INF 1040
Farger og fargerom

- **Temaer i dag :**
  1. Fargesyn og deteksjon av farge
  2. Digitalisering av fargebilder
  3. Fargerom – og overganger mellom dem
  4. Fremvisning og utskrift av fargebilder
  5. Fargetabeller
  6. Noen filformater for digitale bilder
  7. Pseudo-farger og falske farger

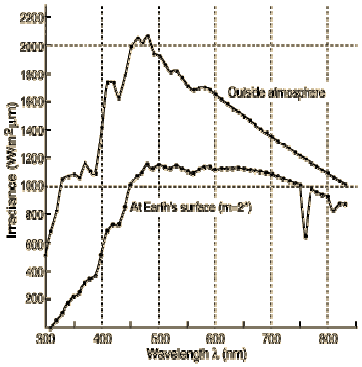
- **Pensumlitteratur:** Læreboka, kapittel 15.

Motivasjon

- Vi kan skille mellom tusenvis av fargenyanser
- Farger gjør det lett å skille mellom objekter
  - Både visuelt
  - Og ved digital bildeanalyse
- Vi må
  - Vite hvilket fargerom vi skal bruke til forskjellige oppgaver
  - Kunne transformere fra ett fargerom til et annet
  - Kunne lagre fargebilder rasjonelt og kompakt
  - Kjenne teknikker for utskrift av fargebilder

Fargen på lyset

- Lyset fra Sola kan best beskrives ved strålingen fra et "svart legeme" med T=5780 K.
- I jordatmosfæren absorberes mye stråling i UV og IR – det meste av enkle molekyler.
- Lyset som slipper ned til bakken avhenger av tid og sted.
- Figuren viser irradiansen i W/m²/μm
  - på toppen av atmosfæren
  - og når sola er 30 grader over horisonten.



Fargen på et objekt

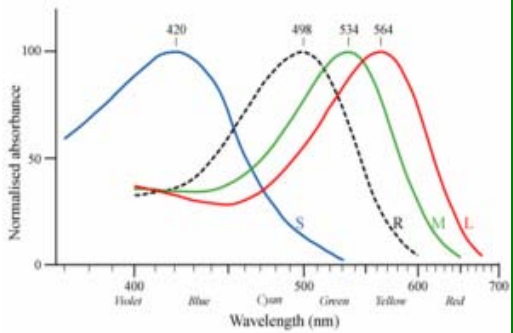
- Objektets farge bestemmes av det lyset som *reflekteres*.
- Objektets farge blir derfor avhengig av
  - Spektral-fordelingen til lyset som faller på objektet
  - Spektralfordelingen til refleksjonen
- Refleksjonsegenskapene bestemmes av
  - Kjemiske pigmenter i overflaten
  - Fysiske overflate-strukturer
- Strukturfarger finnes i naturen
  - kommer snart i en "nano-butikk" nær deg ?

Fargesyn

- Retina er følsom for lys mellom 350 og 760 nanometer (nm = 10^{-9} m)
- Fiolett: 400 - 446 nm
- Blå: 446 - 500 nm
- Grønn: 500 - 578 nm
- Gul: 578 - 592 nm
- Oransje: 592 - 620 nm
- Rød: 620 - 700 nm
- Ved sterk infrarød stråling kan vi oppfatte stråling helt opp til 1000 nm som lys, selv om dette er varmestråling.
- Simultane forskjeller ned til 1 nm i blå-grønt og gult kan sees, mens forskjellen må være minst 10 nm i dyp rødt og fiolett.
- Dette betyr at vi kan skille mellom ca 100 rene farger.

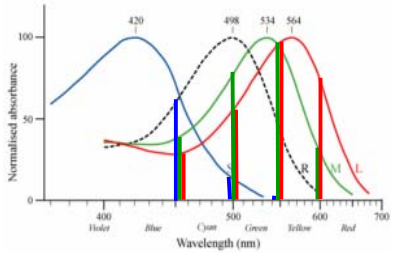
Tre-farge syn

- Tre typer fargefølsomme tapper i retina:
  - S - rundt 420 nm, (2%). Dette er de mest sensitive tappene.
  - L - rundt 564 nm, (65%).
  - M - rundt 534 nm, (33%).
- Tappene analyserer lyset, og finner den dominerende bølgelengden.
- Stavene (R) gir
  - gråtone-syn
  - er ikke sensitive for rødt lys



Tristimulus-verdier

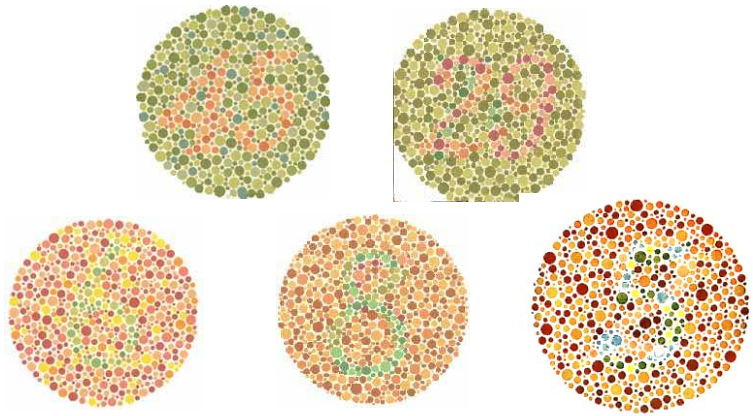
- Fargen reduseres til tre verdier **tristimulus-verdier**
- Mengden av alle slike mulige verdier utgjør vårt perseptuelle fargerom
- Det er noen kombinasjoner av stimuli som ikke er mulige
  - Vi kan ikke stimulere M-tappene uten å få noe respons fra S og / eller L tappene samtidig



