

Signaler, information & bilder, föreläsning 15

Michael Felsberg och Maria Magnusson
Computer Vision Laboratory (Datorseende)
Department of Electrical Engineering (ISY)
michael.felsberg@liu.se,
maria.magnusson@liu.se



Översikt

- 2D Omsampling
 - Uppsampling
 - Nedsampling
 - Rotation
 - Dilation (Expansion)
 - Erosion (Krympning)
 - Kombinationer
- Avståndskartor
- Histogram och tröskelsättning
 - Automatisk tröskelsättning
- Morfologiska operationer
- Teori: Kap. 4, 5.1, 5.2, 6.1, 6.2, 6.5
- Bygger på Maria Magnussons föreläsningar

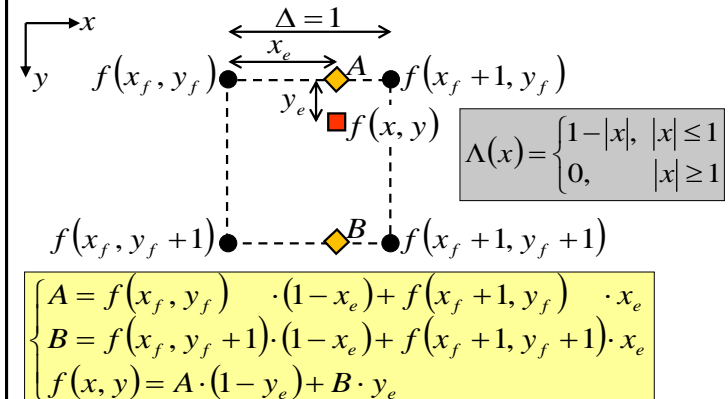


2D Omsampling

- 2D linjär interpolation = bilinjär interpolation
- Metod 1: Börja med 1D linjär interpolation i x-led och fortsätt med 1D linjär interpolation i y-led.
- Metod 2: Utför 2D bilinjär interpolation direkt.

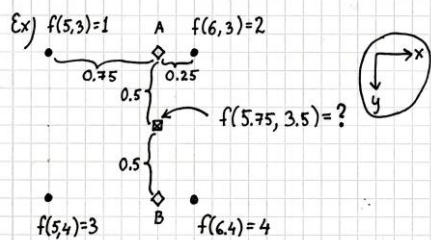


2D bilinjär interpolation, method 1



Ex) 2D bilinjär interpolation, method 1

p. 5



$$A = f(5,3) \cdot \Lambda(0.75) + f(6,3) \cdot \Lambda(0.25)$$

$$A = 1 \cdot 0.25 + 2 \cdot 0.75 = 1.75$$

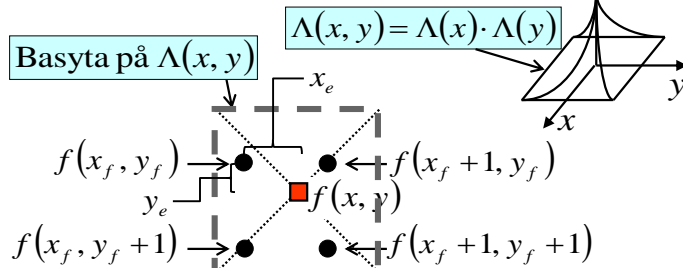
$$B = 3 \cdot 0.25 + 4 \cdot 0.75 = 3.75$$

