NTNU
Institutt for datateknikk
og informasjonsvitenskap

KONTINUASJONSEKSAMEN I EMNE TDT4195
BILDETEKNIKK
LØRDAG 15. AUGUST 2009
KL. 09.00 – 13.00

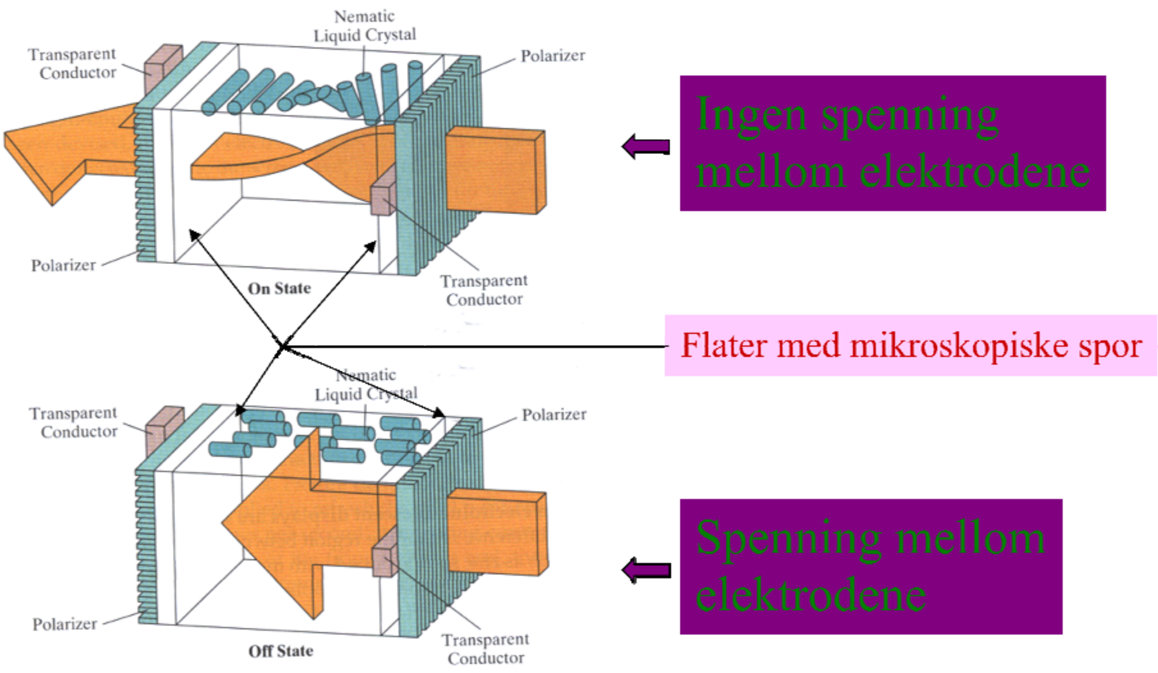
LØSNINGSFORSLAG - GRAFIKK

OPPGAVE 1 Grafikk – diverse spørsmål

a) Fargeoppslagstabeller brukes for å minimere plassen man behøver for bildelageret (lite brukt med dagens hardware). Man kan også bruke fargeoppslagstabeller til å bytte ut hele fargepaletten (ikke nødvendig å nevne for at besvarelsen skal regnes som fullstendig). Se figur 4-1 i læreboka.

b) Man simulerer parallellprojeksjon i matrisen for perspektivprojeksjon ved å plassere projeksjonssenteret i stor avstand fra bildeplanet, se figur 7-43 i læreboka

c)



OPPGAVE 2 Grafikk – transformasjoner

a) Det ventes at kandidaten skal se at det her dreier seg om rotasjon om en akse som er parallell med z-aksen. Det er unødig arbeidskrevende å behandle tilfellet som rotasjon om en vilkårlig akse. Svaret blir:

$$M = T_{origo}^{-1} \cdot R_z \cdot T_{origo} = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 5 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b) Med homogene koordinater $\begin{pmatrix} x_h & y_h & z_h & w_h \end{pmatrix}^T$ får vi de kartesiske koordinatene $\begin{pmatrix} x & y & z \end{pmatrix}^T = \begin{pmatrix} \frac{x_h}{w_h} & \frac{y_h}{w_h} & \frac{z_h}{w_h} \end{pmatrix}^T$. Av dette ser vi at en alternativ formulering av matrisen for uniform skalering kan være:

$$M_s \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{s} \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ \frac{1}{s} \end{bmatrix}$$, som gir kartesiske koordinater $\begin{bmatrix} s \cdot P_x \\ s \cdot P_y \\ s \cdot P_z \end{bmatrix}$

c) Transformasjonen blir en skalering etterfulgt av en translasjon (se figur 6-8 i læreboka).