- En liten andel har nedsatt fargesyn eller er "fargeblinde"
  - Oppfatter farger ved hjelp av to komponenter
  - Grønnblindhet mer utbredt enn rødblindhet

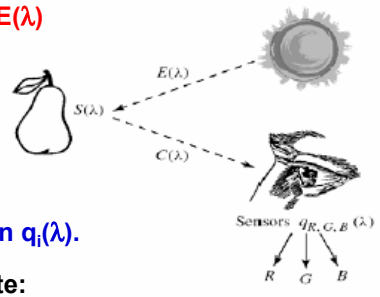
Tester for fargeblindhet

- Med normalt syn ser du tallene 45, 29, 6, 8 og 5
- Ved rød-grønn fargeblindhet ser du tallet 2 nederst til høyre.



Tre integraler gir RGB

- Lys fra en kilde med spektralfordeling E(λ)
  - treffer et objekt med spektral refleksjonsfunksjon S(λ).
  - Reflektert lys detekteres av tre typer tapper med spektral lysfølsomhetsfunksjon $q_i(\lambda)$.



- Tre analoge signaler kommer ut av dette:

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

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

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

RGB primærfarger

- Commision Internationale de l'Eclairage, (CIE) (The International Commision of Illumination)

har definert primærfargene:

- Blå: 435.8 nm
- Grønn: 546.1 nm
- Rød: 700.0 nm

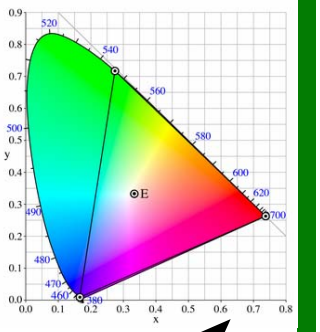
Kromatisitet

- X,Y,Z gir mengden av R,G og B
  - En farge spesifises med trikromatiske koeffisienter:
  - Ser at x+y+z=1
- Den ene parameteren er valgt ekvivalent med luminositet.
- De to andre gir fargen.
- Alle farger som har samme intensitet kan da gjengis i et 2-D kromatisitetsdiagram

$$x = \frac{X}{X+Y+Z}$$

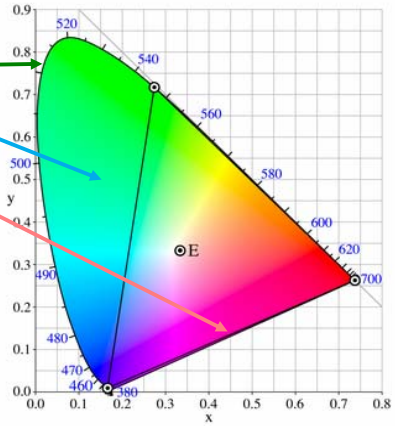
$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$



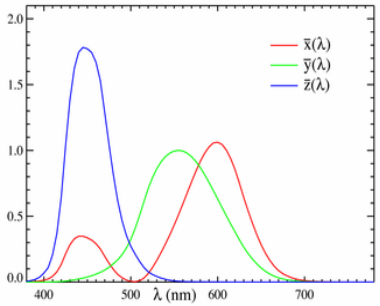
CIE kromatisitetsdiagram

- Mettede farger langs "hestesko"
  - Mindre mettede inn mot midten.
  - Pastellfarger nede til høyre.
- Alle blandinger av N farger ligger innenfor N-kant med de N fargene som hjørner.
  - Alle mulige RGB-farger ligger innenfor markert trekant.



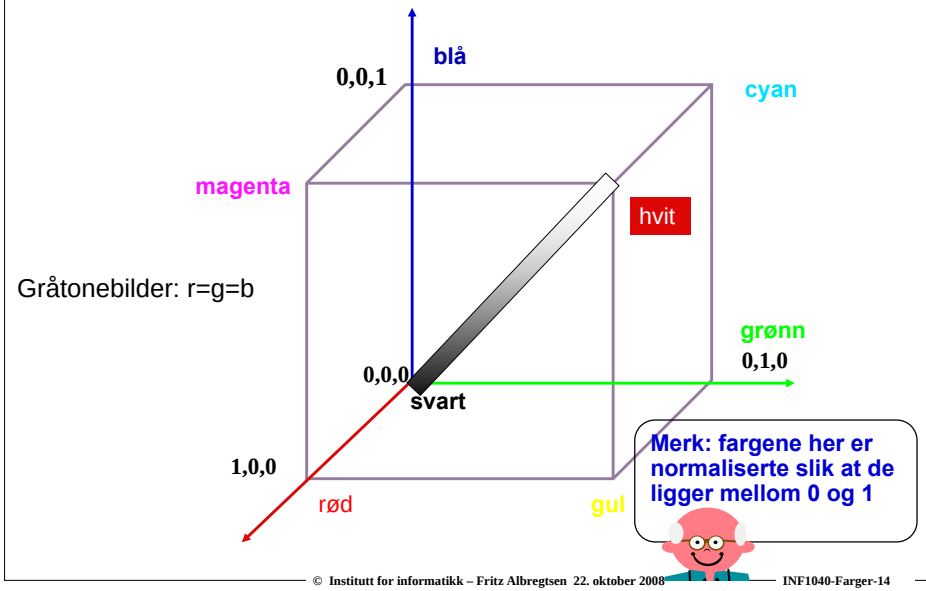
RGB detektorer

- Lysfølsomhet for RGB-detektorer:
- La spektralfordelingen til lyset inn i kamera være C(λ).
- Tre tall bestemmer fargens posisjon i RGB-rommet:



