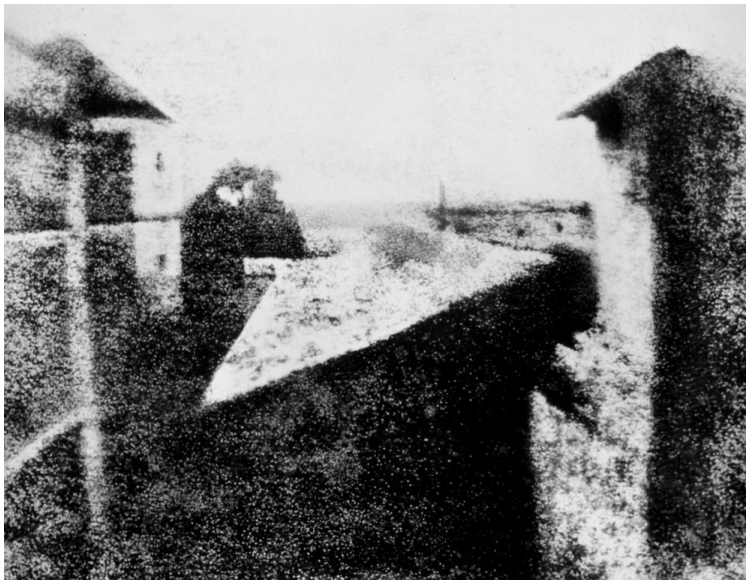
Digitale bilde

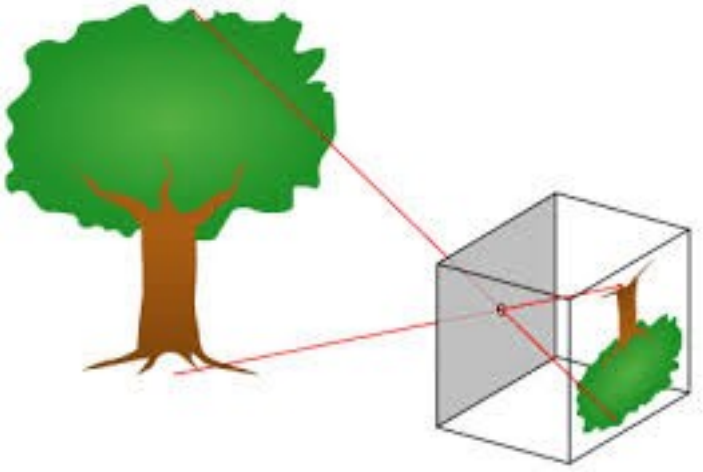
- Bildefangst
  - Kameramodellar
  - Digitalisering
  - Bildestorleik og oppløysing
  - Koordinatsystem
  - Pikseltyper
- Filformat
  - Raster/vektor
  - TIFF
  - GIF
  - PNG
  - JPEG
  - BMP
  - PBM
  - ...

Kameramodell 1: «Pinhole»

Fyrste og enklaste kamera: Boks med hol.
Bilde på bakvegg. Bildet vert rotert 180°.