$$f(5.75, 3.5) = 1.75 \cdot \Lambda(0.5) + 3.75 \cdot \Lambda(0.5)$$

$$= 1.75 \cdot 0.5 + 3.75 \cdot 0.5 = 2.75$$



2D bilinjär interpolation, method 2



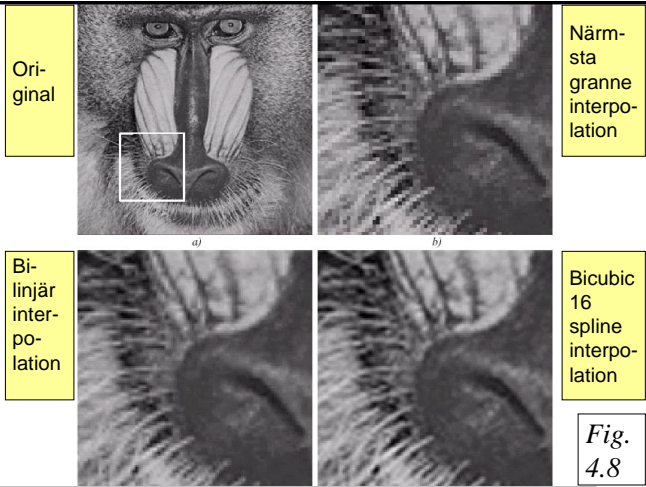
$$f(x, y) =$$

$$= f(x_f, y_f) \cdot (1-x_e)(1-y_e) + f(x_f+1, y_f) \cdot x_e(1-y_e)$$

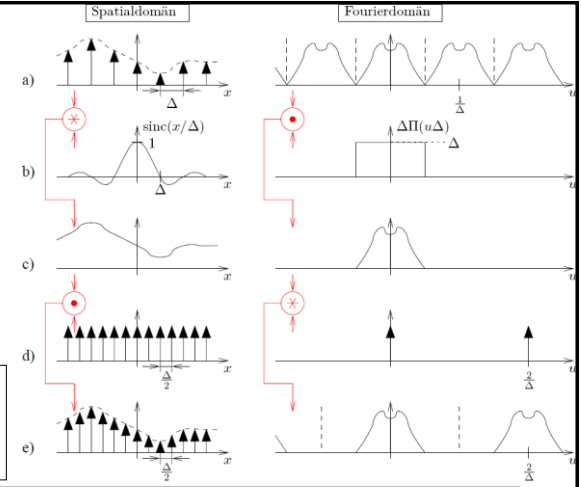
$$+ f(x_f, y_f+1) \cdot (1-x_e)y_e + f(x_f+1, y_f+1) \cdot x_e y_e$$