$$c_i = \int C(\lambda)\, a_i(\lambda)\, d\lambda, \quad i = r, g, b$$

RGB-kuben



Farger og fargerom

- Fargekamera:
  - Vi legger et rutenett over bildet
  - For hvert piksel måles lysintensitet i tre separate bånd i det elektromagnetiske spekteret.
- Husk at for hvert bånd (R, G, B) skal vi :
  1. beregne gjennomsnittsverdien i hver rute
  2. skalere slik at den passer innenfor det tall-området vi skal bruke
  3. kvantiserer verdiene til nærmeste heltalls verdi i tall-området
- RGB -bilder kan også genereres med et monokromt kamera ved å bruke tre filtre etter hverandre som bare slipper gjennom henholdsvis røde, grønne og blå bølgelengder.
  - Hvis vi gjør dette, må kameraet stå helt stille !

Mer om farger

- r,g,b lagres ofte ved 3 · 8 biter = 24 biter

0	63	127	191	255
0	63	127	191	255
0	63	127	191	255
0	63	127	191	255
0	63	127	191	255

0	63	127	191	255
0	63	127	191	255
0	63	127	191	255
0	63	127	191	255
0	63	127	191	255

255	255	255	255	255
191	191	191	191	191
127	127	127	127	127
63	63	63	63	63
0	0	0	0	0

- Vi sier at bildet har 3 bånd:
  - første bånd representerer intensiteten til rødt lys
  - andre bånd: intensiteten til grønt lys
  - tredje bånd: intensiteten til blått lys.
- Fargen til et piksel representeres ved talltrippelet (r,g,b)
- Hvilket snitt gjennom RGB-kuben er dette?

0,0,255	63,0,255	127,127,255	191,191,255	255,255,255
0,0,191	63,63,191	127,127,191	191,191,191	255,255,191
0,0,127	63,63,127	127,127,127	191,191,127	255,255,127
0,0,63	63,63,63	127,127,63	191,191,63	255,255,63
0,0,0	63,63,0	127,127,0	191,191,0	255,255,0

Eksempel RGB-bilde

Bånd 1: R

Bånd 2: G

Bånd 3: B

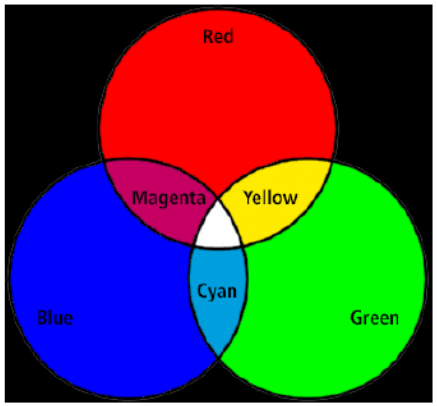
RGB-bildet vist på skjerm

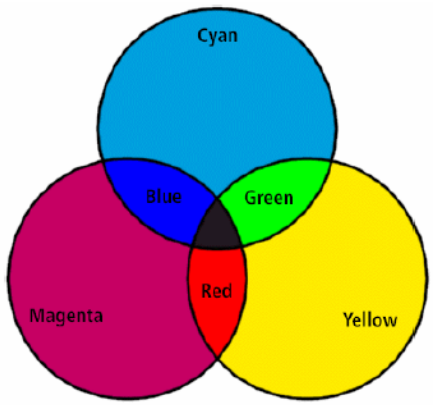
CMYK-fargemodellen

- CMYK- modellen er subtraktiv (start med hvitt, trekk fra farger).
- Alternativ til r,g,b som basisfarger er cyan, magenta, yellow (CMY-modellen) .
  - C = 1 - R eller 255 - R hvis 8-biters ikke-normaliserte bilder
  - M = 1 - G 255 - G
  - Y = 1 - B 255 - B
- RGB er vanlig på display, men CMYK er vanlig på fargeprintere (K er ekstra komponent for svart).
  - Egen komponent for svart
    - fordi full verdi av C, M og Y gir mørk brunt og ikke svart.
  - På ulike printere ser også rene farger ulike ut når de skrives ut.

RGB og CMY

- RGB og CMY er i prinsippet sekundærfarger for hverandre.