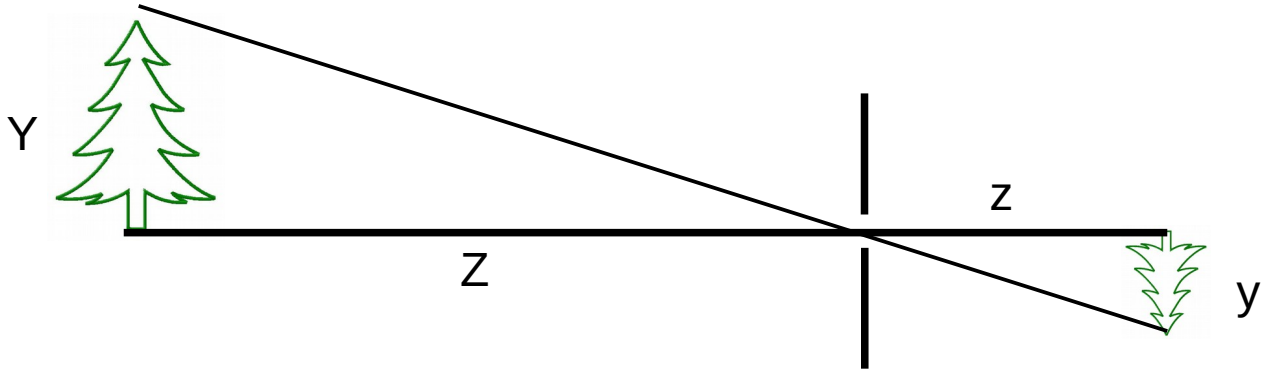


Bilde tatt 1826 med pinhole kamera.



«Pinhole»

Denne kameramodellen har enkel geometri. Sjå figur 1.2 side 5 i boka.
Sett frå sida:



Likeforma trekantar! Dette gjev:

$$\frac{Y}{y} = \frac{Z}{z} \Leftrightarrow \frac{Y}{Z} = \frac{y}{z}$$

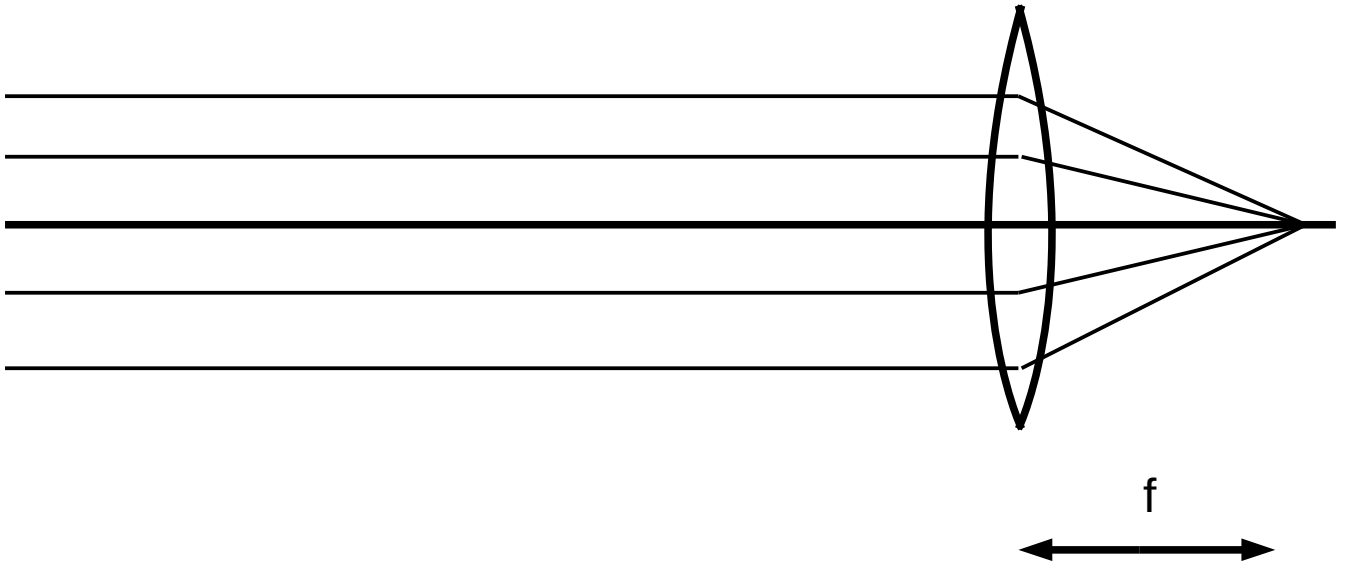
Kjenner vi tre, kan vi rekne ut den fjerde.
(1.1) i boka.

