

INF 2310 – Digital bildebehandling

EN KORT MIDTVEIS-REPETISJON

Før midtveis-eksamen
tirsdag 12. mars 2013 kl 15:00 – 19:00
5. Etg i Veglaboratoriet og SFL (PC-avd) Fysikkbygget

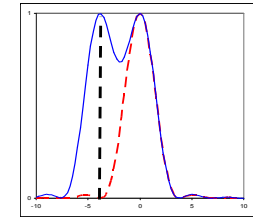
Fritz Albregtsen

Rayleigh-kriteriet

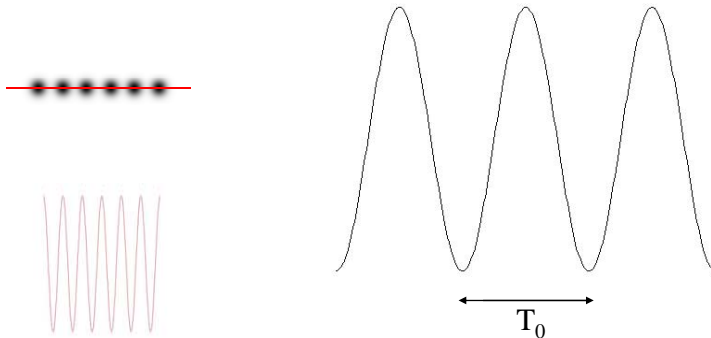
- Anta en "perfekt" linse med aperture-diameter D , og at lysets bølgelengde er λ .
- To punkter i et objekt kan akkurat adskilles i bildet hvis vinkelen mellom dem er gitt ved

$$\sin \theta = 1.22 \lambda / D \text{ radianer.}$$

– Dette er "Rayleigh-kriteriet".

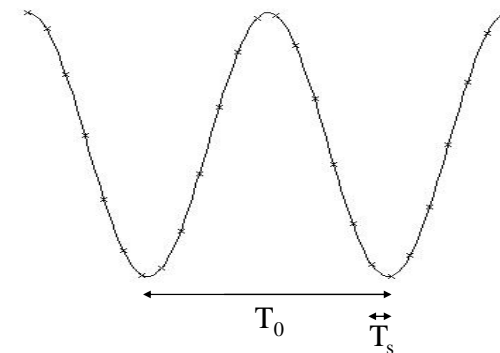


Romlig frekvens



- Periode T_0 (f.eks. i mm eller μm)
- Frekvens $f_0 = 1/T_0$

Sampling av kontinuerlige signaler



- Samplingsperiode T_s
- Samplingsfrekvens $f_s = 1/T_s$ (også kalt samplingsrate)
- Hvor ofte må man sample for å kunne rekonstruere signalet?

Samplingsteoremet (Shannon/Nyquist)

- Anta at det kontinuerlige bildet er båndbegrenset, dvs. det inneholder ikke høyere frekvenser enn f_{\max}
- Det kontinuerlige bildet kan rekonstrueres fra det digitale bildet dersom samplingsraten $f_s = 1/T_s$ er større enn $2 f_{\max}$ (altså $T_s < 1/2T_0$)
- $2 f_{\max}$ kalles Nyquist-raten
- I praksis oversampler vi med en viss faktor for å kunne få god rekonstruksjon.

Undersampling/aliasing

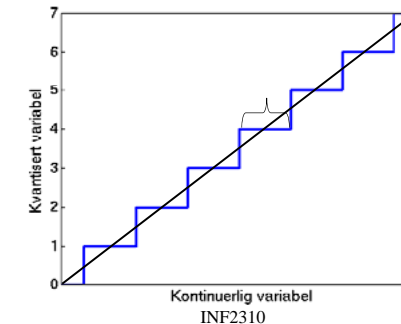
- Undersampling (sample med lavere samplingsrate enn Nyquist-kriteriet) medfører **aliasing**.
- Ved undersampling forvrenges frekvensinnholdet og det digitale bildet inneholder ikke de samme frekvenser som det kontinuerlige bildet.
- Sampling av en sinusoid med for lav samplingsrate gir en diskret sinusoid med lavere frekvens.
- Aliasfrekvens er gitt ved
$$f_a = f_s - f \text{ når } f < f_s < 2f$$

