



Institutt for datateknikk
og informasjonsvitenskap

**EKSAMEN I EMNE TDT4195
BILDETEKNIKK
ONSDAG 2. JUNI 2010
KL. 09.00 – 13.00**

LØSNINGSFORSLAG

OPPGAVE 1 Grafikk – Avbildning (70 poeng)

a) Deloppgaven kan løses på to måter som begge ansees som fullverdige:

- Metoden som er beskrevet i den nye læreboka. Løsning med denne metoden legges fram først.
- Tidligere kull har lært at transformasjonen som transformerer koordinater fra system A til system B er den samme som den transformasjonen som utført i system A bringer aksene i system B til å falle sammen med aksene i system A. Løsning med denne metoden beskrives til slutt.

Basis i verdenskoordinatsystemet er $[e_x, e_y, e_z, Q]^T$ og i kamerakoordinatsystemet $[e_u, e_v, e_n, P]^T$. I innledningen til oppgaven er følgende gitt:

$$\begin{aligned} \vec{e}_u &= -\vec{e}_z \\ \vec{e}_n &= \frac{\sqrt{3}}{2}\vec{e}_x + \frac{1}{2}\vec{e}_y \\ P &= x_c\vec{e}_x + y_c\vec{e}_y + z_c\vec{e}_z + Q \end{aligned} \quad (1)$$

Videre må vi ha:

$$\vec{e}_v = \vec{e}_n \times \vec{e}_u = -\frac{1}{2}\vec{e}_x + \frac{\sqrt{3}}{2}\vec{e}_y \quad (2)$$

Metoden krever uttrykk for basis i verdenskoordinatsystemet ved basis i kamerakoordinatsystemet.

$$\begin{aligned}
 \vec{e}_x &= \gamma_{11}\vec{e}_u + \gamma_{12}\vec{e}_v + \gamma_{13}\vec{e}_n \\
 \vec{e}_y &= \gamma_{21}\vec{e}_u + \gamma_{22}\vec{e}_v + \gamma_{23}\vec{e}_n \\
 \vec{e}_z &= \gamma_{31}\vec{e}_u + \gamma_{32}\vec{e}_v + \gamma_{33}\vec{e}_n \\
 \vec{Q} &= \gamma_{41}\vec{e}_u + \gamma_{42}\vec{e}_v + \gamma_{43}\vec{e}_n + \vec{P}
 \end{aligned} \tag{3}$$

Dette kan skrives på matriseform:

$$\begin{bmatrix} e_x \\ e_y \\ e_z \\ Q \end{bmatrix} = M \cdot \begin{bmatrix} e_u \\ e_v \\ e_n \\ P \end{bmatrix} \tag{4}$$

$$M = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}$$

Av den andre likningen i (1) og likning (2):

$$\begin{aligned}
 \vec{e}_n &= \frac{\sqrt{3}}{2}\vec{e}_x + \frac{1}{2}\vec{e}_y \\
 \vec{e}_v &= -\frac{1}{2}\vec{e}_x + \frac{\sqrt{3}}{2}\vec{e}_y
 \end{aligned}$$

får vi:

$$\begin{aligned}
 \vec{e}_n + \sqrt{3}\vec{e}_v &= \left(\frac{1}{2} + \frac{3}{2}\right)\vec{e}_y = 2\vec{e}_y \\
 \vec{e}_y &= \frac{\sqrt{3}}{2}\vec{e}_v + \frac{1}{2}\vec{e}_n
 \end{aligned} \tag{5}$$

og:

$$\underline{\vec{e}_x} = -2\vec{e}_v + \sqrt{3}\vec{e}_y = -2\vec{e}_v + \sqrt{3}\left(\frac{\sqrt{3}}{2}\vec{e}_v + \frac{1}{2}\vec{e}_n\right) = \underline{-\frac{1}{2}\vec{e}_v + \frac{\sqrt{3}}{2}\vec{e}_n} \tag{6}$$