Kameramodell 2: Tynn linse

«Pinhole» må ha liten lysopning, krev mykje lys. Vi erstattar *opning* med ei *tynn linse*, vi kan da ha større opning (blendar, aperture).

Linse er kjenneteikna ved fokallengde (=brennvidde), f.

Parallelle stråler inn mot tynn linse samlast i eit punkt i avstand f frå (midten av) linsa.

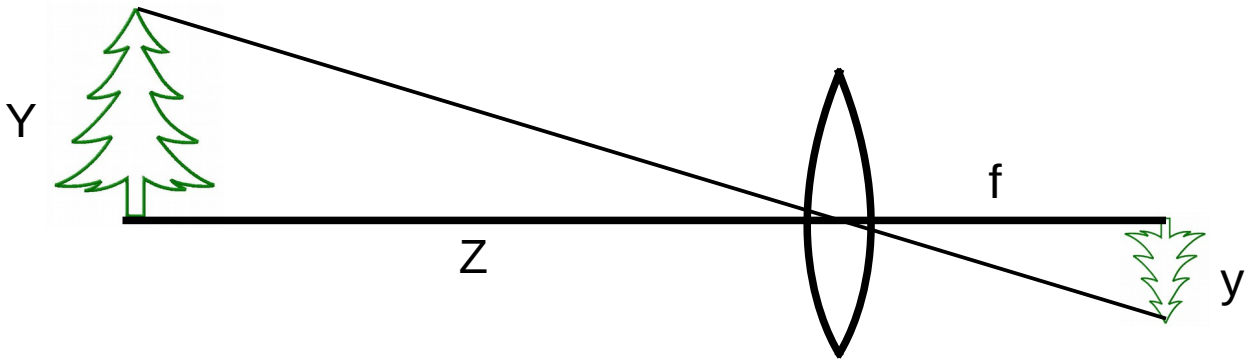


Tynn linse

Vi gjer to tilnærmingar (dvs. feil!):

- Anta at linsa er så tynn at vi kan late som at stråler bøyast i midten av linsa
- Anta at objektet er uendeleg langt unna

Då vert geometrien lik «Pinhole»-geometrien, der bildet må vere i avstand z=f.

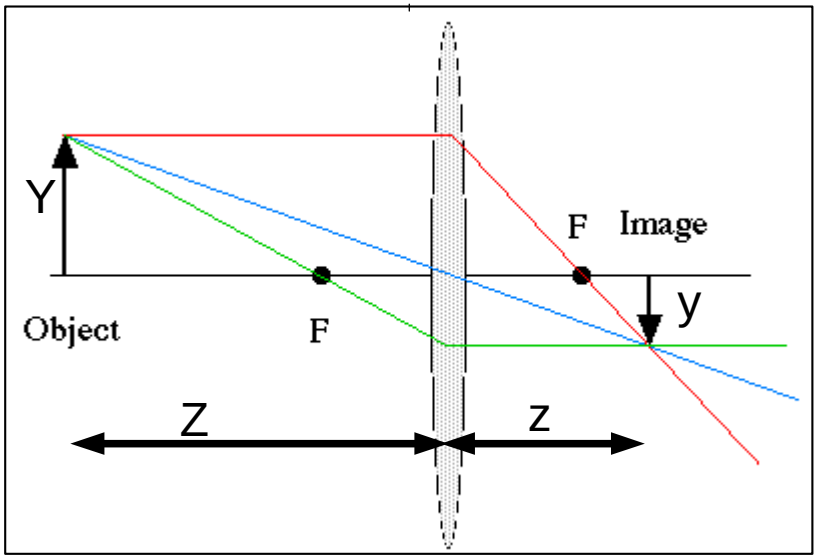


$$\frac{Y}{y} = \frac{Z}{f} \Leftrightarrow \frac{Y}{Z} = \frac{y}{f}$$

Vi har altså 5 verdiar: Y, y, Z, z og f. Men sidan z=f er det reelt berre 4 ukjende. Kjenner vi tre av dei, kan vi rekne ut den fjerde.