Anti-aliasing

- Effekten av aliasing kan reduseres.
- Dette MÅ gjøres FØR samplingen.
- Hvis vi filtrerer bort de høyeste frekvensene først, vil det finnes færre eller ingen frekvenser som kan gi opphav til aliasing.
- Aliasing er en samplings-effekt.

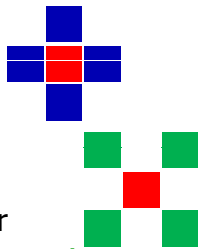
Kvantisering

- Hvert piksel lagres med n biter.
- Pikselet kan da inneholde heltallsverdier fra 0 til $2^n - 1$
- Kvantiseringsfeil
 - Σ (over bildet) av hvert piksels avrundingsfeil



Naboer til piksler

- Et piksel p i posisjon (x,y) har flere nabo-piksler.
- 4-naboene $N_4(p)$ med koordinater $(x+1,y), (x-1,y), (x,y+1), (x,y-1)$
- Diagonal-naboene $N_D(p)$ med koordinater $(x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)$
- Tilsammen utgjør disse 8-naboene til p , $N_8(p)$.



Avstandsmetrikker

- Tre piksler p,q,z med koordinater $(x,y), (s,t)$ og (v,w)
- Avstandsfunksjonen $D(p,q)$ er en **metrikk** hvis
 - $D(p,q) \geq 0$ (*ikke-negativitet*)
 - $D(p,q) = 0$ hvis og bare hvis $p=q$ (*identitet*)
 - $D(p,q) = D(q,p)$ (*symmetri*)
 - $D(p,z) \leq D(p,q) + D(q,z)$ (*trekant-ulikhet*)

Geometriske operasjoner

- Endrer på pikslenes posisjoner
- Første steg i denne prosessen:
 - Transformer pikselkoordinatene (x,y) til (x',y') :
$$x' = T_x(x,y)$$
$$y' = T_y(x,y)$$
 - T_x og T_y er ofte gitt som polynomer.
- Siden pikselkoordinatene må være heltall, må vi deretter bruke interpolasjon til å finne pikselverdien (gråtonen) i den nye posisjonen.

Affine transformasjoner

- Transformerer pikselkoordinatene (x,y) til (x',y') :

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

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

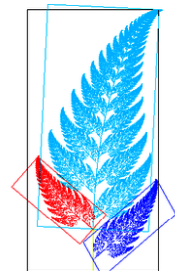
- Affine transformasjoner beskrives ved:

$$x' = a_0x + a_1y + a_2$$

$$y' = b_0x + b_1y + b_2$$

- På matriseform:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ eller } \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 & a_1 \\ b_0 & b_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_2 \\ b_2 \end{bmatrix}$$



Eksempler på enkle transformeringer - I

Transformasjon	a_0	a_1	a_2	b_0	b_1	b_2	Uttrykk
Identitet	1	0	0	1	0	0	$x' = x$ $y' = y$
Skalering	s_1	0	0	0	s_2	0	$x' = s_1x$ $y' = s_2y$
Rotasjon	$\cos\theta$	$-\sin\theta$	0	$\sin\theta$	$\cos\theta$	0	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

