en bewaard (8 bit precisie betekent hier dat voor zowel R, G, als B het min. 0 is en het max. 255 is)
- Het licht dat het oog van de computergebruiker binnengaat is dat wat het scherm uitstoot, daarom moeten we de $E(\lambda)$ -functie kennen.

Gammacorrectie

- RGB waarden in een beeld worden opnieuw geconverteerd naar analoge signalen en sturen de elektronenkanonnen aan in een kathodestraalbuis (CRT)
- We zouden graag hebben dat een CRT systeem licht produceert in lineaire functie met dit voltage, dit is echter niet het geval.
- Uitgezonden licht wel ruwweg proportioneel met het voltage verheven tot een macht, deze macht noemen we **gamma**. (ongeveer 2.2)
- Als de grootte in het rode kanaal R is, dan zendt het scherm licht uit met de proportie van R tot de macht gamma.
- SPD komt overeen met de rode fosforverf dat het doel is van het elektronenkanon voor kanaal elektronenkanon
- **Gammacorrectie** betekent dan dat voor uitzending het signaal tot $1/\gamma$ wordt verheven om de macht op te vangen en zo te komen tot lineaire signalen.
- Zonder deze correctie is het beeld te donker

Color-match functies

- De drie basiskleuren voor additieve kleurenmenging worden de **color primaries** genoemd.
- **Colorimeter**: een toestel dat gebruikt wordt bij het experimenteren met kleuren matching. Door de helderheid van de drie basiskleuren aan te passen moet de proefpersoon een te matchen licht nabootsen.
- Sommige lichten kunnen niet nagebootst worden door een lineaire combinatie van de primaries, daarvoor moet een licht dan van de bovenkant naar de onderkant verplaatst worden en het te matchen licht beschijnen in plaats van het witte scherm.

CIE chromaticiteitsdiagram

- Om te vermijden dat r onder de x-as gaat is een stel fictieve primaries opgesteld, dit heeft geleid tot **de** color-matching functies
- Komen voort uit de rgb color-matching door het uitvoeren een lineaire transformatie (met een 3x3 matrix)
- notatie: $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$
- zodanig gekozen dat y overeenkomt met de luminous-efficiencycurve ($V(\lambda)$)

- Voor een algemene SPD, essentiële colorimetrische informatie (totale respons) om kleur te karakteriseren is volgende set tristimulus-waarden: XYZ (analoog voor Y en Z)

$$X = \int E(\lambda)\bar{x}(\lambda)d\lambda$$

- Y wordt hier de **luminantie** genoemd
- We zitten met drie waarden => 3D voorstelling nodig, maar omdat dit moeilijk voor te stellen is werd een 2D voorstelling gemaakt.
- Hiervoor dienen we de grootte van de vectoren te normaliseren door te delen door te delen door $X + Y + Z$. Hieruit verkrijgen we dan x , y en z (y en z analoog)

$$x = X/(X + Y + Z),$$
 waardoor $x+y+z = 1$ en $z = 1-x-y$
- x , y en z noemen we de **chromaticiteit**.
- Gewoonlijk levert projectie van elke tristimulusvector (X,Y,Z) op het vlak $(1,0,0),(0,1,0),(0,0,1)$ een projectie op in het $z=0$ vlak onder de bissectrice (van linksboven naar rechtsonder)

(Aangezien x en y genormaliseerd zijn, zijn ze altijd kleiner of gelijk aan 1 en aangezien $x + y + z = 1$, is ook hun som altijd kleiner of gelijk aan 1)
- **chromaticiteitsdiagram**: paardenhoef bij monochromatisch licht.
 - o onderste rechte lijn: **line of purples**, verbindt de extremen van het zichtbare spectrum (400 en 700 nm)
 - o de hoof zelf: **spectrum locus** (meetkundige figuur gevormd door het verbinden van een verzameling punten die voldoen aan bepaalde voorwaarden), toont de chromatische waarden (x,y) van monochromatisch licht op elke zichtbare golflengte.
 - o colormatching functie zodanig opgesteld dat $\bar{x}, \bar{y}, \bar{z}$ allemaal dezelfde oppervlakte onder hun curve hebben, daarom geldt dat voor een witte verlichter met alle SPD waarden gelijk aan één (een **equi-energie wit licht**) de chromaticiteitswaarden $(1/3,1/3)$ zijn. Dit noemen we het **witpunt**.

als $E(\lambda)$ altijd 1 is, dan blijft bij XYZ enkel nog $\bar{x}, \bar{y}, \bar{z}$ over onder de integraal en aangezien die eenzelfde oppervlakte hebben onder de curve geldt $X=Y=Z$, maar dan is $x = X/(X+X+X) = 1/3$ (analoog voor y en z)
 - o Verschillende witte spectra kunnen gekozen worden als standaard verlichter, elk met hun eigen witpunt (bv. verlichters A (gloeilamp) en C (poging tot daglicht), daglichttypes D65 en D100, fluorescent F2) -> maar nog steeds rond 1/3
 - o Chromaciteiten op de rand stellen de pure kleuren voor, hoe dichter bij het witpunt, hoe minder **gesatureerd**.
 - o Mix tussen twee lichtkleuren ligt op hun verbindingslijn.
 - o enkel bij **additieve kleurmenging**
 - o **dominante golflengte** = lijn tussen een lichtkleur op de rand en het witpunt (blijft doorlopen door voorbij het punt op de rand), **complementaire dominante golflengte** is de lijn na doortrekking door het witpunt als de dominante golflengte vertrok vanuit een lichtkleur op de line of purples.
 - o **set van complementaire kleuren** voor een gegeven kleur wordt gegeven door alle kleuren op de lijn door het witpunt.
 - o **excitation purity(verzadigingsgraad)**: (uitgedrukt in %) afstanden van het witpunt naar een gegeven kleur over afstanden tot de dominante golflengte van die gegeven kleur (het punt op de rand).

Kleurenmonitorspecificaties

- Deels gespecificeerd door het gewenste witpunt wanneer alle elektronenkanonnen met hun hoogste energie geactiveerd worden.
- We willen een gespecificeerd witpunt bereiken op $R'=G'=B'=1$, met R' de gamma-gecorrigeerde en genormaliseerde versie van R .
- NTSC en SMPTE: Noord-Amerikaanse/Japanse standaard met SMPTE de nieuwere versie (gebaseerd op D65 ipv C)
- EBU: standaard in Europa, gebruikt bij PAL en SECAM systemen.
- Gain-control voor elk elektronenkanon instellen om de chromaticiteiten te bepalen.

