

HOOFDSTUK 1: Inleiding

§1. Wat is multimedia?

definitie

veelzijdig gebruikt + contextafhankelijk

computerwetenschappen:

“computertoepassingen waarin verschillende ‘media’ gebruikt worden

media = spraak, geluid, stilstaand beeld, bewegend beeld, animatie, tekst, ...

+ **interactiviteit**

onderzoeksgebieden

multimediaverwerking en –codering

encoderen en decoderen van audio/beeld/video

compressie van audio/beeld/video

analyse van audio/beeld/video

multimediasystemen en –netwerken

netwerk protocols, internet, ... als én met multimedia

afspeelsystemen (hardware/OS/specs)

client/server en streamen

databanken en multimedia

multimediasystemen en –software

vb. user interface, ...

productie

diverse jobs en functies

technische aspecten

vb. compressie, interactie, implementatie, ...

artistieke aspecten

vb. artiest, GUI designer, schrijver, concept art, ...

lineaire en niet-lineaire multimedia

lineair: geen navigatie, interactiviteit (vaak wel, maar verborgen en niet standaard)

vb. presentatie, film, ...

niet-lineair: aan de hand van interactie met de gebruiker

vb. video games, hypermedia, ...

presentatie

stijlen zeer belangrijk

bepaald door onderzoek naar menselijk zicht
veel onderzoek naar gedaan (vb. marketing)

kleur-principes

niet te veel kleuren

niet als versiering

consistent gebruik van kleuren

contrast

vb. geen roze tekst op rode achtergrond

geen donkere achtergrond met lichte letters, maar omgekeerd

inverse kleuren (voor contrast)

$K = (R, G, B)$

$K' = (255-R, 255-G, 255-B)$

lettertype

groot genoeg lettertype + weinig tekst tegelijkertijd

zeker bij vb. presentatie

schreefhebbend vs. schreefloos

schreefhebbend: lange tekst, ...

schreefloos: presentatie, website, ...

§2. Geschiedenis van multimedia

1945 – Artikel ‘*Memex*’

door Vannevar Bush
al enkele basisconcepten van hypertext
rechtstreekse inspiratiebron voor latere ontwikkeling

1960 – Introductie term ‘*hypertext*’

door Ted Nelson

1967 – Oprichting *Architecture Machine Group* aan MIT

door Nicholas Negroponte

1968 – Voorstelling *On-line System (NLS)*

door Douglas Engelbart
primitieve hypertext-applicatie

1969 – *FRESS*

door Ted Nelson en Andries van Dam (Brown Universiteit)
primitief hypertext systeem en –editor

1985 – Oprichting *MIT Media Lab*

door Nicholas Negroponte en Jerome Wiesner
opvolger van de Architecture Machine Group

1989 – Voorstelling *World Wide Web*

door Tim Berners-Lee en Robert Cailliau (CERN, Geneve)

1990 – Successen *Apple Multimedia Lab*

door Kristina Hooper Woolsey
van Apple en later terug gestopt

1991 – Aanvaarding MPEG-1 als digitaal beeld standaard

voorloper van MPEG-2, MPEG-4,... in jaren ‘90

1991 – Opkomst PDA’s

wijzigend computergebruik t.o.v. MM (mobiel, klein scherm)
geschiedenis:

Palm vs. Compaq

Compaq alliantie met Microsoft => Palm weg

Compaq overgenomen door HP

1992 – Aanvaarding JPEG als standaard voor digitale beeldcompressie

voorloper van JPEG2000

1993 – Productie *NCSA Mosaic*

door National Center for Supercomputing Applications (University of Illinois)
eerste volledige browser
MM in al zijn vormen over hele wereld sturen!

1994 – Productie *Netscape*

door Jim Clarck en Marc Andreessen
geschiedenis:

concurrentie met IE

proces tegen Microsoft <-> laten rekken en iedereen overgestapt

neergang

Mozilla project (Open Source)

1996 – Introductie DVD Video

1998 – XML 1.0 als W3C Recommendation

1998 – Opkomst MP3-spelers

§3. Standaardisatie

W3C-standaarden

standaard: open en gratis beschikbaar

<-> gebruik: NIET altijd gratis (wegens patenten)
verantwoordelijkheid bij gebruiker (disclaimer)

procedure

1. standaardisatiecomité geeft requirements uit voor bepaalde toekomstige standaard
2. markt reageert + dienen elk hun 'oplossing' in markt = bedrijven, universiteiten, onderzoeksgroepen competitieve fase
3. standaardisatiecomité vergelijkt en test ze a.d.h.v. *core-experiments* vaak vele verschillende beter op andere deelaspecten
4. standaardisatiecomité kiest die beste delen + brengt partners samen (proberen)
5. partners werken samen om tot geheel te komen <-> nog steeds veel (eigen)belangen en discussies coöperatieve face
6. testen + publiceren van standaard <-> onderliggend vele patenten

MPEG (ISO)-standaarden

MPEG verplicht patenten van meewerkende entiteiten

standaard: open en tegen (kleine) betaling

<-> gebruik: NIET altijd gratis (wegens patenten)

procedure

5. <analoog W3C>
6. testen + klaar voor publicatie
7. standaardisatiecomité vraagt naar patenten verplicht op ALLE geïntegreerde technologieën patent = teken van engagement + geloof in eigen technologie want toen shitload aan nutteloze + ongebruikte standaarden
8. publicatie

video-standaarden

huidige standaarden

analoge TV: NTSC (VS) en PAL (EU) en SECAM (Fr.) + niet compatibel!
digitale TV: MPEG-2
DVD: MPEG-2
Blu-ray: MPEG-2 of H.264/MPEG-4 ACV

geschiedenis

MPEG-2 = enorm succes
standaardiseren van decoder
= vanuit consument standpunt (vb. mp3 speler)
<-> encoder niet gestandaardiseerd!!
=> concurrentie blijft mogelijk
testen v/e device = bitstream door decoders laten lopen

MPEG-4 AVC

door MPEG (ISO) en ITU (VN)
opvolger MPEG-2 en backwards compatibel
=> geen onmiddellijke overstap voor content providers nodig
tot 40% bitreductie mogelijk
maar niet altijd, want afhankelijk van encoder!

bemerkingen

open vs. gratis

standaarden vaak open = vrij te bekijken (evt. tegen vergoeding)
toch gepatenteerd => voor gebruik misschien betalen!

gepatenteerd vs. te betalen

vaak onderliggende patenten als verdedigingsmechanisme
want alle rechten bij maker (vb. latere versies)
maar toch gratis voor normale gebruikers

§4. Metadata

wat

= 'data over de (multimedia)data'

vaak XML

vb. maker, regisseur, componist, keywords, rating, GPS locatie, ...

nut

zeer belangrijk voor latere archivering/opslag

geldt als fiche voor MM-bestand

gebruikt voor opzoekingen in MM-databanken

als textuele data/info 'op' audio, video en beeld

makkelijker te lezen en te interpreteren

<-> afleiden uit groot, lang en zwaar bestand

aanmaken

bij creatie van het MM-project!!

reden 1:

later moeilijker te achterhalen

dan uit bestand halen of door onderzoek

= arbeidsintensief + kostelijk

reden 2:

later hele archieven tegelijk voorzien

dan zeer dure operatie, dus zelden rendabel

§5. XML technologieën

XML

= 'Extensible Markup Language'

standaard voor definiëren van formele mark-up talen
representatie van gestructureerde gegevens
platte tekst

meta-taal

uitbreidbaar

zelf 'standaarden' maken, afspreken, en vastleggen via DTD en XSD

vereenvoudiging zeer complexe SGML
specifiek voor WWW

te herkennen door computers (parsen) én door mens (leesbaar)

strikte scheiding inhoud <-> opmaak

geen opmaak in XML bestand

opmaak door externe stylesheet (XSL, CSS)

opmaak door transformeren naar XHTML, PDF, WordML, ... (XSLT)