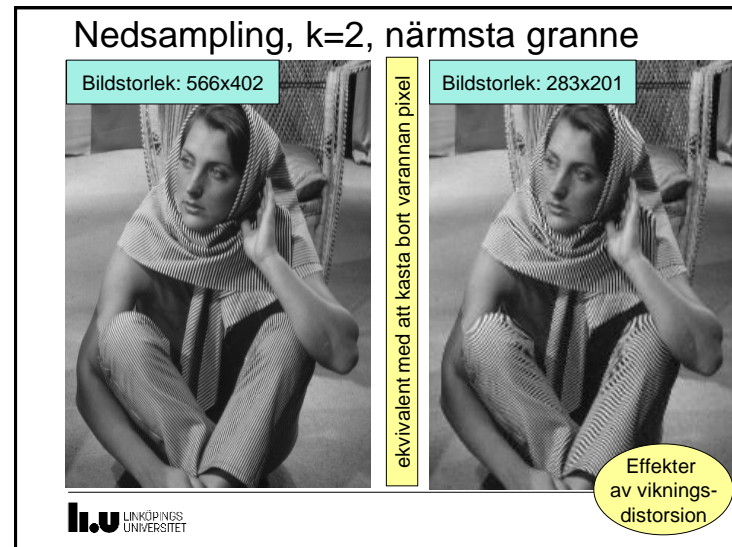
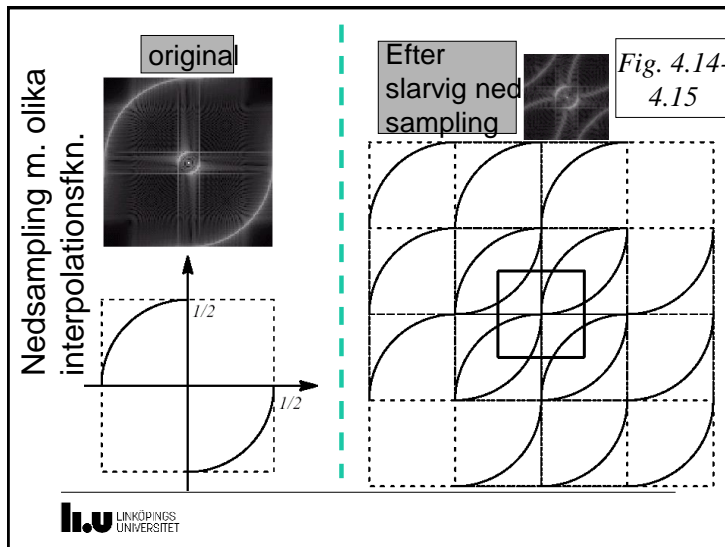
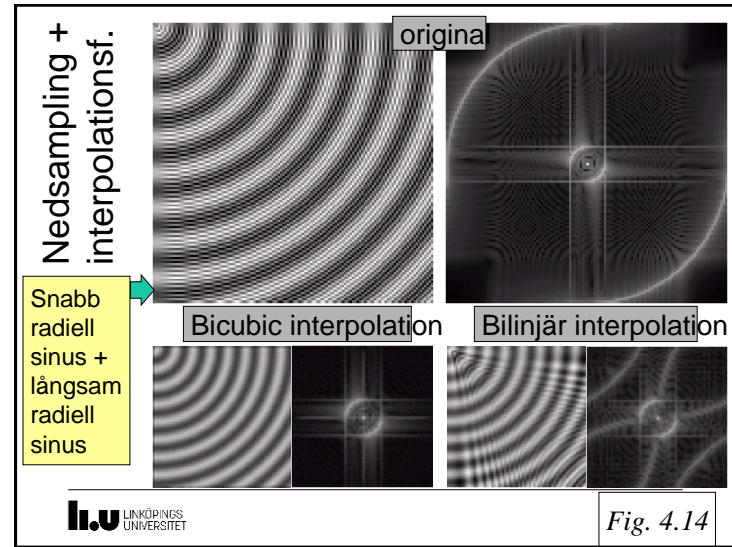
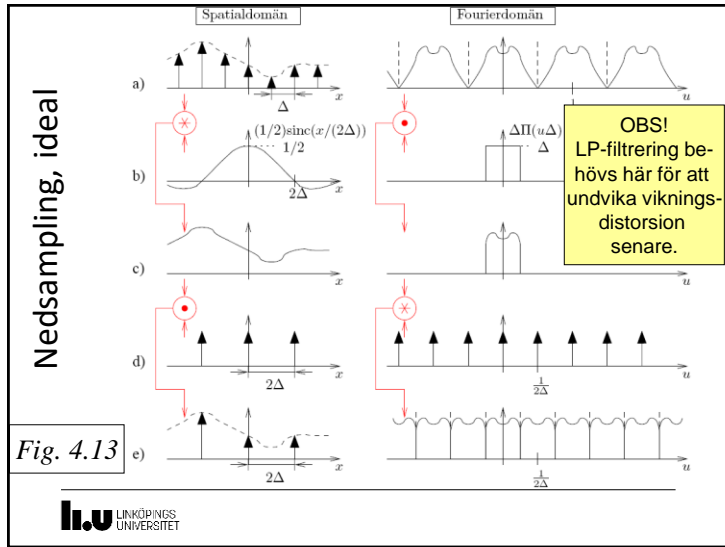


2D uppsampling en faktor 4



Uppsampling, ideal, repetition





Nedsampling, k=2, bilinjär interpolation

Bildstorlek: 283x201

bredd = $2 \cdot k = 4$ pixlar

Detta är ekvivalent med:

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} / 16 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} / 4 * (1 \ 2 \ 1) / 4$$

dvs lågpasfiltrering, följt av att kasta bort varannan pixel

Vigningsdistorser nästan borta

li.u LINKÖPINGS UNIVERSITET

Ideal uppsampling via Fourierdomänen

Bildstorlek: $N \times N$

`np.fft.fftshift(np.fft.fft2(np.fft.ifftshift(f)))`

Nollpadda

Bildstorlek: $2N \times 2N$

`np.fft.fftshift(np.fft.ifft2(np.fft.ifftshift(f)))`

li.u LINKÖPINGS UNIVERSITET

Rotation av en bild

$$R = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

$$\bar{x} = \begin{pmatrix} x \\ y \end{pmatrix}$$

Utbild $f_\alpha(R\bar{x})$

Inbild $f_\alpha(\bar{x})$

α

Fig. 4.10

li.u LINKÖPINGS UNIVERSITET

2 metoder att rotera: Här bara no 2

- För alla \bar{x} i inbilden $f(\)$

Beräkna $R\bar{x}$ i utbilden $f_\alpha(\)$

$f_\alpha(R\bar{x}) := f(\bar{x})$ Forward mapping

värdet $f(\bar{x})$ i inbilden sprids ut i utbilden
- För alla \bar{x} i utbilden $f_\alpha(\)$

Beräkna $R^{-1}\bar{x}$ i inbilden $f(\)$

$f_\alpha(\bar{x}) := f(R^{-1}\bar{x})$ Inverse mapping

värdet $f(R^{-1}\bar{x})$ måste interpoleras fram i inbilden

li.u LINKÖPINGS UNIVERSITET

Rotation enligt metod 2

Observera!
Inbilden ■
och
utbilden ×
är överlagrade.

