Repetisjon før midtveiseksamen i INF2310

March 15, 2019

Sampling og kvantisering - Samplingsteoremet

For å kunne rekonstruere det vi avbilder med største frekvens f_{max}, må samplingsraten $f_s = \dfrac{1}{T_s}$ være større enn $2f_{\mathrm{max}}$:

$$f_s > 2f_{\mathrm{max}} \Leftrightarrow \frac{1}{2}T_{\mathrm{max}} > T_s$$

Sampling og kvantisering - Undersampling/aliasing

- Hvis samplingsraten ikke oppfyller samplingsteoremet, vil vi få aliasing.
- **Anti-aliasing** kan gjøres før sampling for å redusere effekten av aliasing

Sampling og kvantisering - Kvantisering

- Vi kan måle alle mulige verdier under avbildning.
- En digital representasjon av målingene trenger en begrensing på hvor mange verdier vi ønsker å beskrive målingene våres med.
- **Kvantisering** er den prosessen som transformerer verdier i spesifiserte intervall til èn verdi per intervall.

Geometriske operasjoner - Transformasjon av posisjonene til pikslene

Flytter på piksel posisjoner (x, y) til (x', y') ved bruk av transformasjoner $T_x(x, y)$ og $T_y(x, y)$:

$$x' = T_x(x, y)$$
$$y' = T_y(x, y)$$

Affine transformasjoner:

$$x' = a_0 x + a_1 y + a_2$$
$$y' = b_0 x + b_1 y + b_2$$

Koeffsientene a_0, a_1, a_2 og b_0, b_1, b_2 bestemmer hvordan vi skal flytte hhv. $x-$ og $y-$koordinatene.

Geometriske operasjoner - Forlengs-mapping

Mappingen ser på hvilken posisjon (x', y') vi kommer til dersom innbildets koordinater (x, y) blir transformert.

Algorithm 1 Forlengs-mapping ved bruk av affin transformasjon

```
for alle koordinater (x, y) i innbildet do
    x' = a_0 x + a_1 y + a_2
    y' = b_0 x + b_1 y + b_2
    if koordinatene (x', y') passer i utbildet then
        utbilde(x', y') = f(x, y)
    end if
end for
```

- Utbilde har ikke verdier på alle posisjoner (x', y').
- Pikselverdien (x', y') kan bli satt flere ganger
- Transformasjoner av posisjoner gjøres uten garanti for at de brukes i utbilde

Geometriske operasjoner - Baklengs-mapping

Hvis vi kjenner posisjonene (x', y') til utbildet - fra hvilken koordinat (x, y) kom de fra i innbildet?

Vi må gjøre en "motsatt" transformasjon. Kan finne f.eks invers av transform-matrisen.

- Hvis affin transform; Nok å ta invers av transform-matrisen for å finne de inverse koeffsientene a'_0, a'_1, a'_2 og b'_0, b'_1, b'_2.

Algorithm 2 Baklengs-mapping ved bruk av affin transformasjon

```
for alle koordinater (x', y') i utbildet do
    x = a'_0 x' + a'_1 y' + a'_2
    y = b'_0 x' + b'_1 y' + b'_2
    if koordinatene (x, y) passer i innbildet then
        utbilde(x', y') = interpolert verdi innbilde(x, y)
    else
        utbilde(x', y') = 0
    end if
end for
```

- Interpolasjon for å finne $\mathrm{innbilde}(x, y)$, f.eks nærmeste nabo eller bilineær interpolasjon.

Gråtonemapping - Histogram

- Histogram:

$$h(i) = \text{antall piksler med intensitet } i$$

- Normalisert histogram av $m \times n$ stort bilde:

$$p(i) = \frac{1}{m \times n} h(i)$$

Gråtonemapping - Lineær gråtonetransformasjon

$$T[i] = ai + b$$

- **Mer kontrast:** $a > 1$
- **Mindre kontrast:** $a < 1$

- **Lysere bilde:** $b > 0$
- **Mørkere bilde:** $b < 0$

Gråtonemapping - Middelverdi og varians

Middelverdi til med G gråtoner og normalisert histogram $p(i)$:

$$\mu = \sum_{i=0}^{G-1} ip(i)$$

forteller noe om hvilken verdi bildet sannsynligvis inneholdes mest av.