SMIL

'Synchronized Multimedia Integration Language'

doel: MM-presentaties definiëren in XML

verschillende vorm

temporeel: wanneer wat tonen?

spatiaal: waar wat tonen?

nog weinig ondersteund

niet door vb. browsers

wel al door media players

§6. VRML

VRML

= Virtual Reality Modelling Language (“Virmel”)

doel: objecten beschrijven in 3D wereld (VRML world)
bevat enkele elementaire bouwstenen om mee te bouwen
laat deze objecten (met elkaar) interageren

standaard

vrij beschikbaar

<-> gepatenteerd

beschrijft elementaire blokjes + hoe ermee te bouwen + hoe ermee te interageren

succes vanaf MPEG-4

daar objecten apart uit film halen + apart coderen

nood aan “scenebeschrijving”

= hoe staan objecten t.o.v. elkaar

beetje gebaseerd op SMIL

niet enkel wat zit waar en wanneer

ook samenstelling door bouwmogelijkheden

VRML-shapes

soort klassestructuur van type nodes (= hiërarchie)

shape-node

generische node van alle objecten in VRML

vb. Box, Sphere, zijn geometry nodes

material-node

bevat eigenschappen van oppervlakte v/e object

kleur v/h object

transparantie v/h object

texture-node!

texture-node

doel: textures “mappen” op een 3D-object

ImageTexture / MovieTexture

PixelTexture (ook soort figuur, maar pixel per pixel doorsturen)

VRML-world

bevat algemene opties/eigenschappen

camera/point of view

instelbaar

positie, rotatie, hoek

belichting

3 soorten

DirectionalLight

bepaalde richting (evenwijdige lichtstralen)

gehele scene belicht

PointLight

lichtbron 'in' de scene (klein object)

in alle richtingen (diffuus)

SpotLight

lichtbron 'in' de scene (klein object)

in bepaalde richting + breedte

achtergrond

instelbaar

panorama

texture plakken op randen v/d wereld

grote kubus rond de VRML-wereld

interactie en animatie

a.d.h.v. geïnterpreteerde taal bij VRML

=> zeer traag

=> nooit gebruikt

vaak apart via externe toepassing/taal implementeren

HOOFDSTUK 2: Kleuren in multimedia

§1. Kleuren en licht

licht en het elektromagnetisch spectrum

licht = elektromagnetische golf
amplitude -> intensiteit
golflengte -> kleurtint

zichtbaar licht
golflengte 400-700 nm

spectrum of SPD (spectral power distribution) $E(\lambda)$
geeft belang van golflengtes in een lichtbundel weer
vaak gestandaardiseerd (vb. standaard daglichten, fluorlicht, ...)

'gewoon' licht vs. laserlicht

gewoon licht:

verschillende golflengtes
incoherent i/d ruimte
vb. daglicht: bevat alle zichtbare golflengtes

laserlicht:

klein interval golflengtes (theoretisch 1)
coherent i/d ruimte (= bundel blijft lang samen)

opm: fluorlicht

enkele (bredere) pieken <-> 1 lijn bij laser

menselijk zicht

menselijk oog

netvlies met staafjes (rods) en 3 soorten kegeltjes (cones)

staafje

meet intensiteit (enkel grijswaarden)
vooral belangrijk in donkere omstandigheden

kegeltje

meten samen kleur
3 types voor 3 kleurwaarden (R/G/B)
vooral belangrijk in lichte omstandigheden

luminous-efficiency curve $V(\lambda)$

som van gevoeligheidscurves van receptoren R, G en B
geeft relatieve gevoeligheid voor golflengtes in ons oog weer
resultaat: vooral bij rood en groen

gepercipieerde kleurwaarden v/e lichtbundel

= emissie van bron gewogen op onze receptiviteit voor die component

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

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

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

(opmerking: op saturatiefactor na!!)

lichtreflectie en -absorptie op oppervlakten

principe

oppervlakten kunnen licht reflecteren en absorberen
verschillend naar hoeveelheid en per golflengte
dit bepaalde 'kleur' van deze oppervlakte

reflectiefunctie $S(\lambda)$

geeft weer hoeveel van het licht gereflecteerd wordt per golflengte

kleur van een oppervlak

= emissie van de bron gewogen op de reflectiefunctie

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

gepercipieerde kleurwaarden v/e oppervlakte

= emissie van bron gewogen op de reflectiefunctie

en dat resultaat gewogen op onze receptiviteit voor die component

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

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

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

(opmerking: op saturatiefactor na!!)

gammacorrectie

probleem

helderheid van licht uitgedrukt in potentiaal (volt)
maar geen lineair verband!

$$\text{licht} = V_{in}^\gamma \quad \text{met } \gamma \neq 1 \quad (\text{vb. gamma rond 2.2 bij CRT})$$

vb. bij waarde R toont scherm de hoeveelheid licht die bij R^γ hoort.

gevolg

stel 0 is zwart en 1 is wit (bij volt en lichthoeveelheid)
alles ertussen gaat donkerder weergegeven worden
vooral donkergrijs als praktisch zwart

oplossing

aangepaste waarde $R' = R^{1/\gamma}$ doorsturen

zodat na gammacorrectie oorspronkelijke getoond word

$$R'^\gamma = (R^{1/\gamma})^\gamma = R$$

MAAR gamma exact bepalen is zeer duur, dus meestal experimenteel

§2. Color matching functions

color matching functions

doel

- kies 3 (monochrome) hoofdkleuren
- bepaal voor elk (monochroom!) kleur hoeveel je van de hoofdkleuren nodig hebt
relatief t.o.v. verschillende λ
- relatief t.o.v. de drie hoofdkleuren

gebruik

- complex licht definiëren adhv deze 3 hoofdkleuren
- $F_i = \int_0^\infty E(\lambda) \cdot \bar{f}_i(\lambda) d\lambda$ met $E(\lambda)$ de SPD van de lichtbundel

CIE RGB kleuruimte

experimenten

- via colorimeter
- verschillende monochrome testkleuren 'matchen' met R, G en B schakelaar
- daaruit color matching functies $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ en $\bar{b}(\lambda)$ opstellen

probleem

- praktisch niet alle kleuren te vormen met RGB
- wel door bij testkleur een bepaalde hoeveelheid van 1 hoofdkleur bij te voegen
voorgesteld door negatieve functiewaarden
- $\bar{r}(\lambda)$ -functie heeft negatieve waarden

CIE XYZ 1931 kleuruimte

3 voorwaarden

- enkel positieve waarden
- te bepalen vanuit RGB ruimte via een lineaire transformatie
- middelste functie $\bar{y}(\lambda)$ komt overeen met luminous-efficiency curve $V(\lambda)$
was niet zeker dat dit met een lineaire transformatie zou kunnen!
doel: $Y = \int_0^\infty E(\lambda) \cdot \bar{y}(\lambda) d\lambda$ is dan de (waargenomen) luminantie/helderheid

XYZ-waarden

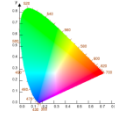
- definiëren een SPD $E(\lambda)$
- $X = \int E(\lambda) \cdot \bar{x}(\lambda) d\lambda$
- $Y = \int E(\lambda) \cdot \bar{y}(\lambda) d\lambda$
- $Z = \int E(\lambda) \cdot \bar{z}(\lambda) d\lambda$

§3. CIE xy chromaticiteitsdiagram

opbouw

reden:

