

---

# INF2310 – Digital bildebehandling

Forelesning 11 - 2020

## Morfologiske operasjoner på binære bilder

Fritz Albregtsen

Repetisjon av grunnleggende mengdeteori

Fundamentale operasjoner

Sammensatte operasjoner

Eksempler på anvendelser

G&W: 9.1-9.5 og deler av 2.6

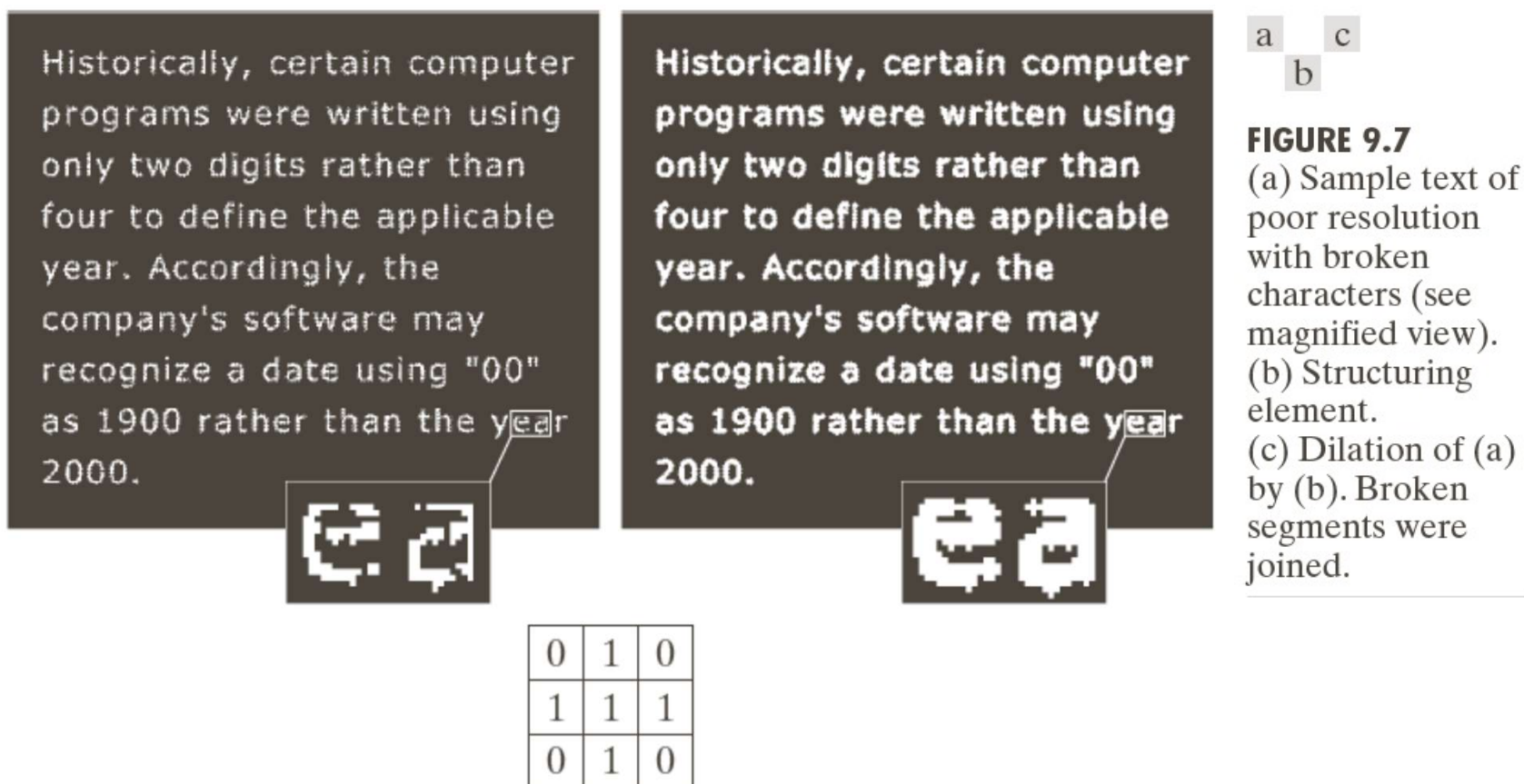
# Introduksjon - bruksområder

---

- ❑ **Brukes** som et steg i behandling og analyse av bilder.
- ❑ **Modifiserer formen** (eng.: *shape*) **til objekter** vba. lokale operasjoner.
- ❑ Kan brukes til å **fjerne uønskede effekter** etter segmentering:
  - Fjerne små objekter (antas å være støy).
  - Glatte omrisset til større objekter.
  - Fylle hull i objekter.
  - Lenke sammen objekter.
- ❑ Kan brukes som et steg for å **beskrive/analysere objekter**:
  - Finne omriss av objekter.
  - Tynne objekter.
  - Finne objekter som inneholder en viss struktur.
  - Finne mønstre i et bilde.
- ❑ Operasjonene er ofte enkle og kan utføres svært raskt.
- ❑ Kan generaliseres til gråtonebilder (med enda flere anvendelser).

# Eksempel: Lenke sammen objekter

- ❑ Morfologiske operasjoner er ofte velegnet til å forbedre en segmentering.
- ❑ Eks.: Lenke sammen defragmentere objekter:



# Litt mengdeteori

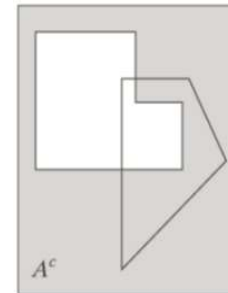
- En **mengde** (eng.: set) består av **elementer**.
  - **Rekkefølgen** av elementene og **antallet** er **ubestemt**.
- Dersom elementet  $a$  er inneholdt i mengden  $A$  skriver vi:  $a \in A$
- Dersom elementet  $a$  ikke er inneholdt i mengden  $A$  skriver vi:  $a \notin A$
- $\emptyset$  er mengden uten noen elementer og kalles **den tomme mengden**.
- $A^c$  er **komplementet til  $A$**  og består av alle elementene som ikke er i  $A$ .
- $\hat{A}$  er refleksjonen av  $A$  ( $180^\circ$  rotasjon)
- **$A$  er en delmengde av  $B$**  dersom alle elementene i  $A$  også er elementer i  $B$ , og dette betegnes:  $A \subseteq B$
- **Unionen av to mengder  $A$  og  $B$**  er mengden som består av alle elementer som er i  $A$  og/eller  $B$ , og dette betegnes:  $A \cup B$
- **Snittet av to mengder  $A$  og  $B$**  er mengden som består av alle elementer som er i både  $A$  og  $B$ , dette betegnes:  $A \cap B$

# Mengder og binære bilder

- La  $A$  være en mengde i  $\mathbb{Z}^2$ .
  - Hvert element i  $A$  er da et punkt  $(a_1, a_2)$  der  $a_1$  og  $a_2$  er heltall.
- Et binært bilde kan beskrives ved forgrunns pikslenes koordinater. Mengden av disse pikslene er en mengde i  $\mathbb{Z}^2$ .

- **Komplementet** til et binært bilde  $f$ :

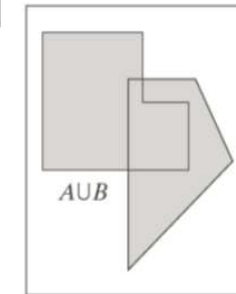
$$h(x, y) = \begin{cases} 1 & \text{hvis } f(x, y) = 0 \\ 0 & \text{hvis } f(x, y) = 1 \end{cases}$$



Komplementet til  $A$

- **Unionen** av to binære bilder  $f$  og  $g$ :

$$h = f \cup g \equiv h(x, y) = \begin{cases} 1 & \text{hvis } f(x, y) = 1 \text{ eller } g(x, y) = 1 \\ 0 & \text{ellers} \end{cases}$$

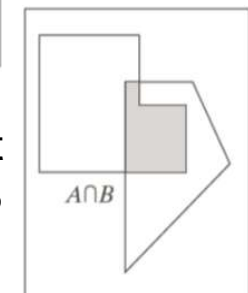


Unionen av  $A$  og  $B$

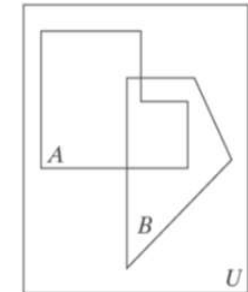
- **Snittet** av to binære bilder  $f$  og  $g$ :

$$h = f \cap g \equiv h(x, y) = \begin{cases} 1 & \text{hvis } f(x, y) = 1 \text{ og } g(x, y) = 1 \\ 0 & \text{ellers} \end{cases}$$