Kameramodell 3 (ikkje i boka)

Kva om objektet er ganske nær kamera, slett ikkje uendeleg langt unna?
Då vil strålene frå eit punkt på objektet samlast ***bak*** fokalpunktet. z > f



Omgrep:

- Linse
- Linseplan
- Optisk akse
- Objekt
- Objektavstand Z
- Objekthøgde Y
- Bilde
- Bildeplan
- Bildeavstand z
- Bildehøgde y
- Fokallengde (Brennvidde) f
- Brennpunkt F
- Forstørring m = y/Y

Linselikninga seier at: $\frac{1}{f}=\frac{1}{Z}+\frac{1}{z}$

Altså: kjenner vi to av desse, kan vi rekne ut den tredje. Men vi har også at $\frac{Y}{y}=\frac{Z}{z}$
Kjenner vi *tre* av Z, z, Y, y og f – kan vi rekne ut dei to siste.

Digitalisering

Digitalisering består av to hovudsteg:

1) **Sampling** : Dele opp kontinuerleg signal i endeleg antal delar, gjere ei måling i kvar del.

   a) Romleg – oftast rektangulær oppdeling av biletflata

   b) Temporalt – lukkartida på digitalkamera/videokamera

2) **Kvantisering** – runde av den målte (kontinuerlege) verdien til næraste verdi som kan representerast med det antal bit (t.d. 8) som er vald.

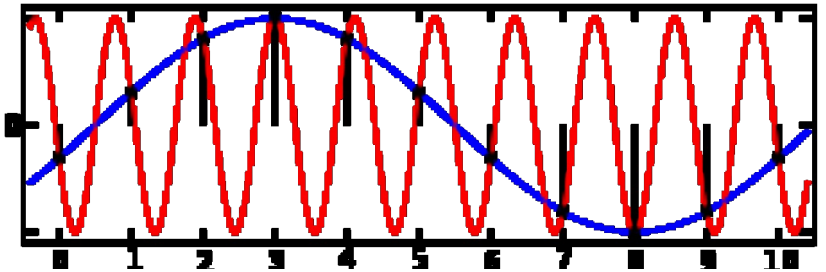
Begge steg introduserer feil, dvs mister informasjon.

Difor ikkje mogeleg å gjenskape originalen eksakt.

Samplingsteoremet

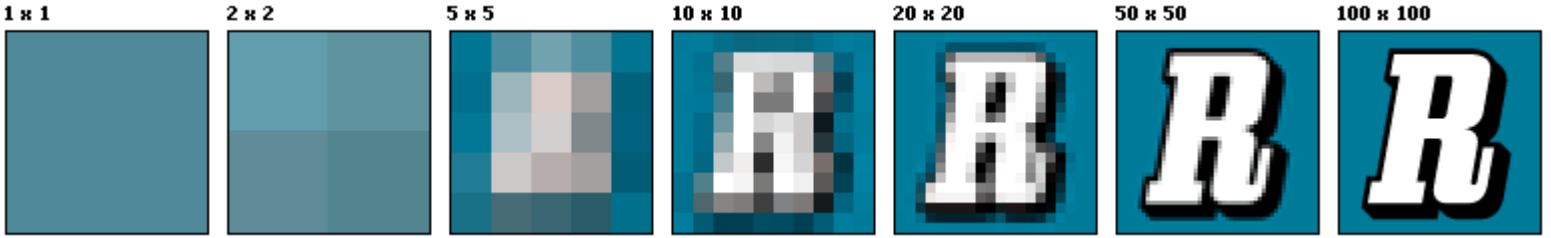
(Nyquists/Shannons samplingsteorem)