Av den første og den tredje likningen i (1) får vi til slutt:

$$\underline{\vec{e}_z = -\vec{e}_u} \quad (7)$$

$$\begin{aligned} \underline{Q} &= -x_c \vec{e}_x - y_c \vec{e}_y - z_c \vec{e}_z + P = \\ &= -x_c \left(-\frac{1}{2} \vec{e}_v + \frac{\sqrt{3}}{2} \vec{e}_n \right) - y_c \left(\frac{\sqrt{3}}{2} \vec{e}_v + \frac{1}{2} \vec{e}_n \right) - z_c (-\vec{e}_u) + P = \\ &= \underline{z_c \vec{e}_u + \frac{1}{2} (x_c - y_c \sqrt{3}) \vec{e}_v - \frac{1}{2} (x_c \sqrt{3} + y_c) \vec{e}_n + P} \end{aligned} \quad (8)$$

Et objektpunkt har representasjonen $a = [x_p \quad y_p \quad z_p \quad 1]^T$ i verdenskoordinatsystemet. I kamerakoordinatsystemet har det samme punktet representasjonen $b = [u_p \quad v_p \quad n_p \quad 1]^T$. Vi får:

$$a^T \begin{bmatrix} e_x \\ e_y \\ e_z \\ Q \end{bmatrix} = a^T \cdot M \cdot \begin{bmatrix} e_u \\ e_v \\ e_n \\ P \end{bmatrix} = b^T \cdot \begin{bmatrix} e_u \\ e_v \\ e_n \\ P \end{bmatrix}$$

som gir:

$$a^T \cdot M = b^T$$

$$\underline{b = M^T \cdot a} \quad (9)$$

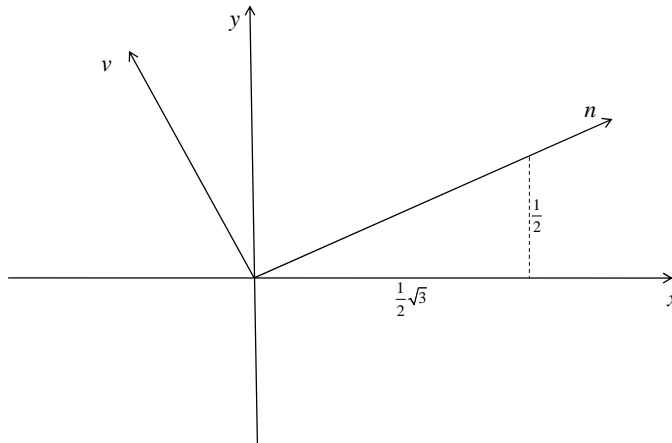
Matrisen som konverterer koordinater i verdenskoordinatsystemet til koordinater i kamerakoordinatsystemet er altså M^T . Fra (4) får vi:

$$M_{w \rightarrow c} = M^T = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \gamma_{31} & \gamma_{41} \\ \gamma_{12} & \gamma_{22} & \gamma_{32} & \gamma_{42} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} & \gamma_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vi setter inn koeffisientene vi har funnet i likningene (5) – (8) og får den søkte matrisen:

$$\underline{\underline{M_{w \rightarrow c}}} = \begin{bmatrix} 0 & 0 & -1 & z_c \\ -\frac{1}{2} & \frac{1}{2}\sqrt{3} & 0 & \frac{1}{2}(x_c - y_c \sqrt{3}) \\ \frac{1}{2}\sqrt{3} & \frac{1}{2} & 0 & -\frac{1}{2}(x_c \sqrt{3} + y_c) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 0 & -2 & 2z_c \\ -1 & \sqrt{3} & 0 & x_c - y_c \sqrt{3} \\ \sqrt{3} & 1 & 0 & -x_c \sqrt{3} - y_c \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (10)$$

I stedet for å utlede uttrykkene for e_x og e_y (likningene (5) og (6)) kan en utnytte at planet $u = 0$ i kamerakoordinatsystemet er parallelt med planet $z = 0$ i verdenskoordinatsystemet. I et plan vinkelrett på z -aksen (og u -aksen) ser projeksjonene av aksene slik ut:



Resultatene i linkningene (5) og (6) kan avledes fra denne figuren. Verdt å merke er også at vinkelen mellom x - og n -aksene er 30° .