Snittet av  $A$  og  $B$



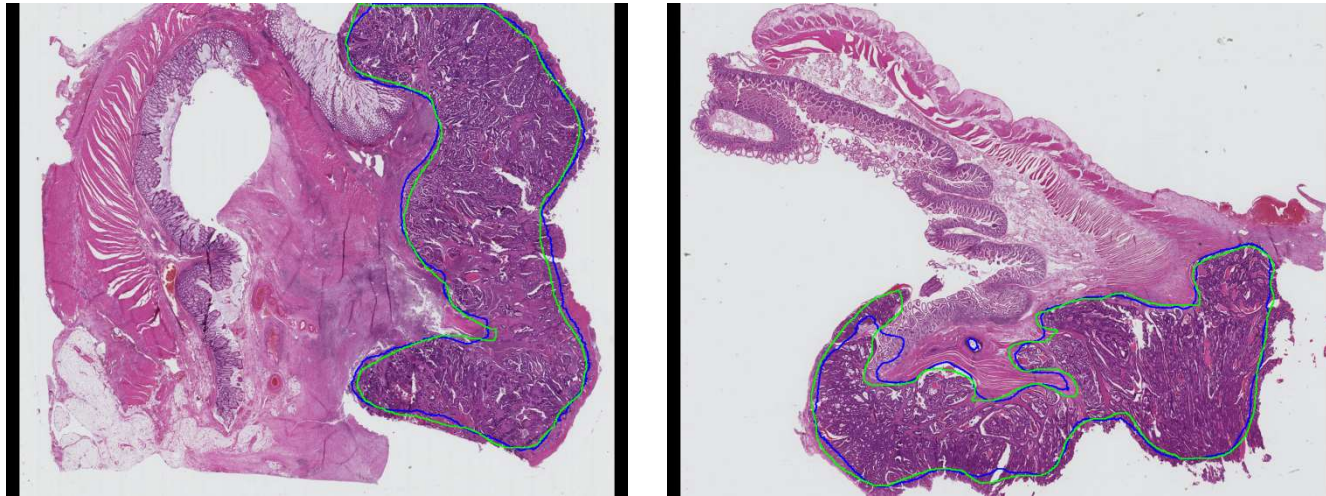
Konturen av to binære bilder,  $A$  og  $B$



I figurene markerer grått med i mengden, og hvitt ikke med. (Fra figur 2.31 i G&W)

# Binært snitt: en anvendelse

- Kan vi automatisere en patologs manuelle omriss av en kreftsvulst, slik at en dyktig patolog kan bruke sin tid på viktigere/vanskeligere ting?
  - **Grønn kurve** = patologens håndtegnede omriss av colon tumor
  - **Blå kurve** = resultat fra konvolusjonsbasert nevralt nett («dyp læring»)



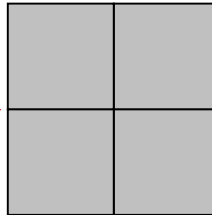
Ole-Johan Skrede, «DoMore!»-Project, ICGI & IFI 2018

- Hvordan måler vi kvaliteten til en slik «*in silico patolog*»?
  - Husk: Bildene har stor bit-dybde, men regionene innenfor omrissene er binære.
  - En mulig metrikk er Dice Similarity Coefficient (DSC) fra 1941:

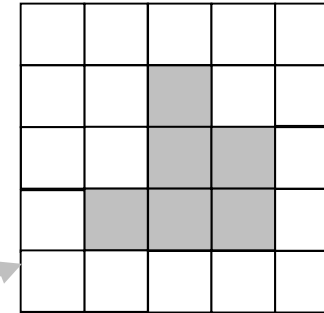
$$DSC = \frac{\text{snitt}}{(\text{midlere areal})}$$

# Tre sentrale begrep

- Et **strukturelement** for et binært bilde er et **naboskap**.

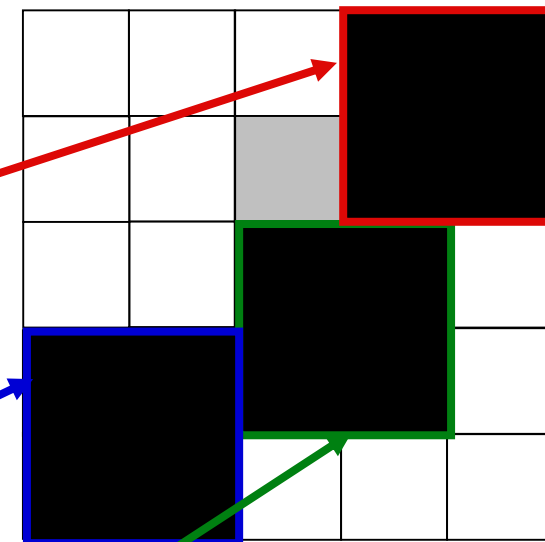


- Typisk definert ved en binær matrise der 1 markerer "med i naboskapet".



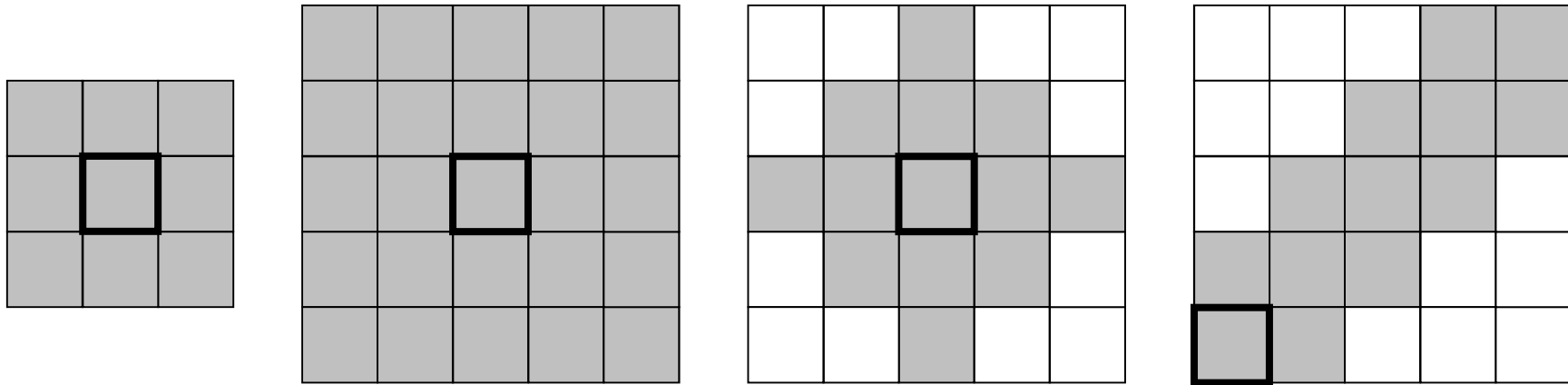
- Når vi fører strukturelementet over det binære bildet vil vi finne:

- Posisjoner der strukturelementet **ikke overlapper** objektet.
- Posisjoner der strukturelementet delvis overlapper objektet, vi sier at elementet **treffer** objektet.
- Posisjoner der strukturelementet ligger inni objektet, vi sier at elementet **passer** inne i objektet.




I figurene markerer grått "med i mengden" (forgrunns piksel), og hvitt "ikke med" (bakgrunns piksel).

# Strukturelementenes form og origo



I figurene markerer  
grått med i nabolskapet  
/ strukturelementet,  
og hvitt ikke med.

- ❑ Strukturelementer kan ha ulik form og størrelse.
- ❑ Må bestemme et **origo**.
  - Origo markerer pikselen som evt. endrer verdi (i utbildet).
  - Origo kan ligge utenfor strukturelementet.
  - Origo bør markeres når man angir strukturelementet, f.eks. ved 



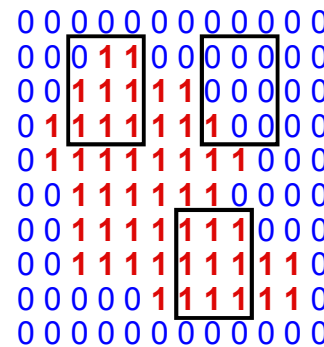
# Passer strukturelementet til det binære bildet?

- ❑ Vi flytter strukturelementet rundt over et binært bilde.

- ❑ Strukturelementet **passer** i posisjonen (x,y) i bildet hvis alle elementer som er  $\neq 0$  i strukturelementet overlapper en pikselverdi  $\neq 0$  i bildet.

- ❑ I denne sammenhengen vil vi alltid:
  - Ignorere pikselverdier som overlapper 0 i strukturelementet.
    - 0 markerer jo «er ikke med i naboskapet».
  - Anta at piksler utenfor bildet er 0.

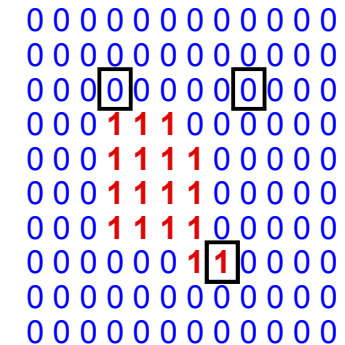
Et bilde



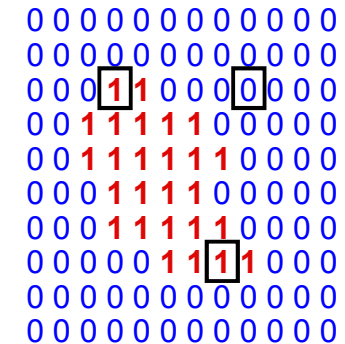
1 1 1  
1 1 1  
1 1 1

To forskjellige strukturelementer

0 1 0  
1 1 1  
0 1 0



To forskjellige resultater



# Erosjon

- Plasser strukturelementet S slik at origo overlapper posisjon (x,y) i inn-bildet f, og beregn ut-bildet g ved å bruke regelen:

$$g(x, y) = \begin{cases} 1 & \text{hvis } S \text{ passer } f \\ 0 & \text{ellers} \end{cases}$$

```

0 1 0 0 0 0 0 1 1 0 0
1 1 1 0 0 0 1 1 1 1 0
0 1 1 1 0 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 0 1 1 1 0
0 1 1 1 1 0 0 0 1 1 1
0 0 1 1 0 0 0 0 0 1 0
    
```

**erodert med**

- Erosjon av et bilde f med strukturelementet S betegnes:

$$f \ominus S$$

```

1 1 1
1 1 1
1 1 1
    
```

**gir**

```

0 1 0
1 1 1
0 1 0
    
```

**gir**

- Mer presist: Erosjonen av mengden A med strukturelementet B er definert som posisjonene til alle piksler z som er slik at B er inkludert i A når origo i B plasseres i z:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

```

0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
    
```

```

0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 1 0 0
0 0 1 0 0 0 1 1 0 0 0
0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 1 1 0 1 0 0 0
0 0 0 1 1 0 0 0 1 0 0
0 0 1 1 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0
    
```