Out-of-Gamut kleuren

- Hoe kunnen we toestel-onafhankelijke kleuren (x,y-chromaticiteitswaarden) genereren, gebruik makende van toestel-afhankelijke kleurwaarden (RGB)
- Uit elke (x,y) willen we die RGB-triple vinden, gegeven een gespecificeerde (x,y,z)
- z vinden we uit $z = 1 - x - y$ en RGB uit de door de produceur gespecificeerde chromaticiteiten.
- Alle waarden van RGB moeten verschillende van nul zijn, anders zien we bv. enkel de chromaticiteit van de rode fosfor.

$$\begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Deze eerste matrix noemen we in het vervolg **M**.

- Als (x,y) gegeven is (genormaliseerde kleuren -> zie chromaticiteitsdiagram), dan moet x,y,z geïnverteerd worden om de juiste RGB te krijgen.
- Als R, G of B negatief is wil dit zeggen dat het kleur zichtbaar is voor mensen, maar niet toonbaar is op het toestel.
- De set van toonbare kleuren noemen we het **gamut** van een toestel, als een kleur niet toonbaar is, dan is hij bijgevolg **out of gamut**.
- Twee methodes om dit te op te lossen:
 - o Het dichtste in-gamut kleur kiezen, dus een kleur op de driehoek die het dichtst bij het echte kleur ligt.
 - o Het dichtste complementaire kleur kiezen, dit kan door een verbinding te maken met het witpunt en dan het kleur op de driehoek te nemen. Dit is meestal de beste oplossing omdat we minder gevoelig zijn aan saturatie dan aan een verschil in tint.
- **Grassman's** wet: het matchen van kleuren is **lineair**. Als we 3 lichten, afkomstig van de drie fosfors combineren kunnen we enkel kleuren maken binnen de convexe set afgeleid van deze drie lichten. Als we namelijk 2 lichten combineren L1 en L2 dan geldt voor hun RGB waarden: $R = R1+R2$, $G=G1+G2$ en $B=B1+B2$. Deze convexe set die we afleiden zal in het geval van schermen een driehoek zijn omdat we bij schermen werken met drie componenten (RGB).

Witpuntcorrectie

- Tot nu toe enkel met xyz (enkel chromaticiteit) gewerkt, terwijl we XYZ (ook "magnitude") willen kunnen mappen naar een toestel-specifieke RGB.
- Daarnaast willen we dat wanneer R, G en B hun hoogste waarde hebben, we het witpunt krijgen.
- De methode om deze twee zaken op te lossen is **witpuntcorrectie**.
 - o Neem $Y(\text{witpunt}) = 1 \Rightarrow (X/Y, Y, Z/Y) = XYZ(\text{witpunt})$
 - o Op zoek gaan naar een set van **correctiefactoren**, zodat als we, wanneer we de "gains" van de elektronenkanonnen vermenigvuldigen met deze waarden we het witpunt (in termen van XYZ) krijgen bij $R=G=B=1$.
 - o De correctiefactoren drukken we uit in de vorm van een diagonaalmatrix D ($=\text{diag}(d1,d2,d3)$), zodat $XYZ(\text{witpunt}) = M*D*(1,1,1)^T$ en aangezien $D*(1,1,1)^T = (d1,d2,d3)^T$ is dit gelijk aan $M*(d1,d2,d3)^T$

XYZ -> RGB

- Transformatiematrix van XYZ naar RGB is T ($=M*D$)
- Dit geeft dan $(X,Y,Z)^T = T * (R,G,B)^T$ (ook voor andere punten dan het witpunt)

Met gammacorrectie

- Meestal wordt er gewerkt met niet-lineaire (i.t.t. hierboven) signalen waarop gammecorrectie is toegepast (deze R', G', B' signalen worden geproduceerd door bv. een camcorder)
- Oplossing:
 - o Eerst de lineaire RGB berekenen met $(X, Y, Z)^T = T^*(R, G, B)^T$ en dan gammacorrectie toepassen.
 - o Meestal niet zo gedaan, meestal wordt de Y-waarde gewoon gebruikt zoals ze is, maar dan voor niet-lineaire signalen (R', G' en B'). Het enige wat verandert is dat we Y nu Y' noemen.
- De betekenis van Y' is dat ze de helderheid van een pixel beschrijft

4.2 Kleurenmodellen in beelden

CMY kleurenmodel : subtractieve kleurmenging

- Voor inkt, geprint op papier: niet rood + groen is geel, maar gele inkt trekt blauw licht van wit licht af en reflecteert rood en groen.
- In plaats van R, G en B als primaries hebben we primaries nodig die overeenkomen met -R, -G en -B, dit zijn C (cyaan), M (magenta) en Y (geel).
- Cyaan filtert rood weg, Magenta filtert groen weg en Geel filtert blauw weg.
- Wit komt enkel voor als het medium wit is.

RGB -> CMY

- $C = 1 - R, M = 1 - G, Y = 1 - B$
- $R = 1 - C, G = 1 - M, B = 1 - Y$

CMYK: undercolor verwijdering

- C, M en Y mengen zou moeten zwart geven, maar in plaats krijgen we een modderig bruin.
- Echte zwarte inkt gebruiken is veel goedkoper dan de kleuren te mengen tot zwart, dit is niet enkel goedkoper maar ook scherper.
- Bereken dat deel van de CMY mengeling dat zwart zou zijn, verwijder het uit de proporties van de kleuren en voeg echt zwart toe in de plaats.

Printer gamuts

- Ideaal gezien zou cyaan, rood volledig moeten wegfilteren en groen en blauw volledig moeten doorlaten. Deze blokkeringsverven worden enkel nagebootst binnen de industrie.
- Dit leidt tot "crosstalk" tussen de kleurenkanalen (=> moeilijk bereikbare kleuren bij printen in te schatten)
- Gamut overlapt NTSC en is een stuk kleiner. (witpunt heet hier nu zwartpunt)
- 6-hoek ipv 3-hoek (naast de drie componenten, ook nog drie combinaties: CM, CY en MY)

4.3 Kleurenmodellen in video

YUV kleurenmodel (oorspronkelijk voor PAL)

- YUV codeert een luminantiesignaal Y, voor gammagecorrigeerde signalen is dit Y' (= de Y' bij witpuntcorrectie, **luma** genoemd)
- **Chrominantie:** verschil tussen een kleur en een witreferentie met dezelfde luminantie, hiervoor gebruiken we de kleurverschillen U en V: $U = B' - Y'$ en $V = R' - Y'$.