Løsning med "gammel" metode:

Den søkte transformasjonen er den samme som den transformasjonen som skal til for å flytte kamerakoordinatsystemet slik at aksene faller sammen med verdenskoordinatsystemets akser:

1. Transler origo i kamerakoordinatsystemet slik at det faller sammen med origo i verdenskoordinatsystemet.
2. Rotere slik at u , v og n -aksene i kamerakoordinatsystemet faller sammen med x , y og z -aksene i verdenskoordinatsystemet.

Translasjonen:

$$M_1 = \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Rotasjonen:

Rene rotasjonsmatriser er ortogonale. Den enkleste måten å få fram den søkte rotasjonsmatrisen på, er å utnytte den egenskapen ved slike matriser at linjevektorene er

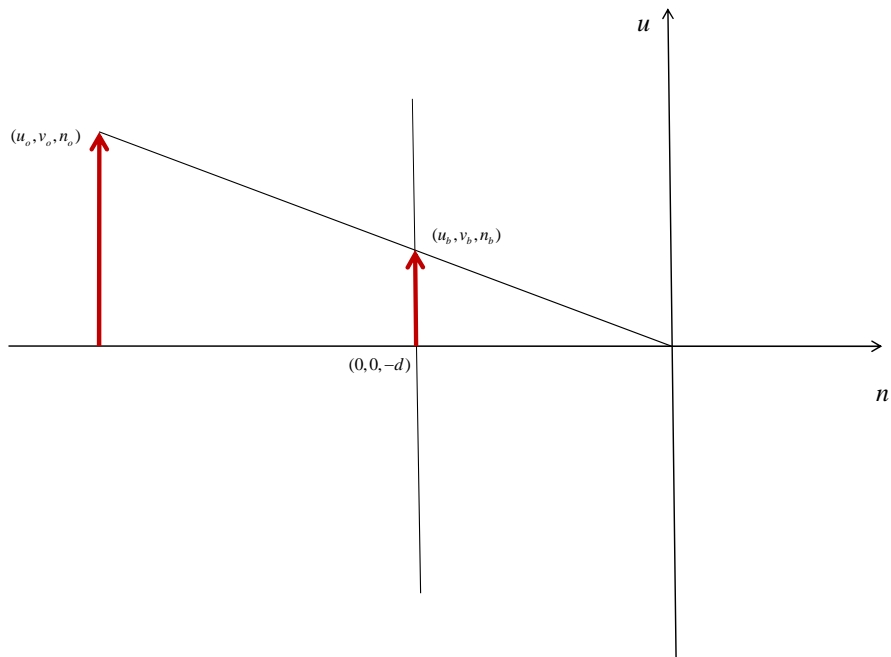
ortonormale og at de dermed transformerer seg selv til ortonormale vektorer med bare en komponent forskjellig fra 0. Dersom vi formulerer akseenhetsvektorene i kamerakoordinatsystemet med sine komponenter langs aksene i verdenskoordinatsystemet, kan disse være de ortonormale linjevektorene i rotasjonsmatrisen. Matrisen vil da rotere aksene i kamerakoordinatsystemet til å falle sammen med aksene i verdenskoordinatsystemet. To av de søkte vektorene i dekomponert form er gitt i oppgaveteksten som gjengitt i de to første uttrykkene i likning (1). Den tredje vektoren er resultatet i likning (2). Rotasjonsmatrisen er:

$$M_2 = \begin{bmatrix} e_{ux} & e_{uy} & e_{uz} & 0 \\ e_{vx} & e_{vy} & e_{vz} & 0 \\ e_{nx} & e_{ny} & e_{nz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Matrisen for konvertering av koordinater (og vektorer) fra verdenskoordinatsystemet til kamerakoordinatsystemet blir da:

$$\begin{aligned} \underline{\underline{M_{w \rightarrow c}}} &= M_2 \cdot M_1 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 0 & 0 & -2 & 2z_c \\ -1 & \sqrt{3} & 0 & x_c - y_c \sqrt{3} \\ \sqrt{3} & 1 & 0 & -x_c \sqrt{3} - y_c \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{aligned} \quad (13)$$

b) I denne deloppgaven arbeider vi i kamerarommet:



Med utgangspunkt i figuren gir likedannede trekkanter oss:

$$\frac{u_b}{u_o} = \frac{-d}{n_o}$$

$$u_b = \frac{u_o}{\left(\frac{n_o}{-d}\right)} \quad (14)$$

Tilsvarende:

$$v_b = \frac{v_o}{\left(\frac{n_o}{-d}\right)} \quad (15)$$

Dessuten har vi:

$$n_b = -d \quad (16)$$

På matriseform:

$$\begin{bmatrix} u_b \\ v_b \\ n_b \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{-d} & 0 \end{bmatrix} \begin{bmatrix} u_o \\ v_o \\ n_o \\ 1 \end{bmatrix} \Rightarrow w = \frac{n_o}{-d}$$

Den søkte avbildningsmatrisen er:

$$M_{proj} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{-d} & 0 \end{bmatrix} \quad (17)$$

c) I kamerakoordinatsystemet blir bildpunktet i homogene koordinater:

$$\begin{bmatrix} u_b \\ v_b \\ n_b \\ w \end{bmatrix} = M_{proj} \cdot M_{w \rightarrow c} \cdot \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \quad (18)$$

I verdenskoordinatsystemet blir bildepunktets posisjon i homogene koordinater:

$$\begin{bmatrix} x_b \\ y_b \\ z_b \\ w \end{bmatrix} = M_{w \rightarrow c}^{-1} \cdot M_{proj} \cdot M_{w \rightarrow c} \cdot \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \quad (19)$$

$M_{w \rightarrow c}^{-1}$ kan bestemmes ved invertering eller på samme måte som vi i deloppgave a) fant $M_{w \rightarrow c}$. Dette er ikke krevd i oppgaven.

OPPGAVE 2 Grafikk – Farger og lys**(70 poeng)**

a) Svar på spørsmålene:

1. I øyet er det tre type av celler som virker ved høye lysnivå (dagslys). Noe forenklet virker disse celletypene på henholdsvis rødt, grønt og blått lys. Tristimulisystemet benytter kombinasjoner av lys av rød, grønn og blå farge for å tilpasse fargeinntrykk etter hva disse celletypene oppfatter.
2. En fargegamut spenner ut et område i det totale fargerommet. Fargegamuten er spesifikk for et bestemt utstyr. De fargene som ligger innenfor dette området, kan framstilles av utstyret. I tristimulisystemet utgjør en RGB-fargegamut en trekant i CIEs kromasitetsdiagram.
3. Additive fargemodeller anvender kombinasjoner av lyskilder for å framstille en bestemt farge. Modellen brukes blant annet av monitører og ved projeksjon. Subtraktive fargemodeller brukes blant annet ved trykking der trykkefargen utgjør et filter som tar fargekomponenter bort fra det hvite lyset som reflekteres fra papiroverflaten.
4. A i RGBA står for alfa. Som fargekomponent benyttes den til å angi grad av transparens. Mulig bruk av transparens er å framstille tåkeeffekter eller til å kombinere bilder for eksempel ved å la et bilde dø ut mens et annet overtar.
5. K i CMYK står for svart. Svart brukes som en fjerde fargekomponent. Grunnen er at det er vanskelig å framstille dekkende svart ved å kombinere cyan, magenta og gul.