# Effekter av erosjon

- ❑ Erodering **krymper** objekter.
- ❑ Piksler fjernes også innenfra, hvis objektet har hull.
- ❑ Erosjon fjerner «små» utstikk i objektets omriss.
  - «Små» er relativt til størrelsen av strukturelementet.
- ❑ Erosjon utvider innbuktninger i objektets omriss.
- ❑ Resultatet er avhengig av strukturelementet.
- ❑ Større strukturelement gir mer erosjon/fjerning.

```

0 0 1 1 1 1 0 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
1 1 1 1 0 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 0 0 0
    
```

**erodert med**

```

1 1 1
1 1 1
1 1 1
    
```

**gir**

```

0 1 0
1 1 1
0 1 0
    
```

**gir**

```

0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0 1 0 0
0 0 1 0 0 0 1 1 1 0 0
0 0 1 0 0 0 1 1 1 0 0
0 0 1 0 0 0 1 1 1 0 0
0 0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
    
```

```

0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 1 1 0 0
0 0 1 1 0 1 1 1 1 0 0
0 1 1 0 0 0 1 1 1 1 0
0 0 1 1 0 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0
    
```

# Iterativ erosjon

- Vi sa at «Større strukturelement gir mer erosjon/fjerning».
- Resultatet av erosjon med et **stort strukturelement** er (nesten) **lik** resultatet av **gjentatt** erosjon med et **mindre strukturelement** med samme form.
- Hvis  $s_2$  er formlik  $s_1$ , men dobbelt så stort, så er:

$$f \ominus s_2 \approx (f \ominus s_1) \ominus s_1$$

```

0 0 1 1 1 1 0 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
1 1 1 1 0 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 0 0 0
    
```

erodert 2 ganger med

1 1 1	0 1 0
1 1 1	1 1 1
1 1 1	0 1 0
<b>gir</b>	<b>gir</b>

0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 1 1 0 0
0 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0

# Anvendelse av erosjon: Kantdeteksjon

- Erodering fjerner piksler langs omrisset av et objekt.
- Vi kan finne kantene av objektene i bildet ved å subtrahere et erodert bilde fra originalbildet:  $g = f - (f \ominus S)$
- Det benyttede strukturelementet avgjør kantens tilkoblingstype:

Et bilde	erodert med	gir	=>	differanse	
<pre> 00111101110 01111111110 01111111110 11110111111 01111111111 01111111110 01111111110 00000111000           </pre>	<pre> 010 111 010           </pre>	<pre> 00000000000 00111101100 00110111100 01100011110 00110111110 00111111110 00000111000 00000000000           </pre>	=>	<pre> 00111101110 01000010010 01001000010 10010100001 01001000001 01000000010 01111000110 00000111000           </pre>	<div style="border: 1px dashed green; padding: 5px;"> <p>Sammenhengende kanter hvis (og bare hvis) man bruker <b>8-tilkobling</b></p> </div>
	<pre> 111 111 111           </pre>	<pre> 00000000000 00011000100 00100011100 00100011100 00100011100 00111111100 00000010000 00000000000           </pre>	=>	<pre> 00111101110 01100111010 01011100010 11010100011 01011100011 01000000010 01111101110 00000111000           </pre>	<div style="border: 1px dashed green; padding: 5px;"> <p>Sammenhengende kanter ved bruk av <b>4-tilkobling</b></p> </div>

# Eksempel: Kantdeteksjon ved erosjon



0 1 0  
1 1 1  
0 1 0

1 1 1  
1 1 1  
1 1 1



Et bilde

erodert med

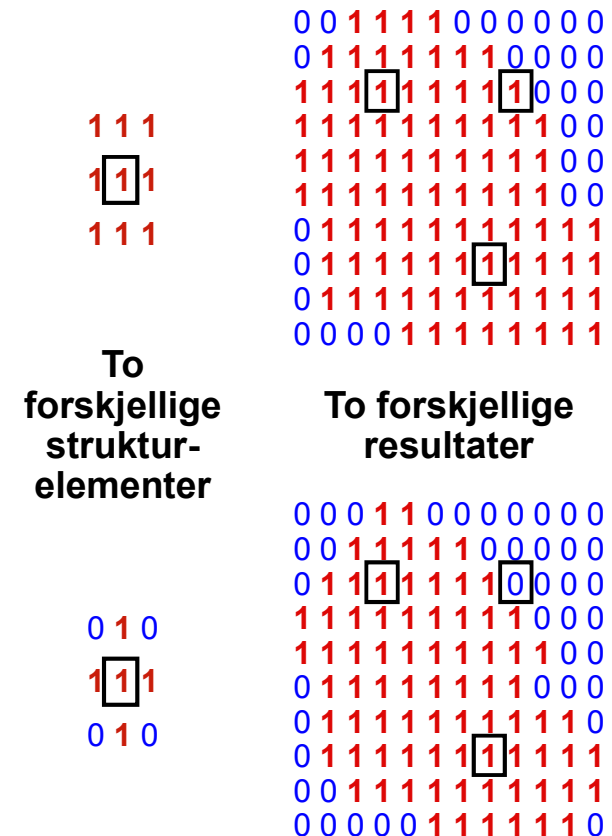
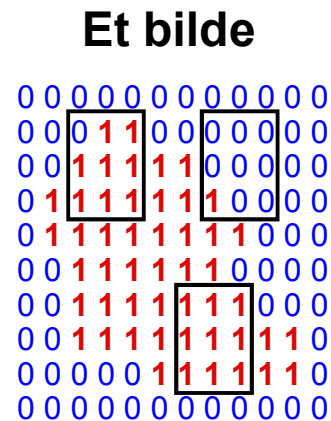
=>

differanse

I bildene markerer  
hvit forgrunn og  
svart bakgrunn.

# Treffer strukturelementet det binære bildet?

- Vi flytter strukturelementet rundt over et binært bilde.
- Strukturelementet **treffer** i posisjonen (x,y) i bildet hvis et element  $\neq 0$  i strukturelementet overlapper en pikselverdi  $\neq 0$  i bildet.
- Her reflekterer vi (roterer 180°) strukturelementet før vi flytter det rundt.



Fortsatt vil vi:

- Ignorere pikselverdier som overlapper 0 i strukturelementet.
- Anta at piksler utenfor bildet er 0.

# Dilasjon (dilatasjon)

- Roter S og plasser det slik at origo overlapper (x,y) i inn-bildet f, og beregn ut-bildet g ved:

$$g(x, y) = \begin{cases} 1 & \text{hvis } S \text{ treffer } f \\ 0 & \text{ellers} \end{cases}$$

```

0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 1 0
0 0 1 0 0 0 1 1 0 0
0 0 0 1 0 1 1 0 0 0
0 0 0 0 1 1 0 1 0 0
0 0 0 1 1 0 0 0 1 0
0 0 1 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0
    
```

- Dilasjon av et bilde f med strukturelementet S betegnes:

$$f \oplus S$$

dilatert med

```

1 1 1
1 1 1
1 1 1
    
```

gir

```

0 1 0
1 1 1
0 1 0
    
```

gir

- Mer presist:  
Dilasjonen av mengden A med strukturelementet B er definert som posisjonene til alle piksler z som er slik at en 180° rotert B har minst ett felles element med A når origo i B plasseres i z:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

```

1 1 1 0 0 0 1 1 1 1 0
1 1 1 1 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 0 1 1 1 1
0 1 1 1 1 1 0 0 1 1 1
    
```

```

0 1 0 0 0 0 0 1 1 0 0
1 1 1 0 0 0 1 1 1 1 0
0 1 1 1 0 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 0 1 1 1 0
0 1 1 1 1 0 0 0 1 1 1
0 0 1 1 1 0 0 0 0 1 0
    
```



# Effekter av dilasjon

- ❑ Dilasjon **utvider** objekter.
- ❑ Dilasjon fyller i hull i objektet.
  - Fyller igjen hullet hvis strukturelementet er stort nok i forhold til hullet.

```

0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 1 1 1 0 0
0 0 0 1 1 1 0 0 1 0 0
0 0 0 1 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0 1 0 0
0 0 0 0 0 1 0 0 1 0 0
0 0 0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
    
```

dilatert med

- ❑ Dilasjon glatter ut innbuktninger i objektets omriss.

```

1 1 1
1 1 1
1 1 1
    
```