«Man kan vise at dersom samplene tas med fast avstand og tilstrekkelig tett, kan det opprinnelige, kontinuerlige signalet gjenskapes fra samplene. Det såkalte samplingsteoremet sier at samplingsraten må være minst dobbelt så høy som den høyeste frekvenskomponenten i det opprinnelige signalet for at signalet skal kunne gjenskapes fra samplene. Dersom denne betingelsen ikke er oppfylt, vil det i samplingsprosessen bli dannet frekvenskomponenter som ikke var til stede i det opprinnelige signalet. Dette kalles aliasing. Frekvenskomponentene som dannes ved aliasing, kan normalt ikke fjernes igjen og vil opptre som støy når signalet rekonstrueres fra samplene.» (Store Norske leksikon)



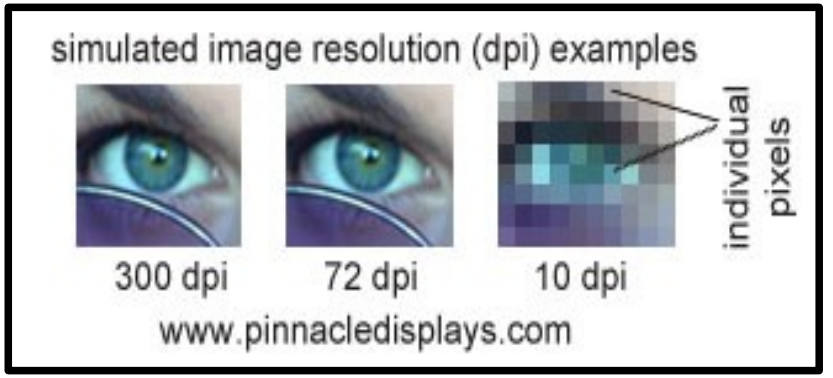
Storleik og oppløysing

Storleik (størrelse): breidd og høgd målt i pikslar.