1) (x, y) givet
3) $f(x', y')$ placeras i punkten (x, y)

2) (x', y') beräknas, varefter $f(x', y')$ interpoleras fram

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = R^{-1} \begin{pmatrix} x \\ y \end{pmatrix}$$

obs: $a < 0$

Fig. 4.11

LINKÖPINGS UNIVERSITET

Rotation i närbild

$f_\alpha(\bar{x}) = f(R^{-1}\bar{x})$

värdet $f(R^{-1}\bar{x})$ måste interpoleras fram i inbilden

Inbild f före rotation

Utbild f_α efter rotation

LINKÖPINGS UNIVERSITET

Rotation med bilinjär interpolation

Rotation -30°

Varför blir apan suddigare efter rotationerna?

Rotation 30°

LINKÖPINGS UNIVERSITET

Slutsatser för omsampling samt rotation

- Den ideala interpolationsfunktionen är en **sinc** som motsvarar en **rektangelfunktion** i Fourierdomänen.
- Vid uppsampling och rotation ska rektangelfunktionen gå till 0 vid **bandgränsen**.
- Vid nedsampling med **k** ska rektangel-funktionen gå till 0 vid **bandgränsen/k**. Den korresponderade interpolations-funktionen blir **k gånger bredare**.
 - Interpolations-funktionen måste utföra **lågpassfiltrering** annars får man vinkningsdistorsion.
- Triangelfunktionen (linjär interpolation): Dess bredd = bredden på sincens huvudlob.
 - Uppsampling: bredden på triangelfunktionen **2 pixlar**
 - Nedsampling: bredden på triangelfunktionen **2-k pixlar**

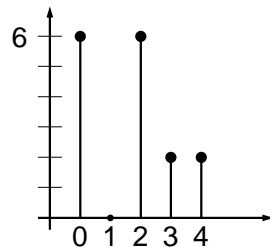
LINKÖPINGS UNIVERSITET

Histogramberäkning på en bild

Bild $f(x,y)$

0	0	4	3
0	0	4	3
0	2	2	2
0	2	2	2

Histogram $p(f)$



Histogrammet p av en bild $f(x,y)$ är en sannolikhetsfunktion som utsäger hur ofta en viss intensitetsnivå $0..Q-1$ förekommer.

```

p = np.zeros(Q)           # Initiera histogram till 0

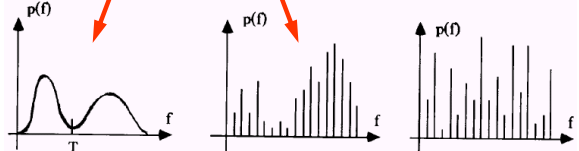
for v in f.flat:         # Beräkna
    p[v] += 1           # histogram

-----
p = p/sum(p)           # Normera histogram
                       # (görs ibland)
    
```

Olika histogram

Kan tröskelsättas

Fig. 5.3



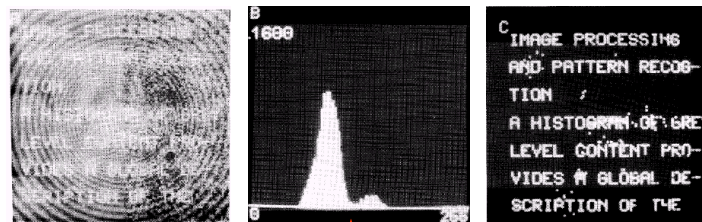
Kontinuerligt, bimodalt

Diskret, bimodalt

Diskret, ej bimodalt

Tröskelsättning

En gråskalebild $f(x,y)$ tröskelsätts enligt: $b(x,y) = \begin{cases} B_1 & \text{om } f(x,y) \geq T \\ B_2 & \text{om } f(x,y) < T \end{cases}$