gir

```

0 1 0
1 1 1
0 1 0
    
```

gir

- ❑ Resultatet er avhengig av strukturelementet.
- ❑ Større strukturelement gir større dilasjons-effekt.

```

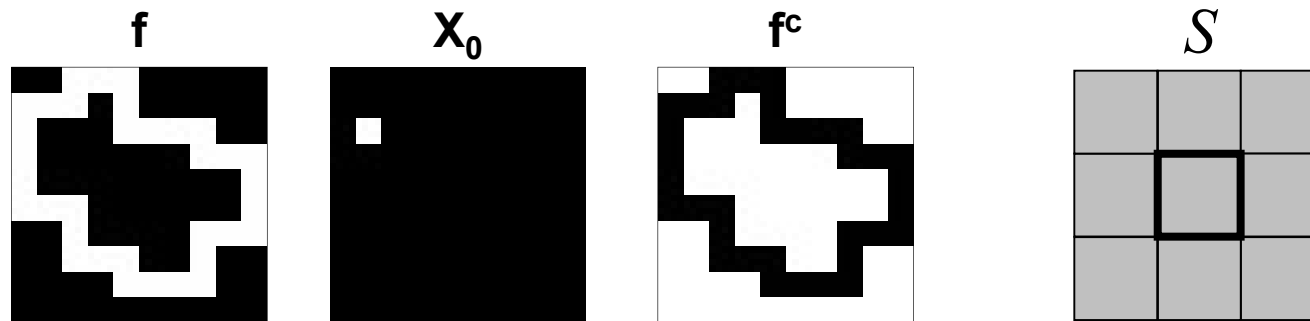
0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 1 0
0 0 0 1 1 1 1 1 1 1 0
0 0 0 0 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 1 1 0
0 0 0 0 0 0 1 1 1 1 0
    
```

```

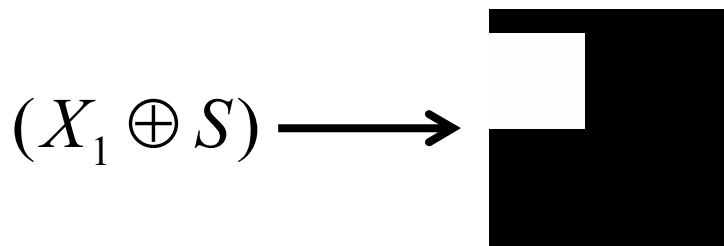
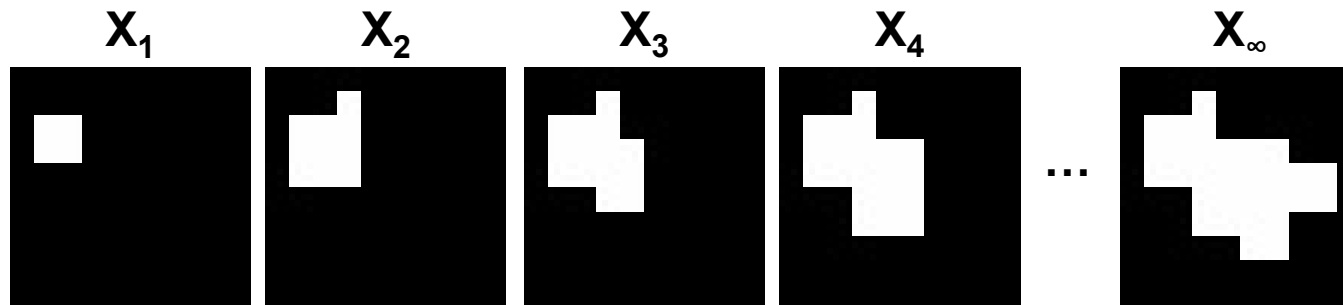
0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 0 1 1 1 0
0 0 0 1 1 1 0 1 1 1 0
0 0 0 0 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 1 1 0
0 0 0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0
    
```

# Anvendelse av dilasjon: Region-fylling

- La  $X_0$  inneholde et punkt i regionen som skal fylles.
- Iterativt beregn  $X_k = (X_{k-1} \oplus S) \cap f^c$  inntil konvergens:



Strukturelement ved 8-tilkoblet region / 4-tilkoblet kant.

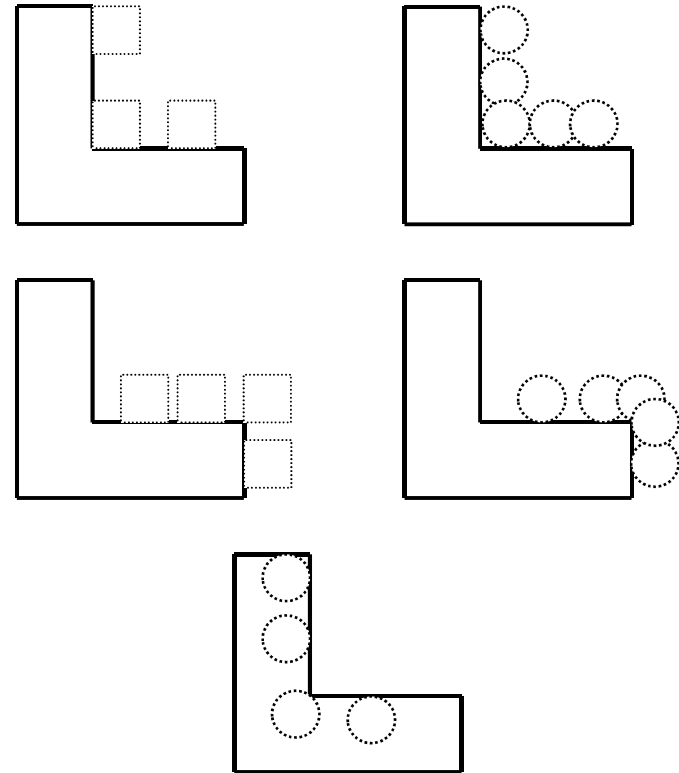


I bildene markerer hvit forgrunn og svart bakgrunn.

(Bildene er hentet fra <http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>)

# Effekter av sirkulære strukturelementer på hjørner

- Både dilatering og erodering med **rektangulære strukturelementer** **bevarer formen til hjørner.**
- **Dilatering** av **konkave** hjørner med sirkulære strukturelementer **bevarer** hjørnenes form.
- **Dilatering** av **konvekse** hjørner med sirkulære strukturelementer **avrunder** hjørnene.
- Omvendt for erosjon:
  - **Avrundede** hjørner ved **erosjon** av **konkave** hjørner.
  - Formen til **konvekse** hjørner **bevares**.



# Dualitet

- **Dilasjon og erosjon er duale** med hensyn til komplementering og reflektering (180° rotasjon), dvs. at dilasjon og erosjon kan uttrykkes ved hverandre:

$$f \oplus S = (f^c \ominus \hat{S})^c$$

$$f \ominus S = (f^c \oplus \hat{S})^c$$

- For å dilatere f med symmetrisk S kan vi erodere komplementet til f med S, og ta komplementet av resultatet.
  - Tilsvarende for å erodere.
- => Dilasjon og erosjon kan utføres av samme prosedyre, forutsatt at vi kan rotere et strukturelement 180° og finne komplementet til et binært bilde.

et bilde	komplementet
0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1
0 0 1 1 0 0 1 1 1 0 0	1 1 0 0 1 1 0 0 0 1 1
0 0 0 1 1 1 0 0 1 0 0	1 1 1 0 0 0 1 1 0 1 1
0 0 0 1 0 0 0 0 1 0 0	1 1 1 0 1 1 1 1 0 1 1
0 0 0 0 1 0 0 0 1 0 0	1 1 1 1 0 1 1 1 0 1 1
0 0 0 0 0 1 0 0 1 0 0	1 1 1 1 1 0 1 1 0 1 1
0 0 0 0 0 0 1 0 1 0 0	1 1 1 1 1 1 0 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1

dilatert med	erodert med
0 1 0	0 1 0
1 1 1	1 1 1
0 1 0	0 1 0

gir	gir
0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1
0 0 1 1 0 0 1 1 1 0 0	1 1 0 0 1 1 0 0 0 1 1
0 1 1 1 1 1 1 1 1 1 0	1 0 0 0 0 0 0 0 0 0 1
0 0 1 1 1 1 1 1 1 1 0	1 1 0 0 0 0 0 0 0 0 1
0 0 1 1 1 1 0 1 1 1 0	1 1 0 0 0 0 1 0 0 0 1
0 0 0 1 1 1 0 1 1 1 0	1 1 1 0 0 0 1 0 0 0 1
0 0 0 0 1 1 1 1 1 1 0	1 1 1 1 0 0 0 0 0 0 1
0 0 0 0 0 1 1 1 1 1 0	1 1 1 1 1 0 0 0 0 0 1
0 0 0 0 0 0 1 0 1 0 0	1 1 1 1 1 1 0 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1

og disse bildene er komplementære.

De to matrisene til høyre er 1 utenfor randen.

## Dualitet og effekter av sirkulære strukturelementer på hjørner

---

- Erosjon er å finne de posisjonene der strukturelementet passer inni forgrunnen.
- Dilasjon er å finne de posisjonene der (det 180° roterte) strukturelementet passer inni bakgrunnen, og så komplementere resultatet.
  - Det er dette den ene dualitetsformelen sier.
- => Siden **erosjonen** av **konkave** hjørner med et **sirkulært strukturelement avrunder** hjørnene, så vil **dilasjonen** avrunde **konvekse** hjørner når vi benytter samme strukturelement.
  - Merk: Et konvekst forgrunns-hjørne er også et konkavt bakgrunns-hjørne.
- Logikken fungerer like bra omvendt vei.

# Dilasjon: Andre egenskaper

- Dilasjon er **kommutativ**.

$$f \oplus S = S \oplus f$$

- Selv om det er en konvensjon at første operand er bildet og andre er strukturelementet, så har dette altså ingen betydning.

- Dilasjon er **assosiativ**.

$$f \oplus (S_1 \oplus S_2) = (f \oplus S_1) \oplus S_2$$

- Hvis S kan dekomponeres, dvs. at S er  $S_1$  dilatert med  $S_2$ , kan vi spare en del regnetid, spesielt hvis  $S_1$  og  $S_2$  er én-dimensjonale.