YIQ

- NTSC er standard for TV og video i USA. Bruker fargesystemet YIQ.
  - Y beskriver luminans, I og Q er krominanskomponentene.
  - samme signalet brukes både på farge- og gråtoneskjermer.
- Overgangen mellom RGB og NTSC's YIQ :

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.623 \\ 1 & -0.272 & -0.648 \\ 1 & -1.105 & 0.705 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

  - Luminans-komponenten $Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$
  - Hue-komponenten $I = 0.596 \cdot R - 0.274 \cdot G - 0.322 \cdot B$
  - Metnings-komponenten $Q = 0.211 \cdot R - 0.522 \cdot G + 0.311 \cdot B$
- Summen av luminans-koeffisientene er 0.299 + 0.587 + 0.114 = 1.000
  - RGB svart (0,0,0) gir NTSC Y = 0
  - RGB hvit (1,1,1) gir NTSC Y = 1
- Både summen av koeffisientene for I og Q er 0.0
  - RGB grå (g,g,g) gir NTSC I = Q = 0

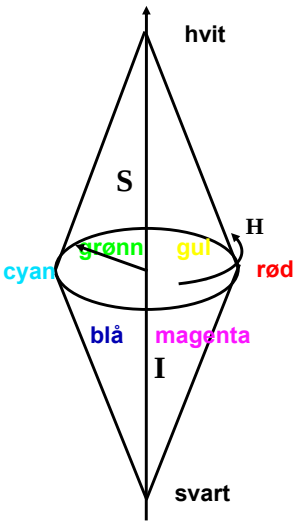
YCbCr-modellen

- Dette er fargemodellen for digital TV og video!
  - Y er luminans (luma)
  - Cb er blå minus luma (B-Y)
  - Cr er rød minus luma (R-Y).
- YCbCr er kun digital, mens RGB kan være både analog og digital.
  - MPEG-kompresjon (i DVD'er, digital-TV og video CD'er) er kodet i YCbCr
  - Digitale videokameraer (MiniDV, DV, Digital Betacam, osv.) gir et YCbCr signal over en digital link som FireWire eller SDI.
  - Den analoge "tvillingen" til YCbCr er YPbPr.

YUV-modellen

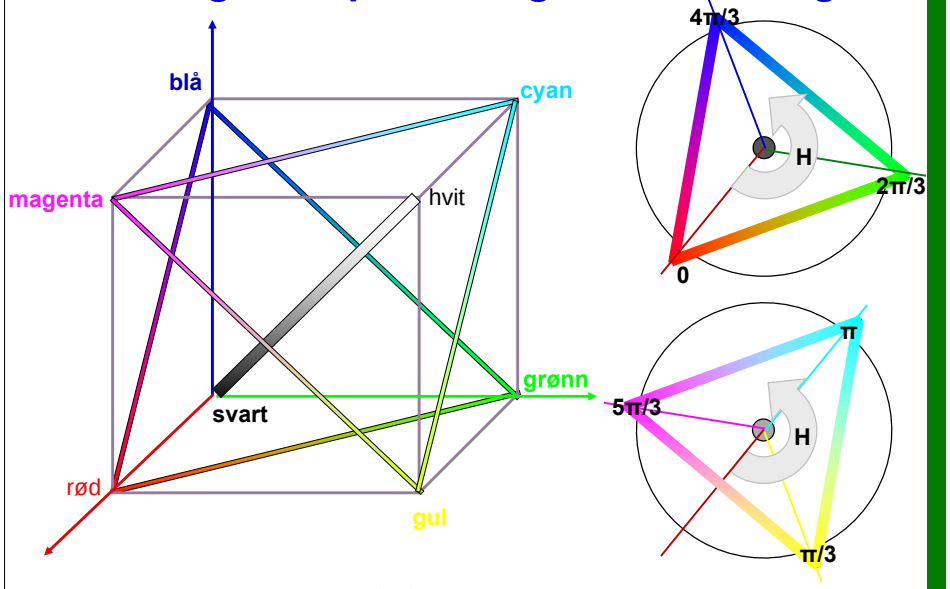
- Brukes analog TV (NTSC, PAL og SECAM).
  - Y representerer intensitet ("luma")
  - U og V er fargedifferansene B-Y og R-Y.
  - Et video-kamera konverterer RGB data som er registrert i fokalplanet til enten
    - "composite analog" (YUV)
    - analog YPbPr
    - digital YCbCr.
  - For framvisning på skjerm må alle disse tre fargerepresentasjonene konverteres tilbake til RGB.

Hue, Saturation, Intensity (HSI)



- Hue: ren farge - angir bølgelengden i det elektromagnetiske spektrum.
- H er en vinkel som ligger mellom 0 og 2π.
- Rød velges oftest som startpunkt.
- Primærfargene ligger ekvidistante rundt sirkelen
  Rød: H = 0, Grønn: H = 2π/3, Blå: H= 4π/3
- Sekundærfargene ligger midt imellom:
  Gul: H = π/3, Cyan: H = π, Magenta: H= 5π/3,
- Hvis vi skalerer H-verdiene til 8-biter får vi:
  - Primærfargene:
    R: H = 0, G: H = 255/3 = 85, B: H = 255·2/3=170
  - Sekundærfargene:
    Gul: H = 255/6 = 42, Cyan : H = 255/2 = 127,
    Magenta: H=255·5/6 = 213.