$f(x,y)$

$p(f)$

Ex) $B_1=1, B_2=0$

$b(x,y)$

Fig. 5.4

En metod att finna en bra tröskel

- Anta att bilden är genererad av
- Bakgrundsprocessen: $P_0, N(\mu_0, \sigma_0)$
 - Objektprocessen: $P_1, N(\mu_1, \sigma_1)$

$$p_0(f) = \frac{1}{\sqrt{2\pi}\sigma_0} \cdot \exp\left[-\frac{1}{2}\left(\frac{f - \mu_0}{\sigma_0}\right)^2\right]$$

$$p_1(f) = \frac{1}{\sqrt{2\pi}\sigma_1} \cdot \exp\left[-\frac{1}{2}\left(\frac{f - \mu_1}{\sigma_1}\right)^2\right]$$

En metod att finna en bra tröskel

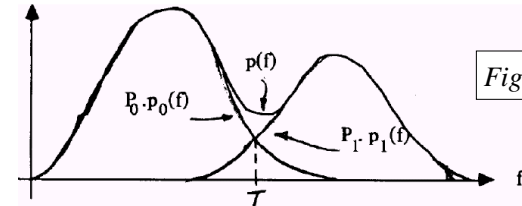


Fig. 5.5

Histogrammet kan då skrivas som:
 $p(f) = P_0 p_0(f) + P_1 p_1(f)$
 Vi söker $f=T$ så att
 $P_0 p_0(f) = P_1 p_1(f)$

Kan leda till t ex **least-error** metoden som inte visas i detalj här.
 En annan populär metod är **Otsus** metod.

Automatisk tröskelsättning med
 mittpunktsmetoden,
 midway method
 (ingen exakt lösning!)

- ① $T^0 = \mu(p(f)), \quad i = 0$
- ② $\mu_0(T^i) = \mu[p(f)|f < T^i]$
- ③ $\mu_1(T^i) = \mu[p(f)|f > T^i]$
- ④ $T^{i+1} = [\mu_0(T^i) + \mu_1(T^i)]/2$
- ⑤ $i := i + 1$

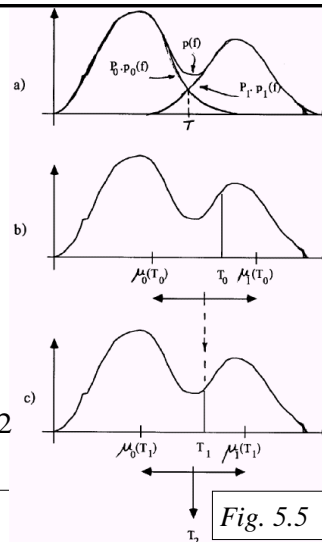
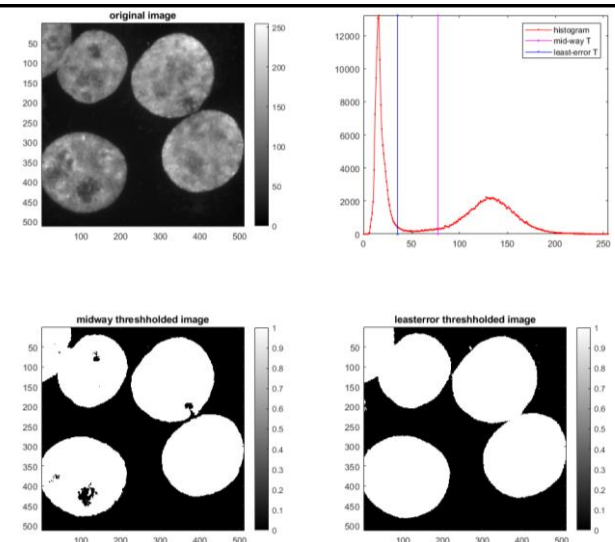