3D (XYZ) data moeilijk weer te geven
dus omzetten naar 2D (xy)



chromaticiteit van een kleur weergeven met twee parameters x en y
<-> andere (Y waarde) staat voor luminantie

berekening

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z}$$

redundantie

$$x + y + z = \frac{X+Y+Z}{X+Y+Z} = 1 \quad \leftrightarrow \quad z = 1 - x - y$$

we houden nog twee vergelijkingen over

structurele eigenschappen

ingekleurd gebied: 'gamut' van het menselijk zicht
= alle zichtbare kleuren

bovenrand: 'spectral locus'
= alle monochrome kleuren

onderrand: 'line of purples'
wel geen voorstelling in monochroom licht

buitenkant: gesatureerde kleuren

centrum: wit
op (1/3, 1/3)

kies twee punten (lichten)
=> alle mengelingen liggen op de lijn tertussen

kies n punten (lichten)
=> alle mengelingen liggen in die n-hoek

dominante golflengte van een kleur
= snijpunt met locus van rechte door het punt en wit

'out of gamut'-kleuren

reden:

kleurenruimte definiëren adhv 3 basiskleuren
EN vanuit 3 punten kan je NOOIT alle zichtbare kleuren bepalen

benadering

hoogste mogelijke saturatie
zelfde dominante kleur
= snijpunt met locus van rechte door het punt en wit

§4. Kleuren weergeven

probleem

omzetting tussen

apparaatonafhankelijke (x, y, z) waarden (chromaticiteitsdiagram)

apparaatafhankelijke RGB-waarden

apparaatspecificaties

Rood, Groen en Blauw in het chromaticiteitsdiagram vastleggen

= hoekpunten van gamut

'White-point' in het chromaticiteitsdiagram vastleggen

waarden

$$x_r, y_r, x_g, y_g, x_b, y_b, x_W, y_W$$

via standaarden

vb. NTSC, SMPTE, EBU

overgang (x,y,z) en RGB

doel: welke chromaticiteit geeft een bepaalde RGB-waarde ECHT

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

doel: welke RGB-waarden moet ik geven om een bepaald (x,y) kleur ECHT te zien

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

probleem: soms negatieve R, G of B waarden

dan out-of-gamut kleur voor de RGB ruimte!

witpuntcorrectie

problemen

niet enkel chromaticiteit mappen, maar ook magnitude
dus XYZ ipv xyz

$R = G = B = 1$ geeft niet de gevraagde (x_W, y_W) chromaticiteit

witpunt magnitude als eenheid nemen

stel $Y(\text{white point}) = 1$

we zoeken drie correctiefactoren d_1, d_2, d_3

zodat wanneer $R = G = B = 1$, dan exacte XYZ voor wit

XYZ waarde te bepalen uit (x_W, y_W) met $Y = 1$

correctie als diagonaalmatrix $D = \text{diag}(d_1, d_2, d_3)$

we bepalen de correctiewaarden

$$\begin{aligned} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{wit}} &= M \cdot \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ &= M \cdot \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \end{aligned}$$

\Leftrightarrow

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

opmerking

moesten correctiefactoren 1 zijn, was er geen correctie gebeurd
kan oplopen tot 30% afwijking!

finale transformatie

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

we krijgen transformatiematrix $T = M \cdot D$

ook voor niet-witpunten gebruiken we deze matrix!

transformatie met gammacorrectie

beste manier

bepaal lineaire signalen

via bovenstaande finale transformatie

pas gamma correctie hierop toe

meest gebruikte manier

pas bovenstaande transformatie toe op niet-lineaire signalen

weinig verlies

belang Y' als helderheid v/e pixel

§5. Kleurmodellen voor printen (CMY, CMYK)

additieve kleuren vs. subtractieve kleuren

additieve kleuren

werkt met lichtbronnen

kleuren van twee verschillende bronnen worden opgeteld

vb. lichtstralen, fosfordeeltjes op monitor die oplichten, ...

subtractieve kleuren

werkt met deels absorberende oppervlakten

we onttrekken kleuren aan (dag)licht

wat we zien is de rest

vb. inkt

CMY kleurmodel

basiskleuren niet meer RGB

maar hun complimenten!

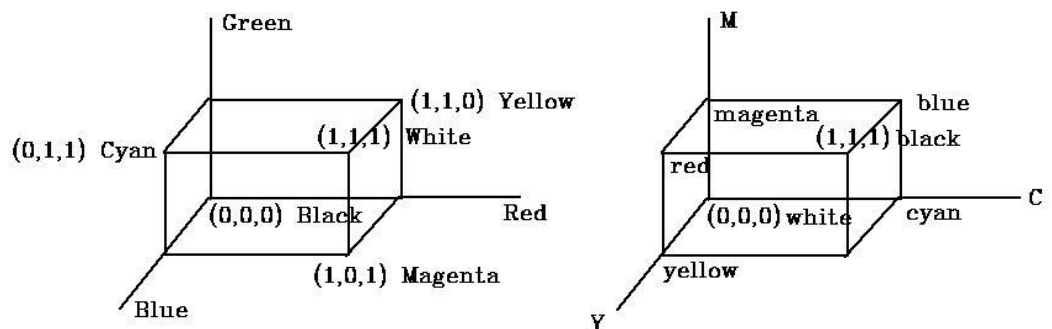
Cyan-Magenta-Yellow

Cyaan = wit – rood = groen + blauw

Magenta = wit – groen = rood + blauw

Geel = wit – blauw = rood + groen

RGB en CMY kubussen



transformatie RGB naar CMY

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

CMYK kleurmodel

apart kleurbakje voor zwart (key)
in plaats van zwart als som van 3 subtractieve basiskleuren (CMY)

reden:

veel meer zwart dan kleur printen
dus apart bakje
anders soort zwartpunt-correctie
onvolmaakt zwart (C+M+Y) geeft vuil bruin
= zeer storend
vgl. witpunt-correctie
<-> onvolmaakt wit stoort niet

transformatie CMY naar CMYK

$$K = \min(\{C, M, Y\})$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \rightarrow \begin{bmatrix} C - K \\ M - K \\ Y - K \end{bmatrix}$$

printer gamut

imperfecties bij additieve kleuren
gamma-correctie, witpunt-correctie zorgen voor imperfectie
voor perfectie: perfect rood, groen en blauw, perfect reflecterend scherm (wit), ...

imperfecties bij subtractieve kleuren (vb. printen)

nog veel erger!

C, M en Y inkt niet perfect (geen 'block dyes')

* blokkeert niet perfect 1 hoofdkleur (vb. 5 procent doorlaten)

* weerkaatst niet perfect de andere 2 (vb. 80 procent doorlaten)

* trage overgang in spectrum

<-> scherpe verticale lijnen op overgangen in spectrum

* vb. cyaan begint nog verder dan rood terug minder te absorberen

materiaal waarop je print

bepaalt mee absorptiekenmerken v/d inkt

we krijgen 'overspraak' (= interferentie)

tussen onze kleuronttrekkers (inktstoffen)

gamut van CMY-printer

gamut is maar beperkt deel van de subtractieve 'spectrum locus'

gamut is vaak een (onregelmatige) 6 hoek

centrum = zwart (vuil zwart)

6 hoeken = C, M, Y, CM, CY, MY

§6. Kleurmodellen voor video (YUV, YIQ, YCbCr)

welke kleurenruimte voor video?

uitgangspunt

video is groot

rekening houden met beperkte bandbreedte

RGB kleurmodel [R=G=B]

klassieke ruimte + vele toestellen werken hiermee

R, G en B kanalen bevatten evenveel info

gehalten in beelden ongeveer evenwaardig

probleem bij beperkte bandbreedte

wat laat ik vallen?