RGB og IHS - primær og sekundærfarger



Mer om HSI

- Saturation: metning – hvor mye grått inneholder fargen
  - Hvis S=0, blir fargen grå uavhengig av hvilken verdi H har. (det vil si at vi ligger et sted på diagonalen i RGB-kuben)
- S ligger normalisert mellom 0 og 1, eller mellom 0 og 255 hvis 8-biters unsigned verdier pr. piksel.
- H og S tilsammen beskriver fargen og kalles kromatisitet
- I: intensitet, ligger mellom 0 og 1 eller 0 og 255.
- HSI-modellen egnet til å beskrive farge
- RGB-modellen egnet til å generere farger
- Konverering fra HSI til RGB: formler finnes

Overganger mellom RGB og HSI (ikke pensum)

- Hvis r,g,b-komponentene er normaliserte slik at de ligger mellom 0 og 1, så blir omregningen fra RGB til HSI slik:

$$H=\begin{cases}\theta & b\le g\\ 360-\theta & b>g\end{cases} \qquad \theta=\cos\left\{\frac{\frac{1}{2}[(r-g)+(g-b)]}{\sqrt{(r-g)^2+(r-b)(g-b)}}\right\} \qquad S=1-\frac{3\min(r,g,b)}{r+g+b} \qquad I=\frac{r+g+b}{3}$$

- Merk at H er udefinert når r = g = b, S er udefinert når I = 0.
- Overgangen fra HSI til RGB kan enklest deles i tre tilfeller:
- Rød-grønn sektor:

$$0<H\le 120$$
$$r=I\left[1+\frac{S\cos H}{\cos(60-H)}\right]$$
$$g=1-(r+b)$$
$$b=I(1-S)$$

Grønn-blå sektor:

$$120<H\le 240$$
$$H=H-120$$
$$r=I[1-S]$$
$$g=I\left[1+\frac{S\cos H}{\cos(60-H)}\right]$$
$$b=1-(r+g)$$

Blå-rød sektor:

$$240<H\le 360$$
$$H=H-240$$
$$r=1-(g+b)$$
$$g=I[1-S]$$
$$b=I\left[1+\frac{S\cos H}{\cos(60-H)}\right]$$

Varianter av HSI

Det finnes ulike varianter av HSI:

- HSB (Hue, Saturation, Brightness)
- HSV (Hue, Saturation, Value)
- HSL (Hue, Saturation, Lightness)

  Forskjellen ligger er stort sett i intensitet eller brightness-komponenten.

- Dessuten kan rekkefølgen variere:

  HSI eller IHS

Eksempler på RGB, CMYK, HSI

	RGB	CMYK	HSI
Rød	(255,0,0)	(0,255,255)	(0, 255, 85)
Gul	(255,255,0)	(0,0,255)	(42,255,170)
Grønn	(0,255,0)	(255,0,255)	(85,255,85)
Blå	(0,0,255)	(255,255,0)	(170,255,85)
Hvit	(255,255,255)	(0,0,0)	(0,0,255)
Lys grå	(192,192,192)	(63,63,63)	(0,0,192)
Mørk grå	(127,127,127)	(128,128,128)	(0,0,127)
Svart	(0,0,0)	(255,255,255)	(0,0,0)

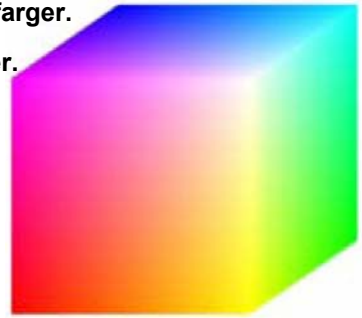
Merk: hvis S = 0, spiller det ingen rolle hva H er !

Men bildet mitt ser ikke likt ut på to skjermer?

- RGB-farger på en skjerm avhenger av skjermens egenskaper, dvs. det samme bilde vist på to skjermer kan se ulikt ut.
- Det samme bildet skrevet ut på to fargeprintere kan se HELT forskjellig ut, fargen avhenger av bl.a. skriveren, fargepatronene, papiret, etc.
- En skjerm kan vise flere farger en en CMYK-printer kan skrive ut (og en CMYK-skriver kan skrive noen farger en RGB-skjerm ikke kan vise).
- Vi sier at RGB og CMYK er utstyrs-avhengige fargerom.
- Det finnes internasjonale standarder for fargerom som er utstyrs-uavhengige. Et slikt system er CIEs XYZ-fargerom.
- Antall stabile, "gjenkjennbare farger" på en skjerm er ganske lite !

Fargesyn

- Vi kan skille mellom ca. 100 rene farger (hue).
- Når fargene også varierer i intensitet, kan vi skille mellom ca. 6 000 farger (hue+intensity).
- For hver av disse, kan vi skille mellom ca. 60 ulike metningsgrader (saturation).
- Vi kan altså skille totalt ca. 360 000 farger.
- Dette kan representeres med 19 biter. ($2^{19} = 524\,288$).
- Lagrer R, G, B komponentene som byte-bilder.
  - totalt 24 biter per piksel.



Fargebilder og fargetabeller

- RGB kan lagres med like mange biter for r, g, b, f.eks (8 + 8 + 8)
- Selv 3 + 3 + 3 = 9 biter gir oss $8 \cdot 8 \cdot 8 = 512$ kombinasjoner, men bare 8 forskjellige nivåer av rødt, grønt og blått, og dermed også bare 8 forskjellige gråtoner.
- Det er ikke sikkert at alle de 512 fargene finnes i bildet.
- Et scene med mange nyanser av én farge vil da se ille ut ! Hvorfor? Jo fordi denne fargen bare får 8 forskjellige nyanser !
- Alternativt kan man bruke 8 biter og fargetabeller.
- Hver rad i tabellen beskriver en r, g, b-farge med 24 biter.
- Tabellen inneholder de 256 fargene som best beskriver bildet.
- I bilde-filen ligger pikselverdiene som tall mellom 1 og 255.
- Når vi skal vise bildet, slår vi bare opp i samme rad som pikselverdien, og finner de tilsvarende r, g, b-verdiene.

