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 ook

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

systeem

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 grijstinten, roze tinten, …

doel: met 1 kleur alle “grijswaarden” van dat kleur kunnen printen

systeem

op elke pixel een dither matrix (n*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 <=> 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: R1 = 0 en alles rechts ervan: R1 = 1
    beschouw deel van figuur met R1 = 0 en R1 = 1 apart!
        + voor deze delen apart hetzelfde 2 x waarde (G1)
    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 Photografic 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 veranderd 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 (Exchangeble Image File)

analoog aan TIFF

specifiek voor digitale camera's

vb. metadata: sportfoto, landschap, …