doel

andere kleurenruimte zoeken

via transformatie v/d kleurenruimte (R' , G' , B')

met prioritaire as(sen)

deze eerst doorsturen

bij beperkte bandbreedte eerst prioritaire doorsturen

YUV kleurmodel [$Y > U=V$]

kleurintensiteit of luminantie (Y) als prioritaire as

reden: mens gevoeliger voor Y-afwijkingen dan kleurtintafwijkingen

in principe kleurfoto -> zwart-wit foto

= zeer veel info

berekening:

$$Y' = 0.299 \cdot R' + 0.587 \cdot G' + 0.114 \cdot B'$$

(zie Y' van vroeger (XYZ) en inclusief gamma correctie)

twee chrominantie-componenten (U en V) als andere assen

chrominantie = verschil tussen kleur en grijswaarde met zelfde luminantie

berekening:

$$U = B' - Y' \qquad V = R' - Y'$$

opmerkingen

indien $R'=G'=B'$ (grijs kleur)

$Y' = R'=G'=B'$ (die grijswaarde)

$U=0, V=0$

probleem

nog geen tweede dominante as t.o.v. derde

YIQ kleurmodel [Y>I>Q]

kleurintensiteit of luminantie (Y) als prioritaire as
analoog aan YUV

situering

bepalen tweederangs prioritaire as
OF bepalen via 'principle component analysis'
techniek uit algebra
uit wolk punten dominante as bepalen
nadeel: complexe berekeningen
nadeel: hoofdassen afhankelijk per video!
OF zoeken naar iets simpel
proefondervindelijk

I en Q als geroteerde versie van U en V over 33°

precies 33° = proefondervindelijk!
zeer snel te berekenen (constante sin(33) en cos(33))
berekening

$$I = \underbrace{(R' - Y')}_{V} \cdot \cos(33^\circ) - \underbrace{(B' - Y')}_{U} \cdot \sin(33^\circ)$$

$$Q = \underbrace{(R' - Y')}_{V} \cdot \sin(33^\circ) + \underbrace{(B' - Y')}_{U} \cdot \cos(33^\circ)$$

YCbCr kleurmodel

nog een variatie op YUV

vaak gebruikt

JPEG beeldcompressie
MPEG videocompressie

HOOFDSTUK 3: Afbeeldingen

§1. Soorten afbeeldingen

1-bit images

1 bit per pixel

0 is zwart, 1 is wit

1-bit monochroom beeld of binair beeld

8-bit "gray-level" images

8 bits per pixel = 1 byte

0 is zwart, 255 is wit

kan ook andere kleur zijn

vb. donkerrood -> wit

ook 'grijswaarden'

24-bit color images

24 bits per pixel = 3 bytes

meestal RGB-waarden, elk 0 tot 255

(0,0,0) is zwart, (255,255,255) is wit

vaak 32-bit versies

alpha-waarde van 1 byte opslaan voor speciaal effect (vb. transparantie)

net 1 woord in 32-bit architectuur

24-bit heeft 17 miljoen kleuren

= meer dan te onderscheiden door menselijk oog

dus kansen voor compressie

8-bit color images

8 bits per pixel = 1 byte

256 kleuren uit het gamma

17 miljoen RGB-waarden mappen op 256 waarden (via LUT)

methode 1: 256 vaste kleuren

kleuren verspreid over spectrum

vaak combinaties van 00/33/66/99/CC/FF

methode 2: 256 'slimme' kleuren kiezen na beeldanalyse

adaptieve kleuren (zie verder)

vb. veel rode tinten bij vuur op foto

§2. Basisbegrippen

representatie vs. presentatie

presentatie

technologie in staat MM weer te geven
hardware (vb. scherm)
software (vb. browser)

representatie

manier van opslaan van MM
in binaire data

representatie vs. formaat

representatie

manier van opslaan van MM
in binaire data

formaat

meer dan representatie
ook interne structuur
ook compressie- en decompressiealgoritmen
enz.

bitmap

2D-array van pixelwaarden met daarin beeld-data
= domweg dumpen v/d pixelwaarden

resolutie

definitie 1 (optica, biologie, ...)

kleinst mogelijk te onderscheiden afstand

definitie 2 (multimedia)

aantal pixels horizontaal en verticaal
zegt niets over nauwkeurigheid!
'field of view' is ook bepalend

§3. Bit-planes

system

beschouw 8-bit waarde als 8 1-bit bit-planes

bit-plane groepeert bits met gelijke significantie i/d byte

planes

plane 0: 0 of 1 in totale som

plane 1: 0 of 2 in totale som

plane 3: 0 of 4 in totale som

...

plane 7: 0 of 128 in totale som

toepassingen

bepalen wat belangrijke info is en wat "ruis" is

versturen via netwerk + bandbreedte op

eerst alle meest significante bits

=> gans het beeld in mindere kwaliteit

pixel per pixel

=> half beeld in volle kwaliteit

§4. Dithering

probleemstelling

bij printen hebben we enkele donkere tinten (vb. zwart)
<-> geen grijs tinten, roze tinten, ...

doel: met 1 kleur alle "grijswaarden" van dat kleur kunnen printen

systeem

op elke pixel een dither matrix ($n \times n$)
praktijk: 16, 67, ... puntposities
elke pixel eigenlijk opsplitsen in n^2 subpixels
vb. 2x2 matrix:
[0 2]
[3 1]

pixelwaarde mappen op $[0, n^2]$ (door ze te delen door factor)
+ printen op subpixel \Leftrightarrow gemapte waarde > matrixwaarde op die subpixel

gevolgen

opslag afbeelding in printer is (tijdelijk) VEEL groter

grootte van dither matrix bepaald mee "resolutie" v/d printer
in DPI (dots per inch)
= aantal inkt-dots die printer op bepaalde oppervlakte kan printen

§5. Look-up tables (LUTs)

principe

mappen v/d 24-bit kleurenruimte (RGB) op kleinere ruimte (vb. 8-bit)

tabel: 8-bit pixelwaarde index -> 24-bit RGB-kleur waarde

toepassingen

- * verkleinen bestandsgrootte
- * eenvoudige afbeeldingen (vb. web-afbeeldingen)
 - <-> foto's met natuurlijke kleuren
- * animaties
 - pixelwaarden blijven hetzelfde
 - look-up tabellen (= de kleurenruimtes) veranderen

opstellen v/d LUT: naïeve methode

verdeel de RGB-ruimte in gelijke delen en kies centrale kleuren
+ menselijk oog gevoeliger voor R en G

praktisch:

R: 8-bit	-> 3-bit (0 tot 7)	$R' = R/(256/8) = R/32$
G: 8-bit	-> 3-bit (0 tot 7)	$G' = G/(256/8) = G/32$
B: 8-bit	-> 2-bit (0 tot 3)	$B' = B/(256/4) = B/64$

problemen

8-bit naar 3-bit geeft afrondingsfouten
maakt geen gebruik v/d info op de foto (vaste kleurentabel)

opstellen v/d LUT: median-cut algoritme (vereenvoudigde versie)

systeem

sorteer van ganse figuur rood waarden + neem mediaan
+ alles links ervan: $R_1 = 0$ en alles rechts ervan: $R_1 = 1$
beschouw deel van figuur met $R_1 = 0$ en $R_1 = 1$ apart!
+ voor deze delen apart hetzelfde 2 x waarde (G_1)
analoog voor blauw
analoog: RGBRGBRG

resultaat

256 indices voor (ongelijke) balken in de 3D RGB ruimte
neem van elke balk het centrale kleur (dat = waarde voor die index/balk)
voor elke pixel centrale kleur met minimale de euclidische afstand

optimalisaties

afrondingsfouten overdragen tussen pixels (compensatie!)
balk nemen die alle kleuren in figuur omvat en telkens langste as nemen

§6. Compressie en codering

compressie

speciale technieken om bestandsgrootte te reduceren

voordeel

bestand word kleiner

nadeel

complexe algoritmen (kosten tijd)

vb. bij live decoderen

groot gevaar voor “transcoderen”

steeds hercoderen voor nieuwe standaarden

=> 1x coderen (vb. bij opslag) is onvoldoende

‘lossless’ vs. ‘lossy’ compressie

lossless of verliesloos

+ geen verlies van data => origineel te construeren

- kleine compressiefactor (2 à 3 max.)

lossy of verlieshebbend

- verlies van data (onomkeerbaar!!)