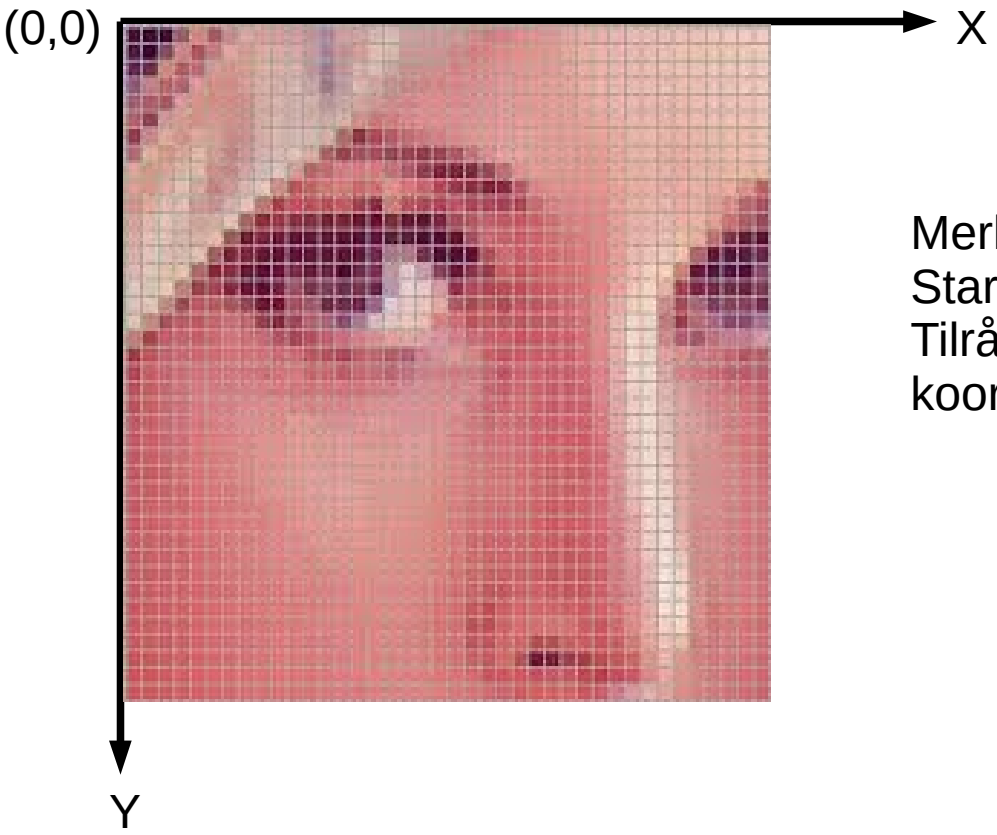


Oppløysing: Tilhøvet mellom pikslar og den verkelege verda.
Målast t.d. i dpi/ppi/ppt (dot/punkt/pikslar per inch/tomme), dpcm (dots per cm).
Kan også seie at eitt piksel er 5x5m. (Svarar til 200 pikslar per km).

Oftast lik oppløysing i begge (alle?) dimensjonar.
Nyttast til å gje eigenskapar ved monitor og printer.
Kan også gje eigenskapar ved ***nokre*** bilde.



Koordinatsystem for bilde



Merk at y-akse går nedover.
Startar i origo, (0,0).
Tilrår sterkt å nytte x for horisontal koordinat, og y for vertikal.

Pikselverdiar / pikseltyper

Med k bit kan ein altså lagre 2^k ulike verdiar. Kor mange bit trengs?
Sjå tabell 1.1 side 11 i boka:

Gråtonebilde			
Kanalar	Bits/piksel	Verdiområde	Bruk
1	1	0..1	Binære bilde:dokument, fax, strektegning
1	8	0..255	Generelt gråtonebilde
1	12	0..4095	Høg kvalitet gråtonebilde
1	14	0..16383	Profesjonell kvalitet gråtonebilde
1	16	0..65535	Høgaste kvalitet: medisin, astronomi, ...

Pikselverdiar / pikseltyper II

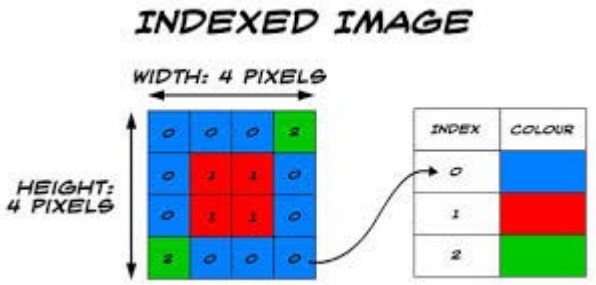
Fargebilde			
Kanalar	Bits/piksel	Verdiområde	Bruk
3	3x8	$[0..255]^3$	Generelle RGB fargebilde
3	3x12	$[0..4095]^3$	Høg kvalitet fargebilde
3	3x14 el 3x16	$[0..16383]^3$ el. $[0..65535]^3$	Prof. kvalitet fargebilde
4	4x8	$[0..255]^4$	CMYK for printing
1	8 (el. meir)	0..255	Indeksert/palett-bilde

Spesielle bilde			
Kanalar	Bits/piksel	Verdiområde	Bruk
1	16	-32768..32767	Heiltal med forteikn
1	32	$+/- 3{,}4 \cdot 10^{38}$	Kommatal (flyttal)
1	64	$+/- 1{,}8 \cdot 10^{308}$	Høg kvalitet kommatal (flyttal)

Indeksert bilde

Pikselverdiane fortel ikkje direkte kva for gråtone eller farge pikselet skal ha.

Pikselverdien nyttast som indeks i tabell som fortel kva for gråtone / farge. Tabellen kallast oppslagstabell / Look Up Table / LUT.



Om bildet har 8-bits pikslar må tabellen vere på 256x3 byte. For å endre fargane i eit bilde endrar ein berre desse 768 vediane, ist. for millionar av pikslar.

