

H2

Video Transitions

- overgangen.

- Typen:

* cuts: abrupte overgang

* wipes: deel v.d. pixels worden abrupt vervangen in functie van de tijd. (LTR, RTL, TTB, BTB)

o dissolves:

* cross dissolve: elke pixel wordt geleidelijk veranderd

$$D = (1 - \alpha(t)) \cdot A + \alpha(t) \cdot B$$

met $\alpha(t)$: transitie familie

$$\alpha(t) = k(t), \quad k(t_{\max}) = 1$$

A en B, de kleurvectoren v. respectievelijk

de eerste en de tweede video.

* dither dissolve: bepaald door $\alpha(t)$, veranderen er ~~abrupt~~ meer en meer pixels abrupt van video 1 naar video 2.

Video Formats

- NTSC	30 fps	USA
- PAL	25 fps	EUROPA
- SECAM	25 fps halve netspanning	FRANKRIJK

VRML

- Virtual Reality Modeling Language
- interpreted door browser
- VRML Shaper:

VRML bestaat uit nodes in een hiërarchie die een scene met een of meerdere objecten beschrijven.

VRML heeft basis geometrische vormen, die men kan combineren tot meer complexe objecten. (box, cylinder, cone en sphere)

- VRML World

- licht:
- directional: licht in een bepaalde richting over heel de wereld
 - point: licht in alle richtingen vanuit een punt
 - spot: licht vanuit een bepaald punt in een bepaalde richting

H3

Image Data Types.

1-Bit Images.

elke pixel opgeslagen als 1 bit. "on" en "off" bits
1-bit monochroom: bevat 1 kleur.

8-Bit Gray-Level Images.

elke pixel opgeslagen als 1 byte $[0, 255]$

monochroom: bevat 1 kleur, niet noodzakelijk wit.

bitmap: twee dimensionale array van pixel waarden.

resolutie: aantal pixels in de afbeelding. bv. 1600×1200 (4:3 aspect ratio)

set v 8 1-bit bitplanes

bij punten wordt de techniek van dithering gebruikt.

dithering: intensiteits resolutie verminderen voor spatiale resolutie, pixel waarden vervangen door een groter patroon. $m \times m$ dither matrix kan $m^2 + 1$ niveaus van intensiteit weergeven. (m x m dither matrix)

we moeten pixelwaarden $[0, 255]$ hermappen naar $[0, m^2]$.

dit doen we door de pixelwaarde te delen door $(256/m^2 + 1)$ en af te kappen na de komma. Als deze waarde groter is dan de dither matrix entry, dan printen we daar en dot.

De grootte v.e. dithered image is groot, als oplossing kan men opteren voor ordered dithering, waarbij de dither matrix als het ware over de afbeelding wordt geshoven en de dot geprint wordt als de pixelwaarde op die plaats (herleid naar $[0, m^2]$) groter is als de waarde in de dither waarde.

24-bit Color Images

elke pixel voorgesteld door 3 bytes, die RGB voorstellen.

ondersteunt $256 \times 256 \times 256$ kleuren.

meestal opgeslagen als 32 bit afbeelding, met een extra byte aan data voor elke pixel.

8-bit Color Images

slechts 256 kleuren, opgeslagen in een color lookup table.

een pixelwaarde is geen kleur op zich, maar wordt gebruikt als index in de color lookup table, die dan 3 bytes heeft (RGB).

slechts 256 kleuren, dus belangrijk om te kiezen welke kleuren we opnemen in de LUT.

naïve methode: RGB-cube verdelen in gelijke stukken in elke dimensie. De centers v.d. resulterende kleine cubes dienen als entrees in de color LUT, terwijl hersehelen v.d. RGB ranges $[0, 255]$ naar de geschikte intervallen resulterend in 8 bit codes. Mensen zijn meer gevoelig voor R en G dan voor B, we ~~kan~~ herschalen R en G dus naar $[0, 7]$, met 3 bits en B naar $[0, 2]$, met 2 bits, om zo 8 bit codes te bekomen.

nadeel: "edge artifacts"

median-cut: adaptief partitionerings schema, dat probeert om het meest bits te spenderen, door naar de kleuren geclusterd zijn.

(1) ~~split~~ sorteer de R bytes waarden, ~~sort~~ ook median, splits op 1

(2) voor de 2 gemiddelde sets v. RGB waarden:

sorteer op G byte waarden en splits op median 011

(3) voor de 4 gemiddelde sets v. RGB waarden:

sorteer op B byte waarden en splits op median 011

(4) herhaal (1), (2), (3) (1), (2) om zo 256 sets te bekomen.

Dus voor R en G gebruiken we 3 bits, voor B 2, zodat een 8 bit kleur index bekomen naar de pixels, de corresponderende 24-bit kleuren zijn de centers v.d. resulterende kleine kleur cubes.

Popular File Formats

GIF : Graphics Interchange Format

8-bit color images

interlacing: 4-pass display process

pass 1: ry $0+k \cdot 8$

pass 2: ry $4+k \cdot 8$

pass 3: ry $2+k \cdot 4$

pass 4: ry $1+k \cdot 2$

file format :

GIF signature (6 bytes)

Screen descriptor (7 bytes)

Global color map

⋮

Image descriptor

Local color map

Raster area

⋮

GIF terminator

} repeated 1 to n times
→ LZW compression

screen descriptor

image descriptor

JPEG Joint Photographic Experts Group

- Lossy!
- buit beperkingen v.h menselijk zicht uit.
mensen zijn minder gevoelig voor hoge spatiale frequenties.
(= veel verandering v pixel waarden binnen kleine pixeloppervlakte)
dit geldt zeer sterk voor kleur informatie.
- bevat dus gedecimeerde kleur informatie

PNG Portable Network Graphics

omvateunt tot 48 bits aan kleur informatie
bevat gamma-correction informatie en alpha-channel informatie
interlacing: Adam7 : 7-pass op een 2 dimensionale manier.

TIFF Tagged Image File Format

lijktomende informatie in de vorm van tags
origineel lossless, maar nu ook met JPEG tags, zodat
voor lossy JPEG compression kan worden gekozen.

EXIF Exchange Image File

voor digitale cameras, bevat dan ook veel informatie
over de camera en de condities waaronder de foto werd
genomen.