b) Typer av lyskilder som brukes i datagrafikk, er:

- Punktllys
- Lyskilder med utbredelse
- Spotlys
- Fjerne lyskilder
- Bakgrunnslys

Punktllys:

Lys fra en punktformig lyskilde svekkes med kvadratet av avstanden d til lyskilden. Belysningen av et punkt er:

$$E(p, p_0) = \frac{1}{d^2} I(p_0) \quad \text{med } d = |p - p_0| \quad (20)$$

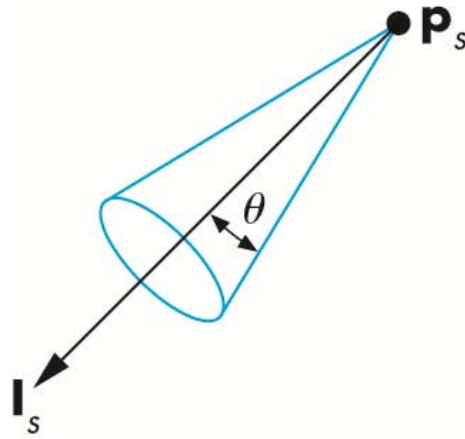
der p er objektpunkt, p_0 er lyskildens posisjon og $I(p_0)$ er lyskildens styrke.

Lyskilder med utbredelse:

$$E(p, p_0) = \min\left(1, \frac{1}{a + bd + cd^2}\right) I(p_0) \quad (21)$$

At faktoren for lysstyrken ikke kan bli større enn 1, sikrer at en ikke får forsterkning av lyskilden i det tilfellet at nevneren i brøken skulle bli liten. Leddet bd modellerer utstrekningen av lyskilden mens a er en gardering mot at brøken blir urimelig stor.

Spotlys:



Lysstyrken avtar med økende vinkel θ :

$$I_s \propto \cos(\theta) = s \cdot l_s \quad (22)$$

Fjerne lyskilder:

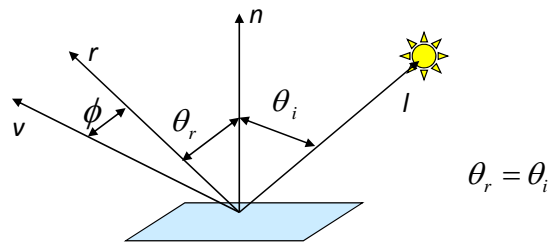
Lyskilden har fast styrke og karakteriseres ved stråleretningen:

$$p_0 = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \quad (23)$$

Bakgrunnslys:

Bakgrunnslys er retningsuavhengig og har en fast styrke.

- c) For et perfekt speil er refleksjonsvinkel θ_r lik innfallsvinkel θ_i . For en mindre perfekt blank flate spres lyset i en kjegle rundt retningen for den perfekte refleksjonen.



Lysstyrken avtar med økende avviksvinkel ϕ . I Phongs refleksjonsmodell benyttes refleksjonskoeffisienten:

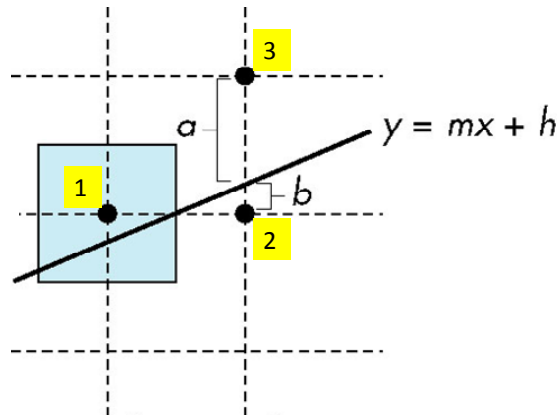
$$R_{i\lambda} = k_\lambda \cos^\alpha(\phi) \quad (24)$$

der k_λ og α er konstanter. Flater med stor spredning rundt den ideelt reflekterte strålen har liten α . Dess bedre flaten er som speil, dess større er α . Refleksjonskoeffisienten er bølgelengdeavhengig. Det vil i praksis si at det benyttes egne verdier for hver av RGB-komponentene.