Filformat

Det er mange filformat å velje mellom. Kva for eit skal du velje –

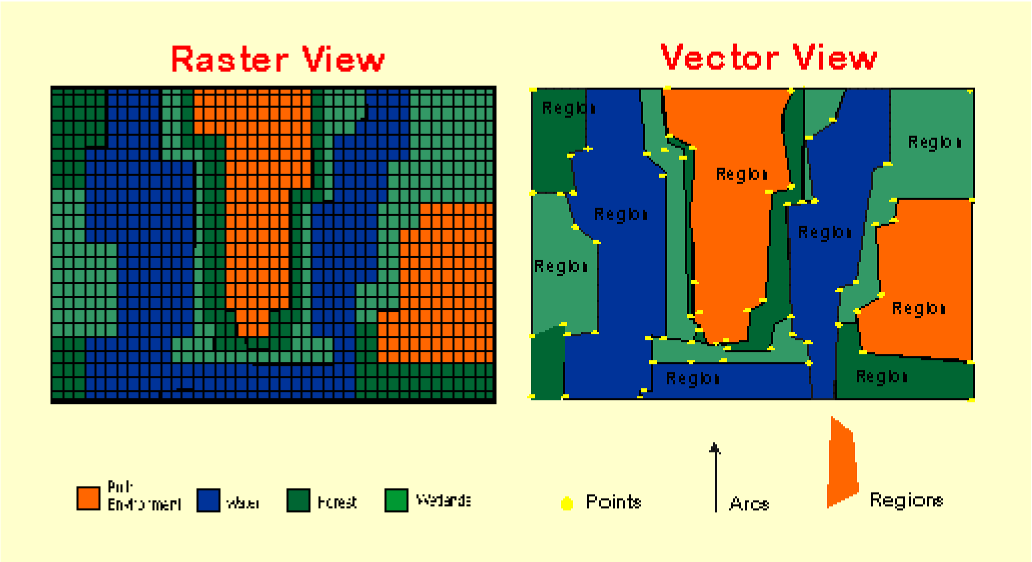
- for å lagre dine eigne bilde?
- for å utvikle applikasjon som handterer bilde?

Vurdér:

- Kva slags bilde? Svart-kvitt, gråtone eller fargar? Naturlege eller kunstige? Kor store kan dei bli? Er irreversibel (lossy) kompresjon aktuelt?
- Kor mykje lagringsplass har du? Kor mange bilde? Skal bilda flyttast over nettverk – ein eller fleire gonger?
- Skal bilda utvekslast med andre, eller lagrast lenge, slik at eit opent, standard filformat er viktig?
- Kva for fagfelt er det snakk om – somme fagfelt har sine eigne, viktige filformat.

Raster eller vektor?

Vektor: Her lagrast bildet som ei mengde enkle geometriar, t.d. polygon. Nokre bilde er egna til å lagre som vektor, andre som raster. Enkle bilde med store homogene område er godt egna som vektor, som grafikk, kart m.m.



Vektorformat:

CGM – Computer Graphics Metafile
SVG – Scalable Vector Graphics
- og mange proprietære.

Mens eit naturleg bilde som konverterast til vektor, vil oftast *mangedoble* filstorleiken.

TIFF – Tagged Image File Format

- Omfattande (fleksibelt) format, mykje bruka.
- Støttar gråtone, indeksert og true-color, mange pikseltyper.
- Kan ha mange ulike bilde på same fil.
- Støttar mange former for kompresjon, tapsfri og irreversibel.
- Utvidbart: kan definere nye «taggar», men dette er ikkje standardisert.
- Difor nesten umogeleg å lage programvare som støttar *alt* i TIFF - «unsupported tag» feilmelding kan forventast.

GIF – Graphics Interchange Format

- Mykje nytta på web
- Gråtonar og indeksert (maks 8 bit), ikkje true-color
- Kan innehalde transparens
- Støttar animasjon – bildesekvens
- Komprimerer tapsfritt (LZW), ikkje irreversibelt. Godt egna til grafikk
- Patentkrangel og avgiftskrav rundt kompresjonsalgoritma fram til 2006.
- Bruken er på veg ned – PNG tek over. Betre teknisk, og opent.