H4 Color in Image and Video

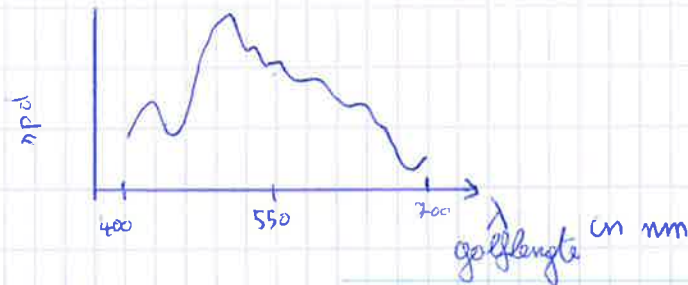
Light and Spectro

licht: elektromagnetische golf, kleur gekarakteriseerd door golflengte λ
meten met spectrophotometer

korte golflengte: blauwachtig | langgolflengte: roodachtig
wit licht bevat alle kleuren v.d. regenboog.

zichtbaar licht 400-700 nm

SPD: Spectral Power Distribution of spectrum toont de
relatieve hoeveelheid lichtenergie op elke golflengte. $E(\lambda)$



Human Vision

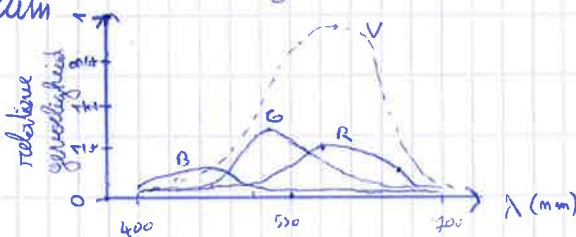
licht valt op het netvlies, dat bestaat uit staafjes en kegeltjes.

Stafjes vormen bij weinig licht een beeld uit grijstinten. Er zijn 3 soorten kegeltjes, gevoelig voor R, G en B, die bij meer licht in werking treden.

Spectral Sensitivity of the Eye

Het oog is het meest gevoelig voor licht in het midden v.b.

zichtbaar spectrum



$V(\lambda)$: totale gevoeligheid
luminous-efficiency function

oog: 6 miljoen kegeltjes R:G:B · 40:20:1

spectral sensitivity functions:

$$q(\lambda) = [q_R(\lambda), q_G(\lambda), q_B(\lambda)]^T$$

Het antwoord in elk kleurkanaal v.h. oog is proportioneel met het aantal afgeworde neuronen. Al het licht dat valt in het niet nul gedeelte v.d. kegelfunctie zal bijdragen tot het antwoord.

Het totale antwoord is dus de som over al het licht dat op het netvlies valt, waarvoor de kegel gevoelig is, gemogen t.o.v. de gevoeligheid voor die golflengte.

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

$$G = \int E(\lambda) q_G(\lambda) d\lambda$$

$$B = \int E(\lambda) q_B(\lambda) d\lambda$$

Image Formation

bevoerenstaande vergelijkingen gelden enkel bij het kijken naar een lichtbron. Meestal kijken we echter naar licht dat door een oppervlak werd gereflecteerd. Verschillende oppervlakken reflecteren verschillende hoeveelheden licht of verschillende golflengtes. $S(\lambda)$: reflectie functie

Beeldvorming: = licht v. bron met SPD $E(\lambda)$

- gereflecteerd door oppervlak met spectraal reflectie functie $S(\lambda)$
- gefilterd door de kegeltjes in ons oog met kegel functie $q(\lambda)$

Het kleursignaal $C(\lambda)$ is het product v.d. lichtbron en de reflectie.

$$C(\lambda) = E(\lambda) \cdot S(\lambda)$$

spectral power distribution
 $E(\lambda)$



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

$$G = \int E(\lambda) \cdot S(\lambda) q_G(\lambda) d\lambda$$

$$B = \int E(\lambda) \cdot S(\lambda) q_B(\lambda) d\lambda$$

$S(\lambda)$

$C(\lambda)$
color signal



$q(\lambda)$
cone functions

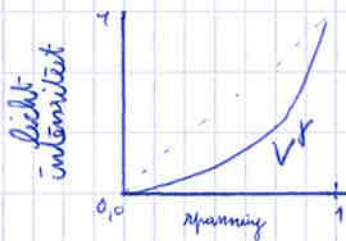
surface spectral reflection function

Gamma Correction

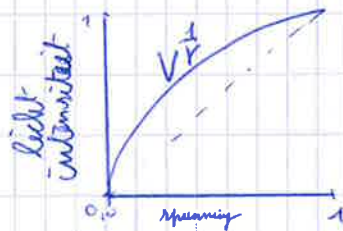
De RGB-waarden v. elke pixel worden omgeset naar een analoge spanning in een CRT-scherm om de elektronenkanonnen aan te sturen. Deze spanning bepaald de lichtintensiteit, jammergenoeg niet met een lineair verband. De lichtintensiteit is licht proportioneel met de spanning tot een macht. Deze macht noemen we "gamma" met symbool γ . Het beeld dat we te zien krijgen op het CRT-scherm, wijkt dus af v.h. originele beeld. Kleurinten worden namelijk te donker weergegeven.

We lossen dit probleem op door het signaal te geven te corrigeren voor we het affineren.

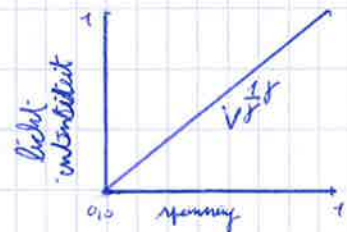
Signaal gegenereerd door CRT: V^γ , met V het voltage en γ de gamma fact. We kunnen dit signaal corrigeren door $V^{\frac{1}{\gamma}}$ aan het CRT te geven. Deze zal licht uitstralen proportioneel met $V^{\frac{1}{\gamma} \cdot \gamma}$, ofwel V . Aldus een lineair verband.



geen gamma correctie



gamma correctie

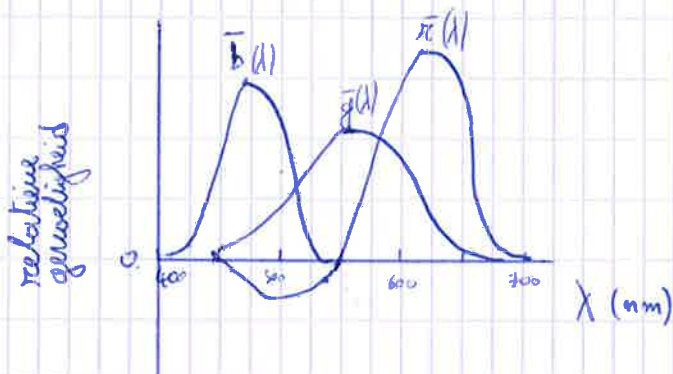
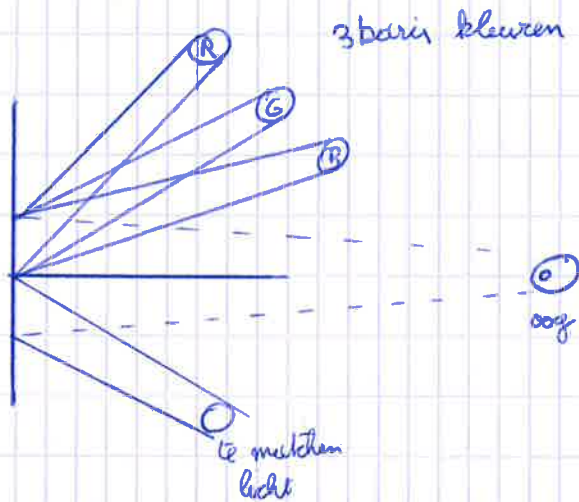