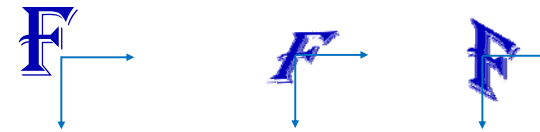
12.02.2013

INF2310

13

Eksempler på enkle transformeringer - II

Transformasjon	a_0	a_1	a_2	b_0	b_1	b_2	Uttrykk
Translasjon	1	0	Δx	0	1	Δy	$x' = x + \Delta x$ $y' = y + \Delta y$
Horisontal "shear" med faktor s_1	1	s_1	0	0	1	0	$x' = x + s_1y$ $y' = y$
Vertikal "shear" med faktor s_2	1	0	0	s_2	1	0	$x' = x$ $y' = s_2x + y$



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

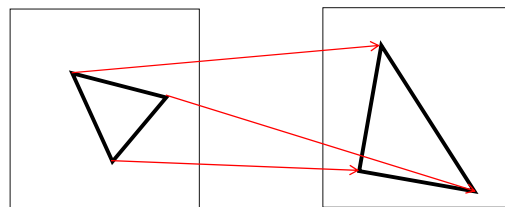
12.02.2013

INF2310

14

Alternativ måte å finne transformkoeffisientene

- En affin transform kan bestemmes ved å spesifisere tre punkter før og etter avbildningen



inn-bildet

resultat-bildet

- Med disse tre punktparene kan vi finne de 6 koeffisientene; $a_0, a_1, a_2, b_0, b_1, b_2$
- Med flere enn 3 punktpar velger man den transformasjonen som minimerer (kvadrat-)feilen summert over alle punktene (mer om dette senere)

12.02.2013

INF2310

15

Forlengings-mapping

for all x', y' do $g(x', y') = 0$

$a_0 = \cos\theta$
 $a_1 = -\sin\theta$
 $b_0 = \sin\theta$
 $b_1 = \cos\theta$

for all x, y do

$x' = \text{round}(a_0x + a_1y)$
 $y' = \text{round}(b_0x + b_1y)$
if (x', y') inside g
 $g(x', y') = f(x, y)$
end

Eksempel:

Enkel rotasjon ved transformen:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Flytter de posisjonstransformerte pikselposisjonene til nærmeste pikselposisjon i utbildet.

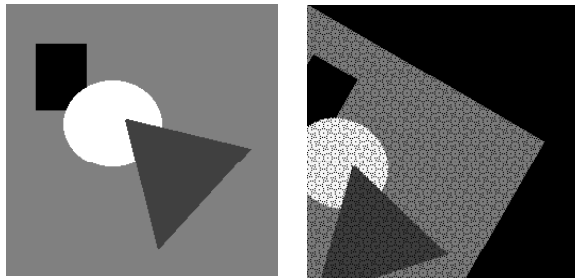
Skriver innbildets $f(x, y)$ inn i $g(x', y')$

12.02.2013

INF2310

16

Forlengings-mapping, forts.



Problemer:

- Ikke alle utpikslar får verdi (hullene i bildet)
- Unødig beregning av pikselkoordinater som allikevel ikke blir synlige (ender utenfor utbildet)
- Samme utbilde-piksel kan bli satt flere ganger

12.02.2013

INF2310

17

Baklengs-mapping

```
a0 = cos (-θ)
a1 = -sin (-θ)
b0 = sin (-θ)
b1 = cos (-θ)
```

for alle x',y' do

```
x = round(a0x'+a1y')
```

```
y = round(b0x'+b1y')
```

```
if (x,y) inside f
```

```
g(x',y') = f(x,y)
```

```
else
```

```
g(x',y')=0
```

```
end
```

Samme eksempel som ved forlengings-mappingen.

NB: (x,y) rotert med θ ga (x',y')
(x',y') rotert med -θ gir (x,y)

Resample bildet.

Her; for hvert utbilde-piksel, invers-transformér, og velg nærmeste piksel fra innbildet.