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & (-0.114 + 1) \\ (-0.299 + 1) & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

- Voor een grijze pixel, met $R'=G'=B'$, heeft Y' de waarde van die pixel omdat $0.299+0.587+0.114 = 1 \Rightarrow 0.299*R'+0.587*R'+0.114*R'=R'$ en is de chrominantie $(U,V) = 0$ (duidelijk te zien op matrix)
- In de echte implementatie worden U en V herschaald zodat ze een handiger minimum en maximum hebben.
 - o component: $\pm 0.5 * \text{maximum van } Y'$
 - o composiet: $-1/3 \rightarrow +4/3$
- 0 is niet het minimum voor U en V , er geldt:
 - o $U > 0$: blauw
 - o $U < 0$: geel (-blauw)
 - o $V > 0$: rood
 - o $V < 0$: cyaan (-rood)
- Veel bandbreedte voor Y (want veel gevoeliger aan), nl. 5.5 Mhz en weinig voor U en V (slechts 1.3 MHz elk)

YIQ kleurenmodel (gebruikt voor NTSC)

- variatie op YUV (ook chrominantie 0 bij grijze pixels), waarbij het ene kleurenkanaal (I) wel meer informatie bevat dan het andere, rekening houdend met onze visuele perceptie.
- U en V gedraaid over 33 graden:
 - o $I = V * \cos(33) - U * \sin(33)$
 - o $Q = V * \sin(33) + U * \cos(33)$
- U is ruwweg richting oranje-blauw en Q is ruwweg richting paars-groen
- Origineel ook bandbreedte in zelfde hiërarchie, (4.2, 1.5 en 0.6), nu echter beide 1 (MHz)
- $[Y', I, Q]^T = [R', G', B']^T$

YCbCr kleurenmodel (component video)

- Gebruikt bij JPEG en MPEG en sterk gerelateerd aan YUV
- Cb is U , maar met $0.5 * B'$, soms verschoven zodat de waarden tussen 0 en 1 liggen
- Cr is V , maar met $0.5 * R'$ (+0, 0.5, 0.5)
- In de praktijk 8-bit codering met Y' in $[16, 219]$
- Cb en Cr hebben een offset van 128 en een bereik van ± 112 (+16, 128, 128)

Hoofdstuk 5 : Fundamentele concepten in video

5.1 Types videosignalen

Component video

- Voor high-end videosystemen: drie **verschillende videosignalen** voor de rode, groene en blauwe beeldvlakken. Elk kleurenkanaal wordt als een afzonderlijk signaal verzonden.
- De meeste computersystemen gebruiken component video met afzonderlijke signalen voor R, G en B. Dit hoeft echter niet, ook YIQ of YUV zijn hier mogelijkheden.
- Voor elk kleurafzonderingsschema geeft component video de **beste kleurenreproductie** omdat er geen "crosstalk" mogelijk is tussen de drie kanalen (ze zijn immers fysiek van elkaar gescheiden) i.t.t. de volgende 2 types.
- Vergt echter veel **bandbreedte** en een goede **synchronisatie** van de drie componenten.

Composite video (1 signaal)

- Zowel luminantie (intensiteit), als chrominantie (kleur) worden gemengd in **één** enkele **draaggolf**. Chrominantie is hierbij de compositie van de twee kleurencomponenten: U en V of I en Q.
- Bij NTSC worden I en Q gebundeld in een chromasignaal dat door een kleuren subcarrier op de hoge frequentie van het videosignaal wordt gezet. Op de lage frequentie komt het luminantiesignaal. Ten slotte opsplitsen aan de kant van de ontvanger.
- Gebruik van enkel **één** kabel met gemengde kleurensignalen, optioneel nog sync/audio signalen over dezelfde kabel.
- Sowieso enige crosstalk tussen luminantie en chrominantie.

S (eparated) - Video (2 signalen)

- Twee draden: één voor luminantie en de andere voor een (composiet) chromasignaal => minder crosstalk tussen de kleurinformatie en de cruciale grijs-schaalinformatie.
- Luminantie krijgt hier zijn eigen signaal omdat zwart-wit informatie voor de visuele perceptie het belangrijkste is.

5.2 Analoge video

- Wordt weergegeven door de functie $f(t)$, een tijdsvariërend.
- 2 manieren van scannen:
 - o progressief: elk tijdinterval het volledige beeld overlopen, rij per rij (typ. 1/72 s)
 - o interlaced (vooral in tv): Eerst de oneven lijnen, daarna de even lijnen. Dit zorgt voor even en oneven velden die samen een frame vormen.
 - het even veld start in het midden en stopt rechtsonder, het oneven veld start linksboven en stopt in het midden.
 - de scanlijnen lopen niet horizontaal door de toepassing van een klein voltage dat de elektronenstraal naar beneden weegt over de tijd.
 - Het terugkeren van onder naar boven noemt de **verticale retrace**, het terugkeren van rechts naar links, de **horizontale retrace** (ondertussen blanking = helderheid op juiste punt brengen).
 - Zigzag tracing gebeurt niet omdat je dan een klein verschil krijgt in beweging aan de randen en een steeds groter verschil tijdens het naar links bewegen, het is veel minder storend om verspreide fouten te hebben.
 - Vermindert het (waargenomen) flikkeren wanneer het volledige frame niet snel genoeg doorgestuurd kan worden.

- Bij snelle actie kan het verschil in tijd tussen de even en de oneven scanlijnen zorgen voor een wazig effect.
- Soms is het nodig om de framerate te veranderen, te herschalen of stilstaande beelden te extraheren. Dan moeten we eerste **de-interlacen**. De gemakkelijkste manier is om gewoon één van de velden weg te laten en de scanlijnen van het andere veld te dupliceren. (Andere methodes laten wel toe data uit beide te behouden)
- CRT schermen moeten 50-70 keer per seconden flitsen om vloeiend beeld te kunnen voorzien, dit hangt vast aan de netfrequentie van 50 Hz (60 in Amerika) en een framerate van 25 fps bij digitale video (30 in Amerika, dit is de helft van de netfrequentie)
- Voltage is slechts 1-dimensionaal, hoe aanduiden wanneer een nieuwe videolijn begint? Als voor zwart een zeer kleine waarde boven nul wordt gebruikt kan voor een nieuwe lijn 0 gebruikt worden.
- Tijdens de horizontale retrace ook een sync die ongeveer 30% van het signaal gebruikt!

NTSC video

- Standaard in Noord-Amerika en Japan, gebruikt een 4:3 **aspect ratio** (= hoogte-breedte verhouding) en 525 scanlijnen @ 30 fps (om historische redenen 29,97)
- Gebruik van interlacing, dus 262,5 lijn/veld
- Horizontale veegfrequentie is $525 * 29,97 = 15,734$ lijnen/s, dus elke lijn duurt $1/15,734$ s = $63,6 \mu\text{s}$, waarvan 10,9 voor de retrace en 52,7 voor het actieve lijnsignaal.
- Aan het begin van elk veld worden 20 lijnen voorzien voor blankinginformatie => slechts 485 actieve videolijnen per frame.
- Ongeveer 1/6 van het raster (aan de linkerzijde) wordt gehouden voor horizontale retrace en sync.
- Pixels vallen vaak tussen scanlijnen in, zelfs bij het niet interlaced scannen worden slechts nog ongeveer 70% van de lijnen getoond, bij interlaced zelfs maar 50%.
- In de randen wordt geen beelddata bewaard, maar bv. wel audiokanaal data, ondertitels..
- Geen vaste horizontale resolutie => vastleggen hoeveel keer het signaal gesampled wordt voor weergave (1 sample komt overeen met 1 pixel output). De pixelklok wordt gebruikt om horizontale videolijnen te verddelen in samples, hoe hoger de frequentie, hoe meer samples per lijn. (VHS < st. 8mm < Hi-8 mm < S-VHS < betamax)
- Gebruik van het YIQ kleurenmodel: **quadratuur modulatie** nodig om het spectraal overlappende deel van I en Q signalen te combineren in een enkel C signaal