PNG – Portable Network Graphics

(Utt. «ping»?) Laga for å erstatte GIF grunna lisenskrangel. Også betre teknisk:

- True-color (opp til 3x16 bit)
- Gråtone (opp til 16 bit)
- Indeksert (opp til 256 fargar)
- Alpha-kanal for *gradert* transparens
- Store bilde – opp til 2^{30} x 2^{30} pikslar
- Tapsfri kompresjon med PKZIP
- Berre eitt bilde per fil
- Bruk dette på web når du ynskjer tapsfri kompresjon!

JPEG – Joint Photographic Experts Group

Eigentleg ein standard for irreversibel kompresjon, baka inn i mange format og program. Egna til naturlege bilete.

Hovudsteg i algoritma:

1) Konvertere frå RGB til YC_bC_r (som separerer gråtonebildet frå fargane) og deretter nedsampling av fargekomponentane
2) Cosinustransform av blokker på 8 x 8 pikslar, og kvantisering av resultatet. Dei høgaste frekvensane (raske endringar) kuttast mest.
3) Tapsfri kompresjon (oftast Huffman) av dette.

Komplisert algoritme. Ikkje egna til strekteikningar, tekst, grafikk. Tek berre opp til 8-bit pikslar. 8x8 blokkene vert merkbare ved kraftig kompresjon.

JPEG-format

- JFIF – JPEG File Interchange Format. Mykje bruka format som implementerer JPEG standarden. Fastset det som er ubestemt i JPEG.
- EXIF – Exchangeable Image File Format. Brukast til å lagre bilde i digitalkamera. Tek vare på kameraparametre (blendar, lukkartid, tidspunkt, GPS-posisjon, ...) ved hjelp at TIFF-taggar.
- JPEG-2000. Forbetring (og dermed endring) av JPEG med t.d. opp til 64x64 blokker, wavelet-transform ist. for cosinus. Gjev noko betre kompresjon, men likevel lite bruka så langt.

BMP – Windows Bitmap

Enkelt format mykje bruka på Windows plattform.

- Støttar binære, gråtone, indeksert, og true-color bilde.
- Tapsfri kompresjon («runlength»).
- Mindre fleksibelt enn TIFF.

PBM – Portable Bitmap Format

Eigentleg *familie* av format (PNM – Portable Anymap):

- PBM – Portable Bitmap til binære bilete (1-bit / svart/kvitt)
- PGM – Portable Greymap til gråtonebilde (8-bit)
- PPM – Portable Pixmap til RGB fargebilde (3x8bit)

Familien er også kalla Netpbm. Støttar både binært koda bilde, og koda på lesbart tekstformat slik at du kan lage eit bilde med tekstbehandlar. Lesbart format tek mykje større plass.

- Ingen kompresjon
- tekstformat egna til små bilde (ikonar)

Bits & bytes

Bildefil vert oftast lest og skrive med ferdige biblioteksmodular (klasser...)

I spesielle tilfelle må du kanskje skrive eigen programvare til dette. Tenk over:

- *Big endian* vs. *little endian*. Når du har t.d. eit 32-bits heiltal som skal skrivast til fil – i kva rekkefylgje skal dei 4 bytane skrivast? MSB (most significant byte) fyrst (big endian), eller LSB (least significant byte) fyrst (little endian)? Intel-prosessorar nyttar little endian, dei fleste andre big endian. Kan gje kaos.
- Filheader. Alle(?) format er definert med ein header før piksledata. Headeren fortel t.d. kva for pikseltype som er nytta, bildestorleik, og andre metadata. Nokre format har fast storleik på header, andre meir fleksibelt (t.d. TIFF).
- Korleis vite kva format som er nytta? To løysingar:
  - Fil-etternavnet. .jpg, .pbm osv.
  - Ein «signatur», dei fyrste 2++ byte av fila har spesiell verdi.