Fig. 5.5

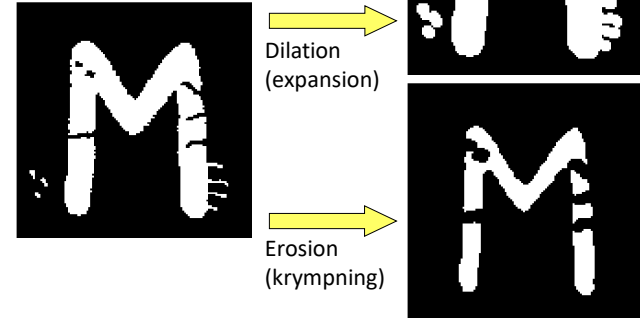
Thresholding example



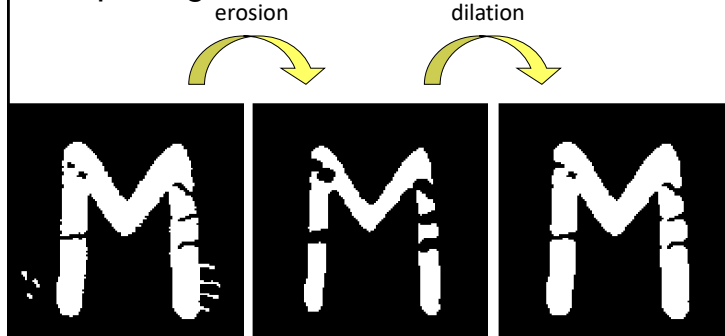
Morfologiska operationer

- Grundläggande morfologiska operationer
 - Dilation (Expansion)
 - Erosion (Krympning)
- Opening (Öppning) = Erosion + Dilation
- Closing (Slutning) = Dilation + Erosion

Grundläggande morfologiska operationer



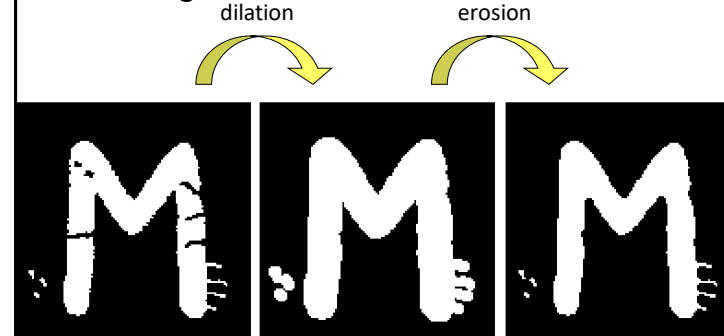
Opening = erosion + dilation



Tar bort små objekt och utskott (spurs)