$$C = I \cos(F_{sc}t) + Q \sin(F_{sc}t)$$
- Dit gemoduleerd chromasignaal staat ook bekend als de kleuren **subcarrier** met grootte $\sqrt{I^2 + Q^2}$ en fase $\tan^{-1}(Q/I)$. De frequentie van dit signaal is $F_{sc} = 3.58$ MHz
- Het volledige composietsignaal is dan de compositie van dat chromasignaal samen met nog een luminantiesignaal Y. (van de 6 MHz: 4,2 voor Y, 1,6 voor I en 0,6 voor Q)
- Bij **decoderen** eerst Y en C afzonderen (Y zit op een lage frequentie, C op een hoge), Y kan verkregen worden door het gebruik van een **low-pass filter**.
- Daarna kan het chromasignaal gedemoduleerd worden om I en Q op te splitsen, om I te extraheren: $C * 2 \cos(F_{sc} * t) = I + I * \cos(2F_{sc} * t) + Q * \sin(2F_{sc} * t)$ => opnieuw een low-pass filter toepassen om de 2 hogere frequenties ($2F_{sc} * t$) weg te smijten en I over te houden. $C * 2 \sin(F_{sc} * t)$ voor Q.
- audioband op de hoogste frequentie (5.75), dicht bij het kleursignaal => interferentie met kleursignaal zeer goed mogelijk.

PAL video

- Standaard gebruikt in West-Europa, China, India..
- 625 scanlijnen, 25 fps, 4:3 aspect ratio en interlacing
- Er wordt gebruikgemaakt van de YUV kleurenruimte en 8 MHz bandbreedte, 5,5 voor Y en 1.8 voor U en V afzonderlijk. De frequentie van de subcarrier (f_{sc}) is ongeveer 4,43 MHz

- Om de beeldkwaliteit te verbeteren hebben chroma signalen een tegengesteld teken in opeenvolgende scanlijnen. (=> Phase Alternating Line)
- Maakt het gemakkelijk om aan de kant van de ontvanger een comb filter (een uitgestelde versie van het signaal bij zichzelf toevoegen) te gebruiken. Het gemiddelde van opeenvolgende lijnen wordt genomen zodat het chromasignaal zichzelf gaat verwijderen en enkel het Y-signaal overblijft.

SECAM video

- 625 scanlijnen, 25 fps, 4:3 aspect ratio en interlacing
- Enkel kleurenschema verschilt met PAL: elke scanlijn ofwel U, ofwel V, niet tegelijk.
- Ook 8 MHz, maar 6 voor Y en 2 voor zowel U, als V.

5.3 Digitale video

- Voordelen:
 - o Kan **bewaard** worden op digitale toestellen of in geheugen, klaar om **bewerkt** (ruis verwijderen, knippen en plakken) te worden en geïntegreerd te worden in multimedia-applicaties.
 - o **Onmiddellijke toegang** mogelijk, niet-lineaire video bewerken wordt gemakkelijk.
 - o **Herhaald opnemen** doet de beeldkwaliteit niet verminderen
 - o Gemakkelijke **encryptie** en betere **tolerantie** tegenover **ruis** op het kanaal.
- Bij moderne digitale video, meestal **component video**, in combinatie met YCbCr.

Chroma subsampling

- Mensen zien kleur met een veel lagere spatiale resolutie (lijnen per inch) dan zwart/wit => bandbreedte besparen door chromasignaal te decimeren.
- Verschillende schema's:
 - o 4:4:4, geen chroma subsampling toegepast, voor elke pixel zowel Y, als Cb, als Cr.
 - o 4:2:2: horizontale subsampling van Cb en Cr met een factor 2, elke tweede Cb en elke tweede Cr wordt verzonden, maar alle Y's. bv. Cb₀,Y₀ - Cr₀,Y₁ - Cb₂,Y₂ - Cr₂,Y₃ -...
 - o 4:1:1: horizontale subsampling, maar nu met een factor 4
 - o 4:2:0: horizontale en verticale subsampling met een factor 2, hier liggen de chromapixels theoretisch gezien tussen de rijen en kolommen. Gebruikt bij MPEG en JPEG.

CCIR standaarden

- CCIR-601 voor component video => is nu ITU-R-601, een internationale standaard voor professionele videotoeepassingen (wordt bijvoorbeeld toegepast bij DV video. Gebruikt ook een interlaced scan.
- CIF (Common Intermediate Format): een formaat voor lagere bitrates (ongeveer VHS kwaliteit), zonder interlacing (progressieve scan)
- QCIF (Quarter CIF): letterlijk een kwart van de grootte van CIF.
- Alle resoluties in (Q)CIF deelbaar door 8 en (behalve 88) ook deelbaar door 16 => gemakkelijk voor blokgebaseerde videocodering.
- CIF is een compromis tussen PAL (helpt van zijn aantal actieve lijnen) en NTSC (framerate)
- NTSC/PAL: 720x480 luminantie (helpt voor chrominantie, horizontaal), 4:2:2 kleurensampling, respectievelijk 60 en 50 velden per seconde en beide interlaced.
- CIF 352x288, QCIF en chrominantie de helft (per dimensie, dus eigenlijk totale grootte / 4), 4:2:0 subsampling en 30 velden per seconde (want progressieve scan).

HDTV

- Niet hoofdzakelijk een verhoging in precisie, maar eerder een vergroting van het kijkveld, voornamelijk in de breedte.
- Eerste generatie in de late 70's, daarna een verbetering (hybride analoog/digitale technologie) in de 90's. 1125 scanlijnen, interlaced (60 velden per seconde) en 16:9 aspect ratio.
- Compressie nodig, want anders meer dan 20 MHz bandbreedte nodig, eventueel over meerdere kanalen na compressie.
- SDTV (standaard definitie): NTSC of hoger
- EDTV (enhanced) :
 - 704x480 (704 actieve pixels, 480 actieve lijnen), 16:9/4:3, 60 interlaced of 60-30-24 (fps) progressief.
 - 640x480, 4:3, 60 interlaced of 60-30-24 (fps) progressief
- HDTV (hoge):
 - 1920x1080, 16:9, 60 interlaced of 30-24 progressief
 - 1280x720, 16:9, 60 interlaced, 30-24 progressief
- Ook aantal velden per seconde beperkt tot 60 omdat niet alle TV's meer aankunnen.
- MPEG-2 compressie voor video, AC-3 voor audio en ondersteuning voor 5.1 Dolby surround.
- Belangrijkste verschillen:
 - 4:3 -> 16:9 **aspect ratio** (breder!)
 - interlaced -> progressief (omdat interlacing zorgt voor gezaagde randen bij bewegende objecten en zorgt voor flikkering langs horizontale randen).
- Nadelen:
 - blokartefacten bij zeer grote resoluties
 - minder signaalsterkte per pixel, maar zelfde ruis => verhouding wordt slechter (de ruis van het scherm zelf wordt wel kleiner, maar transmissieruis blijft dezelfde) Die ruis kan beperkt worden door filtering, maar aangezien randen ook hoge frequenties hebben worden die minder scherp door de filtering.