Fargetabell / oppslagstabell (LUT)

- Gråtone/fargeavbildningen utføres som oppslag i en tabell
- LUT - Look Up Table
- Innholdet i bildefilen endres ikke, LUT-operasjonen utføres på datastrømmen mellom hukommelsen (databufferet) og skjermen

  $$v_{out} = LUT(v_{in})$$

- Hvis vi ønsker endring i bildet: Oppdatér bare G verdier i LUT (ikke $N \cdot M$ verdier i bildet)
- Q: Kan vi lage et negativt fra et positiv på denne måten ?

Fargetabell

Pikselverdi	RGB-verdi
0	0,0,0
1	255,0,0
2	255,255,0
3	0,255,0
.	255,100,0
.	.
.	.
.	.
254	0,100,255
255	255,255,255

Disse verdiene ligger lagret på bildefilen

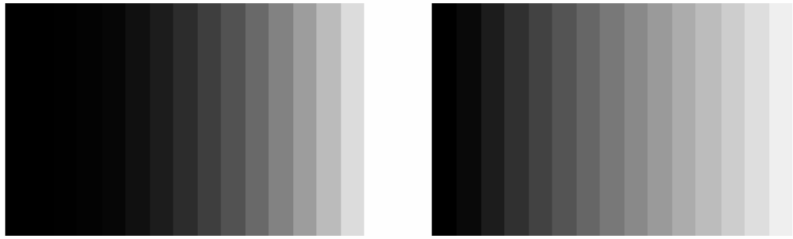
Disse verdiene vises på skjermen

- Kan vise 24 biters RGB-verdier på 8 biters skjerm
- Eller vise pseudofarger fra et gråtonebilde
- Pikselverdiene fra 0 til 255 tilordnes et RGB-triplet
- Ved framvisning leses pikselverdien
- Pikselverdien viser til et linjenummer i tabellen som inneholder RGB-fargene.

Gamma-korreksjon

Pikselverdien gjengis ikke lineært som intensitet på skjermen, men proposjonal med inputsignalet opphøyet i eksponenten γ.

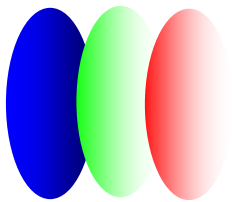
- $\gamma = 2.2$ => Pikselverdi f = 0.5 gjengis som 0.21. (Typisk for NTSC).
- TV-systemer forhåndskorrigerer bildene før de sendes ut.
  - $R' = R^{1/\gamma}$.
  - Lysmengden fra skjermen $(R')^{\gamma} = R$



Alfa-kanal

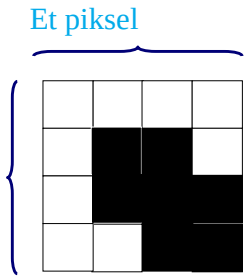
- α i (RGBα) eller (αRGB) spesifiserer om fargene (RGB) i bildet er helt eller delvis transparente.
- Verdier av α fra 0 (helt transparent) til 255 (helt ugjennomsiktig).
- Hensikten med en "alfa-kanal" er at man kan la en bakgrunn synes gjennom et bilde.
- Bakgrunnen kan bestå av forskjellige grafiske elementer, eller av et annet bilde.
- Teknikken kalles "alpha blending", og kan både brukes til
  - å vise tekst og grafikk sammen med et bilde
  - "blending" av to bilder, to bildesekvenser, eller stillestående bakgrunn med en video-sekvens.
  - Finnes i Adobe Photoshop, Paint Shop Pro, GIMP ....
- Hvis vi legger et bilde oppå en bakgrunn, blir resultatet

  (bildefargen · α + (bakgrunnsfargen · (255 - α)) / 255.
  - Resultat lik bakgrunn for α = 0
  - Resultat midt mellom for og bakgrunn for α = 127
  - Resultat lik forgrunn for α = 255.



Utskrift av gråtonebilder

- Problem: printere er binære, skriver svart eller ingenting
- Løsning: printeren jobber på et finere grid (bruker halvtoner)
- Virker fordi: øyet gjør en glatting av intensitetsverdier, slik at et gjennomsnitt vises
- Utfordring: hvordan lage mønstere av binære piksler som utgjør en gråtone
  - "Patterning" bruker n^2+1 verdier fra n x n rutenett
  - Ordnet "Dithering" terskler med en matrise
  - "feil-diffusjon" fordeler feilene ved terskling



"Dithering"

- Terskler gråtonebildet mot en "dither-matrise"
- Dither-matrisen D_n
  - inneholder $2^n \cdot 2^n$ elementer
  - deler gråtoneskalaen fra 0 til 255 inn i $(2^n)^2$ ekvidistante trinn.

$$D_2 = \begin{bmatrix} 0 & 128 & 32 & 160 \\ 192 & 64 & 224 & 96 \\ 48 & 176 & 16 & 144 \\ 240 & 112 & 208 & 80 \end{bmatrix}$$