+ grotere compressiefactor

opm: verlies van data is niet meteen verlies kwaliteit

vb. niet merkbaar

vb. ruis verwijderen (eigenlijk ‘betere’ kwaliteit!)

niet gecomprimeerd of gecomprimeerd archiveren

tot nu toe steeds gecomprimeerd (‘is nodig’)

<-> nu tegenbeweging + hevig debat

bij deep-archiving ongecomprimeerd opslaan

= indien niet meer aan te passen (vb. beeldarchieven VRT)

daardoor onafhankelijk van standaarden

tweede gecomprimeerde versie voor bevraging

steeds opbouwen vanuit ongecomprimeerde basisfile

naar de standaarden van moment/toepassing

§7. GIF-formaat (Graphics interchange format)

256-kleuren (8bit)

goed voor figuren of tekeningen
minder voor foto's met natuurlijke kleuren

4-pass interlacing

systeem

regels opslaan in volgende volgorde

0, 8, 16, ...

4, 12, 20, ...

2, 6, 10, ...

1, 3, 5, 7, ...

toepassing

grootte komt in buurt v/d bandbreedte
mindere kwaliteit + ganse figuur
<-> volle kwaliteit + halve figuur

4-pass

4 keer ganse figuur doorlopen

geavanceerde technieken

error concealing

= effect van verloren lijnen proberen beperken
vb. lege lijnen afleiden uit reeds ontvangen
aankomstzijde

error resilience

= reeds anticiperen op problemen bij verzenden
vb. spatiaalresolutie verkleinen => 'ruimte' om alles n* door te sturen
verzendzijde

bestandsstructuur

opbouw

- GIF signature (6 bits: GIF87a)
- screen descriptor
- global color map
- image descriptor]
- local color map] n keer herhaald
- raster area (pixelwaarden)]
- GIF terminator

subbeelden

- elk deel kan eigen LUT definiëren
 - <-> moet niet!
- voordeel: betere (kleur)kwaliteit
- nadeel: overheid aan LUT's => minder compressie
- vb. foto zee + strand in twee knippen

screen descriptor

bevat algemene info over het (globaal) beeld

hoogte en breedte

achtergrondkleur

eventueel

indexnummer in globale LUT

eigen (globale) LUT gedefinieerd of default LUT gebruiken (m)

kleurresolutie (cr)

standaard 8-bit

te verkleinen!

bits / pixel

standaard 8-bit

te verkleinen!

image descriptor

bevat algemene info over het subbeeld

locatie subbeeld (X, Y), hoogte en breedte

lokale LUT gebruiken of globale gebruiken

interlacing of niet

bits / pixel

enkel indien lokale LUT gebruikt!

§8. JPEG-formaat (Joint Photographic Experts Group)

2 varianten

verliesloze variant

bijna niet gebruikt

verlieshebbende variant

meest gebruikte standaard

eigenschappen

gebruikt beperkingen van menselijk oog

spatiale frequentie (1/m)

“hoe snel verandert een eigenschap i.f.v. afstand”

bepaald door Fouriertransformatie van $f(x, y)$

berekent frequentieinhoud

hoge spatiale frequenties

vooral aan randen

komen weinig voor over het algemeen (meestal geleidelijker)

menselijk oog veel gevoeliger voor afwijkingen in LAGE spatiale frequentieinhoud

=> JPEG gooit hoge grotendeels weg

<-> indien te veel: zichtbare fouten, vooral aan randen (“artefacten”)

compressiefactor

gebruiker mag zelf compressiefactor/kwaliteit kiezen!

§9. PNG-formaat (Portable Network Graphics)

pattentvrij

<-> GIF

tot 48 bits per pixel

<-> GIF: maximum 8-bit!

transparantie ondersteund (alpha-waarde)

<-> JPEG

compressie

enkel lossless

minder compressie dan JPEG

zeker bij foto's

geen artefacten aan randen

<-> JPEG

toepassingen

figuren met scherpe randen

webfiguren

letters, egale vlakken, rechte lijnen, ...

progressief weergeven

per 8*8 blok v/h beeld (= per 64 pixels)

7 passes en telkens enkele pixels getoond

§10. TIFF-formaat (Tagged Image File Format)

doel

metadata en MM-data samenbrengen

systeem

container/wrapper van data en metadata

1 format signifier tag

bepaald formaat, compressietechnieken, ...

vb. gif, jpeg, png, ...

andere infotags

bevatten metadata

EXIF (Exchangeable Image File)

analoog aan TIFF

specifiek voor digitale camera's

vb. metadata: sportfoto, landschap, ...

HOOFDSTUK 4: Video

§1. Videosignalen

component video [analoog]

principe

3 analoge signalen fysiek gescheiden
splitsing in RGB, YCbCr (digitaal) of YPbPr (analoog), ...

eigenschappen

geen crosstalk (interferentie)

=> beste kleurreproductie

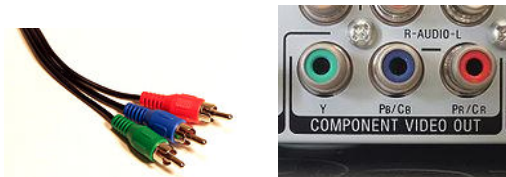
duurder

=> meestal in professionele omgeving (vb. studios)

gebruikt meer bandbreedte

goede synchronisatie nodig

YPbPr analoge kabels (3x RCA connector) en YPbPr/YCbCr video-systeem



composite video [analoog]

principe

3 analoge signalen op 1 draad
signalen moduleren en demoduleren

eigenschappen

wel crosstalk (interferentie)

=> kans op lichte kleurmisvorming

systeem: zie analoge video

NTSC, PAL, SECAM werken hierop

composite video analoge kabel (1x RCA connector)



S-video (seperated- of (foutief) super-video) [analoog]

principe

3 analoge signalen op 2 draden

signalen moduleren en demoduleren

1 voor luminantie en 1 voor chrominantie (2 signalen)

luminantie apart, want belangrijkste

eigenschappen

compromis tussen composite en component

geen crosstalk tussen kleur en luminantie

belangrijkste

wel nog crosstalk tussen kleursignalen onderling

minder belangrijk

NTSC, PAL, SECAM werken hierop

S-video analoge kabel (Mini-DIN connector)



(4 kabels: 2x signaal en 2x aarding)

VGA (Video Graphics Array) [analoog]

eigenschappen

gescheiden kleuren : RGB

voor computerscherm

VGA analoge kabel (D-subminiature connector)