Hoofdstuk 6 : Basis van digitale audio

6.1 Digitalisatie van geluid

Wat is geluid?

- een fenomeen met golven, net zoals licht, maar macroscopisch, met luchtmoleculen die samengedrukt en terug vergroot worden door het toedoen van een fysisch toestel.
bv. een speaker vibreert voorwaarts en achterwaarts waardoor er een **longitudinale** drukgolf ontstaat die we als geluid ervaren.
- Omdat geluid een drukgolf is neemt het continue waarden aan, i.p.v. gedigitaliseerde.
- Heeft ook de gewone golfeigenschappen en het gewone golfgedrag (alhoewel het een longitudinale golf is) zoals **reflectie** (afketsen), **refractie** (verandering in hoek bij het binnenkomen in een medium met een andere dichtheid) en **diffractie** (rond een obstakel buigen) => maakt het ontwerp van **surround** geluid mogelijk.
- Op elk 3D punt kan de druk gemeten worden en daarna met een **transducer** omgezet worden in voltagelevels.

Digitalisatie

- Een analogo signaal is continu, daarom moeten we het eerst omzetten in een reeks van (liefst gehele, voor de efficiëntie) getallen.
- Terwijl beelden afhangen van zowel x als y in het 2D-vlak is amplitude enkel afhankelijk van de tijd en dus 1D.
- Aangezien de grafiek wel 2D is, moeten we het signaal samplen in elke dimensie.
- **Samplen** is het meten van de kwantiteit waarin we geïnteresseerd zijn (meestal gelijk opgesplitste intervallen).
 - o Wanneer enkel gelijk verdeelde intervallen gebruikt worden heet het **sampling**. De snelheid waarmee dit gebeurt noemen we de **sampling frequency**. (typisch tussen 8 kHz en 48 kHz -> zie nyquist theorema)
 - o Sampling van de amplitude of het voltage noemen we **quantisatie**. Ook niet uniforme quantisatie rates mogelijk. Typische uniforme rates zijn 8 en 16 bit (bij 8 bit opdelen in 256 gelijke intervallen)

Nyquist theorema

- Signalen kunnen ontbonden worden in een som van sinusfuncties
- frequentie is een absolute maatstaf, toonhoogte (pitch) is relatief, een perceptuele, subjectieve kijk op geluidskwaliteit. Deze worden gelinkt door de noot A boven middel C (stemtoon van orkesten) op exact 440 Hz te zetten.
 - o Een **octaaf** erboven is er opnieuw een A noot. Een octaaf komt overeen met het verdubbelen van de frequentie. Als de middelste A (A4 of A440) op 440 Hz staat, dan staat de volgende op 880 Hz.
 - o **Harmonieën**: Een serie muzikale tonen met frequenties die gehele **vermenigvuldigingen** zijn van de frequentie van een **fundamentele** toon.
 - o Als we ook niet gehele vermenigvuldigingen toelaten -> meer complexe geluiden
- **Nyquist theorema** stelt hoe frequent we moeten samplen in **tijd** om het originele geluid te reproduceren.
 - o Als de samplingrate = echte frequentie, dan krijgen we een constante met 0 frequentie.
 - o Samplen op 1.5x echte frequentie dan krijgen we een incorrecte frequentie (**alias**) die lager is dan de echte. (helft van de echte, afstand tussen 2 pieken is dubbel) => bij graphics veel wordt veel moeite gedaan om aliassen te voorkomen door anti-aliasingmethodes (elk artefact dat niet behoort tot het originele signaal is een alias)

- **theorema:** Als een signaal **bandgelimiteerd** is (dus bovengrens en ondergrens) heeft, $f_1 = \text{onder}$ en $f_2 = \text{boven}$, dan is er een samplingrate van ten minste $2 \cdot (f_2 - f_1)$ nodig.
- Stel dat we een **vaste samplerate** hebben, dan kunnen we het signaal niet langer reproduceren wanneer de frequentie hoger is dan de helft ervan. Meestal antialiasing filter om ervoor te zorgen dat de frequentie op of onder de helft is.
- De frequentie gelijk aan de helft van de **nyquist rate** noemen we de **nyquist frequentie**.
- Hoogst mogelijke signaalfrequentie component heeft dezelfde frequentie als de sampling zelf.
- Als de samplingfrequentie tussen de ware en $2x$ de ware frequentie ligt dan geldt dat de aliasfrequentie gelijk is aan de samplingfrequentie - de ware frequentie.
- Bij het doorsturen van een te hoge samplerate => foutrobuuster en minder ruis door uitmiddeling, maar meer data.
- Te hoge frequenties worden teruggeklapt op het signaal zelf.

Signaal-ruis verhouding (SNR)

- De verhouding van het correcte signaal en de ruis, meet de kwaliteit van een signaal.
- Wordt uitgedrukt in dB (tiende van een bel)
- $SNR = 10 \cdot \log_{10}(V^2(\text{signaal}) / V^2(\text{ruis})) = 20 \cdot \log_{10}(V(\text{signaal})/V(\text{ruis})) \rightarrow \text{volt}^2 = \text{Watt (vermogen)}$
- Kracht van een signaal is proportioneel met het kwadraat van het voltage. Als het signaal 10 keer de ruis is, dan is de SNR 20 dB (kwadraat naar voor gebracht)
- dB is een eenheid voor krachtverhoudingen: Als we de kracht van tien vloten tegenover 1 meten dan is de krachtverhouding 10 dB.
- **relatieve eenheid**, de 0 moet eerst vastgelegd worden.