globaal resultaat

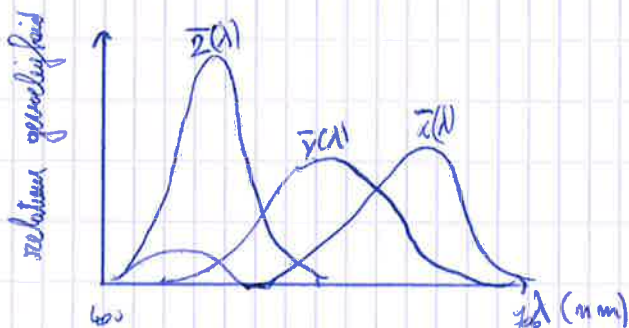
Color-Matching Functions

probleem: kleur specificeren en reproduceren

Dat is uit een experiment met een colorimeter result gebruikt om color matching functions te krijgen.



$\bar{x}(\lambda)$ heeft een negatief deel! Er werden dan ook een set v. fictieve basis kleuren bedacht, deze geven aanleiding tot de color matching functions, met enkel positieve waarden. Ze zijn het resultaat v.e. 3×3 matrix transformatie van de \bar{x} , \bar{y} , \bar{z} curves en worden genoteerd als $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ en $\bar{b}(\lambda)$. $\bar{y}(\lambda)$ valt samen met de luminance-efficiency curve $V(\lambda)$.



CIE Chromaticity Diagram.

Voor een algemene SPD $E(\lambda)$ wordt de essentiële colorimetrische informatie benodigd om een kleur te karakteriseren gegeven door een set v. tristimulus X , Y en Z gedefinieerd als

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

$$Y = \int E(\lambda) \bar{y}(\lambda) d\lambda \quad \text{luminantie}$$

$$Z = \int E(\lambda) \bar{z}(\lambda) d\lambda$$

3D-ruimte is moeilijk voor te stellen.

We stellen vast dat het verhoging v.d. helderheid enkel een scalar effect heeft op de tristimulus X , Y en Z . We kunnen dus een 2D diagram bedenken door de grote v.d. vectoren X , Y en Z eruit te filteren. Dit doen we door ze te delen door hun som.

$$x = X / (X + Y + Z)$$

$$y = Y / (X + Y + Z)$$

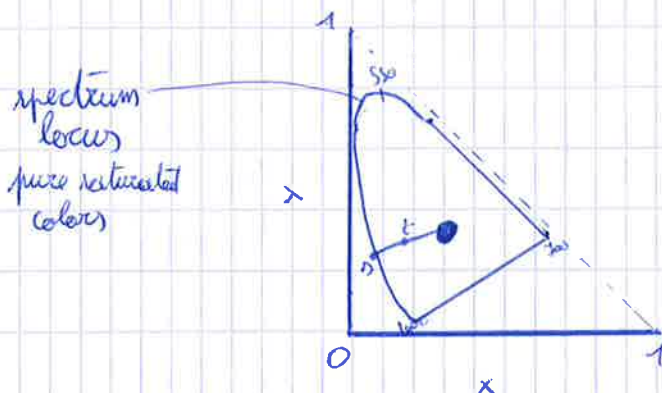
$$z = Z / (X + Y + Z)$$

En dus geldt

$$z = 1 - x - y$$

x, y noemen we chromaticities

We projecteren dus elke tristimulus vector (X, Y, Z) op het vlak $(1, 0, 0), (0, 1, 0), (0, 0, 1)$.



- $(\frac{1}{3}, \frac{1}{3})$ white point
- λ is de dominante golflengte v. t. complementaire kleuren v. t. vindt men op de lijn tussen E en het whitepoint

Color Monitor Specifications

monitor specificaties bestaan uit de gegevens, door de maker
reest gekozen chromaticities naar de monitors forser, samen met
het standaard benodigde white point.

	red		green		blue		white point	
system	x_r	y_r	x_g	y_g	x_b	y_b	x_w	y_w
NTSC								
SMPTE								
EBU								

Out-of-Gamut Colors

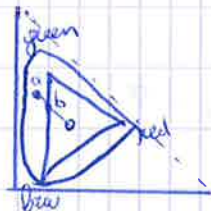
hoe apparaat onafhankelijke kleuren genereren, gespecificeerd door
(x,y) chromaticity waarden, gebruikt makend een apparaat
specifieke RGB kleur waarden.

$$(x, y) \rightarrow (x, y, 1-x-y) \rightarrow (R, G, B)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Leftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix}^{-1} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

RGB zijn apparaat specifiek en na berekening kan er een
negatieve waarde voorkomen. Dit kleur is zichtbaar naar
menen, maar kan niet worden weergegeven door het apparaat.
We noemen dit kleur out-of-gamut. We opteren voor het weergeven
v.h. dichtste complementaire in gamut kleur.

Orange is color matching linear is, valt voor een monitor elk in
gamut kleur binnen een driehoek



a valt buiten gamut
b is het dichtst in gamut
complementaire kleur.

White-Point Correction

2 problemen:

- * ipm $(x, y) \rightarrow R, G, B$, willen we eigenlijk $(X, Y, Z) \rightarrow R, G, B$ hebben, dus met magnitude inbegrepen.
- * als $R=G=B=1$, dan hebben we de verkeerde waarden (x, y) voor het white point.

Aflossingsmethode.

* neem de white point magnitude aan Y als eenheid

$$Y_{\text{whitepoint}} = 1$$

, door $X_{\text{white}} = \frac{x_w}{y_w}$

$$Y_{\text{white}} = \frac{y_w}{y_w} = 1$$

$$Z_{\text{white}} = \frac{z_w}{y_w}$$

(x_w, y_w, z_w) zijn geg. door specs.