OPPGAVE 3 Grafikk – Rasterisering

(80 poeng)

- a) Bresenhams algoritme reduserer oppgaven med å rasterisere en rett linje til en prosedyre med heltallsinkrementering. Likningen for den rette linjen er $y = mx + h$.



Vi forutsetter at linjen starter og slutter i punkt med heltallskoordinater henholdsvis (x_1, y_1) og (x_2, y_2) og at $x_2 > x_1$.

Metoden kan med nødvendig tilpasning brukes for alle verdier av stigningsforholdet m . Som illustrasjon her ser vi på linjer der $0 \leq m \leq 1$. Vi tegner da med enhetssteg, $\Delta x = 1$, i x -retningen. Se ovenstående figur. Siste piksel som er tegnet er merket 1. Neste piksel vil være enten det som er merket 2 eller det som er merket 3. Den grunnleggende ideen i Bresenhams algoritme er:

- Definisjon av desisjonsvariabelen $d = b - a$. Dersom $d < 0$ passerer linjen nærmest punkt 2 og dette velges. I motsatt fall tegnes punkt 3.
- Redefinisjon av desisjonsvariabelen til $d = (x_2 - x_1)(b - a)$ gjør d til et heltall.
- Beregning av ny desisjonsverdi for neste skritt beregnes ved inkrementering.

- b) I Cohen-Sutherlands algoritme for linjeklipping brukes utkastingskodene til å bestemme om en linje trivielt kan forkastes eller aksepteres. Algoritmen kan brukes på rektangulære klippevinduer. Planet deles i ni regioner ved å forlenge vinduets kanter:

1001	1000	1010	$y = y_{\max}$
0001	0000	0010	
0101	0100	0110	$y = y_{\min}$
$x = x_{\min}$ $x = x_{\max}$			

Utkastingskoden er en firesifret binær kode. Regionen med kode 0000 er klippevinduet. Linjenes endepunkter gis kode etter den regionen de ligger i.

- Dersom begge kodene til et linjestykke er 0, ligger hele linjen inne i vinduet og blir trivielt akseptert
- Ellers, dersom en bitvis og-operasjon gir et resultat forskjellig fra 0, ligger hele linjestykket utenfor vinduet og blir forkastet
- Ellers må linjestykket klippes mot en (forlenget) vinduskant og det gjenværende linjestykket testes

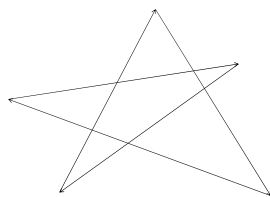
- c) Krysningstesten:

Vi har et punkt som skal undersøkes for å finne ut om det ligger inne i eller utenfor en polygon. En stråle skytes fra punktet og antall krysninger med polygonens kanter telles. Dersom tellingen ender opp med et odde tall, ligger punktet inne i polygonene. Ellers ligger det utenfor.

Vindingstesten:

Polygonens kanter gis retning slik at et hjørne er endepunkt for en kant og startpunkt for neste. Også her sender vi ut en stråle fra punktet som skal testes. Mest hensiktsmessig er strålen langs en skannlinje. Vi nullstiller en teller og teller opp for krysninger med kanter i retning med klokka og teller ned for krysning med kanter i retning mot klokka. Dersom telleren ender opp med 0, ligger punktet utenfor polygonen. Ellers ligger punktet inne i polygonen.