Återställer storleken

Closing = dilation + erosion



Fyller igen små hål och sprickor

Återställer storleken

Opening följt av closing

opening closing

Closing följt av opening

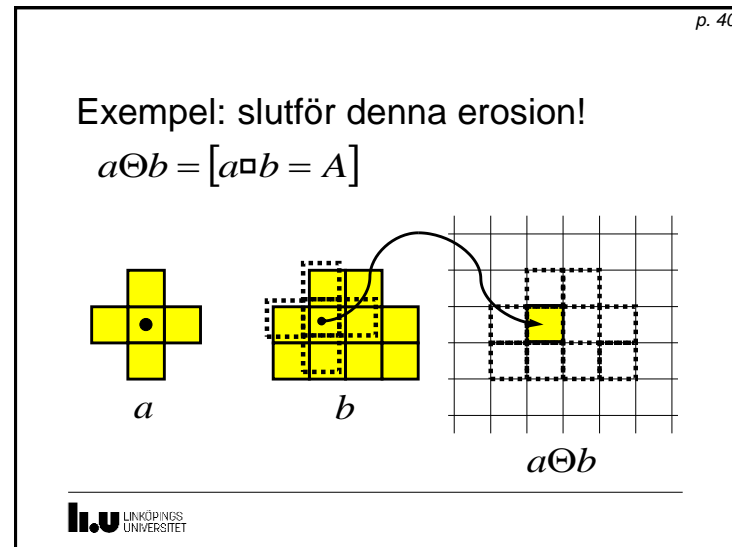
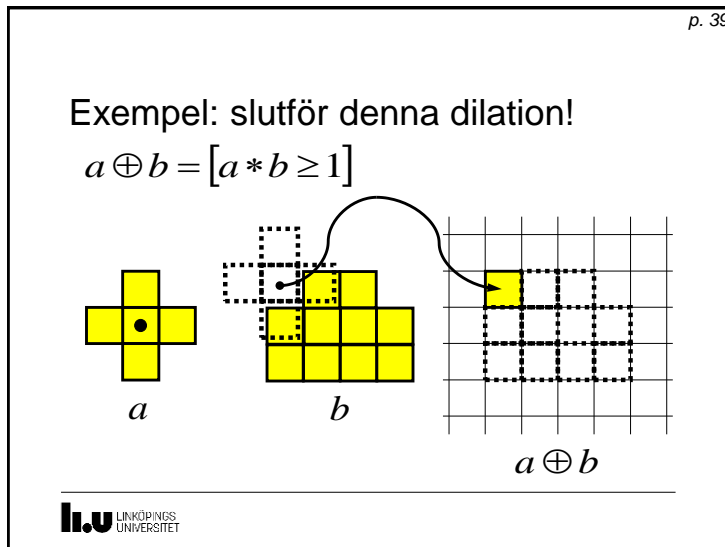
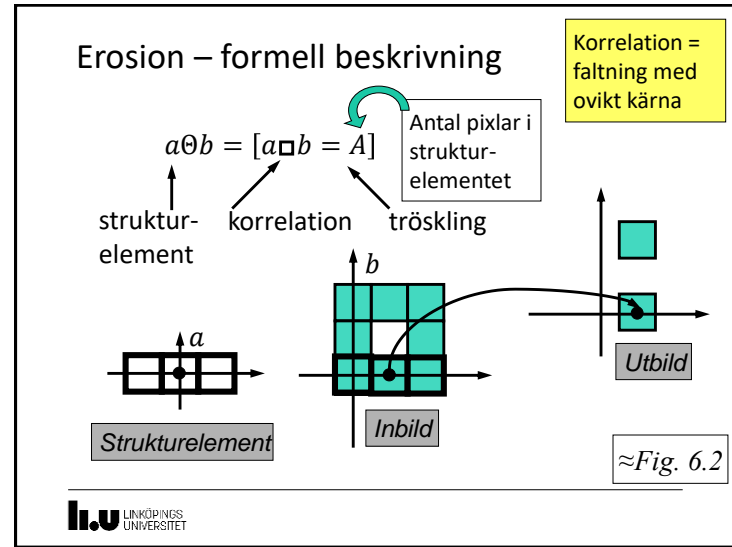
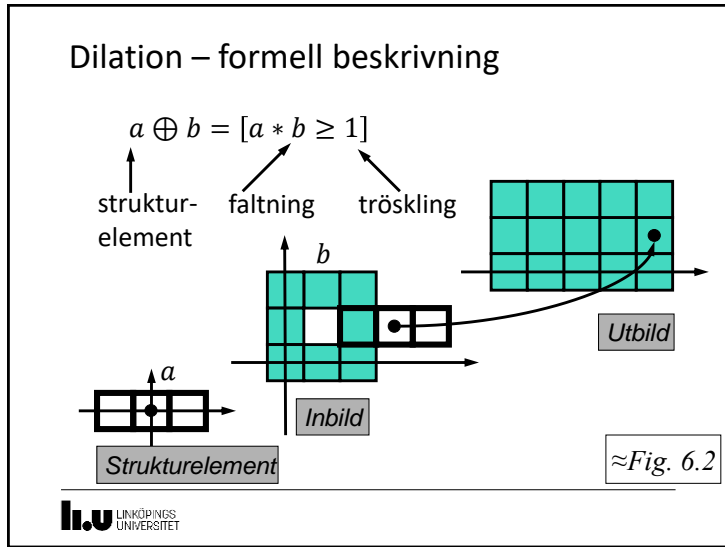
closing opening

Nästan samma resultat!

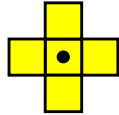
Opening följt av closing Closing följt av opening

Strukturelement

- Binära filterkärnor brukar kallas strukturelement. De vanligaste strukturelementen är illustrerade nedan. Origo är markerat med en punkt.



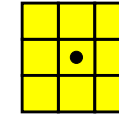
Strukturelementet $d^{(4)}$



Expansion med detta strukturelement:
Alla sidogrannar till objektet 1-ställs

Krympning med detta strukturelement:
Alla sidogrannar till bakgrunden 0-ställs

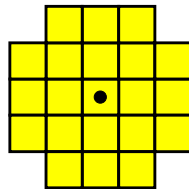
Strukturelementet $d^{(8)}$



Expansion med detta strukturelement:
Alla sido- och hörngrannar till objektet 1-ställs

Krympning med detta strukturelement:
Alla sido- och hörngrannar till bakgrunden 0-ställs

Strukturelementet $d^{(okt)}$



Detta strukturelement ger en jämnare (oktagonal) expansion/krympning. Samma effekt kan erhållas genom att omväxlande använda $d^{(4)}$ och $d^{(8)}$.