DVI (Digital Visual Interface) [analoog en digitaal]

eigenschappen

overgang van analoog naar digitaal
heeft aansluitingen voor beide
digitale dual link mogelijk voor hoge resoluties
voor computerscherm

types

DVI-I (integrated): beide
single link en dual link

DVI-D: enkel digitaal
single link en dual link

DVI-A: enkel analoog

DVI kabel



HDMI (High Definition Multimedia Interface) [digitaal]

eigenschappen

audio- en videosignalen
in ongecomprimeerde digitale vorm
voor vele compatibele digitale bronnen (dvd speler, versterker, scherm, PC, ...)
elektrisch compatibel met DVI
geen signaalconversie bij gebruik adapter => geen dataverlies

HDMI kabel



DisplayPort [digitaal]

eigenschappen

audio- en videosignalen
niet compatibel met HDMI of DVI
royalty free (<-> HDMI)

DisplayPort kabel



§2. Analoge video

progressief en interlaced scanning

progressief scannen

elk tijdsinterval volledig frame inladen (rij per rij)

interlaced scannen (TV en sommige andere apparaten)

eerst oneven scannen, dan even lijnen

=> 2 fields die samen 1 frame maken

rij per rij, lichtjes dalend en van boven naar onder

horizontal retrace (teruggaan van links naar rechts)

vertical retrace (teruggaan naar boven)

2x tijdsverlies

later pauzes gebruikt voor teletext!

nadeel

2 opeenvolgende lijnen hebben steeds Δt verschil

lijnen 0,2,4,... komen overeen en lijnen 1,3,5,... komen overeen

gevolg: niet gebruiken indien

veel beweging

kleine variaties

(vb. rimpeling in de zee, anders geen golvend effect meer)

effect: soort flikkering + onscherpte

moduleren en demoduleren

vb. NTSC

PAL en SECAM analoog

kleurmodel: YIQ

modulatie: kwadratuur amplitude modulatie (QAM)

1 signaal in-phase (I) met draaggolf

1 signaal op kwadratuur draaggolf (Q) = 90° out of phase met draaggolf

signaal filteren

op frequentie

3 soorten

laagdoorlaatfilter (low-pass filter)

hoogdoorlaatfilter (high-pass filter)

banddoorlaatfilter (band-pass filter)

vanaf 'cut-off' frequentie

encoderen

stap 1: I en Q moduleren in 1 chroma signaal C

$$C = I \cdot \cos(F_{sc} t) + Q \cdot \sin(F_{sc} t)$$

met F_{sc} drager frequentie (grootteorde enkele Mhz)

stap2: C en Y samenvoegen

$$\text{compositie} = Y + C$$

dit wordt als signaal verstuurd + bij aankomst gedemoduleerd

decoderen

stap 1: Y en C scheiden

frequentie Y: 0 Mhz

frequentie C: F_{sc} (enkele Mhz)

low-pass filter toepassen = Y afsplitsen

stap2: I en Q demoduleren uit C

I demoduleren:

stap a:

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

stap b:

frequentie I: 0 Mhz en $2F_{sc}$ (enkele Mhz)

frequentie Q: $2F_{sc}$ (enkele Mhz)

low-pass filter toepassen = I afsplitsen

Q demoduleren:

stap a:

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

stap b:

frequentie I: $2F_{sc}$ (enkele Mhz)

frequentie Q: 0 Mhz en $2F_{sc}$ (enkele Mhz)

low-pass filter toepassen = Q afsplitsen

vergelijking van TV systemen

	NTSC	PAL	SECAM
kleurmodel	YIQ	YUV	YUV
Frame Rate (Hz)	30	25	25
	reden: VS: 60 Hz <-> EU: 50 Hz netspanning + signalen makkelijk synchroon met halve bronfrequentie opm: field rate (zie interlacing) wel 60 en 50 Hz		
# scan Lines	525	625	625
	opm: bij NTSC uitbreiden adhv interpolatie		
Channel Width (MHz)	6.0	8.0	8.0
Bandwidth allocation (Mhz)			
Y	4.2	5.5	6.0
I/U	1.6	1.8	2.0
Q/V	0.6	1.8	2.0
	opm: YUV: Y < U = V <-> YIQ: Y < I < Q opm: som is groter dan totale breedte wegens overlapping componenten reden: bandbreedte = zeer duur!		

§3. Digitale video

digitaal vs. analoog

- flexibel stockeren op digitale apparaten of in computergeheugen
 - alomtegenwoordig
- flexibele niet-lineaire toegang
- geen degradatie bij veelvuldig gebruik
- makkelijkere encryptie
 - digitale encryptie staat zeer ver
- makkelijkere compressie
 - geavanceerde compressiestandaarden
 - vb. MPEG-2, MPEG-4, ...
- toleranter t.o.v. ruis
- makkelijker te bewerken
 - vb. effecten invoegen, bewerken, monteren, ...
 - vb. herschalen voor lage bandbreedtes
- goedkope hardware en software

chroma subsampling

4:4:4

- geen subsampling (volledige info)
- toepassing: digital cinema
- toepassing: opslaan originele video (diep archiving)
- te groot voor alledaagse bandbreedte + opslag

4:2:2

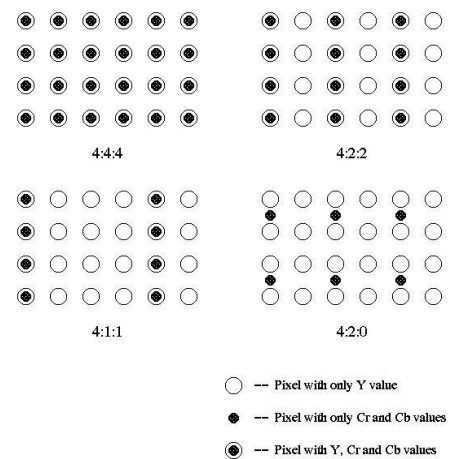
- horizontale subsampling
 - maar om de 2 kolommen chroma-waarden opslaan
- geen verticale subsampling

4:1:1

- horizontale subsampling
 - maar om de 4 kolommen chroma-waarden opslaan
- geen verticale subsampling

4:2:0

- horizontale subsampling van chrominantie
 - maar om de 2 kolommen chroma-waarden opslaan
- verticale subsampling
 - chroma-waarden als interpolatie van 2 pixels
- toepassing: algemeen gebruik (zeer belangrijk)



digitale video specificaties

CCIR-601

situering

doel: interlaced analoge video signalen digitaal encoderen
moeder van alle standaarden
door Consultative Committee for International Radio (CCIR)
later beheerd door Video Coding Experts Group (VCER)
< International Telecommunications Union (ITU)
2 verschillende versies (voor NTSC en PAL)

kenmerken

	NTSC	PAL
luminantie:	720x480	720x576
chrominantie:	360x480	360x576
subsampling:	4:2:2	4:2:2
aspect ratio:	4:3	4:3
frame rate:	60 Hz	50 Hz
interlaced:	yes	yes

CIF en QCIF

situering

doel: standaardformaat voor lage bandbreedtes (vb. video conferencing)
door International Telegraph and Telephone Consultative Committee (CCITT)
< International Telecommunications Union (ITU)
compromis tussen NTSC (framerate) en PAL (half aantal active Lines)

kenmerken

	CIF	QCIF
luminantie:	352x288	176x144
chrominantie:	176x144	88x72
subsampling:	4:2:0	4:2:0
aspect ratio:	4:3	4:3
frame rate:	30 Hz	30 Hz
interlaced:	no	no

HDTV specificaties

situering

doel: grotere en bredere schermen

kenmerken

meestal breedbeeld: 16:9 <-> 4:3
groter aantal pixels: 1280x720 of 1920x1080
verschillende frame/field rates, zowel progressief als interlaced