Skaleringen:
Kantene i det normaliserte klippevinduet har lengde 2, men skal ha lengde gitt av området på skjermen, det vil si $x_{\max} - x_{\min}$ og $y_{\max} - y_{\min}$.

Translasjonen:
Sentrum av det normaliserte klippevinduet er origo. For å få det skalerte vinduet til rett plass på skjermen, må koordinatene i x- og y-retningen gis tillegg på henholdsvis:

$$\Delta x = x_{\min} + \frac{x_{\min} - x_{\max}}{2} = \frac{x_{\min} + x_{\max}}{2} \text{ og } \Delta y = y_{\min} + \frac{y_{\min} - y_{\max}}{2} = \frac{y_{\min} + y_{\max}}{2}$$

Vi får:

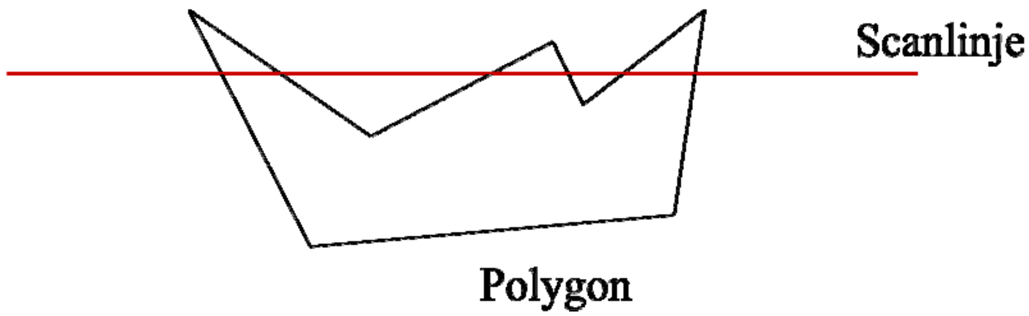
$$M_{normalsquare,viewport} = \begin{bmatrix} 1 & 0 & \frac{x_{\min}+x_{\max}}{2} \\ 0 & 1 & \frac{y_{\min}+y_{\max}}{2} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{x_{\max}-x_{\min}}{2} & 0 & 0 \\ 0 & \frac{y_{\max}-y_{\min}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{x_{\max}-x_{\min}}{2} & 0 & \frac{x_{\min}+x_{\max}}{2} \\ 0 & \frac{y_{\max}-y_{\min}}{2} & \frac{y_{\min}+y_{\max}}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