Eksempel:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} = [1 \ 1 \ 1 \ 1 \ 1] \oplus \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

# Erosjon: Andre egenskaper

---

- Erosjon er **IKKE** kommutativ:

$$f \ominus S \neq S \ominus f$$

- Erosjon er heller **IKKE** assosiativ, men  
suksessiv erosjon av bildet  $f$  med  $A$  og så med  $B$   
er ekvivalent med erosjon av bildet  $f$  med  $A$  **dilatert** med  $B$ :

$$(f \ominus A) \ominus B = f \ominus (A \oplus B)$$

- Passer det med denne tidligere påstanden (side 12)?  
«Hvis  $s_2$  er formlik  $s_1$ ,  
men dobbelt så stort,  
så er  $f \ominus s_2 \approx (f \ominus s_1) \ominus s_1$ »

# Åpning

- ❑ **Erosjon** av et bilde **fjerner** alle strukturer som ikke kan inneholde strukturelementet, og «**krymper**» alle andre strukturer.
- ❑ Hvis vi **dilaterer** resultatet av en erosjon med samme strukturelement, vil de strukturene som «**overlevde**» erosjonen bli omtrentlig **gjenskapt**.
- ❑ Dette er en **morfologisk åpning**;

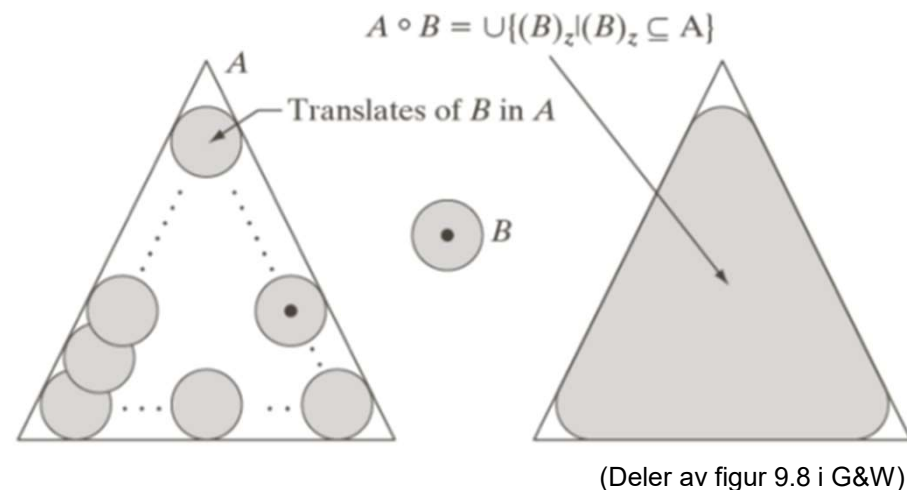
$$f \circ S = (f \ominus S) \oplus S$$

- ❑ Navnet kommer av at operasjonen kan skape en åpning (et mellomrom) mellom to strukturer som bare henger sammen ved en tynn «bro», uten å krympe disse to strukturene i noen betydelig grad.
  - Bare erosjon kan også skape en slik åpning/mellomrom, men vil også krympe begge strukturene.



# Geometrisk tolkning av åpning

- Tenk at strukturelementet definerer **størrelsen og formen til spissen av en tusj penn**.
- Det er bare tillatt å **fargelegge innenfor objekter**.
  - Et par detaljer: Man må holde tusjen vinkelrett på tegneflaten og med samme rotasjon som strukturelementet.
- **Åpningen er resultatet av å fargelegge så mye man har lov til**.
- For runde strukturelementer: Konvekse hjørner blir avrundet, konkave hjørner beholdes.
  - Akkurat som ved dilasjon (skyldes at enhver åpning avsluttes med en dilasjon).



Åpning er **idempotent**:

$$(f \circ S) \circ S = f \circ S$$

dvs. at gjentatte anvendelser med samme strukturelement gir ingen endring.

# Lukking

---

- **Dilasjon** av et bilde **utvider** strukturer, **fyller** i **hull og innbuktninger** i omrisset.
- Hvis vi **eroderer** resultatet av en dilasjon med samme strukturelement, vil strukturene **stort sett** få **gjenskapt** sin opprinnelige størrelse og form, men **hull og innbuktninger** som ble fylt igjen ved dilasjonen vil **ikke gjenoppstå**.
- Dette er en **morfologisk lukking**;

$$f \bullet S = (f \oplus S) \ominus S$$

- Navnet kommer av at operasjonen kan lukke en åpning mellom to strukturer som bare er adskilt med et lite gap, uten at de to strukturene vokser i noen betydelig grad.
  - Bare dilasjon kan også lukke en slik åpning, men vil også forstørre begge strukturene.

# Geometrisk tolkning av lukking

□ Vi kan benytte **samme metafor** som for åpning:

- Strukturelementet definerer størrelsen og formen til spissen av en tusjpenn.
- Man holder tusjen vinkelrett på tegneflaten og fargelegger så mye man har lov til.

□ **Denne gangen** er det bare tillatt å **fargelegge utenfor objekter**.

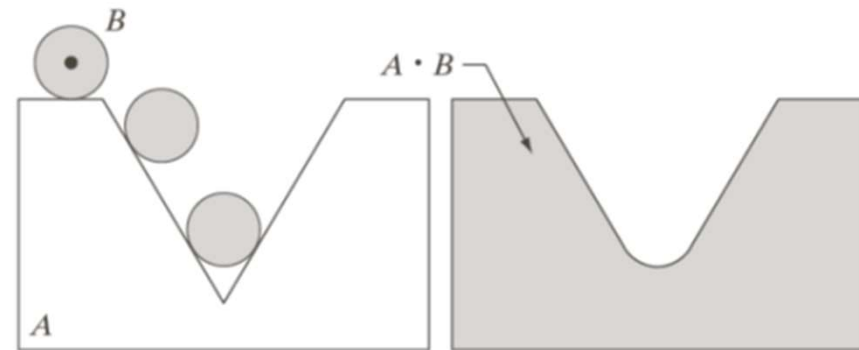
- En detalj: Denne gangen skal tusjen holdes speilvendt ( $180^\circ$  rotert) av strukturelementet.

□ **Lukkingen er det som ikke fargelegges**.

- Denne gangen fargelegger vi altså bakgrunnen, sist fargela vi forgrunnen.

□ For runde strukturelementer:  
Konkave hjørner blir avrundet,  
konvekse hjørner beholdes.

- Akkurat som ved erosjon (skyldes at enhver lukking avsluttes med en erosjon).

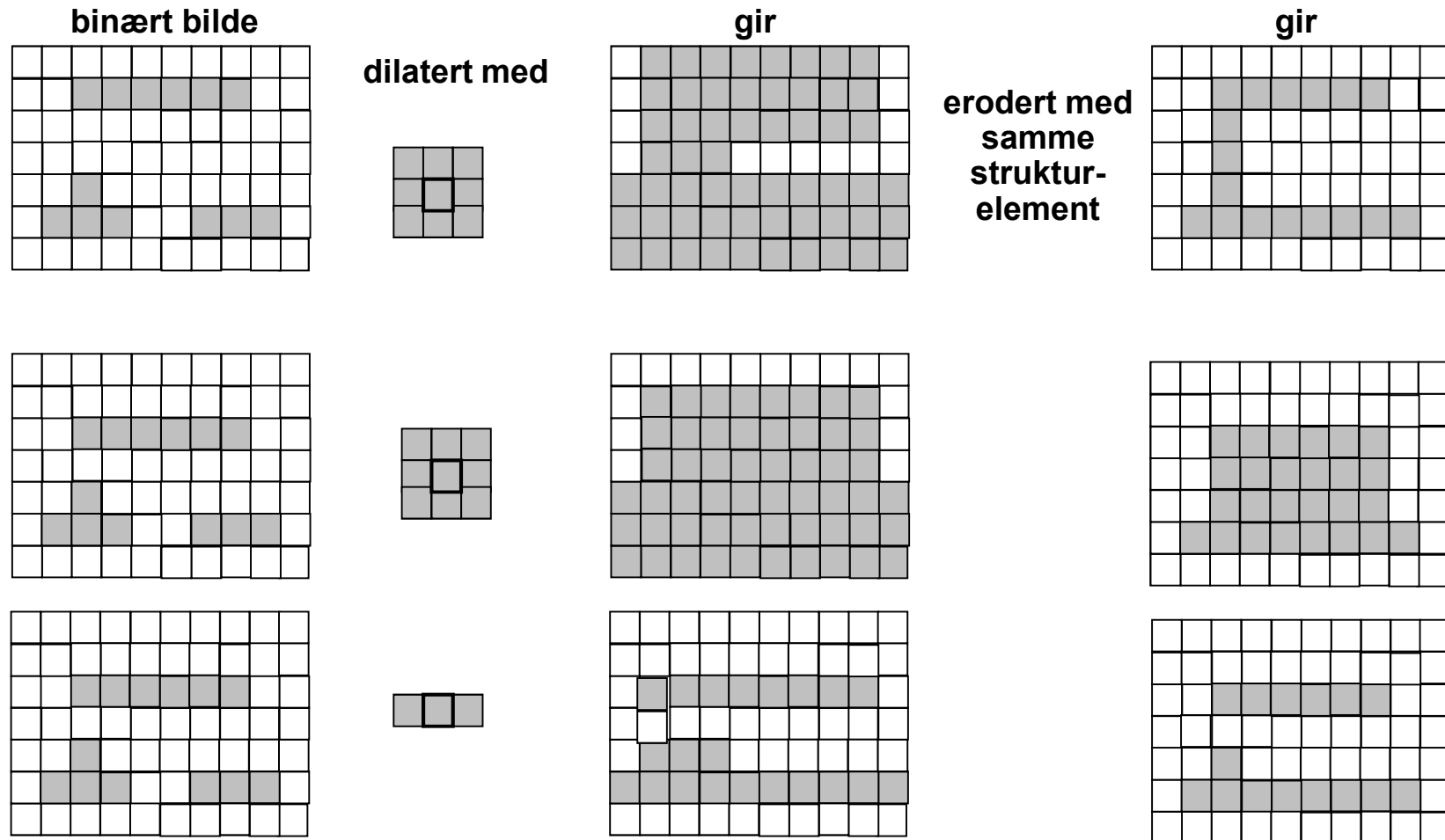


(Deler av figur 9.9 i G&W)

Også lukking er **idempotent**:

$$(f \bullet S) \bullet S = f \bullet S$$

# Lukking lukker små åpninger



- Strukturelementets størrelse og form, og strukturenes mellomrom er avgjørende for resultatet.

I figurene markerer grått forgrunn og hvitt bakgrunn.