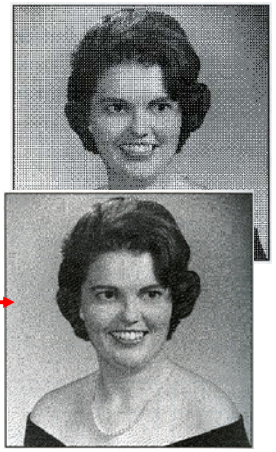
- Forstørr opp bildet med en faktor 2^n.
- Matrisen legges som en maske over bildet
- Elementene i matrisen fungerer som terskler.
- Hvis pikselverdien > terskelen => hvit, ellers svart.
- Gir et tilsynelatende gråtonebilde som
  - Består av svarte og hvite punkter
  - Har samme størrelse som original-bildet
  - Har systematiske mønstre for hver gråtone.

Feil-diffusjon

- Retter opp systematiske feil som innføres ved dither-terskling.
- En terskel = 128 vil avbilde en gråtoneverdi som 0 (svart) eller 255 (hvit)
  - OK hvis pikselverdi nær 0 eller 255
  - hvis pikselverdi nær terskelverdien blir feilen stor.
- Diffusjon sprer feilen over flere nabopiksler

$$\begin{bmatrix} .. & .. & .. \\ .. & P & 7/16 \\ 3/16 & 5/16 & 1/16 \end{bmatrix}$$

- Dette forbedrer det visuelle resultatet
  - Begrensninger:
    - Kan ikke kan spre feilen utenfor bildets grenser
    - Gråtoner kan ikke ende under 0 eller over 255.



Utskrift av fargebilder

- CMYK-modell brukes
- Halvtonemønstre i bestemte vinkler (ulik for hver farge) må brukes til å lage fargemønstere
- Prinsipp: øyet kombinerer de fire fargene slik at ingen brå fargeoverganger ses
  - Hver farge skrives ut i et spesielt symmetrisk mønster

Overganger mellom små og store "endianere"

- To piksler med 3·8 biters RGB "big endians":(R1G1B1)(R2G2B2) ex:(0,255,0),(0,255,0)
  avlest som "little endians" blir til (G1R1R2)(B1B2G2) (255,0,0),(0,0,255)

- La en LUT inneholde 256 farger
  - LUT'en – som inneholder 256 · 3 byte (RGB) vil bli utsatt for effekten ovenfor.
  - Samtidig vil to og to piksler i bildefilen bytte plass
- La en LUT inneholde $2^{16} - 1 = 65535$ linjer (farger) a 16 biter.
  - Nå blir ikke lenger to og to piksler i bildefilen byttet om.
  - Men pikselverdien vil peke til feil sted i fargetabellen.
- Anta 16 biter = 2 byte RGB: 5 + 6 + 5 biter
  - 50% grå svarer til (16, 32, 16) i en (5 + 6 + 5) biters LUT, med bitmønster

    1000010000010000
  - Bytter vi om på bytene får vi

    0001000010000100 (2, 4, 4) dvs (0.0625, 0.0625, 0.125) på en skala fra 0 til 1.

    50% grått er blitt til en ganske dunkel blåfarge.

Bildeformater

Vanligvis:

- Header – kan være ascii eller binære verdier

< magic number > < tittel > < bredde = n > <høyde = h > < #bånd = k > < bildetype >.....

- Pikselverdier – binære verdier (som oftest)

$x_1\ x_2$ ..... x_n linje 1 – bånd 1
...
$x_1\ x_2$ ..... x_n linje h – bånd 1
$x_1\ x_2$ ..... x_n linje 1 – bånd 2
...
$x_1\ x_2$ ..... x_n linje h – bånd 2
$x_1\ x_2$ ..... x_n linje 1 – bånd 3
...
$x_1\ x_2$ ..... x_n linje h – bånd 3
.....
.....
$x_1\ x_2$ ..... x_n linje 1 – bånd k
...
$x_1\ x_2$ ..... x_n linje h – bånd k

Formattyper

- Software-spesifikke

  XITE BIFF-format, ENVI, MacPaint, Windows BMP, HIPS-format

- Utvekslingsformater
  - GIF (Graphic Interchange Format)
  - PNG (Portable Network Graphics)
  - JFIF (JPEG File Interchange Format)
  - TIFF (Tagged Image File Format)
  - PGM (Portable Grey Map)
  - FITS (Flexible Image Transport System)
- MPEG: standard for video (mer om den senere)

GIF

- GIF har mest historisk interesse pga WWW og HTML.
- GIF var første bildeformat som kunne håndteres av nettlesere.
- "GIF standard" er begrenset til 8 biters fargebilder (LUT)
  - passer best for bilder eller grafikk med få og distinkte farger.
- GIF finnes i to utgaver: GIF87a, og GIF89a.
  - Den siste gir mulighet for enkel animasjon.
- GIF bruker LZW-algoritmen for kompresjon (mer om det senere).