OPPGAVE 3 Grafikk – skannkonvertering

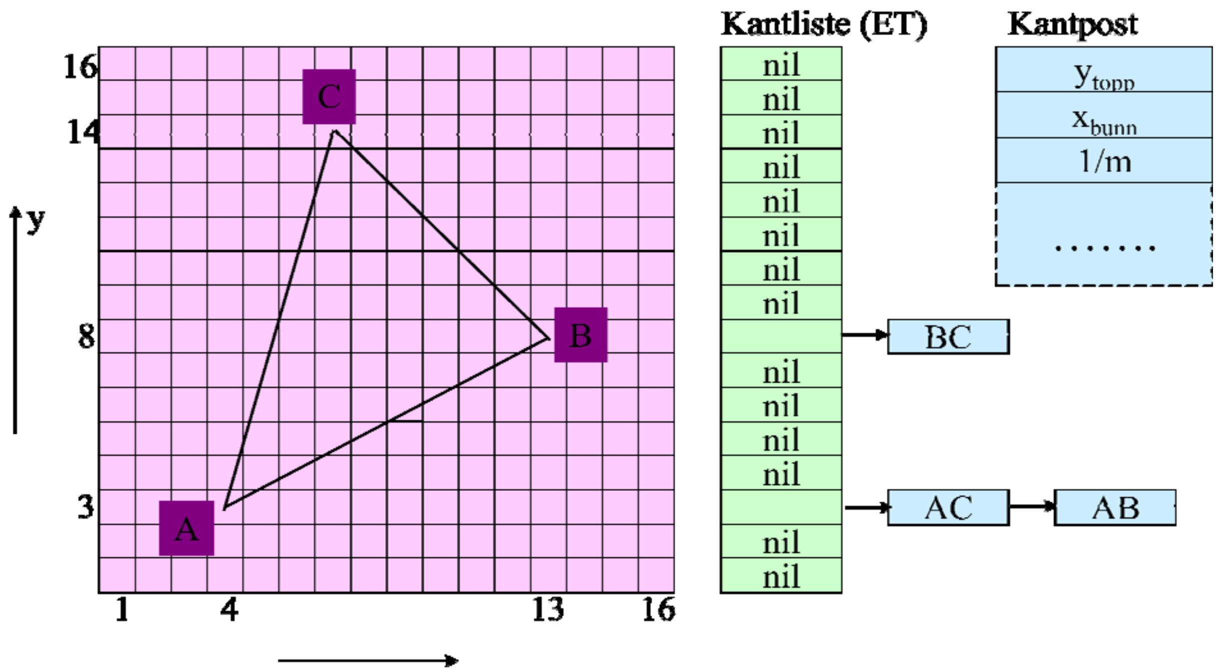
Skannlinjealgoritmen for fylling av polygoner fyller som navnet tilsier polygonen linje for linje langs skannlinjen. Grunnleggende er en metode for å bestemme om pikselen som skal tegnes, ligger inne i eller utenfor polygonen. En vanlig algoritme er:

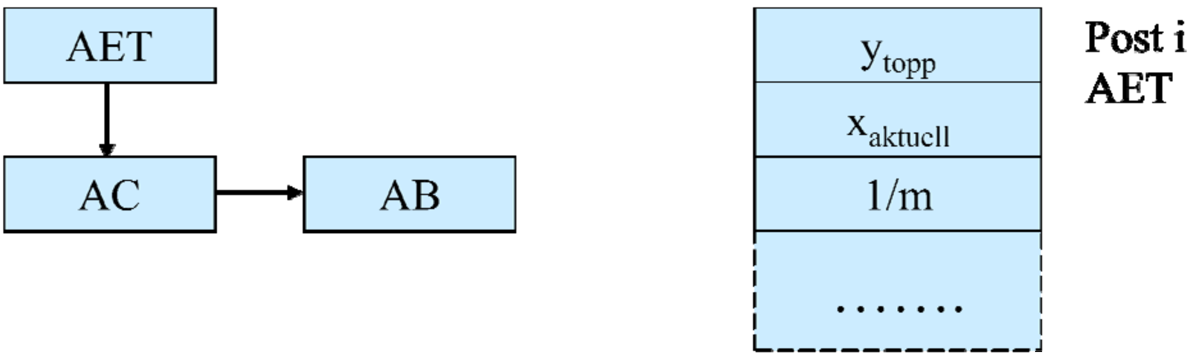
- Starter fra et sted utenfor polygonen
- Initierer en kantkrysseteller til 0
- Går bortover skannlinjen og teller opp kantkrysseselleren med 1 for hver kant som krysses
- Vi er inne i polygonen når kantkrysseselleren har som verdi et odde tall og utenfor når den er et partall.