# Dualitet mellom åpning og lukking

- **Lukking** er en **dual** operasjon til **åpning** med hensyn til komplementering og reflektering (180° rotering), og omvendt:

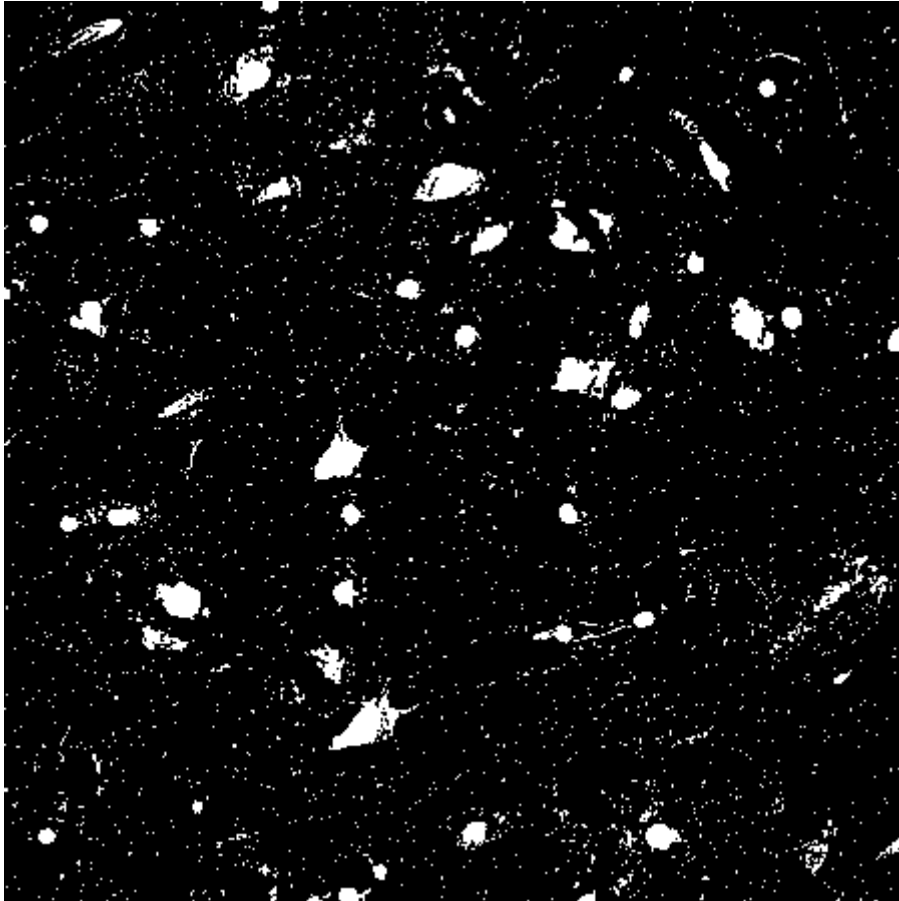
$$f \bullet S = (f^c \circ \hat{S})^c \quad f \circ S = (f^c \bullet \hat{S})^c$$

- Lukking kan utføres ved å komplementere bildet, åpne det med det speilvendte (180° rotere) strukturelementet, og ta komplementet av resultatet.
  - Tilsvarende for åpning.
- Vi kan altså utføre begge operasjonene med kode bare for den ene, hvis vi har kode for å speilvende og komplementere et binært bilde.
- **Lukking** er en **ekstensiv** transformasjon (pikslar legges til).
- **Åpning** er en **antiekstensiv** transformasjon (pikslar fjernes).

$$f \circ S \subseteq f \subseteq f \bullet S$$

# Eksempel: Støyfjerning med åpning

---



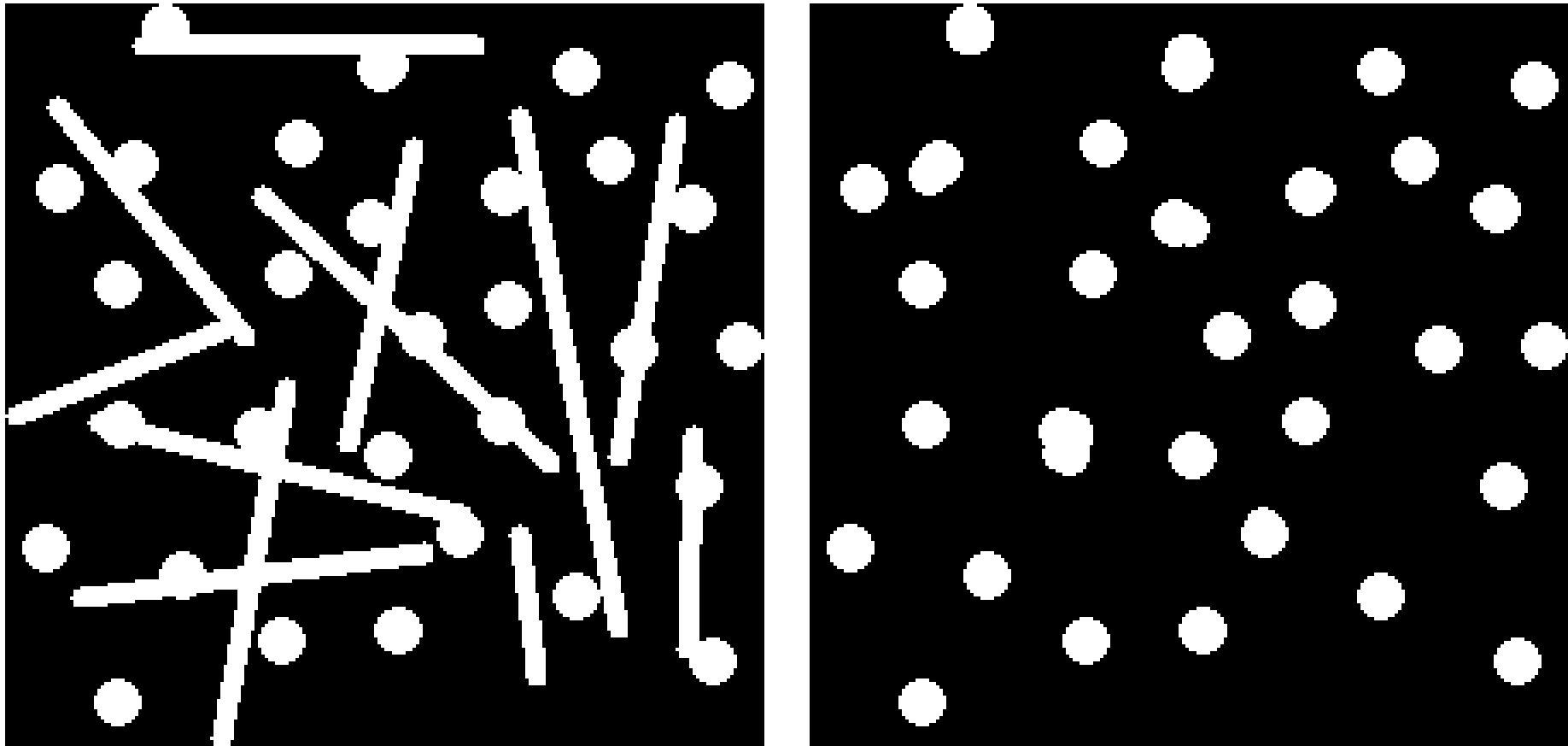
**Åpning med et (7x7) sirkulært strukturelement**

I bildene markerer hvit forgrunn og svart bakgrunn.

(Bildene er hentet fra  
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/>)

