

## INF 2310, Obligatorisk oppgave nr 1, våren 2010

Dette oppgavesettet er på tre sider og består av to bildebehandlingsoppgaver.

Besvarelsen av denne og neste obligatoriske oppgave må være godkjent for at du skal få anledning til å gå opp til endelig skriftlig eksamen i kurset.

Besvarelsene kan utarbeides i smågrupper på opptil to studenter, men det er ikke noe i veien for å arbeide alene. Studenter i samme smågruppe kan levere identisk besvarelse, men samarbeidet må framgå av navnene på forsiden av besvarelsen.

Av side 1 skal det fremgå hvem som har utarbeidet besvarelsen. Her skal det også settes inn følgende formulering:

*Jeg/vi har lest og forstått reglene som er gitt i dokumentet  
«Krav til innleverte oppgaver ved Institutt for informatikk»  
på <http://www.ifi.uio.no/studinf/skjemaer/egenerklaring.pdf>*

Det forventes at arbeidet er et resultat av egen innsats. Å utgi andres arbeid for sitt eget er uetisk og kan medføre sterke reaksjoner fra Ifis side.

Den skriftlige rapporten, programkode og resultatbildene skal sendes per e-post til gruppelæreren. Filene skal være pakket til én fil, f.eks. med zip, med følgende navn: *inf2310-oblig1-brukernavn.zip* der brukernavn byttes ut med ditt eget brukernavn. Filen sendes som vedlegg i en e-post til [martiert@student.matnat.uio.no](mailto:martiert@student.matnat.uio.no). Husk å sette «INF2310 Oblig1» i subjekt-feltet! Ved ønske om annen leveringsform, f.eks. å kunne levere rapporten på papir, kontaktes gruppelærer.

Bildene det refereres til vil være å finne under:

<http://www.uio.no/studier/emner/matnat/ifi/INF2310/v10/undervisningsmateriale/bilder/>

Oppgaven utleveres mandag 22. februar 2010.  
Innleveringsfrist er fredag 12. mars 2010.

## Oppgave 1 Preprosessering av portrett for ansiktsgjenkjenning

*Viktig: Dere skal her programmere transformene “fra bunnen av”, altså ikke bruke ferdige programpakker.*

Denne oppgaven går ut på å klargjøre (preprosessere) portrettbilder slik at det senere kan sendes til en ansiktsgjenkjenning algoritme.

Bildet vi skal jobbe med er *portrett.png*. Som vi ser er bildet både skjevt og det har lav kontrast. For å kunne sammenligne dette bildet med andre bilder i en database, er man nødt til å standardisere både geometrien og kontrasten. I denne oppgaven skal vi gjøre dette på følgende måte:

1. Standardisere kontrasten ved å gjøre en histogramutjevning.
2. Standardisere geometrien ved å utføre en affin transform slik at øyne og munn passer over en forhåndsdefinert maske. Masken er gitt ved filen *geometrimaske.png*.

Lag en implementasjon av både forlengs- og baklengstransformasjon. Ved baklengstransformasjonen, prøv ut både nærmeste nabo og bilineær interpolasjon.

Dere skal altså ende opp med et resultatbilde som er like stort som maske-bildet (512x600), har god kontrast, og hvor øyne og munn finnes på de samme pikselkoordinatene som i maske-bildet.

Hva skal leveres:

- 1) Mellom-resultat-bildet etter histogramutjevningen.
- 2) Forklaring på hvordan dere fant koeffisientene til den affine transformen.
- 3) Resultat-bilder for både forlengs- og baklengstransformasjonen, samt nærmeste nabo og den bilineære interpolasjonen.
- 4) Kommentarer/forklaringer på eventuelle forskjeller i resultatene ved forlengs- og baklengsmapping, og ved nærmeste nabo og bilineær interpolasjon.
- 5) Programkode legges i en egen mappe.

**NB!**

*Dere kan benytte ferdige Matlab-funksjoner til å lese/skrive fra/til fil.*

*Histogramutjevningen og den geometriske transformen MÅ dere implementere selv.*

## Oppgave 2 Gradient-deteksjon

Anta at vi har et bilde med noe støy. Dere har sett at vi kan finne kanter i slike bilder ved hjelp av gradient-operatorer. Vi skal her bare se på gradientmagnituden - ikke på gradient-retningen.

a) En mulig gradient-operator er den vanlige Sobel-operatoren:

$$H_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad H_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

En ulempe med denne operatoren er at den kan gi for brede kanter i forhold til størrelsen på strukturene i bildet, for eksempel for tynne linjer.

b) Alternativt kan input-bildet filtreres med en modifisert Sobel-operator som består av fire filtermasker:

$$H_{x1} = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 2 & -2 \\ 0 & 1 & -1 \end{pmatrix}, \quad H_{x2} = \begin{pmatrix} 1 & -1 & 0 \\ 2 & -2 & 0 \\ 1 & -1 & 0 \end{pmatrix}, \quad H_{y1} = \begin{pmatrix} -1 & -2 & -1 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad H_{y2} = \begin{pmatrix} 0 & 0 & 0 \\ -1 & -2 & -1 \\ 1 & 2 & 1 \end{pmatrix}$$

For hver piksel summeres bidragene fra de fire filtrene til en gradient-magnitude:

$$G(x, y) = [g_x^2(x, y) + g_y^2(x, y)]^{1/2}$$

der

$$g_x^2(x, y) = g_{x1}^2(x, y) + g_{x2}^2(x, y), \quad g_y^2(x, y) = g_{y1}^2(x, y) + g_{y2}^2(x, y)$$

$$g_{x1} = H_{x1} * f, \quad g_{x2} = H_{x2} * f, \quad g_{y1} = H_{y1} * f, \quad g_{y2} = H_{y2} * f$$

Programmer en implementasjon av både alternativ a: «vanlig Sobel-operator» og b: «modifisert Sobel-operator». Sammenlign resultatet ved bruk av de to alternativene på bildet *shapes\_noise15.png*.

Drøft hvordan de to alternativene vil oppføre seg på linjer i binære bilder, der linjetykkelsen er henholdsvis 1, 2 og 3 piksler.

Hva skal leveres:

- 1) Resultat-bildet fra alternativ a).
- 2) Resultat-bildet fra alternativ b).
- 3) Drøfting av resultatene fra henholdsvis alternativ a) og b) på bildet.
- 4) Drøfting av resultatet av alternativ a) og b) på tynne linjer i binære bilder.
- 5) Programkode legges i en egen mappe.

**NB!** 2D-konvolusjonen må dere implementere selv (med «for-løkker»).

Lykke til!