§4. Video-overgangen

situatie

signaleren scène-overgang

overgang van ene fysieke video naar andere

kunnen impliciete semantische waarde hebben

cut

abrupte overgang

laatste frame van A met meteen eerste frame van B erachter

zeer eenvoudig en veel gebruikt

wipe

pixels in steeds groeiende regio vervangen door die van B

vb. L->R, R->L, wijzers v/e klok, vanuit centrum, ...

eigenlijk speciaal geval van Dither dissolve (type II)

met specifieke geometrische eigenschappen

dissolve

elke pixel wordt "over time" vervangen van A naar B

2 types

Cross Dissolve (type I)

aandeel A en B gelijk over alle pixels

elke pixel gewogen som van pixelwaarden van A en B

+ aandeel B stijgt

$$D_t = (1 - \alpha(t)) * A + \alpha(t) * B$$

D_t , A, B vectoren van RGB-ruimte

$\alpha(t)$ transformatiefunctie

met $\alpha(0) = 0$ en $\alpha(t_{\max}) = 1$

meestal lineair: $\alpha(t) = k*t$

Dither Dissolve (type II)

aandeel A en B niet gelijk over alle pixels

elke pixel is ofwel A-waarde, ofwel B-waarde
+ steeds meer pixels zijn B-waarde

abrupte overgang op pixelniveau

welke pixels eerst is in te stellen

fade-in en fade-out

speciale cross-dissolve (type I)

video A of B is egaal zwart of egaal wit

vele andere

vb. slide of push, ...

HOOFDSTUK 5: Digitale audio

§1. Geluid

geluidsgolf

wat

drukwisselingen in bepaald medium (geen luchtledige)
longitudinaal (in richting van energie)

eigenschappen

golflengte (en dus frequentie)

bepaald toonhoogte

toonhoogte vastgelegd (vb. $\lambda = x$ Hz)

octaven: steeds frequentie $\times 2$ (vb. $\lambda' = 2x$ Hz)

amplitude

bepaald sterkte van geluidsgolf

maximale druk in een punt

relatief t.o.v. referentiedruk (0 dB zie verder)

hebben typische golfeigenschappen

reflectie (= weerkaatsten op oppervlakten)

diffRACTIE (= buigen rond object)

van richting veranderen bij overgang tussen mediums

analoge geluidsgolf

weergave als goniometrische functie \leftrightarrow eigenlijk longitudinale golf

druk in een punt meten in functie van de tijd

$f(t)$

gevolg

geluid gedefinieerd adhv 1 variabele (A), afhankelijke van 1 variabele (t)

\leftrightarrow beelden: afhankelijk van 2 variabelen (x,y)

\leftrightarrow video: afhankelijk van 3 variabelen (x,y,t)

geluid vs. spanning, vermogen, ...

geluidssignaal als spanning V

grafiek van geluidsgolf gewoon interpreteren als grafiek van spanning

dit is wat gebeurt in een microfoon

daar omzetting van druksignaal naar elektrisch signaal

vermogen V^2 van een geluidssignaal

reden: spanning fluctueert rond 0

=> op lange termijn spanning altijd 0

daarom kwadrateren

=> dan maat voor intensiteit

speciale geluidssignalen

goed geluidssignaal

geluidsgolf als som van sinussen (en cosinussen) met termen:
gehele veelvouden van de grondfrequentie als frequentie = 'harmonieken'
elk met hun frequentiecomponenten (coëfficiënten bij die term)
complexe geluidsgolf als som van de harmonieken

bandgelimiteerd ('braaf') geluidssignaal

fouriergetransformeerde heeft beperkte bandbreedte
dus frequenties van signaal tussen f_1 en f_2

digitale audio vs. digitale video

digitale video

zowel opnemen als weergeven kan digitaal én analoog
vb. kijken naar discrete pixels
volledig digitaal systeem mogelijk

digitale audio

zowel opnemen als weergeven steeds analoog
vb. uit digitale of analoge waarden een analoge geluidsgolf maken
enkel intern digitaal opslaan/bewerken/verwerken
<-> I/O altijd analoog

decibel (dB)

wat?

geen eenheid!
logaritmische schaal om verhoudingen aan te duiden
vooral van vermogens
meestal één v/d twee als constante referentiewaarde

eigenschappen

0 dB is verhouding 1 (dus gelijkheid, dus de referentiewaarde zelf)
+10 dB -> verhouding 10 (dus 10x groter dan referentiewaarde)
+20 dB -> verhouding 100 (dus 100x groter dan de referentiewaarde)

formules

$$L = 10 \cdot \log_{10} \frac{I_1}{I_0} \text{ dB} \quad \text{of m.a.w.} \quad \frac{I_1}{I_0} = 10^{L/10}$$

met I_1 een (vermogens)grootte en met I_0 de referentiewaarde

veelgebruikte toepassing: geluidsniveau

verhouding van de geluidsintensiteit t.o.v. geluidsdrempel (net hoorbaar geluid)

$$L = 10 \cdot \log_{10} \frac{J}{J_0} \text{ dB} \quad \text{of} \quad L = 20 \cdot \log_{10} \frac{p}{p_0}$$

met J de geluidsintensiteit en p de geluidsdruk ($J = p^2$)

§2. Digitaliseren van geluid

digitaliseren van de geluidsgolf

principe

- beide assen digitaliseren of discretiseren door ze te samplen
 - tijd en amplitude
- 'samplen' of bemonsteren
 - hoeveelheid meten in bepaalde intervallen
 - meestal van gelijke grootte
 - meestal gebruikt voor tijd
 - <-> kan voor alle grootheden
 - ook gewoon 'samplen' genoemd
 - sampling frequentie (Hz) geeft grootte intervallen aan

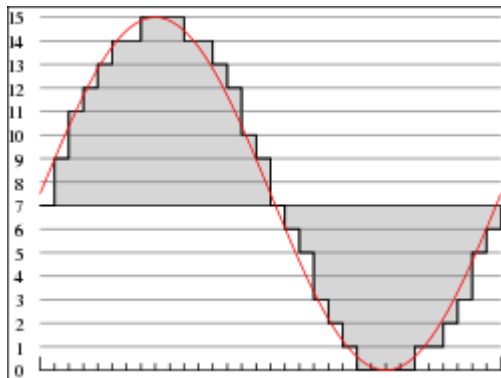
tijd-as

- 'samplen'
 - doel: maar op bepaalde tijdstippen waarde meten <-> alle tijdstippen

amplitude-as

- 'kwantiseren'
 - doel: maar bepaalde kleine 'categorieën' amplitudes overhouden <-> alle waarden

gevolg



parameters

- welke sampling rate (tijdas) [1]
- welke nauwkeurigheid voor kwantisatie en uniform verdeeld? [2]

bepalen sampling rate [1]

Nyquist stelling

gegeven een goed en bandgelimiteerd audiosignaal
volledig analog signaal te reconstrueren adhv discreet aantal tijdstippen
ALS sampling rate $\geq 2 \times$ intervalbreedte van frequenties
of sampling rate $\geq 2 \times (f_2 - f_1)$

praktisch: oogpunt signaal (of sampler maker)

gegeven een bepaald bandgelimiteerd signaal
heeft bepaalde bandbreedte
welke sampler mijn signaal volledig reconstrueren
 \Rightarrow al diegene met als sampling rate minimaal 2x mijn bandbreedte
die grens = Niquist rate
vb. bandbreedte signaal 50Hz \Rightarrow maak sampler van (minimum) 100Hz

praktisch: oogpunt sampler (of sampler gebruiker)

gegeven een sampler
heeft bepaalde sampling rate
welke signalen kan mijn sampler volledig reconstrueren
 \Rightarrow al diegene met als bandbreedte maximaal $\frac{1}{2}$ x mijn sampling rate
die grens = Niquist frequency
vb. sampling rate 10kHz \Rightarrow alle signalen met bandbreedte onder 5kHz OK