* nu zoeken we een set van 3 correctie factoren, zodat als de gains v.d. 3 elektronen kanonnen vermenigvuldigd worden met deze correctie factoren, we exact de white point waarden $X_{\text{white}}, Y_{\text{white}}, Z_{\text{white}}$ krijgen als $R=G=B=1$.

$$XYZ_{\text{white}} \equiv M \cdot (d_1, d_2, d_3)^T$$

↳ matrix van fosfor chromaticities
geg. door monitor specificatie

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{white}} = M \cdot \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

en dus

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = M^{-1} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{white}}$$

XYZ to RGB Transform

Met $T = M \cdot D$

, met d_1, d_2 en d_3 de met bekende correctie factoren.

$$T = M \cdot \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}$$

dan

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = T \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

en dus

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = T^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Transform with Gamma Correction

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = T^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

veranderde! lineaire RGB ~~waarden~~ signalen, maar meestal hebben we nog aan met lineaire R' G' B'.

We zouden op de bekende RGB gamma correctie kunnen toepassen, maar meestal gebeurt het anders.

Color models in Images.

Subtractive Color: CMY Color Model.

in CRT: additief color

in printing: subtractief color.

RGB wordt $-R -G -B$
C M Y

cyan magenta yellow

RGB to CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Undercolor Removal: CMYK

$C+M+Y$ zou zwart moeten zijn, maar is vaak iets lichter bruin. Zwarte inkt is ook goed koper dan de mix. We halen dus dat deel v. C M en Y dat samen zwart maakt eruit.

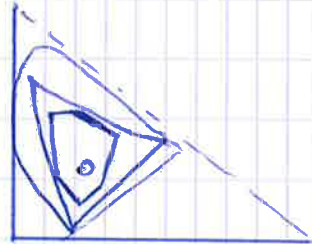
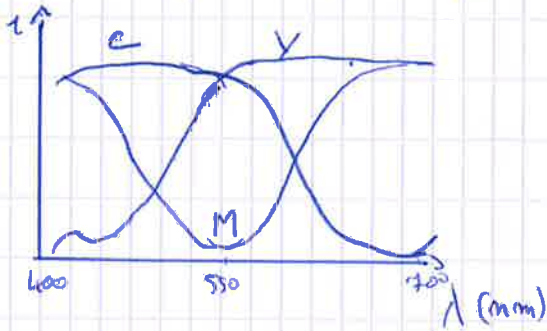
$$k = \min \{ C, M, Y \}$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} C-k \\ M-k \\ Y-k \end{bmatrix}$$

Printer Egemat.

Transmissie curves voor C, M en Y ink overlopen en dit geeft aanleiding tot cross talk tussen de kleur kanalen en maakt het moeilijk printless kleuren te waarneffen.

doorlaatbaarheid



Color Models in Video

YUV Color Model

gebruikt in PAL analoge video

encodeerd een luminantie signaal Y' (voor gemiddeld corrected signaal).

kleur is geencodeerd als het verschil tussen de kleur en een referentie wit bij dezelfde luminantie

$$U = B' - Y'$$

$$V = R' - Y'$$

Deze worden nog herschaald

$$U = 0,492 (B' - Y')$$

$$V = 0,877 (R' - Y')$$

Het chrominantie signaal C wordt gegeven door compositie van U en V

$$C = U \cdot \cos(\omega t) + V \cdot \sin(\omega t)$$

met ω de NTSC color frequentie

YIQ Color Model

- * gebruikt in NTSC kleur tv broadcasting.
- * omvat betere prioriteit van signalen met betrekking tot de menselijke perceptie v. kleuren.
- * Een versie van YUV met zelfde Y' en met U en V gerooteerd over 33°

$$I = 0,877 (R' - Y') \cos(33^\circ) - 0,492 (B' - Y') \sin(33^\circ)$$
$$Q = 0,877 (R' - Y') \sin(33^\circ) + 0,492 (B' - Y') \cos(33^\circ)$$

YCbCr Color Model

- * standaard voor component digital video
- * gebruikt in JPEG en MPEG compressie
- * gelijkt of YUV. ~~Geen Q , maar met een coefficient van $0,5$~~

$$C_b = \left((B' - Y') / 1,772 \right) + 0,5$$
$$C_r = \left((R' - Y') / 1,402 \right) + 0,5$$

(H5) Fundamenteel Concepten in Video

Types of Video Signals

* Component Video

- 3 fysiek aparte signalen
- beste kleur reproductie, neemt geen overslag
- hogere bandbreedte en meer synchronisatie nodig

* Composite Video

- kleur (chrominantie) en intensiteit (luminantie) in 1 signaal
- chrominantie is samengesteld uit U en V of I en Q en wordt door een chroma subcarrier aan de kant v.d. hoge frequentie v.h. signaal, dat het deelt met het luminantie signaal geplaatst.
- interferentie is onvermijdelijk.

* S-Video

- compromie tussen bovenstaande
- 2 fysiek aparte signalen
 - 1 voor de belangrijkste luminantie
 - en 1 voor de samengestelde chrominantie

Analog Video

* analogoog signaal $f(t)$ stelt een tijdsafhankelijke afbeelding voor.

* progressieve scanning:

tekent in elk tijdsinterval ($1/25$ sec) alle rijen v.v. frame

* interlaced scanning

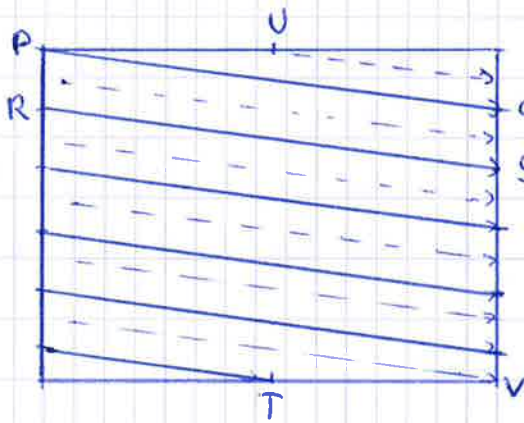
oneven lijnen eerst

$P \rightarrow Q, R \rightarrow S, \dots \rightarrow T$

even lijnen erin

$U \rightarrow, \dots \rightarrow V$

\Rightarrow oneven en even velden die samen 1 frame vormen



voordeel: verminderd flikkering
nadeel: beweging bij snelle beweging
objecten

* deinterlacing: onegeert een veld en dupliceert de lijnen v.h. ander veld.