Varians til med G gråtoner og normalisert histogram $p(i)$:

$$\sigma^2 = \sum_{i=0}^{G-1} p(i)(i-\mu)^2$$

forteller noe om hvor mye spredning det er mellom verdiene og middelverdi til bildet.

Gråtonemapping - Hvordan en gråtonetransformasjon påvirker bildets middelverdi og varians

Se på hvordan

$$\mu_T = \sum_{i=0}^{G-1} T[i]p(i)$$

og

$$\sigma_T^2 = \sum_{i=0}^{G-1} p(i)(T[i] - \mu)^2$$

blir uttrykt ved bildets opprinnelige middelverdi μ og varians σ^2

Gråtonemapping - Ikke-lineære transformasjoner

- **Logaritmisk mapping:** $T[i] = \log(i)$,
  strekker ut mørke områder
- **Eksponensiell mapping:** $T[i] = \exp(i)$,
  strekker ut lyse områder
- **Power-law:** $T[i] = i^{\gamma}$ for en gitt γ

Histogrambaserte operasjoner - Kumulativ histogram

Det normaliserte kumulative histogram til et $m \times n$ bilde med G gråtoner og histogram $h(i)$:

$$
\begin{aligned}
c(j) &= \frac{1}{m \times n} \sum_{i=0}^{j} h(i) \\
&= \sum_{i=0}^{j} p(i)
\end{aligned}
$$

Histogrambaserte operasjoner - Histogramutjevning

- Ønsker å flytte histogramsøyler til et bilde slik at de dekker et ønsket intervall av gråtoner. Kontrasten maksimeres.

Vi må finne en transform som transformerer bildets histogram til et uniformt histogram så godt det lar seg gjøre!

- Normalisert kumulativt histogram til uniformt histogram $c_{\text{ønsket}}(i) = \frac{i}{G}$.
- Vil finne $T[i]$ slik at $c_{bilde}(i) = c_{\text{ønsket}}(T[i]) = \frac{T[i]}{G}$.
- Dette gir:

$$T[i] = Gc_{bilde}(i)$$

Histogrambaserte operasjoner - Histogramtilpasning

Formen på ønsket histogram kan være hva som helst.

- Bildets normaliserte kumulative histogram $c_{\mathrm{bilde}}(i)$
- Ønsket normalisert kumulativ histogram $c_{\mathrm{ønsket}}(i)$
- Vi må finne en transformasjon $T[i]$ slik at $c_{\mathrm{bilde}}(i) = c_{\mathrm{ønsket}}(T[i])$ holder så godt som mulig

Algorithm 3 Histogramtilpasning

1: **for** $i = 0, \ldots, G-1$ **do**
2: Finn j der $|c_{\mathrm{bilde}}(i) - c_{\mathrm{ønsket}}(j)|$ er minst for $j = 0, \ldots, G-1$.
3: Sett $T[i] = j$
4: **end for**

Histogrambaserte operasjoner - Lokal gråtonetransformasjon

- Utfører en gråtonetransformasjon over mindre områder i bildet.
- Transformasjonen er gjerne avhengig av pikselverdiene innad i området.
- Kan standardisere middelverdi og varians innad et område slik at bildet får lik lyshet og kontrast.

Naboskapsoperasjoner I - Filter

Brukes til å prosessere bilde, og definerer bildet blir prosessert.

$$\text{Filter} = \text{operasjon} + \text{naboskap}$$

Naboskapsoperasjoner I - Konvolusjon

1. Rotèr filteret 180°.
2. Der et er noe overlapp mellom bildet og filteret; punktvis multipliser verdiene og summèr de resultarende verdiene etter punktvis multipkasjon. Lagre resultatet i en matrise.
3. Gjenta steget over frem til alle piksler der det er overlapp mellom bildet og filteret har blitt besøkt.

Matematisk formulering:

$$f * h = \sum_{s=-a}^{a} \sum_{t=-b}^{b} h(s,t) f(x-s, y-t)$$

Naboskapsoperasjoner I - Bilderandproblemet

Hvordan skal bildet utvides når filteret havnet utenfor?