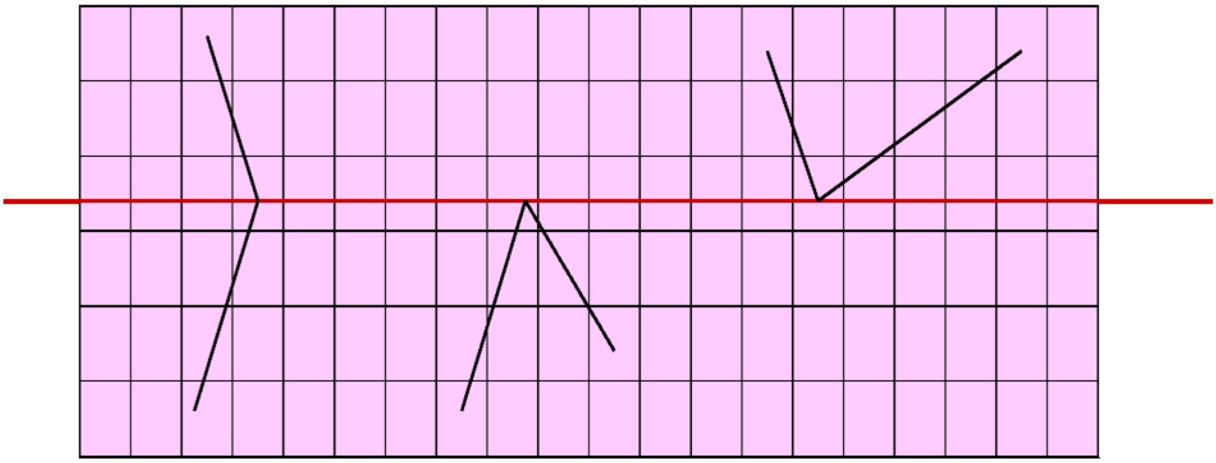
Skannlinjealgoritmen:

- Sorter kantene etter nedre y-verdi
- For hver y-verdi (skannlinje) lag en liste av kanter sortert etter økende nedre x-verdi.
    - Samlingen av disse listene utgjør kantlisten (Edge Table - ET)
    - Lage en tom AET (Active Edge Table)
- For hver skannlinje:
    - Overfør kantposten for vedkommende skannlinje til den aktive kanttabellen (AET)
    - Sørg for at AET forblir sortert med hensyn på $x_{aktuell}$.
    - Fyll skannlinjen ved bruk av paritetsregelen
    - Øk y med 1 (neste skannlinje)
    - Fjern elementer fra AET der $y=y_{topp}$ (kanten er ferdigbehandlet)
    - Inkrementer $x_{aktuell}$ for hver av kantene i AET *
    - Sorter om nødvendig AET med hensyn på $x_{aktuell}$.





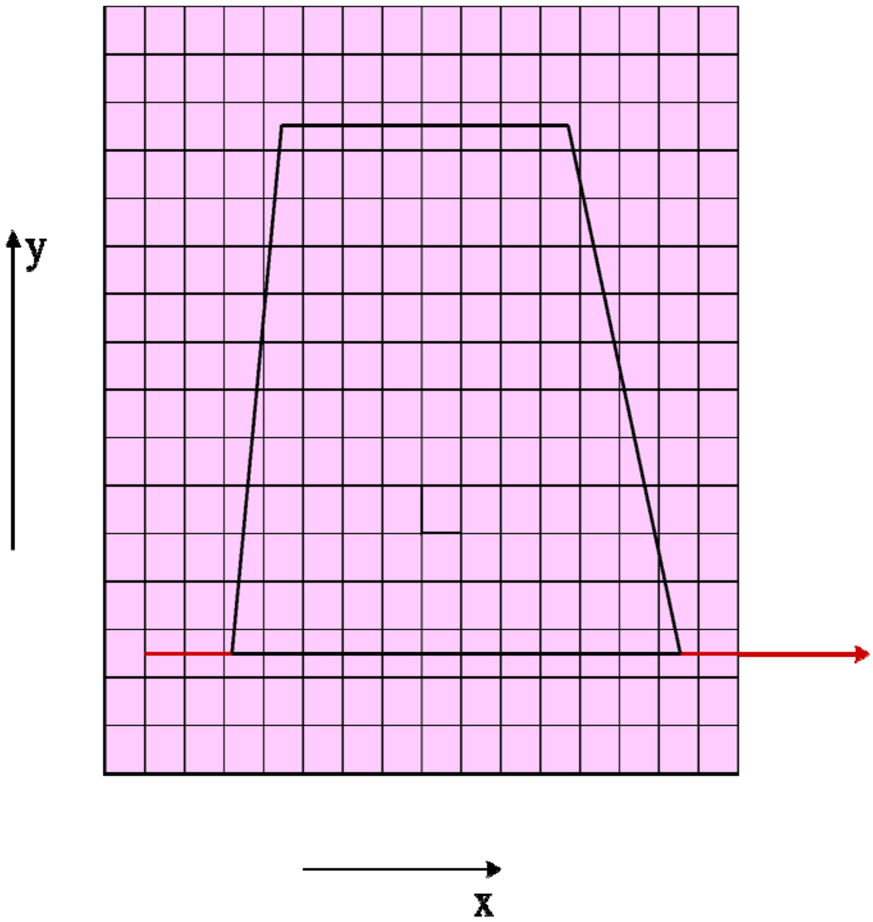
Problemet med hjørner er at skannlinjen når den treffer nøyaktig i hjørnet krysser to kanter. Kantkryssetelleren økes dermed med 2. I nedenstående figur vil dette føre til feiltegning i tilfellet helt til venstre. Dersom polygontegnefargen er på ved kryssing av kanten, vil den forbli på og tegning vil fortsette med feilaktig å gi bakgrunnen polygonfarge. Tilsvarende vil det dersom bakgrunnsfargen er på ved kryssingen av hjørnet, feilaktig ble tegnet med bakgrunnsfarge inne i polygonen. I tilfellet i midten og det til høyre, vil det ikke oppstå problemer.