* horizontal retrace: Q na R

* vertical retrace: T na U en V na P

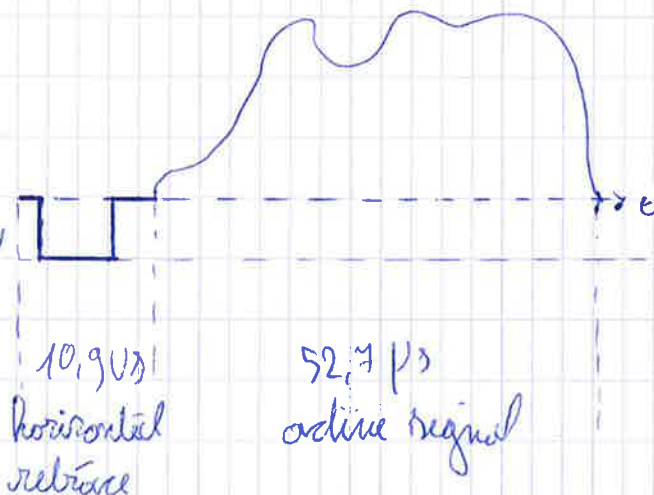
* hoe weten we wanneer een nieuwe lijn begint in het één dimentionale continue spanningsignaal?

kwart: iets boven 0V

start nieuwe lijn: 0V "blacker than black"

white (0,714V)

Black (0,055V)
Blank (0V)
sync (-0,286V)



electronic signal for one NTSC scan line

NTSC Video.

- * TV standaard in Noord-Amerika en Japan.
- * 4:3 beeldverhouding, 525 scanlijnen aan 30 fps.
- * interlaced scanning, met 2 velden van 262,5 lijnen met aan het begin v. elk veld 20 lijnen blanking informatie en aan het begin v. elke lijn $\frac{1}{6}$ v. s. lijn voorbehouden voor blanking en controle informatie. Dit leidt tot een totaal van 485 zichtbare lijnen.



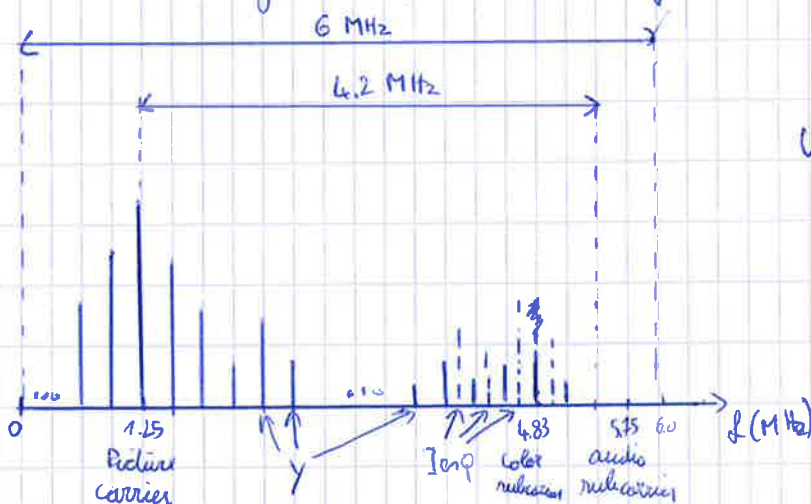
- * NTSC video is een analoge signaal zonder vaste horizontale resolutie. Hierdoor moeten we dus beslissen hoeveel keer het signaal gesampled moet worden. Elk sample correspondeert met een pixel oefft. Een pixel clock verdeelt elke horizontale lijn in samples. Hoe hoger zijn frequentie hoe meer pixels of samples per lijn.

- * gebruikt het YIQ color model.

quadrature modulation om I en Q signalen in 1 chroma signaal C te combineren.

$$\text{color subcarrier } C = I \cos(F_c t) + Q \sin(F_c t)$$

met magnitude $\sqrt{I^2 + Q^2}$ en fase $\tan^{-1}(Q/I)$ en $F_c \approx 3,58 \text{ MHz}$



interleaving Y en C signals

* Decoding the composite signal

- Y v. C scheiden door middel v. o. low pass filter
- uit C : I en Q extraheren.
om I uit C te halen:

rekansregeltjes:

$$2 \cos^2(x) = 1 + \cos(2x)$$

$$2 \sin^2(x) = 1 - \cos(2x)$$

$$2 \sin(x) \cos(x) = \sin(2x)$$

$$\begin{aligned} C \cdot 2 \cos(F_{rc} t) &= I 2 \cos^2(F_{rc} t) + Q 2 \sin(F_{rc} t) \cos(F_{rc} t) \\ &= I \cdot (1 + \cos(2F_{rc} t)) + Q 2 \sin(F_{rc} t) \cos(F_{rc} t) \\ &= I + I \cdot \cos(2F_{rc} t) + Q \sin(2F_{rc} t) \end{aligned}$$

gebruik nu een low pass filter om hieruit I te halen

om Q eruit te halen.

$$\begin{aligned} C \cdot 2 \sin(F_{rc} t) &= I 2 \cos(F_{rc} t) \sin(F_{rc} t) + Q 2 \sin^2(F_{rc} t) \\ &= I \sin(2F_{rc} t) + Q (1 - \cos(2F_{rc} t)) \\ &= I \sin(2F_{rc} t) + Q - Q \cos(2F_{rc} t) \end{aligned}$$

low pass filter Q hieruit.

Digital Video

Voordelen:

makkelijk op te slaan en bewerken

directe access

meeruudelig kopiëren zonder verlies v. kwaliteit.

makkelijk encrypteerbaar en betere tolerantie tegen noise.

Chromasubsampling

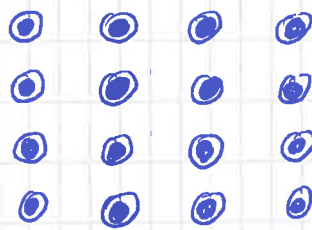
Elke pixel heeft een Y, een G en een B waarde.

menren zijn echter hoofzakeelij gevoelig voor spaciële resolutie in zwart-wit en veel minder naar kleur.

We kunnen dus leven met minder G en B waarden.

● : G en B waarden

○ : Y waarde



4:4:4

no chroma subsampling



4:2:2



4:1:1



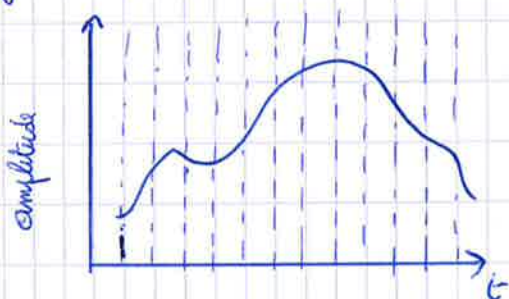
4:2:0

(H6) Basics of Digital Audio

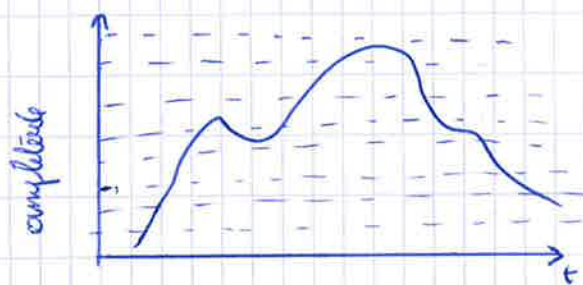
What is sound?