PNG-formatet

- Laget pga. patentproblemene med GIF
- Støtter:
  - gråtonebilder med max 16 biter
  - 8 biters fargebilder med fargetabeller
  - RGB med opptil 16 biter pr. kanal
- Har kompresjon (koding)
- Alpha-bånd

JPEG

- JPEG er i dag det mest brukte bildeformatet
- de facto standard for kompresjon og lagring av komprimerte bilder.
- Man kan sette
  - enten en kompresjonsfaktor
  - eller en kvalitets-parameter,

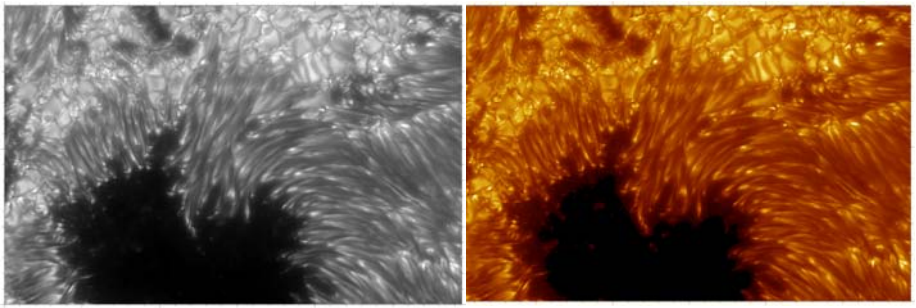
  og så overlate komprimeringen til programvaren.
- Vi kommer tilbake til JPEG-kompresjon senere.

TIFF = "Tagged Image File Format"

- Åpner for å hekte på tilleggsinformasjon om bildet - "tags"
  - hva slags kompresjon som er brukt
  - oppslagstabeller, osv.
- TIFF kan lagre en rekke forskjellige typer bilder:
  - bitplan
  - gråtonebilder
  - 8 biters fargebilder (med LUT)
  - 24 biters RGB
  - bilder som er komprimert uten informasjonstap
  - og JPEG-bilder komprimert med informasjonstap

Pseudo-farger

- *Pseudo-fargebilder* er egentlig gråtonebilder der man har tilordnet hver gråtone en RGB-farge ved hjelp av en oppslagstabell (LUT).



(©Royal Swedish Academy of Sciences).

Falske farger

- NOAA AVHRR

kanal 1:

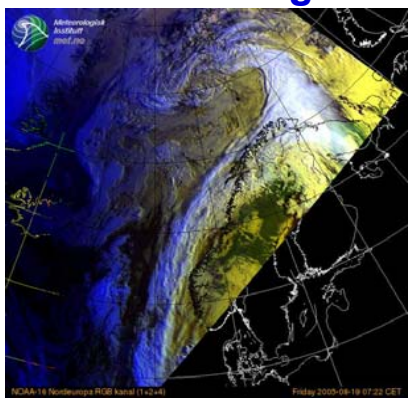
580-680 nm

kanal 2:

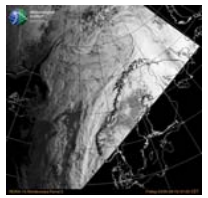
725-1000 nm

kanal 4:

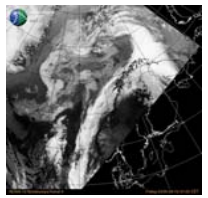
1030 – 1130 nm



Kanal 1+2+4 som RGB



Kanal 2

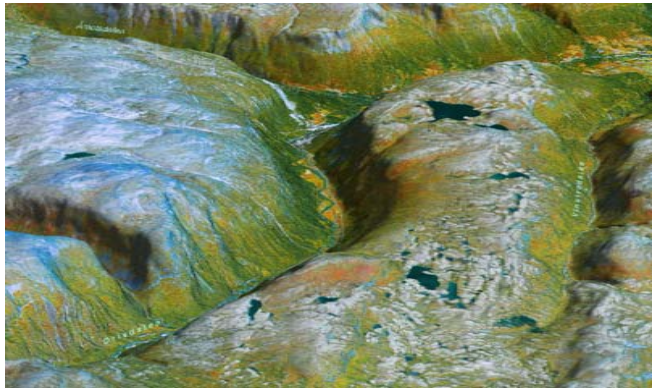


Kanal 4

- vist som RGB-bilde (Meteorologisk Institutt)
  - kanalene er ikke RGB (700, 546.1, 435.8). Altså *falske farger*.

Digitale bilder på terrengmodell

- Flerbånds satellittbilder kan gi en fargemessig korrekt gjengivelse av vann, jordbruksområder, skog, snaufjell og snø/is,
- Resultatet kan ”legges oppå” en terrengmodell som gir høyden til hvert piksel og så gjengis tredimensjonalt.



”Verdens beste satellittbilde”.Geodatasenteret AS 2005.

Dagens oppsummering

- Vi oppfatter farger fra fiolett (400 nm) til rødt (700 nm).
- RGB brukes til bildefangst (kamera) og til framvisning (skjerm).
- CMYK brukes for å skrive ut bilder på papir.
- YIQ, YCbCr og YUV brukes i TV- og videosystemer.
- LUT’er brukes for å kunne ha en kort ordlengde i bildefilen og en full fargerepresentasjon i LUT’en.
- Vi har sett på
  - Gammakorreksjon for å endre kontrast
  - Alfakanal for ”blending” av bilder
  - Utskrift med halftoning, patterning, dithering og feildiffusjon
  - Problemer med ”endians”
  - Bruk av pseudofarger og falske farger
- Neste gang: Digital video og digital bildeanalyse.