Signaal-quantisatie-ruis verhouding (SQNR)

- enkel gequantiseerde waarden worden opgeslaan bij digitale signalen, de precisie van elke sample wordt bepaald door het aantal bits/sample (typisch 8 of 16, dit komt overeen met een verdeling in 256 of 16 miljoen samples).
- Naast de ruis reeds aanwezig in het originele, analoge signaal wordt hier bijkomende ruis aan toegevoegd door de quantisatie: **quantisatie-ruis**, dit zijn de afrondingsfouten veroorzaakt door een beperkt aantal samples - bv. 256 (al is dit niet echt ruis, maar komt ook willekeurig voor van sample tot sample)
= waarde van het analoge signaal op een bepaald tijdstip - dichtstbijzijnde waarde van een quantisatieinterval. (max. helft van interval)
- Voor een quantisatieprecisie van N bits per sample, signaal tussen $-2^{(N-1)}$ en $+2^{(N-1)}$. Als het analoge signaal tussen $-V_{\text{max}}$ en $+V_{\text{max}}$ ligt dan representeert elk quantisatielevel een voltage van $2 \cdot V_{\text{max}} / 2^N = V_{\text{max}} / 2^{(N-1)}$.
- **Dynamisch bereik** is de verhouding van maximum tot minimum van de absolute waarde van het signaal. V_{max} wordt gemapt op $2^{(N-1)}$, $-V_{\text{max}}$ op $-2^{(N-1)}$ en V_{min} op 1 (kleinste positieve signaal, niet verborgen door de ruis).
- **Peak signal-to-noise ratio** (voor max signaal en max ruis): $PSQNR = 20 \cdot \log_{10}(V(\text{signaal})/V(\text{quantisatie-ruis})) = 20 \cdot \log_{10}(2^{(N-1)} / (1/2)) = 20 \cdot N \cdot \log(2) = 6,02 \cdot N \text{ dB}$
=> elke bit voegt ongeveer 6 dB resolutie toe.
- 8 bits komt ongeveer overeen met telefoonkwaliteit, 16 bits met cd-kwaliteit

Lineaire en niet lineaire quantisatie

- **Lineair formaat:** samples worden typisch uniform gequantiseerd bewaard. Maar door het beperkt aantal bits, misschien beter om rekening te houden met de eigenschappen van menselijke waarneming.
- **Niet-uniforme quantisatie:** kiest meer nauwkeurigere levels waar mensen het scherpst horen. We quantiseren hoe luid een signaal is, zijn grootte, amplitude.

- **Weber's wet:** hoe meer er is, hoe meer proportioneel gezien toegevoegd moet worden om een verschil te maken.
- De factor waarmee toegevoegd moet worden aan een bepaalde stimulus om dezelfde respons te krijgen blijft dezelfde, deze factor noemen we k : **$dr = k * (1/s) ds$**
Hierbij is dr de verandering in respons, k de factor, s de huidige stimulus en ds de verandering in stimulus (als we deze wet opleggen wordt quantisatie zo geoptimaliseerd dat gelijk waargenomen verschillen leiden tot gelijke verschillen in respons) =>
(integratie) $r = k*\ln(s) + C$ of bij uitrekenen van $r = 0$ krijgen we **$r = k*\ln(s/s_0)$** met s_0 het laagste stimulusniveau waarop nog respons is ($r = 0$ als $s = s_0$)
- Bij niet-lineaire quantisatie willen we uniforme stappen langs de r -as krijgen, daarvoor moeten we eerst het analoge signaal van de s -ruimte naar de theoretische r -ruimte brengen en dan de **resulterende waarden uniform quantiseren**. Gevolg is dat op de s -as de quantisatiestappen bij lagere waarden meer geconcentreerd zijn dan bij hogere waarden (want minder gevoelig aan luide geluiden). Bij hoge waarden dient één quantisatiestap in r dan voor veel stimuluswaarden.
- Een wet die voor deze codering zorgt bij audio noemen we de μ -wet (gewoonlijk $\mu=100$ of 255), een gelijkaardige regel is de A -wet (voor Europese telefonie, gewoonlijk $A = 87,6$)
- Om s tussen -1 en 1 te laten variëren: delen door die piekstimulus s_p
- **companding:** 1) μ -wet of A -wet transformatie toepassen, 2) de resulterende waarden **uniform** quantiseren (non-lineaire transformatie tegenover de input) => meer compressie bij hogere amplitudes. 3) na transmissie terug naar analogo geconverteerd en door een **expander** circuit gestuurd om de logaritmische transformatie ongedaan te maken.
- Bij een vast aantal bits voor het encoderen van signalen zal er relatief gezien minder ruis zitten om de kleinere signalen (betere SNR).
- We horen ook het meest stillere geluiden, dus ook in die zin zijn er meer bits voorzien, daar waar het meeste geluid voorkomt.

Audio filtering

- Voor conversie van analogo naar digitaal en voor het sampelen wordt een audiosignaal meestal eerst **gefilterd** op **ongewenste frequenties**. Deze frequenties hangen af van toepassing tot toepassing.
 - o Voor spraak tussen de 50 Hz en de 10 kHz houden, de rest blokkeren met een **band-pass filter** (filtret hoge en lage frequenties weg)
 - o Muzieksignalen typisch tussen 20 Hz en 20 kHz
- Bij terug converteren van digitaal naar analogo kunnen er opnieuw hoge frequenties verschijnen in de output door sampling en quantisatie (in theorie bevatten deze discontinue signalen alle mogelijke frequenties), vandaar het toepassen van een **lowpass** filter aan de DA kant.

Kwaliteit v.s. datarate

- Enkel gebruik van niet-lineaire quantisatie bij telefonie (A -wet in Europa)
- dubbel zoveel bandbreedte bij stereo
- sample rate van 22,05 KHz => 22050 samples/s en bv. 16 bits/sample => $16*22050$ bps
- resulterende **datarate** nog x2 voor stereo: $16*22050*2 = 705600$ bps

6.2 MIDI: Muziek Instrument Digitale Interface

- simpele scriptingtaal en hardware setup (defaultgeluiden van de geluidskaart worden gebruikt)
- Codering van gebeurtenissen die staan voor de productie van geluiden (inclusief bv. toonhoogte, duur en volume van een bepaalde noot)
- Standaard in de muziekindustrie om toestellen zoals synthesizers en geluidskarten aan te sturen die muziek produceren.
- Ondersteund door de meeste synthesizers, dus geluiden zijn uitwisselbaar.
- MIDI-interface meestal ondersteund door de geluidskaart (moet ook AD en DA converters)

- MIDI kanalen gebruikt voor afzonderlijke berichtgeving: 16 kanalen die de laatste vier bits van een bericht vormen, normaalgezien wordt aan elk kanaal een specifiek instrument gelinkt. Switchen is echter mogelijk.
- Systeemberichten voor algemene informatie (verandering in tuning of timing), als de eerste vier bits op 1 staan wordt het bericht als een system common bericht gezien.
- Bij een "speel geluid"-bericht niet voor zijn kanaal negeert het synthetisch instrument het bericht gewoon, bij meerdere berichten tegelijk => meer dan 1 noot tegelijk spelen (als het meerstemmig is)
- timbre vs stem: timbre is MIDI-terminologie voor het te emuleren instrument (de kwaliteit van het geluid), een multi-timbre instrument of geluidskaart is er een die veel instrumenten tegelijk kan simuleren.
Stem is elke soort timbre en toonhoogte dat de toonmodule tegelijk kan produceren.
- Patches zijn sets controleinstellingen die een bepaald timbre specificeren, deze worden georganiseerd in banks (databanken). Mapping om kanalen te verbinden met patches (instrumenten).
- Midibytes bevatten een start en stop bit (dus in totaal 10 bytes).