Dersom vi bruker disse testene til polygonfylling, vil vindingstesten helt fylle stjernen i figuren nedenfor mens krysningstesten ikke vil fylle midtpartiet i stjernen:



- d) Baksiddefjerning er å utelukke polygoner som vender bort fra øyepunktet. Polygoner som skal testes, har en flatenormal. Dersom skalarproduktet av denne normalen og en normal fra øyepunktet til polygonen er positivt, vender polygonen bort og kan utelukkes.

Baksiddefjerning er en beregningsmessig billig måte til å eliminere polygoner som ikke er synlige. Derfor kan det være hensiktsmessig å forprosessere en scene med baksiddefjerning før en anvender en mer beregningstung algoritme for bestemmelse av synlige flater.

OPPGAVE 4 Bildebehandling – Grunnleggende begreper (80 poeng)

- a) The psf is the image of an ideal point light source formed by the lens system. The lens system is assumed to be focused correctly. The point spread function is the response by the lens system to a delta function.
- b) The capturing of an image requires quantisations in space and brightness.
- c) Precision is improved when the quantisation is made more fine.

For spatial quantisation this means that the size of each pixel should be reduced, ie. the image should be based on a more fine pixel grid. For example, the pixel dimensions should be increased from 512x512 to 2048x2048.

For brightness quantisation the dynamic range of brightness should be increased. This means that the number of bits needs to be increased from the usual value of 8. A 16 or 32 bit integer pixel or a floating point representation of a pixel would give an increase in dynamic range of brightness. Similar comments apply to all of the colour planes of a colour image.

- d) Show a chain of steps similar to: capture; pre-processing; segmentation; feature extraction; pattern recognition.

OPPGAVE 5 Bildebehandling – Regionbaserte metoder (80 poeng)

- a) Thresholding is an example of a classification procedure. This is usually a two class problem, but may be a choice between a small number, e.g. 3, 4 classes.

Thresholding is usually considered to be a segmentation technique in image processing, though it is, in fact, classifying pixels, usually as foreground or background.

Thresholding is a decision based on the value of a numerical attribute of the pixel. If the attribute is more than T , then the pixel is class 1 e.g. foreground, if it is less than or equal to T then the pixel is class 2, eg. background.

When the numerical attribute is the pixel's brightness value, then a simple model of image formation is being assumed, i.e. that the foreground pixels are brighter than the background. The validity of this model may be strengthened by setting up the image capture operation appropriately.

It is possible to calculate the numerical attribute of each pixel in many different ways. These can take account of local variations in the image. In principle, remote variations in pixel value could also be included, but this is rare.

The response to local variations is often implemented as a local variation of the threshold value.

- b) Simple thresholding, dynamic thresholding, Calvard-Ridler thresholding, Otsu's method.
- c)
- d) Euler number, area, moments, compactness, signature, Fourier descriptor...

OPPGAVE 6 Bildebehandling – Fourierdomenemetoder (60 poeng)

- a) $f(x, y)$ is assumed to be periodic, with period N , in both x and y .
- b)

$$G(u, v) = \sum_0^{N-1} \sum_0^{N-1} e^{-\frac{2\pi j(ux+vy)}{N}} f(x - x_0, y - y_0)$$

Substitute $xx = x - x_0$ and $yy = y - y_0$

$$G(u, v) = \sum_{-x_0}^{N-1-x_0} \sum_{-y_0}^{N-1-y_0} e^{-\frac{2\pi j(u(xx+x_0)+v(yy+y_0))}{N}} f(xx, yy)$$

$$G(u, v) = e^{-\frac{2\pi j(ux_0+vy_0)}{N}} \sum_{-x_0}^{N-1-x_0} \sum_{-y_0}^{N-1-y_0} e^{-\frac{2\pi j(u\cdot xx+v\cdot yy)}{N}} f(xx, yy)$$

The next step is to use periodicity of the transform to correct the range of summations. It is unlikely that the students will get this far.

- c) The summation indices start from zero and increase. These are exactly the zero frequency and low frequency terms. Thus low pass filtering can be implemented by taking the first few terms in the summation.