Longitudinale druk golf. Lucht moleculen worden samengekeurd of uit elkaar getrokken. Longitudinale druk golf met normale golf eigenschappen zoals reflectie, breking en afbuiging. Druk is omhoog en neemt continue waarden aan. De druk op een bepaald punt meten en omzetten in voltage.

Digitisation



sampling



quantisation

- waarden veranderen over tijd in amplitude
- amplitude is een continue hoeveelheid.
- sampling: meten v.d. hoeveelheid waaraan we geïnteresseerd zijn, meestal op gelijkverdeelde intervallen.
- quantisation: sampling in de amplitude dimensie, uniform of niet uniform.
- sampling rate: snelheid waarmee samples worden genomen.
- De mens zijn oor hoort: 20 Hz tot 20 kHz
- menselijke stem kan tot 4 kHz produceren, waardoor worden vaak sampling rates gebruikt van 8 kHz tot 40 kHz.
- Zowel sampling in tijds dimensie als quantisatie geeft aanleiding tot discretisatie fouten.

Nyquist Theorema

frequentie : absolute waarde

pitch : relatief, tonhoogte

↳ link : moet A boven middel C op exact $4:4:3$ H₂ zetten.

1 octaaf hoger : dubbele frequentie.

harmonies : elke rij muziek noten waarvan de frequentie een integer vermenigvuldiging is v.e. grondtoon.

Nyquist theorema :

Om correct te samplen moeten we een sample snelheid gebruiken die gelijk is aan minstens twee maal de maximum frequentie v.b. inhoud v.d. signaal.

Deze snelheid noemen we de Nyquist-rate en deze is gelijk aan 2 maal de Nyquist-frequentie.

Meer algemeen :

Als een signaal bandgelimiteerd is door f_1 en f_2 , dan moet de sample snelheid minstens $2(f_2 - f_1)$ zijn.

Oaangerien het niet-mogelijk is om frequenties te herstellen die hoger zijn dan de Nyquist-frequentie, bevatten veel systemen een anti-aliasing filter. Deze filtert frequenties hoger dan de Nyquist-frequentie weg voor dat het signaal naar de sampler gaat.

Signal-to-Noise Ratio (SNR)

Analoge systemen gaan steeds gepaard met ruis of noise.

SNR: verhouding tussen vermogen v signaal en vermogen ruis, waarbij vermogen proportioneel is met V_{signaal}^2 en wordt uitgedrukt in dB.

$$SNR = 10 \log_{10} \frac{V_{\text{signaal}}^2}{V_{\text{ruis}}^2} = 20 \log_{10} \frac{V_{\text{signaal}}}{V_{\text{ruis}}}$$

Signal-to-Quantization-Noise Ratio (SQNR)

ruis in het originele analoge signaal is er ook nog de kwantisatie fout. Dit zijn afrondingsfouten gemaakt bij het omzetten van waarden uit continue interval naar een beperkt aantal discrete waarden.

Quantisatienuis is gedefinieerd als het verschil tussen de waarde v.h. analoge signaal of een bepaald tijdstip en het dichtste kwantisatie niveau.

$$\begin{aligned} SQNR &= 20 \log_{10} \frac{V_{\text{signaal}}}{V_{\text{quas noise}}} = 20 \log_{10} \frac{2^{N-1}}{\frac{1}{2}} \\ &= 20 \cdot N \cdot \log 2 = 6.02N \text{ (dB)} \end{aligned}$$

met N het aantal bits gebruikt om een gequantiseerde waarde voor te stellen. 1 extra bit zorgt dus voor 6 dB extra.

Linear and Nonlinear Quantization

Wet v. Weber. : verband tussen impuls en response

verband tussen de kracht van fysieke impulsen op de menselijke zintuigen en de intensiteit v.d. daarbijkorende gewaarwording.

$$\Delta \text{response} \propto \frac{\Delta \text{stimulus}}{\text{stimulus}}$$

met k een constante die de proportionaliteit aangeeft

$$dr = k \cdot \left(\frac{1}{s}\right) \cdot ds \quad \text{met response } r$$

stimulus s

door integreren

$$r = k \ln s + C \quad \text{met } C \text{ constant}$$

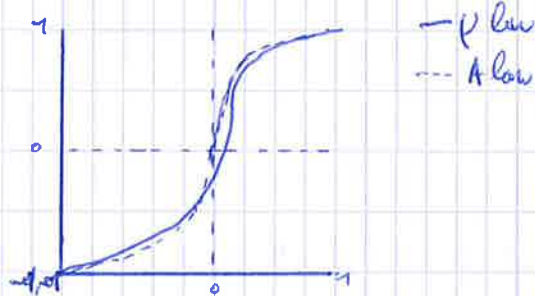
of

$$r = k \ln \left(\frac{s}{s_0}\right)$$

met s_0 het laagste niveau van stimulus, die een response veroorzaakt.

Niet lineaire quantisatie werkt door een analoge signaal in de s ruimte om te zetten naar de theoretische r ruimte. En dan uniform te quantiseren.

2 vbi: μ law
A law



signaal \rightarrow logaritmische compressor \rightarrow AD converter, uniform

signaal \leftarrow expander \leftarrow DA

Quantisation and Transmission of audio

Coding of audio.