Räkneregler

Expansion är kommutativ.
(Jämför med faltning.)

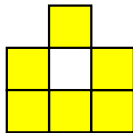
$$a \oplus b = b \oplus a$$

Krympning är inte kommutativ.
(Jämför med korrelation.)
Antag att a och b är olika.
Om a ryms i b så ryms b ej i a .

$$a \ominus b \neq b \ominus a$$

Vilka grannar är sammanhängande?

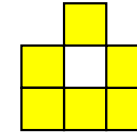
- $d^{(4)}$ -konnektivitet: pixlar sitter ihop om de har angränsande sida
- $d^{(8)}$ -konnektivitet: pixlar sitter ihop om de har angränsande sida eller hörn



- $d^{(4)}$ -konnektivitet: 2 objekt
- $d^{(8)}$ -konnektivitet: 1 objekt

Konflikt mellan objekt och bakgrund

- $d^{(4)}$ -konn. objekt \Leftrightarrow $d^{(8)}$ -konn. bakgrund
- $d^{(8)}$ -konn. objekt \Leftrightarrow $d^{(4)}$ -konn. bakgrund



- $d^{(4)}$ -konnektivitet: 2 objekt, sammanhängande bakgrund
- $d^{(8)}$ -konnektivitet: 1 objekt med hål

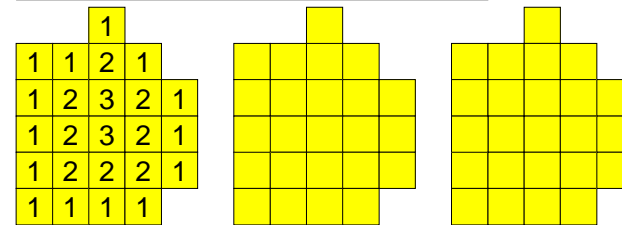
Strukturelementet kan användas för att beräkna avstånd approximativt. Det finns olika metriker.

Strukturelementet appliceras upprepade gånger på objektet (här en punkt). Notera iterationsnumret. Det ger ett avstånd till objektet. Vid varje iteration ökas avståndet med 1.



Avståndskartor **inuti** objekt. $d^{(4)}$ klar. Utför $d^{(8)}$ och $d^{(8)}$ avståndskartering!

$d^{(4)}$: alla sido-grannar påverkas
 $d^{(8)}$: alla sido- och hörngrannar påverkas



$d^{(4)}$ -metrik $d^{(8)}$ -metrik $d^{(8)}$ -metrik

Euklidisk avståndskarta

- Ger exakt avståndsmått.
- Kan inte implementeras med strukturelement.
- Implementerad i MATLAB (phyton?) tillsammans med $d^{(4)}$ och $d^{(8)}$ avståndskartor.
- $d^{(4)}$ och $d^{(8)}$ avståndskartor är snabbare att beräkna än en euklidisk avståndskarta.
- En ganska snabb implementering föreslogs av Breu et al., 1995.

$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
1	0	1	2
$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$

Avstånd i digitala bilder

Man använder inte alltid det korrekta **euklidiska** avståndet. Det är enklare att använda approximativa avstånd, t ex:

- $d^{(4)}$ -metrik
- $d^{(8)}$ -metrik
- $d^{(\text{okt})}$ -metrik (omväxlande $d^{(4)}$ - och $d^{(8)}$ -metrik)

En metrik är ett avståndsmått mellan 2 punkter p och q som måste uppfylla:

- $d(p,q) \geq 0$ med $d(p,q) = 0$ om $p = q$
- $d(p,q) = d(q,p)$
- $d(p,q) \leq d(p,r) + d(r,q)$

Avståndsmått

- Euklidiskt avstånd: $d(x, y) = \sqrt{x^2 + y^2}$
maxfel: 0%
- $d^{(4)}(x, y) = |x| + |y|$
maxfel: 41%
- $d^{(8)}(x, y) = \max(|x|, |y|)$
maxfel: -41%
- $d^{(\text{okt})}(x, y) = \max\left(|x|, \left\lfloor \frac{2}{3}(|x| + |y| + 1) \right\rfloor, |y|\right)$
maxfel: 11%, förutom $(\pm 1, \pm 1)$