- Kan utføre konvolusjon kun der det er fullt overlapp. Får et mindre utbilde enn innbildet.
- Kan utvide randen ved bl.a
  1. Nullutviding/zero-padding: Legge på 0-ere langs rand
  2. Bruke nærmeste piksel verdi
  3. Speile bildet om kantene langs randen
  4. Anta bildet er periodisk

  Kan nå få et utbilde like stort som innbildet.

Naboskapsoperasjoner I - Lavpassfiltre

Demping av hurtige variasjoner i bildet.

- Ønsker å redusere støy og finne større objekter

Naboskapsoperasjoner I - Seperable filtre

- Mål: Redusere mengden operasjoner og derav kjøretid.
- Uttrykke et 2D filter som konvolusjon av 1D - filtre

Naboskapsoperasjoner II - Høypassfiltre

Fremheve skarpe variasjoner i bildet, som bl.a kanter og andre detaljer.

Naboskapsoperasjoner II - Approksimasjon av gradient

- Numerisk derivasjon
- Gjøre støy-robust ved å lavpassfiltrere langs èn retning, for så approksimere derivert langs ortogonal retning.

Finner de horisontale kantene til bildet f ved $g_x = h_x * f$, der h_x er et høypassfilter som tilnærmer derivert langs horisontal retning.

Finner de vertikale kantene til bildet f $g_y = h_y * f$, der h_y er et høypassfilter som tilnærmer derivert langs vertikal retning.

- Gradient-magnitude: $M(i,j) = \sqrt{g_x^2(i,j) + g_y^2(i,j)}$
- Gradient-retning: $\theta(i,j) = \text{atan}\left(\frac{g_y(i,j)}{g_x(i,j)}\right)$

Naboskapsoperasjoner II - Laplace

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Null-gjennomgang: Punkt der andrederivert er lik 0, og har ulik fortegn før og etter punktet den er lik 0.
- Laplace-operatoren gir andrederivert til en funksjon i to dimensjoner.
- Kan approksimere laplace operatoren ved numerisk tilnærming på derivasjon.
- Laplace er sensitiv til støy, og kan kombineres ved å konvolvere med et Gauss filter. Får en "Laplacian-of-Gaussian" operator.

Naboskapsoperasjoner II - Cannys algoritme

Tanken er: best mulig deteksjon, god kant-lokalisering i for av èn respons for kant.

1. Lavpassfiltrer bildet med en Gauss gitt en σ.
2. Finn gradient-magnituden og gradient-retning.
3. Tynning av gradient-magnitude som står ortogonalt på kant. Deler typisk inn i fire retninger. Tar så naboene som står ortgonalt til kant og tester om de har høyere verdi enn senterpikselen. Hvis èn av naboene har større verdi, settes senterpikselen lik 0.
4. Hysterese-tersklingen. velg to terskler T_l og T_h der $T_l < T_h$.
   - I det tynnede gradient-magnitude bildet, markèr alle piksler som har verdi større eller lik T_h.
   - For alle piksler i det tynnede magnitude bildet som har verdier til og med T_l til T_h, sjekk om hver enkelt piksel har en markert 4- eller 8-nabo. Hvis den har en markert 4- eller 8-nabo, merkes piskelen og.
   - Fortsett frem til ingen nye piksler markeres

Fargerom, fargebilder og bildebehandling i farger - Representasjon av farger

Forskjellige representasjoner av farger.
Vi har sett bl.a på

- RGB: Står for Red, Green og Blue. Forteller oss om hvor mye av hhv. rødt, grønt og blått bildet består av.
- HSI: Står for Hue, Saturation og Intensity.
  - *Hue* beskriver dominerende farge/bølgelengde
  - *Saturation* beskriver hvor sterk fargen er
  - *Intensity* beskriver hvor lys/mørk fargen er.

HSI er nyttig for å kunne beskrive fargen, og hva som karakteriserer den.

For å unngå å introdusere nye farger ved f.eks histogramutjevning eller filtrering på et fargebilde, gjør vi gjerne transformasjonen på I-komponenten av HSI-representasjonen til fargebildet.