Pulse Code Modulation

PCM in general:

basis technieken om een digitaal signaal te maken v. e. analoge signaal

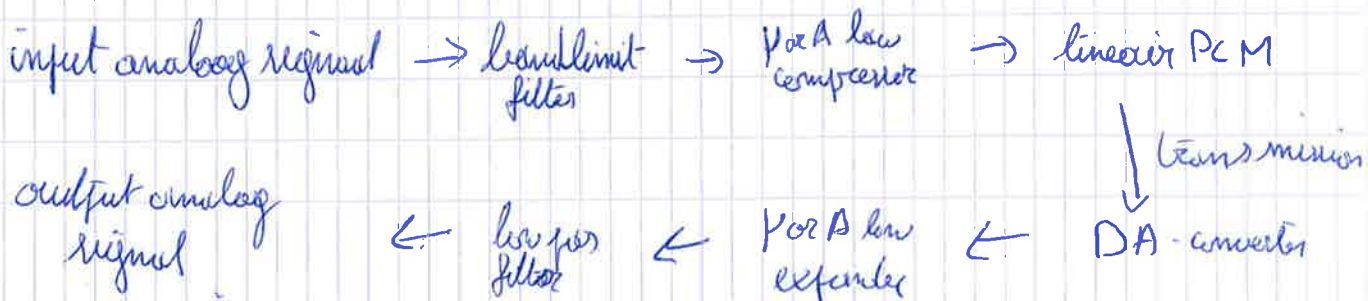
- sampling : kies een sampling rate en reproduceer een waarde of elk sample tijdstip
- quantisation : kies intervallen in de magnitude domein en rondt elke waarde binnen een bepaald interval naar de representatieve waarde van dit interval. (reconstruction level)

compression scheme heeft 3 stappen

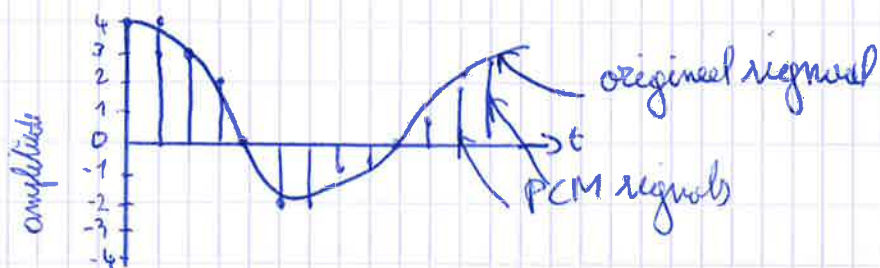
- transformatie
- loss
- coding

PCM in Speech Compression

- signalen worden eerst getransformeerd naar een logaritmische reeks door gebruik v. p law.
- daarna wordt uniforme quantisatie gebruikt.
gerekend hiervoor is dat we kleinere reeksmetingen in volume kunnen aangeven ~~aan~~ aan de lage volume kant v. spraak. Dus daar waar we dit meest willen.
- in telephonie willen we frequenties tot 4 kHz kunnen reconstrueren we gebruiken dus een sampling rate van 8 kHz
Indien we companding gebruiken kunnen we de grote v.d. sample reeks 8 bit maken zonder waarneembare verlies aan kwaliteit met maximaal of een bitrate van 64 kbps.



tot 4 kHz reconstrueren, dus eerst alle hogere laagfilteren met
lowpass filter.

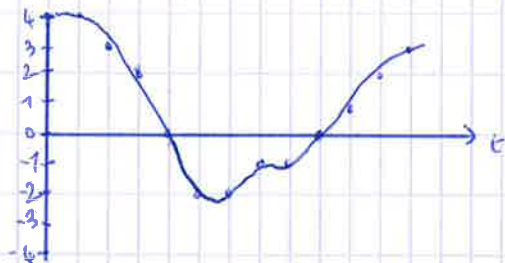


↓ DA converter



dit signaal bevat naast componenten
v.h. origineel ook een theoretische
oneindig set van hoge frequentie
componenten.

↓ lowpass filter



Differential Coding of audio

audio wordt weinig opgelezen in ruifel PCM formaat, maar in een formaat dat verschillen gebruikt.

Het voordeel is dat het histogram van een signaal in verschillen veelal veel gefiechter zal zijn dan het histogram v.h. origineel signaal.

Als we code woorden geven aan deze verschillen kunnen we korte geven aan veel voorkomende verschillen en langere aan minder voorkomende.

We kunnen dit op 2 manieren doen lossers (losser predictive coding) of lossy (DPCM).

Lossers Predictive Coding.

Lossers: decoder produceert zelfde signaal als de originele verschillen transmisseren. We voorspellen het volgende sample als zijnde gelijk aan het huidige en sturen niet het sample, maar het verschil door. (We zouden ook complexe voorspellingen kunnen maken)

eenwaardige predictor

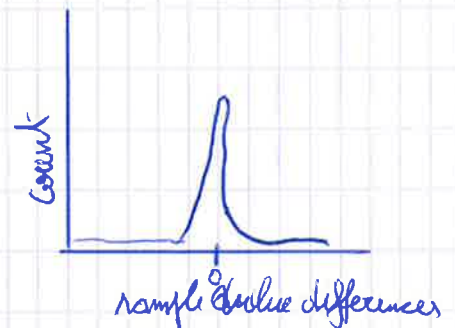
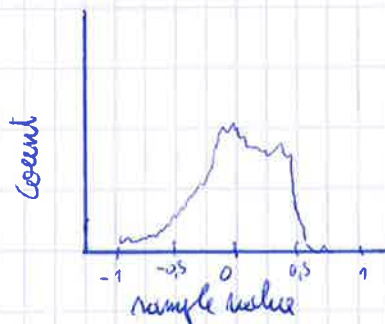
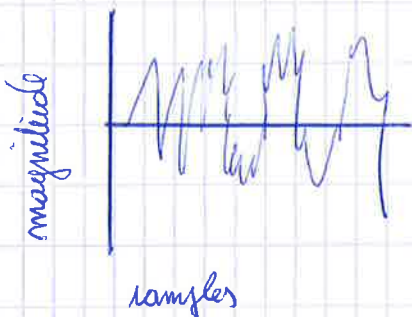
$$\begin{aligned} \hat{f}_m &= f_{m-1} \\ e_m &= f_m - \hat{f}_m \end{aligned}$$

\hat{f}_m is de voorspelde waarde
 e_m is de daarbij gemachte fout.

lineaire predictor

$$\hat{f}_m = \sum_{k=1}^{254} a_{m-k} f_{m-k}, \text{ truncated or rounded}$$

dit is de hoop e_m zo klein mogelijk te maken, en het histogram van alle e_m (wat we uiteindelijk doorsturen) zo gefiechter mogelijk te maken. Zodat we korte code woorden kunnen geven aan veel voorkomende waarden.



Probleem: wat als enkele woorden eenleiding geven tot enkele exceptioneel grote verschillen?

oplossing: definieer 2 nieuwe codes: SU en SD die staan voor shift up en shift down.

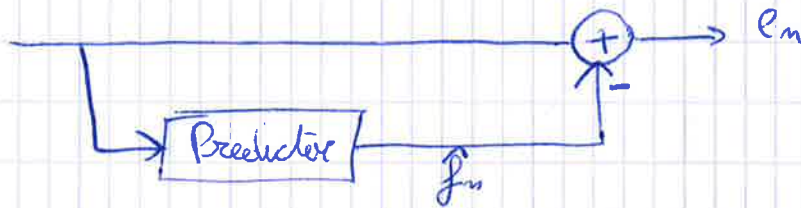
stel samples $\in [0, 255] \rightarrow e_n \in [-255, 255]$

definieer SU en SD als shifts met 92.