Dette problemet kan løses på en av følgende måter:

- Ta bort den øverste pikselen fra alle kantene
- Forskyve skannlinjen litt i vertikal retning slik at den ikke treffer rett i hjørnene
- Undersøk om det er kant både over og under skannlinjen og i så fall ta hensyn til det (for eksempel ved å fjerne øverste piksel fra kanten som ligger nederst)

Problemet med horisontale linjer er at det ikke er diskrete skjæringspunkter med skannlinjen.



Dette problemet løses ved å la være å tegne horisontale kanter. Dette realiserer ved å utelukke slike kanter fra kantlisten (ET).

OPPGAVE 4 Bildebehandling – Fourier-transformer og frekvensdomenefiltrering

a) $\underline{F^*(u)} = \frac{1}{N}\sum_{x=0}^{N-1}\exp(-2\pi(-j)ux/N)\mathrm{f}\left(x\right) = \frac{1}{N}\sum_{x=0}^{N-1}\exp(-2\pi j(-u)x/N)\mathrm{f}\left(x\right) = \underline{F(-u)}$

b) $2\pi ux/N = 24\pi x/N$

$u = 12$ is the fundamental.

The harmonics are $u = 12n$.

c) exp or 0.

OPPGAVE 5 Bildebehandling – grunnleggende bildebehandling og filtrering

a) The point spread function of an optical system is the output when it images an ideal point light source on its optical axis.

b) The information content is not increased. The image may be easier to process and is better conditioned for some algorithms.

c) Let T be the Fourier transform. Let $f(x)$ and $g(x)$ be signals in the space domain and $F(u)$ and $G(u)$ be the respective Fourier transforms. Thus $F(u) = T(f(x))$, $G(u) = T(g(x))$.

Let $*$ denote convolution then the convolution theorem states that

$$T(f * g) = F\,G$$

d) The image is blurred. This means that higher spatial frequencies are attenuated relative to the lower spatial frequencies.

OPPGAVE 6 Bildebehandling – segmentering, beskrivelse og gjenkjennelse

a) $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

b) An edge enhanced image has increased intensity in locations where the input image had a transition in intensity. This means that the edges are thick.

The edges need to be thinned and any gaps need to be closed.

Several different steps can achieve this:

1: threshold
2: thin
3: close gaps using heuristics
4: remove hairs using heuristics

or

1: grey level thinning / non-maximal suppression
2: close gaps using heuristics
3: remove hairs using heuristics

c) A chain code is a quantisation of direction into a small number of directions, eg 4, 8 or 16, based on approximating part of a curve by a straight line segment.

The chain code is a sequence of these discrete steps that usually forms a closed path around a shape.

The chain code can represent a sequence of pixels in the same resolution as the pixel grid. The code obtained this way tends to carry a large amount of quantisation noise.

An alternative is to base the chain code on a grid that is more coarse than the original digitisation. The chain code will then have less quantisation noise, but will represent a less precise version of the shape.

d) A linear decision function is a threshold, line, plane or hyper-plane drawn in feature space in 1-D, 2-D, 3-D or more than 3-D respectively.

The decision function partitions the feature space into two half spaces. If a feature point lies in one half space then its identity can definitely be assigned to a class: if it is in the other half space, then its identity can be established by considering other decision surfaces.