# Eksempel: Form-separering ved åpning

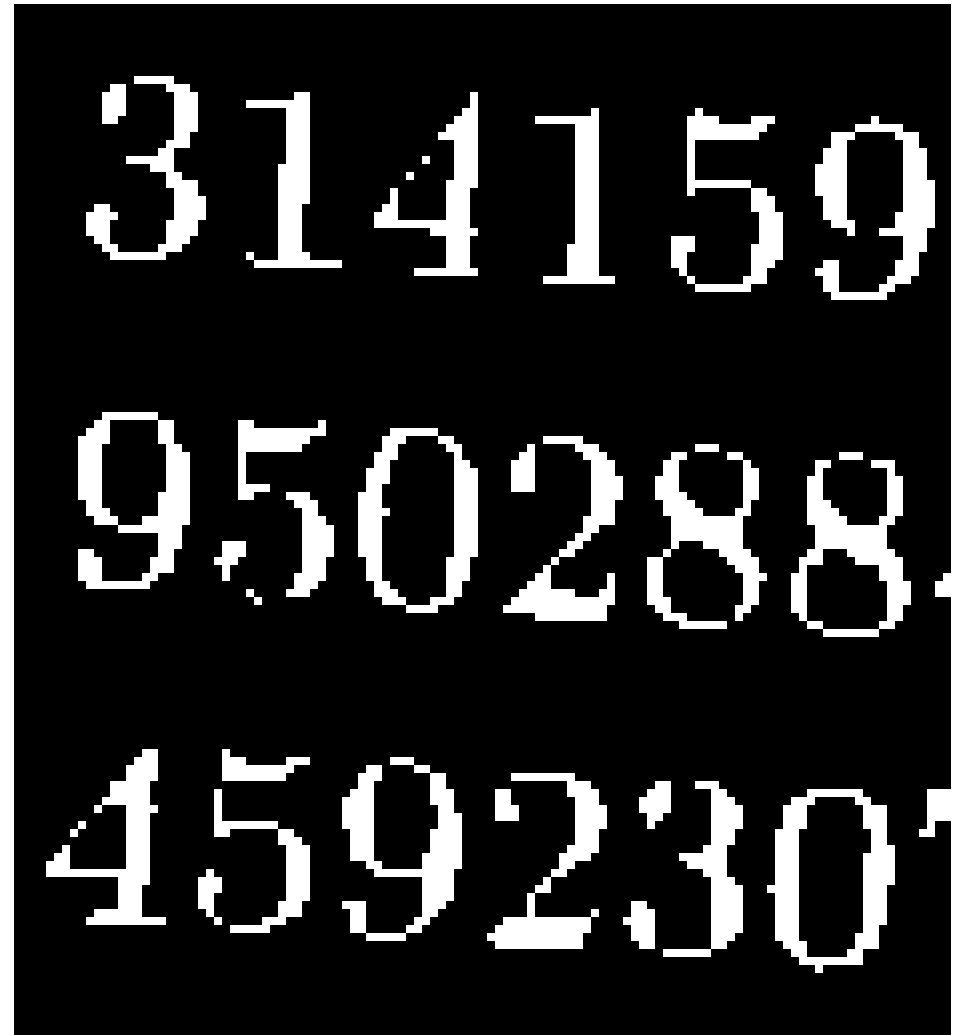
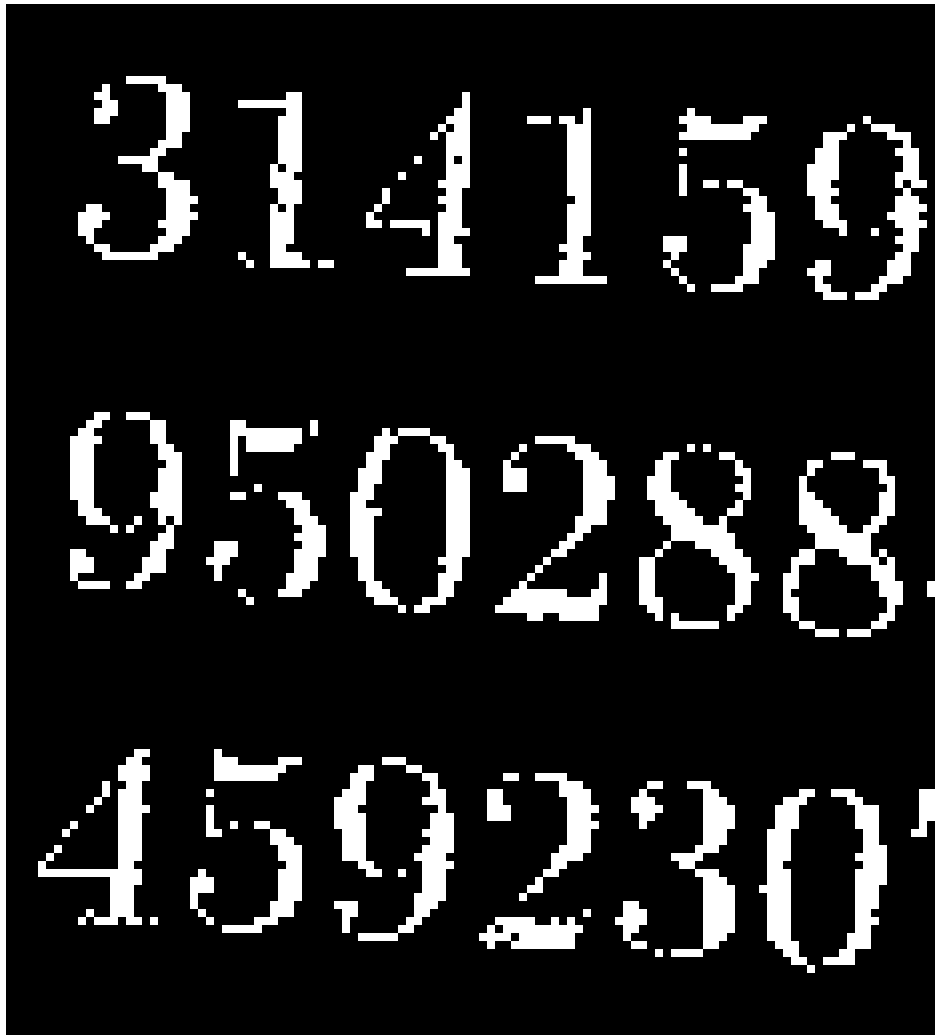
---



**Åpning med et sirkulært strukturelement**

# Eksempel: Filtrering ved lukking

---

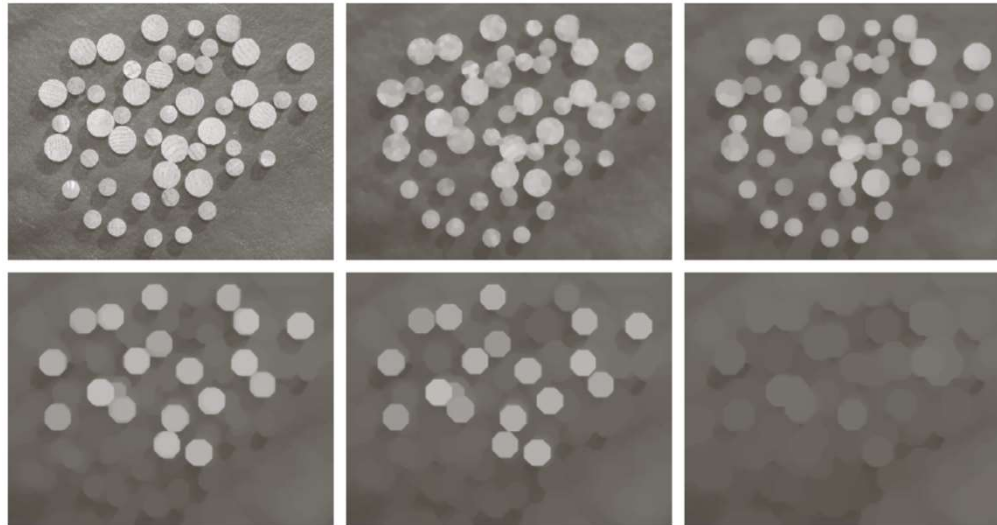


Lukking med et (3x3) kvadratisk strukturelement

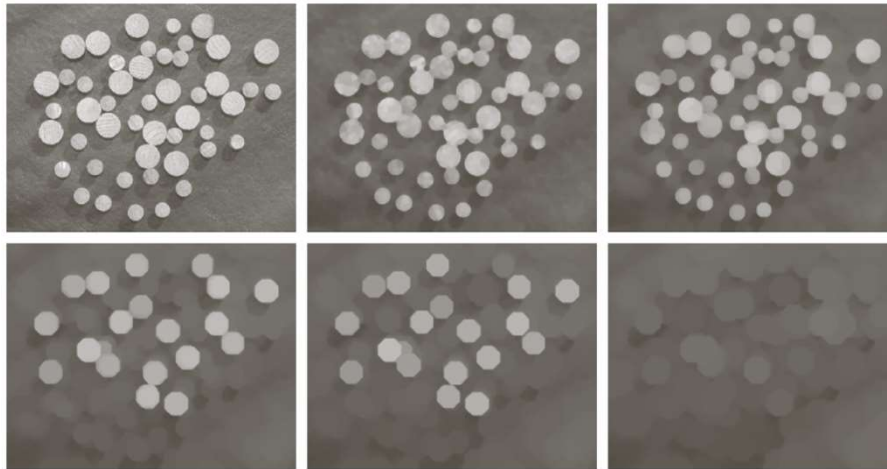


# Example application: granulometry

- ❑ Granulometry: determine the size distribution of particles in an image.
- ❑ Assumption: objects with regular shape on a background.
- ❑ Principle: perform a series of openings with increasing radius  $r$  of structuring element
- ❑ Compute the sum of all pixel values after the opening.
- ❑ Compute the difference in this sum between radius  $r$  and  $r-1$ , and plot this as a function of radius.