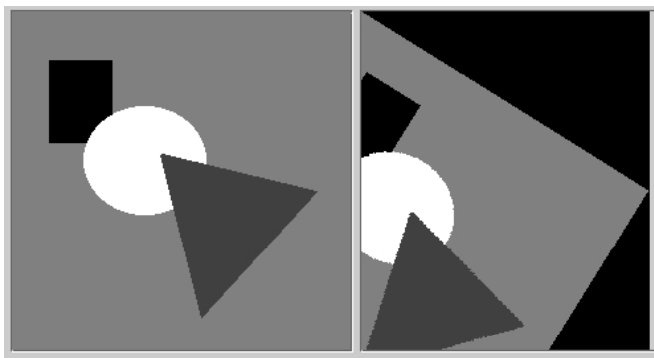
For hver pikselposisjon i ut-bildet: Hent pikselverdi fra innbildet.

12.02.2013

INF2310

18

Baklengs-mapping, forts.



12.02.2013

INF2310

19

Bilineær interpolasjon - I

- Anta at vi kjenner gråtoneverdien i fire nabo-punkter
- Ønsker å estimere gråtonen i et mellomliggende punkt.
- Gjør to lineære interpolasjoner i én retning først, f.eks i x-retning:

$$f(x, y_1) \approx \frac{(x_2 - x)}{(x_2 - x_1)} f(x_1, y_1) + \frac{(x - x_1)}{(x_2 - x_1)} f(x_2, y_1)$$

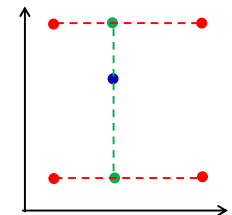
og

$$f(x, y_2) \approx \frac{(x_2 - x)}{(x_2 - x_1)} f(x_1, y_2) + \frac{(x - x_1)}{(x_2 - x_1)} f(x_2, y_2)$$

- Så én interpolasjon i ortogonal retning:

$$f(x, y) \approx \frac{(y_2 - y)}{(y_2 - y_1)} f(x, y_1) + \frac{(y - y_1)}{(y_2 - y_1)} f(x, y_2)$$

- Resultatet er uavhengig av rekkefølgen.
- Den interpolerte flaten er kvadratisk (krum), men lineær langs linjer parallelle med aksene.



12.02.2013

INF2310

20

Bilineær interpolasjon - II

- Bilineær interpolasjon når f er kjent i $(0,0)$, $(0,1)$, $(1,0)$ og $(1,1)$:

$$f(x, y) \approx (1-y)f(x,0) + yf(x,1)$$

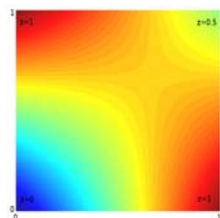
der

$$f(x,0) \approx (1-x)f(0,0) + xf(1,0)$$

og

$$f(x,1) \approx (1-x)f(0,1) + xf(1,1)$$

$$\Rightarrow f(x, y) \approx (1-x)(1-y)f(0,0) + x(1-y)f(1,0) + (1-x)yf(0,1) + xyf(1,1)$$



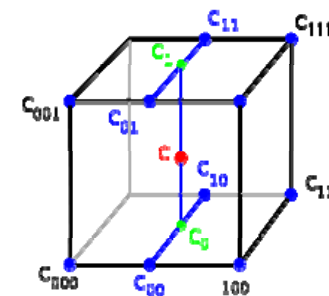
- Eller i matrisenotasjon:

$$f(x, y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

Trilineær interpolasjon

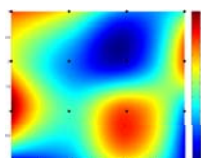
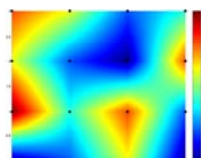
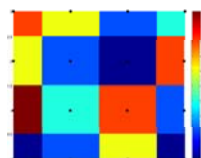
- Utvidelsen fra 2D til 3D kalles *trilineær* interpolasjon, og er en lineær interpolasjon mellom resultatene av to bilineære interpolasjoner.

- Resultatet er igjen uavhengig av rekkefølgen.