6.3 Quantisatie en transmissie van audio

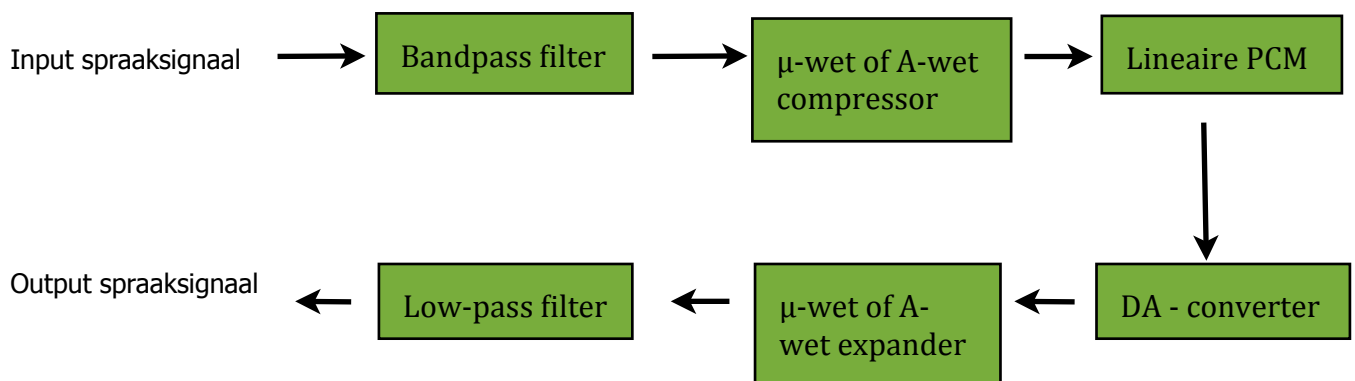
Coderen van audio

- Coderen van data = quantisatie + transformatie van de data
 - o Voor audio wordt de μ -wet vaak gecombineerd met een algoritme om temporele redundancy te gebruiken.
 - o Verschillen in signalen tussen huidige en volgende signalen: kleinere signaalwaarden en histogram van pixelwaarden in een veel kleiner bereik.
 - o Resultaat van de variantie tussen waarden te verminderen is dat de bitstream korter is voor meer waarschijnlijke waarden.
 - o **PCM** (pulse code modulation): gequantiseerde, gesampelde output produceren
 - o **DPCM** (differential PCM): versie met verschillen
 - o **DM** (delta modulation): ruwe, maar efficiënte variant van DPCM
 - o **ADPCM** (adaptive DPCM): veranderlijke (adaptieve) versie
- **Coderen** is altijd lossless t.o.v. **comprimeren** (lossy): coderen is het weglaten van redundante gegevens terwijl je in staat blijft het origineel perfect te reconstrueren, bij comprimeren worden niet/slecht waarneembare zaken weggelaten.

Pulse Code Modulation

- Basistechniek om digitale signalen te krijgen uit analoge zijn **sampling** en **quantisatie**.
- **Quantisatie**: selecteren van breekpunten in de amplitude (de grootte) en dan elke waarde binnen het interval mappen op de representatieve outputlevels.
 - o set van intervalgrenzen: **decision boundaries** en de representatieve waarden: **reconstruction levels**.
 - o De grenzen voor inputintervallen van de quantizer worden allemaal gemapt op een zelfde outputlevel form, de **coder mapping**.
 - o De representatieve waarden, de output van de quantizer, zijn een **decoder mapping**.
 - o Eventueel nog toepassen van **compressie** zodat minder bits worden gebruikt voor de meest verspreide bitwaarden.
- **Compressieschema**:
 - o **Transformatie** van de inputdata naar een nieuwe representatie die gemakkelijker of meer efficiënt te comprimeren is.
 - o Eventueel **verlies** van informatie introduceren, **quantisatie** is de hoofdstap bij lossy.
 - o **Codering** door een codewoord aan elk outputlevel toe te kennen, dit kan een vaste lengte hebben of een variabele lengte => vorming van een binaire bitstream
- Voor audio signalen beschouwen we PCM als digitalisatiemethode:

- differentiële codering: Lossless predictive coding en DPCM
- betere compressie: ADPCM
- Bij **spraakherkenning**: bandbreedte voor spraak tussen 50 Hz tot 10 kHz => volgens Nyquist theorema samplen op 20 kHz.
 - Zonder companding minimum sample size ongeveer 12 bits (tussen telefonie en FM-radio)
 - Voor mono-spraak krijgen we dan een bitrate van 240 kbps (= 12*20)
 - Met companding minimum sample size ongeveer 8 bits en een bitrate van 160 kbps (=8*20)
 - Standaardbenadering van telefonie veronderstelt een maximale frequentie van slechts 4 kHz, dus samplen op 8 kHz en een bitrate van 64 kbps (=8*8)
- Omdat enkel geluid tot 4 kHz beschouwd wordt, is al het overige geluid ruis, hoge frequenties verwijderen van het analoge signaal met een **band-limiting/bandpass** filter (blokkeert de hoge en de lage frequenties)
- Naar digitaal omzetten -> trapfunctie verkrijgen, bevat ook scherpe hoeken, waardoor er frequentiecomponenten in zitten die niet behoren tot het originele signaal (je krijgt een theoretisch oneindige set hoge-frequentie componenten door Fourier analyse), deze zijn irrelevant. Daarom moet de output van DA converter nog door een low-passfilter passeren (enkel waarden tot het originele maximum blijven behouden)



Differentiële codering van audio

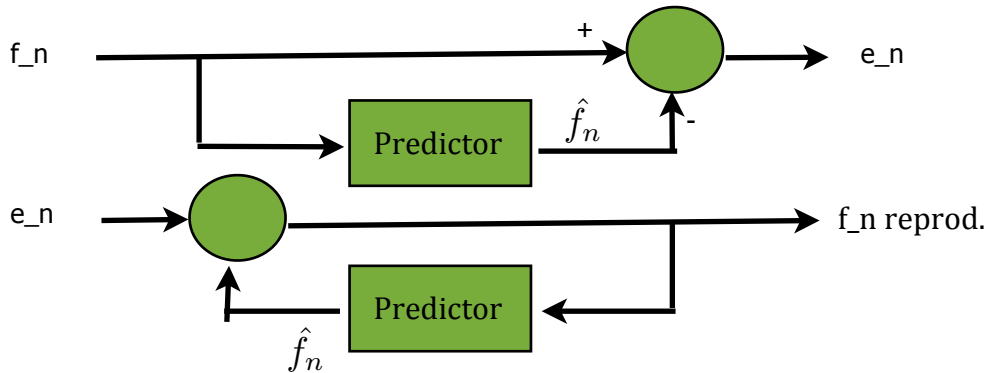
- Audio wordt vaak niet opgeslagen in eenvoudige PCM, maar in een vorm die verschillen exploiteert (over het algemeen kleinere getallen waardoor er minder bits bewaard moeten worden)
- Als een tijdsafhankelijk signaal enige consistentie heeft over de tijd (temporele redundancy), dan zal het "verschilsignaal" (trekt het huidige signaal af van het vorige) een meer gepiekt histogram hebben met een maximum rond 0.
- Gebruik van korte codewoorden voor veel voorkomende waarden en lange codewoorden voor zeldzame waarden.