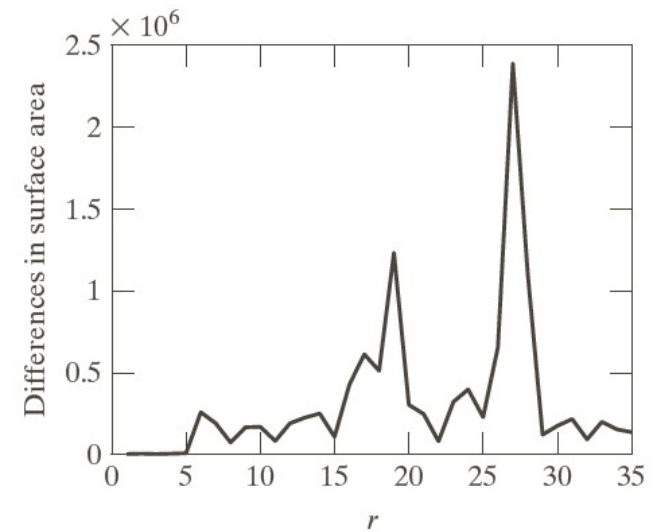


# Example - granulometry



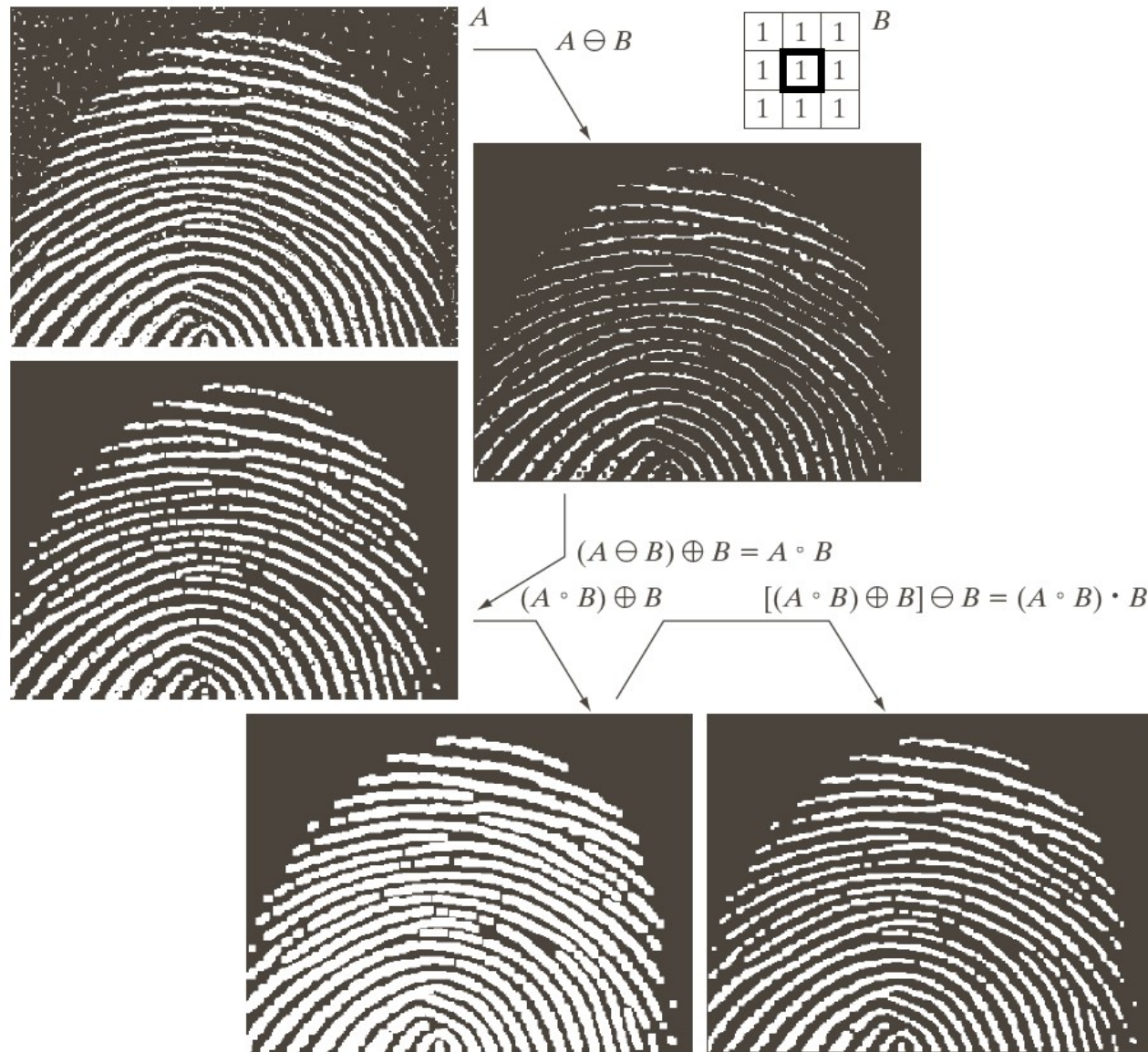
a b c  
d e f

**FIGURE 9.41** (a)  $531 \times 675$  image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)



**FIGURE 9.42** Differences in surface area as a function of SE disk radius,  $r$ . The two peaks are indicative of two dominant particle sizes in the image.

# Eksempel: Filtrering ved åpning og lukking




a	b
d	c
e	f

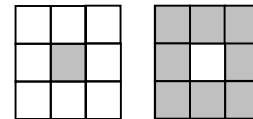
**FIGURE 9.11**  
 (a) Noisy image.  
 (b) Structuring element.  
 (c) Eroded image.  
 (d) Opening of  $A$ .  
 (e) Dilation of the opening.  
 (f) Closing of the opening.  
 (Original image courtesy of the National Institute of Standards and Technology.)

# «Hit-or-miss»-transformasjonen

- Tilbake til den opprinnelige situasjonen: Bilde  $f$  og strukturelement  $S$
- Men strukturelementet  $S$  er nå definert ved et par  $[S_1, S_2]$  av binære strukturelementer som ikke har noen felles elementer.
- «Hit-or-miss»-transformasjonen av  $f$  med  $S = [S_1, S_2]$  er definert som:

$$f(*)S = f(*) [S_1, S_2] = (f \theta S_1) \cap (f^c \theta S_2)$$

- En **forgrunns** **piksel** i ut-bildet **oppnås** kun hvis:
  - **$S_1$  passer forgrunnen** rundt pikselen **og**
  - **$S_2$  passer bakgrunnen** rundt pikselen.
- Kan brukes til å finne/behandle bestemte mønstre i et bilde, f.eks. til å:
  - Finne bestemte strukturer.
  - Fjerne enkeltpikslar. 
  - Benyttet i “tynning” (om to slides).



# Eksempel: «Hit-or-miss»

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0
0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Et bilde A

```

0 1 0
1 1 1
0 1 0

```

Strukturelement  
 $S_1$

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Resultat etter erosjon med  $S_1$

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1
1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1
1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1
1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

$A^c$  - komplementet til bildet  
(er 1 utenfor randen)

```

1 0 1
0 0 0
1 0 1

```

Strukturelement  
 $S_2$

```

1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1
1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1
1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1

```

$A^c$  erodert med  $S_2$

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

«Hit-or-miss»-resultatet

Logisk AND av de  
to delresultatene

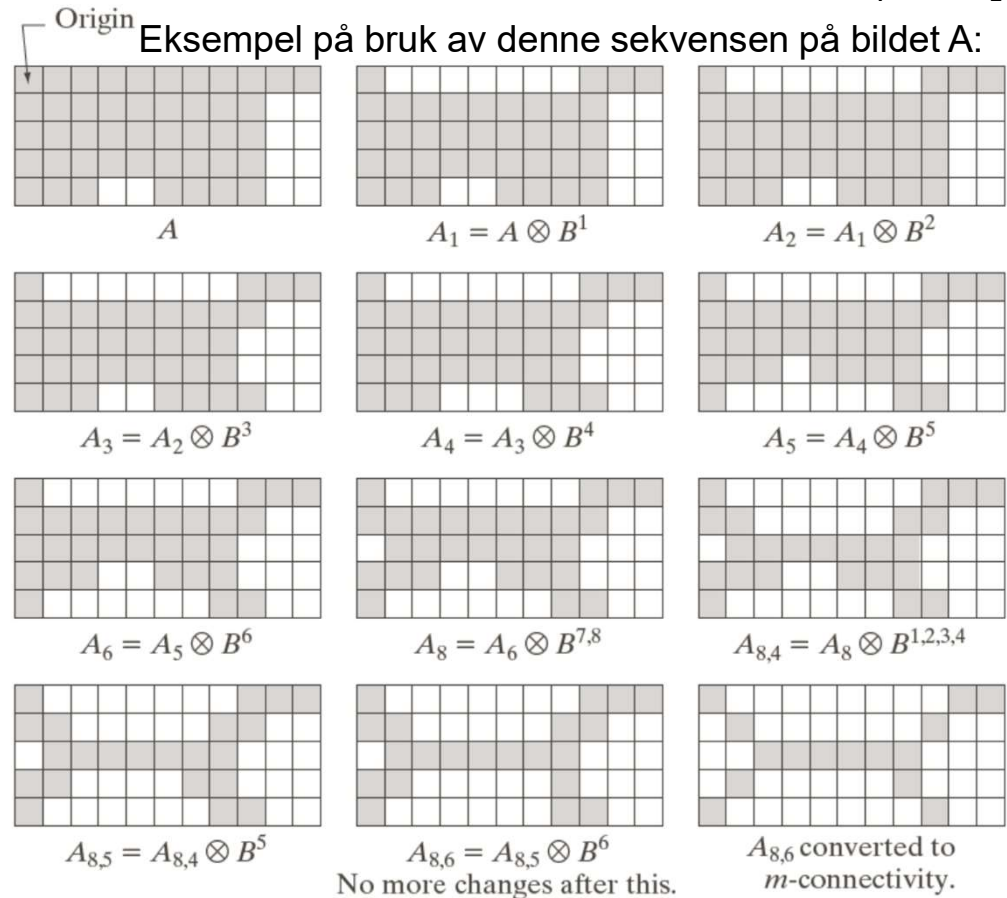
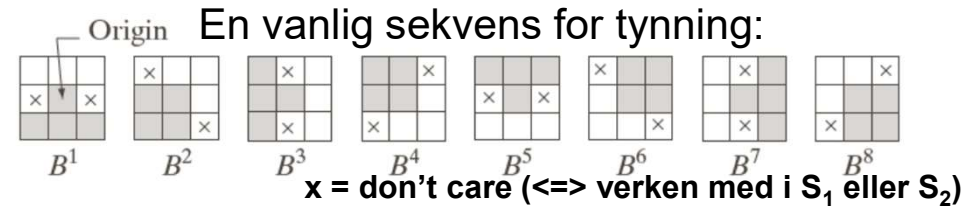
# Morfologisk tynning

- Morfologisk tynning:

$$f \otimes S = f - (f(*)S)$$

**gjentatt** med en sekvens av strukturelementer,  $S^k$  for  $k=1, \dots, n$ , inntil **ingen** av strukturelementene **skaper noen endring**.

- **Fjerner** grovt sett **alle piksler utenom** de som:
  - er isolerte,
  - definerer utstrekningen av et objekt, **eller**
  - trengs for å ikke dele et objekt.



(Figur 9.23 i G&W)  
Grå markerer forgrunn og hvit bakgrunn.

# Oppsummering

---

- Strukturelement (med origo)
- **Erosjon**
- **Dilasjon**
- Dualitet
- Åpning (erosjon etterfulgt av dilasjon)
- Lukking (dilasjon etterfulgt av erosjon)
- Hit-or-miss
- Tynning
- + Tenk over sammenligninger med filtrering