Interpolasjon – en sammenligning

- Nærmeste nabo gir 2D trappefunksjon, med diskontinuitet midt mellom punktene.
- Bi-lineær interpolasjon bruker $2 \times 2 = 4$ piksler. Derivert er ikke kontinuerlig over flaten.
- Bi-kubisk interpolasjon gir glattere flater enn bilineær, men er mer regnekrevende; bruker $4 \times 4 = 16$ piksler.



Normalisert histogram

- Vi har at $\sum_{i=0}^{G-1} h(i) = n \times m$
- Det normaliserte histogrammet er:

$$p(i) = \frac{h(i)}{n \times m}, \quad \sum_{i=0}^{G-1} p(i) = 1$$

- $p(i)$ kan ses på som en sannsynlighetsfordeling for pikselverdiene i
 - "Uavhengig" av antall piksler i bildet
- Man kan si en del om bildet ut fra denne sannsynlighets-tetthetsfunksjonen

Kumulativt histogram

- Hvor mange piksler har gråtone mindre enn eller lik gråtone j ?

$$c(j) = \sum_{i=0}^j h(i)$$

- Normalisert kumulativt histogram:

$$\frac{c(j)}{n \times m}$$

(Sannsynligheten for at en tilfeldig piksel er mindre eller lik gråtone j)

Lineær avbilding

- Lineær strekking

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

$$g(x, y) = a f(x, y) + b$$

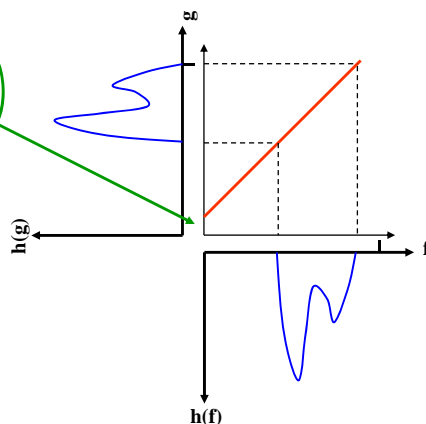
- a regulerer kontrasten, og b "lysheten"
- $a > 1$: mer kontrast
- $a < 1$: mindre kontrast
 - Q: Når og hvordan påvirker a middelveien?
- b : flytter alle gråtoner b nivåer
- Negativer: $a = -1$, $b = \text{maxverdi for bildetype}$

Endre "lysheten" (brightness)

- Legge til en konstant b til alle pikselverdiene

$$g(x, y) = f(x, y) + b$$

- Hvis $b > 0$, alle pikselverdiene øker, og bildet blir lysere
- Hvis $b < 0$, bildet blir mørkere
- Histogrammet flyttes opp eller ned med b
- **Middelveien endres!**

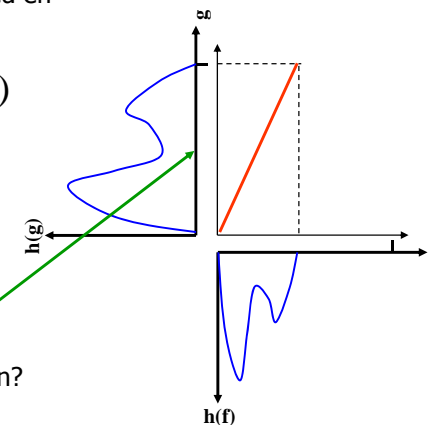


Endre kontrasten

- Multiplisere hver pikselverdi med en faktor a :

$$g(x, y) = a f(x, y)$$

- Hvis $a > 1$, kontrasten øker
- Hvis $a < 1$, kontrasten minker
- Eks: Bruke hele intensitetsskalaen
- Q: Hva skjer med middelveien?