Lossless Predictive Coding

- Predictive coding wil zeggen dat de verschillen worden doorgezonden (volgende sample voorspellen als hetzelfde als de huidige sample) -> verschillen vinden en deze verzenden, gebruik makend van een PCM systeem.
- De verschillen van gehele getallen zijn opnieuw gehele getallen, noteer het inputsignaal als set van waarden \hat{f}_n , dan geldt: $\hat{f}_n = f_{n-1}$ en voor de fout: $e_n = f_n - \hat{f}_n$.
- Soms meerdere vorige waarden samen voor een betere voorspelling, typisch gebruik van een lineaire voorspeller:

$$\hat{f}_n = \sum_{k=1}^{2 \text{ to } 4} a_{n-k} f_{n-k}$$

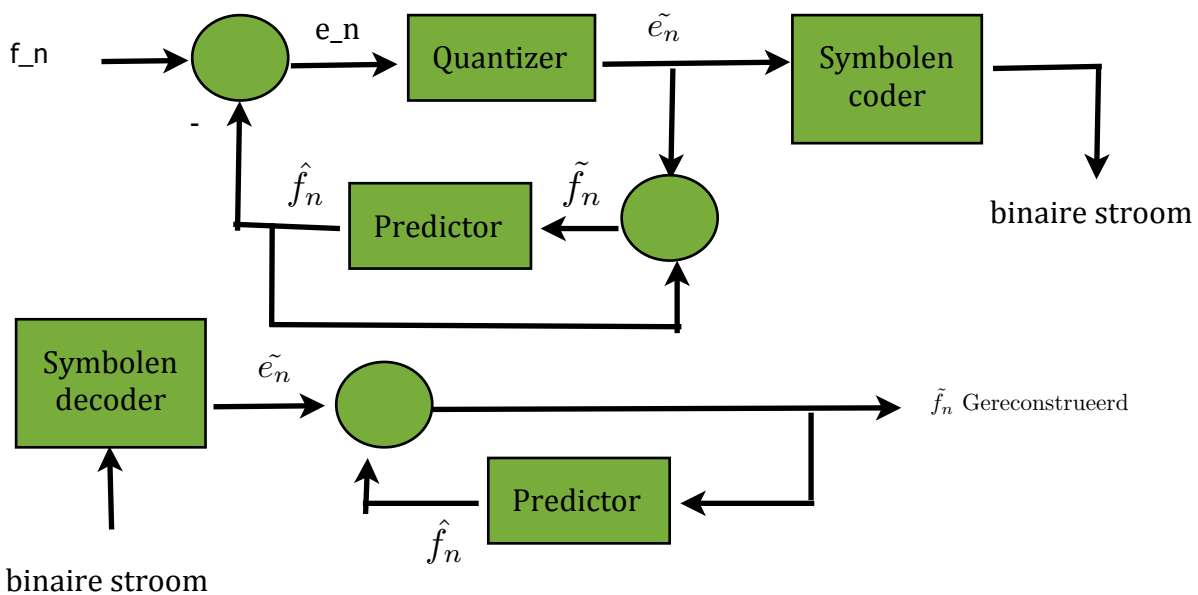
- Als de gehele waarden tussen 0 en 255 lagen, dan kunnen de verschillen tussen -255 en 255 liggen waardoor het **dynamisch bereik** (verhouding maximum tot minimum) verdubbeld => meer bits nodig om verschillen door te sturen
 - o Oplossing: twee nieuwe codes definiëren - SU en SD (shift-up, shift-down)
Deze codewoorden kunnen we dan gebruiken voor een gelimiteerde set signaalverschillen (bv. -15..16)
 - o Signalen binnen dit bereik gewoon doorsturen, anders een reeks van codes bestaande uit SU en SD (SU is dan +32 en SD -32 => 100 = SU,SU,SU,4)
- f_n kan bv. door gemiddelde vorige waarden berekend worden (afroonden naar beneden)



- Eigenlijk errorwaarden doorgestuurd

DPCM (Differential Pulse Code Modulation)

- Zelfde als Lossless Predictive Coding, maar dan met een kwantisatiestap (dus zonder de lossless)
 - o Lloyd-Max quantizer is een schema om de beste quantizer stappen voor een niet-uniforme quantizer te vinden.
 - o \tilde{f}_n is hier gequantiseerd, gereconstrueerd signaal
 - o Vorm de voorspelling -> vorm een fout e_n door de voorspelling af te trekken van het signaal -> quantiseer de fout tot \tilde{e}_n -> verzend het codewoord voor die gequantiseerde fout -> reconstrueer een gequantiseerd signaal door de voorspelling op te tellen bij de gequantiseerde fout.



- Predictor altijd gebaseerd op de gereconstrueerde, gequantiseerde versie van het signaal. Dit is zo omdat de encoder niets mag gebruiken dat niet ook aanwezig is aan de zijde van de decoder. Stel dat we het echte signaal f_n gebruiken, dan kan de quantisatiefout zich gaan opstapelen in plaats van zich te centreren rond 0.
- Voorspelde waarde zelfde aan decoderzijde op voorwaarde dat dezelfde voorspeller gebruikt wordt.

DM (Delta Modulatie)

- Sterk vereenvoudigde versie van DPCM, vaak gebruikt als snelle AD-converter.
- **Uniforme** delta modulatie: een enkele gequantiseerde foutwaarde, ofwel positief, ofwel negatief.
 - o Een 1-bit coder, produceert output dat het originele signaal volgt in een trapfunctie:
 - $\hat{f}_n = \tilde{f}_{n-1}$ (pure vertrageningsfunctie)
 - $e_n = f_n - \hat{f}_n = f_n - \tilde{f}_{n-1}$
 - $\tilde{e}_n = k$ als $e_n > 0$, anders $-k$
 - $\tilde{f}_n = \hat{f}_n + \tilde{e}_n$
 - o Snelle veranderingen kunnen opgevangen worden door de samplingrate veel te verhogen.
- **Adaptieve DM:** k kan ook aanpasbaar gemaakt worden indien er bijvoorbeeld met stijle curves gewerkt wordt.