SNR: Signal to Noise Ratio (Signaal-ruisverhouding)

signaal vs. ruis

signaal = alles wat we willen horen
ruis = al de rest
doel: signaal \ggg ruis

Signal to Noise Ratio

verhouding ertussen (in dB)
maat voor kwaliteit van signaal (hoe hoger, hoe beter)
berekening

$$SNR = 10 \cdot \log_{10} \frac{V_{signal}^2}{V_{noise}^2} = 20 \cdot \log_{10} \frac{V_{signal}}{V_{noise}}$$

met V de spanning (of druk)

met V^2 het vermogen of de intensiteit van het signaal

SQNR: Signal to Quantization Noise Ratio

kwantisatie geeft afrondingsfout

dus ook ruis (want iets dat we niet willen horen)

afhankelijk van aantal bits N om waarden op te slaan

want maximaal 2^N verschillende waarden

quantisation noise (kwantisatieruis)

= verschil tussen echte (analoge) waarde op een (gesampled) tijdstip

en waarde van dichtste intervalwaarde

maximaal half interval

Signal to Quantization Noise Ratio

verhouding tussen signaal en kwantisatieruis (in dB)

maat voor kwaliteit van kwantisatie (hoe hoger, hoe beter)

berekening

$$SQNR = 10 \cdot \log_{10} \frac{V_{signal}^2}{V_{quan_noise}^2} = 20 \cdot \log_{10} \frac{V_{signal}}{V_{quan_noise}}$$

met V de spanning (of druk)

met V^2 het vermogen of de intensiteit van het signaal

lineaire en niet-lineaire kwantisatie [2]

situering

lineaire kwantisatie is zeer dom!

want bij stille volumes quantization noise veel groter waargenomen

zie SQNR met constante $V_{quan_noise}^2$

dan kleinere verhouding bij kleine signaalintensiteit

=> daar fijner kwantiseren

principe: gelijke fouten moeten overeenkomen met gelijke perceptie (respons)

signaalruimte en responsruimte

verband tussen signaalruimte en responsruimte

$$dr = k \cdot \frac{1}{s} ds$$

met r respons en s absolute prikkel (constante k)

en geïntegreerd

$$r = k \cdot \ln s + C$$

of anders gesteld

$$r = k \cdot \ln s/s_0$$

met s_0 laagste prikkel die respons veroorzaakt ($s = s_0 \rightarrow r = 0$)

systeem

eerst signaal omzetten van signaalruimte naar responsruimte

via A-law of μ -law

daar uniform kwantiseren!

audio filtering

inputkant

doel: zorgen dat elk signaal zeker kan gesampled worden

band-pass filter toepassen om bandbreedte te beperken tot correct interval
(zie Nyquist frequentie)

outputkant

probleem: discontinue functie bij Fouriertransformatie oneindig grote frequenties
wiskundig altijd zo!

< digitale opslag

low-pass filter toepassen om alleen lagere frequenties door te laten

specificaties van verschillende audio applicaties

specificaties

	Telefoon	FM Radio	CD	DVD
sample rate (kHz)	8	22	44	192 (max)
bits per sample	8	16	16	24 (max)
# channels	1	2	2	6 (max)
data rate (kB/sec)	8	88	176	1200 (max)
frequentieband (kHz)	0.2-3.4	0.02-11	0.005-20	0-96 (max)

opmerkingen

data rate = samples per sec x bits per sample x #channels

vb. FM Radio: $22000 \text{ 1/s} \times 16 \text{ bits} \times 2 = 704000 \text{ bps} = 88000 \text{ B/s} = 88 \text{ kB/s}$

§3. Coderen van geluid

algemeen

opnemen van geluid

1. samplen + niet-uniforme kwantisatie
2. transformatie

input data omzetten naar andere representatie

doel: makkelijker en betere compressie

2. coderen

codewoord aan elk outputniveau toekennen

2 soorten: vaste lengte of variabele lengte (vb. Huffman codering)

lossy- en lossless audio coding

enkel over extra (compressie)verlies bij coderen <-> steeds verlies bij digitaliseren

PCM: Pulse Code Modulation

gewoon waarden opslaan (geen transformatie)

wat we in deel 2 gebruikt hebben

differentiële codering

uitgangspunt: verschil tussen waarde en vorige meestal klein

indien signaal beetje consistent in de tijd

= temporele redundantie

principe: sla verschil op met vorige ipv steeds absolute waarden

gevolg 1: verschilwaarden kleiner dan absolute waarden

=> kleinere representatie

gevolg 2: meer gepiekt histogram met maximum rond 0

=> meer voordeel bij variabele lengte codering

Lossless Predictive Coding

principe

voorspelt waarden op een vaste wijze (adhv vorige waarden)

slaat fouten tegenover voorspelling op

systeem

inputwaarden f_n komen binnen

voorspelde waarde $\hat{f}_n = F(f_{n-1}, f_{n-2}, f_{n-3}, \dots)$ bepalen

$$\hat{f}_n = f_{n-1} \quad (\text{vb. gelijk aan vorige, eenvoudigst})$$

$$\hat{f}_n = \sum_{k=1}^{2 \text{ tot } 4} a_{n-k} f_{n-k} \quad (\text{vb. lineaire voorspeller in functie van enkele vorige})$$

fout e_n tussen echte en voorspelde waarde berekenen en encoderen

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

latere verliesloze reconstructie mogelijk

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

ook als geen temporele redundantie of slechte schatter (enkel minder compressie)

DPCM: Differential PCM

principe

we passen lossless predictive coding toe

=> we krijgen reeks verschillen

deze waarden gaan we ook nog eens kwantiseren!

DUS nog eens extra verlies invoeren!

systeem

inputwaarden f_n komen binnen

voorspelde waarde $\hat{f}_n = F(\tilde{f}_{n-1}, \tilde{f}_{n-2}, \tilde{f}_{n-3}, \dots)$ bepalen

fouten e_n berekenen

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

kwantiseren naar foutwaarden \tilde{e}_n en coderen

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

latere verliesloze reconstructie NIET mogelijk

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

dus verlieshebbende codering

<-> nog andere compressiemethoden mogelijk

DM: Delta Modulation

principe

vereenvoudigde versie van DPCM (zeer snel)

kwantiseren naar twee waarden \tilde{e}_n

voor stijgend en dalend

1 bit voor foutwaarde

systeem

gelijkheid voorspellen (met aangepaste lossy versie!!)

$$\hat{f}_n = \tilde{f}_{n-1}$$

fouten e_n berekenen

$$e_n = f_n - \hat{f}_n = f_n - \tilde{f}_{n-1}$$

kwantiseren naar foutwaarden \tilde{e}_n en coderen

$\tilde{e}_n = +k$ als $e_n > 0$ en $\tilde{e}_n = -k$ anders
met k een constante

latere verliesloze reconstructie NIET mogelijk

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

voorbeeld

$f_i = \{10, 11, 13, 15\}$ en $k = 4$

$\tilde{f}_i = \{10, 14, 10, 14\}$ dus nog relatief dicht bij echte waarden

slechte situaties

- bijna constant signaal bij relatief grote k -> geblokt
 - snel stijgende signalen (veel sneller dan k) -> kan niet volgen
- oplossing: zeer veel samplen (vele keren Nyquist rate)

adaptive DM

uitgangspunt:

- k laten aanpassen aan stijlheid van de curve
- zo slechte gevallen proberen wegwerken

gevolg:

- kleine fluctuaties ($\pm ct$) -> kleine k -> geen grote blokken
- grote fluctuaties (stijl) -> grote k -> sneller meekunnen