Justering av μ og σ^2

- Gitt inn-bilde med middelværdi μ og varians σ^2
- Anta en lineær gråtone-transform $T[i]=ai+b$
- Ny middelværdi μ_T og varians σ_T^2 er da gitt ved

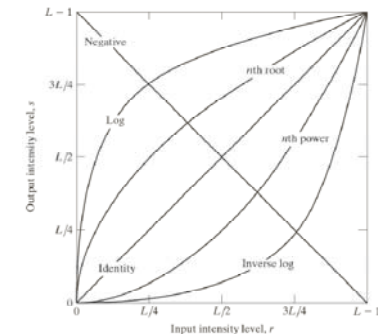
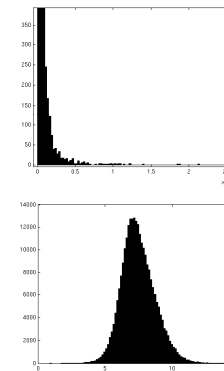
$$\mu_T = \sum_{i=0}^{G-1} T[i] p(i) = a\mu + b$$

$$\begin{aligned} \sigma_T^2 &= \sum_{i=0}^{G-1} T[i]^2 p(i) - \left(\sum_{i=0}^{G-1} T[i] p(i) \right)^2 \\ &= \sum_{i=0}^{G-1} (a^2 i^2 + 2aib + b^2) p(i) - \left(\sum_{i=0}^{G-1} (ai + b) p(i) \right)^2 \\ &= a^2 \left(\sum_{i=0}^{G-1} i^2 p(i) - \left(\sum_{i=0}^{G-1} i p(i) \right)^2 \right) = a^2 \sigma^2 \end{aligned}$$

- Dvs.
 - $a = \sigma_T / \sigma$, $b = \mu_T - a\mu$
- Vi kan altså
 - velge nye μ_T og σ_T^2 ,
 - beregne a og b,
 - anvende $T[i]=ai + b$ på inn-bildet
 - og få et ut-bilde med riktig μ_T og σ_T^2

Logaritmiske transformasjoner

- Hvilken av transformasjonene til høyre er brukt her?



(Fig 3.3 i DIP)

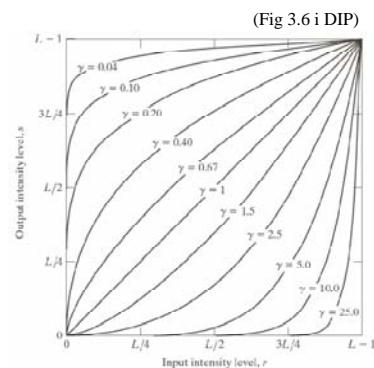
Power-law (gamma)-transformasjoner

- Mange bildeproduserende apparater har et input/output-forhold som kan beskrives som:

$$s = ci^\gamma$$

der s er ut-intensiteten ved en input i

- Kan korrigeres ved gråtonetransformen $T[i] = i^{1/\gamma}$
- Generell kontrast-manipulasjon
 - Brukervennlig med kun én variabel



(Fig 3.6 i DIP)

Algoritme for histogramutjevning

- For et $n \times m$ bilde med G gråtoner:
 - Lag array p , c og T av lengde G med initialverdi 0
- Finn bildets normaliserte histogram
 - Gå igjennom bildet piksel for piksel.
 - Hvis piksel har intensitet i , la $p[i]=p[i]+1$
 - Deretter skalér, $p[i] = p[i]/(n*m)$, $i=0,1,\dots,G-1$
- Lag det kumulative histogrammet c
 - $c[0] = p[0]$, $c[i] = c[i-1]+p[i]$, $i=1,2,\dots,G-1$
- Sett inn verdier i transform-array T
 - $T[i] = \text{Round}((G-1)*c[i])$, $i=0,1,\dots,G-1$
- Gå igjennom bildet piksel for piksel,
 - Hvis bildet har intensitet i , sett intensitet i utbildet til $s=T[i]$