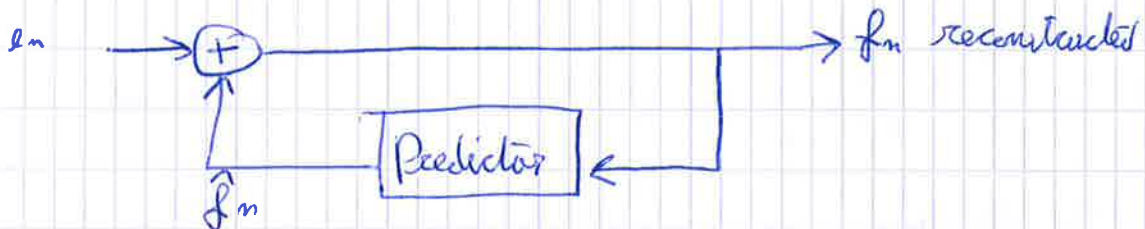
dan kunnen we coderwoorden toekennen aan $[-15, 16]$.

en voor e_n die daarbuiten vallen stellen we een code responentie op bestaande uit shift en een waarde in $[-15, 16]$.

schematisch
encoder
 f_n



decoder



Predictor moet dus op zijn minst f_{n-1} even bijhouden

DPCM Differential Pulse Code Modulation

- * Zelfde als predictive coding, maar met quantisatie stap. en is dus lossy. Quantisatie gebeurt zoals bij PCM en kan uniform of nonuniform zijn.
- * Lloyd-Max quantiser, gekend of kleinste kwadraten minimalisatie berekent de beste set van ~~met~~ nonuniforme quantisatie stappen.
- * nomenclatuur

f_n : original signal
 \hat{f}_n : predicted signal
 \tilde{f}_n : gereconstrueerde, gequantiseerde signaal.

* hoe?

- neem e_n door door het verschil & nemen van het signaal en het voorgaande signaal.
- quantiseer de error e_n naar \tilde{e}_n
- codeer met entropie codering \tilde{e}_n

$$\hat{f}_n = \text{functie van } (\tilde{f}_{n-1}, \tilde{f}_{n-2}, \dots)$$

$$e_n = f_n - \hat{f}_n$$

$$\tilde{e}_n = Q[e_n]$$

transmit codewoord (\tilde{e}_n)

$$\text{reconstructie: } \tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

Neuk hierop op dat de predictor steeds gekend is of de gereconstrueerde, gequantiseerde reëel-t. signaal, omdat de decoder enkel hierover beschikt.

De "distortie" is de gemiddelde kwadratische fout

$$\left[\sum_{n=1}^N (\tilde{f}_n - f_n)^2 \right] / N$$

Voor sprake kunnen we de kwantisatie stap aanpassen on the go, door het gemiddelte en de variantie te schatten voor een bepaalde hoeveelheid ~~signaal~~ sample waarden en de kwantisatie stap voor die blok aan te passen.

op tijdstap i een blok van N waarden f_n en de kwantisatie doet minimaliseren

$$\min \sum_{n=i}^{i+N-1} (f_n - Q[f_n])^2$$

Ongereguleerd signaal verschillen enorm gefiekt zijn, kunnen we dan distributie modellen met an bepaalde probability distribution functies.

$$l(x) = \left(\frac{1}{\sqrt{2\sigma^2}} \right) e^{\frac{-\sqrt{2}|x|}{\sigma}}$$

Als kunnen nu dus kwantisatie stappen aanpassen voor een kwantisator met niet uniforme stappen door te veronderstellen dat signaal verschillen, d_n , betrokken worden van zijn distributie en dan stappen te kiezen ~~en~~ zodat

$$\min \sum_{n=i}^{i+N-1} (d_n - Q[d_n])^2 l(d_n)$$

geminimaliseerd wordt.

Dit kleinste kwadraten probleem ^{iteratief} kan opgelost worden door gebruik te maken v.d. Lloyd-Max kwantisator.

DM Delta Modulation.

* Een vereenvoudigde versie van DPCM.

Uniform-Delta DM.

We gebruiken enkel 1 gequantiseerde error waarde (positief of negatief).
De gecodeerde output volgt dus het origineel signaal of een
vrij gelijkrijze manier.

$$\begin{aligned}\hat{f}_m &= \tilde{f}_{m-1} \\ e_m &= f_m - \hat{f}_m = f_m - \tilde{f}_{m-1} \\ \tilde{e}_m &= \begin{cases} +k & \text{if } e_m > 0 \\ -k & \text{otherwise} \end{cases} \\ \tilde{f}_m &= \hat{f}_m + \tilde{e}_m\end{aligned}$$

Adaptieve DM

wanneer de curve v.h. signaal heel steil stijgt of daalt, kan de trapsgewijze
het signaal niet bijhouden.

We kunnen k aanpassen naar de huidige eigenschappen v.h. signaal.

ADPCM Adaptive Differential Pulse Code Modulation.

Egaat nog verder in het adaptief aanpassen v.d. coder aan de input.
DPCM: quantizer en predictor, beide kunnen worden aangepast.

we kunnen de quantizer adaptief aanpassen, door het aanpassen v.d. stap grootte en de decision boundaries in een nonuniform quantizer.
dit kan op 2 manieren

- gebruik makend van eigenschappen v.h. input signaal (forward adaptive quantization)
- gebruik makend v.d. grootte v.d. gequantiseerde error. (backward adaptive quantization)

we kunnen ook de predictor aanpassen. (adaptive prediction coding)

$$\hat{f}_m = \sum_{i=1}^M a_i \tilde{f}_{m-i} \quad \text{met } M \text{ de orde v.d. predictor.}$$

gebruiken aanpak v. minimaliseren van bleekst kwadratisch om zo de beste a_i te vinden

$$\min \sum_{n=1}^N (f_n - \hat{f}_n)^2 \quad , \text{ maar omdat } \hat{f}_n \text{ afhankelijk is v. gequantiseerde is dit te moeilijk.}$$

Dus gebruiken we

$$\min \sum_{n=1}^N \left(f_n - \sum_{i=1}^M a_i f_{n-i} \right)^2.$$

differentieer naar elke a_i en gelijk stellen aan 0, levert een lineair stel vgl (M) dat makkelijk op te lossen is.