Histogramtilpasning

- Histogramutjevning gir flatt histogram
- Kan spesifisere annen form på resultathistogrammet:
 1. Gjør histogramutjevning på innbildet, finn $s=T(i)$
 2. Spesifiser ønsket nytt histogram $g(z)$
 3. Finn den transformen T_g som histogramutjevner $g(z)$ og inverstransformen T_g^{-1}
 4. Inverstransformer det histogramutjevnete bildet fra punkt 1 ved $z=T_g^{-1}(s)$

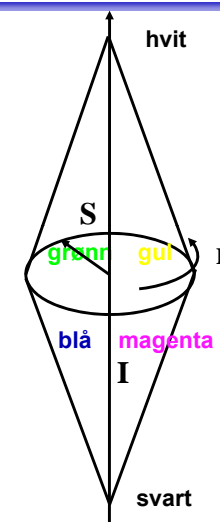
12.02.2013

INF2310

33

Histogramutjevning av RGB-bilder

- Histogramutjevning på hver komponent (r,g,b) uavhengig av hverandre
 - Kan føre til endring i fargetonene i bildet
- Alternativt benytte HSI:
 - Transformér bildet fra RGB til HSI
 - Gjør histogramutjevning på I-komponenten
 - Transformer HSI_{ny} tilbake til RGB



12.02.2013

INF2310

34

Lokal GTT: to muligheter

- Utfør lokal GTT som gir samme kontrast over hele bildet
 - Histogramspesifikasjon
 - Beregn kumulativt histogram i et vindu sentrert om (x,y)
 - Endre senterpikselen ved den resulterende transformen
 - Lineær standardisering av σ til σ_0
 - Beregn $\mu(x,y)$ og $\sigma(x,y)$ i et vindu sentrert om (x,y)
 - Transformer $f(x,y)$ til $g(x,y)$ med en lineær transform som gir nytt standardavvik σ_0 innenfor vinduet

$$g_1(x, y) = \mu(x, y) + (f(x, y) - \mu(x, y)) \frac{\sigma_0}{\sigma(x, y)}$$

12.02.2013

INF2310

35

Lokal GTT – forts.

- Ønsker vi lokal GTT som også gir en ny middelvei μ_0 , så bruker vi transformen

$$g_2(x, y) = \mu_0 + (f(x, y) - \mu(x, y)) \left(\frac{\sigma_0}{\sigma(x, y)} \right)$$

- Men dette vil gi et "flatt" bilde
- Parameteren β kan styre hvor kraftig vi endrer μ :
 - $\beta = 0 \Rightarrow$ uforandret middelvei over hele bildet
 - $\beta = 1 \Rightarrow$ lik middelvei over hele bildet

$$g_3(x, y) = \beta \mu_0 + (1 - \beta) \mu(x, y) + (f(x, y) - \mu(x, y)) \left(\frac{\sigma_0}{\sigma(x, y)} \right)$$

12.02.2013

INF2310

36

Lokal GTT – forts.

- Hva er karakteristisk for homogene områder i et bilde?

$$\sigma(x, y) = 0$$

- Her får vi problemer, fordi vi har σ i nevneren:

$$g_3(x, y) = \dots + (f(x, y) - \mu(x, y)) \frac{\sigma_0}{\sigma(x, y)}$$

- Innfører parameteren δ :

$$g_4(x, y) = \beta \mu_0 + (1 - \beta) \mu(x, y) + (f(x, y) - \mu(x, y)) \left(\frac{\sigma_0}{\sigma(x, y) + \delta \sigma_0} \right)$$

- Lokal pikselverdi-mapping gir økt regnearbeid.

Lokal GTT - Implementasjon

- Lokal kontrast-endring er regnekrevende
 - Histogramspesifikasjon: Beregne nytt lokalt kumulativt histogram for hver piksel
 - Lineær transform:
Beregne ny μ og σ for hver piksel
- Benytt overlappet mellom vinduene i det man flytter til neste piksel
 - Løpende oppdatere både histogrammet, μ og σ
 - Alternativ: Bruk median og percentil.