INF2310 – Digital bildebehandling

Forelesning 14

# Morfologiske operasjoner på binære bilder

Andreas Kleppe

Repetisjon av grunnleggende mengdeteori

Fundamentale operatorer

Sammensatte operatorer

Eksempler på anvendelser

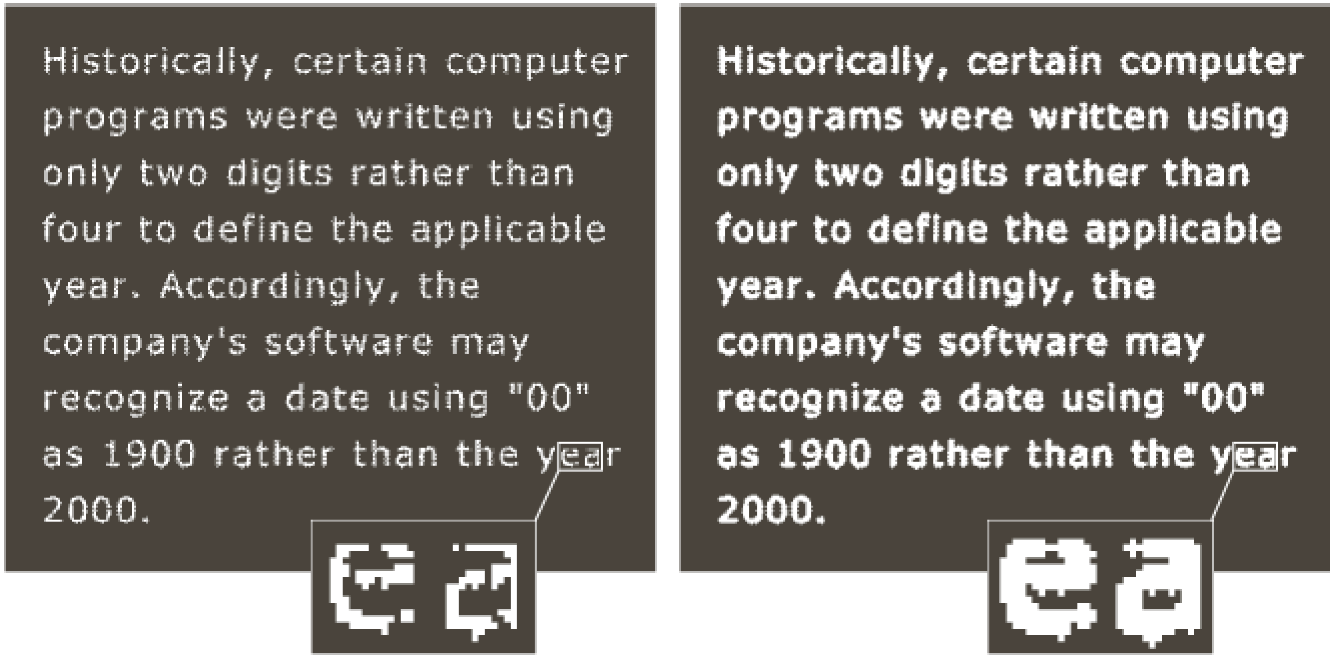
G&W: 9.1-9.5.2 og 9.5.5, og deler av 2.6.4 og 9.5.9

# Introduksjon

- **Matematisk morfologi** brukes ofte som et steg i analyse av bilder.
- **Modifiserer formen** (eng.: *shape*) **til objekter** v.b.a. 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 av 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 (har da enda flere anvendelser).

# Eksempel: Lenke sammen objekter

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




| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**FIGURE 9.7** (a) Sample text of poor resolution with broken characters (see magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

# Litt mengdeteori

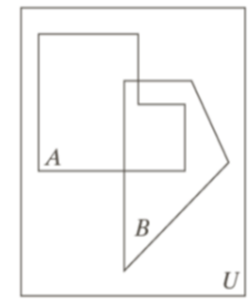
- En **mengde** (eng.: *set*) består av **elementer**.
  - **Rekkefølgen** av elementene og **antallet** av hver element er **ubestemt**.
- Dersom elementet a er inneholdt i mengden A skriver vi: $a \in A$
- Dersom elementet a ikke er inneholdt i mengden A skriver vi: $a \notin A$
- $\varnothing$ er 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.

- **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 forgrunnspikslenes koordinater. Mengden av disse pikslene er en mengde i $\mathbb{Z}^2$.

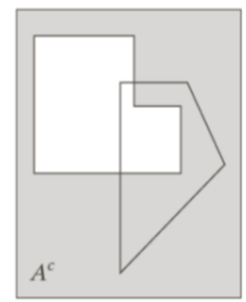
Konturen av to binære bilder, A og B




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

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

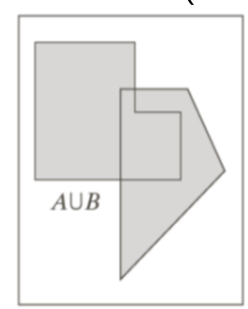
$$h(x,y) = \begin{cases} 1 \text{ hvis } f(x,y) = 0 \\ 0 \text{ hvis } f(x,y) = 1 \end{cases}$$




Komple-mentet til A

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

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

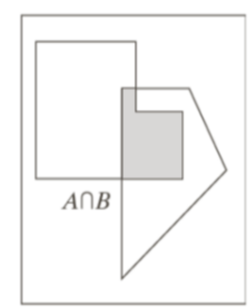



Unionen av A og B

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

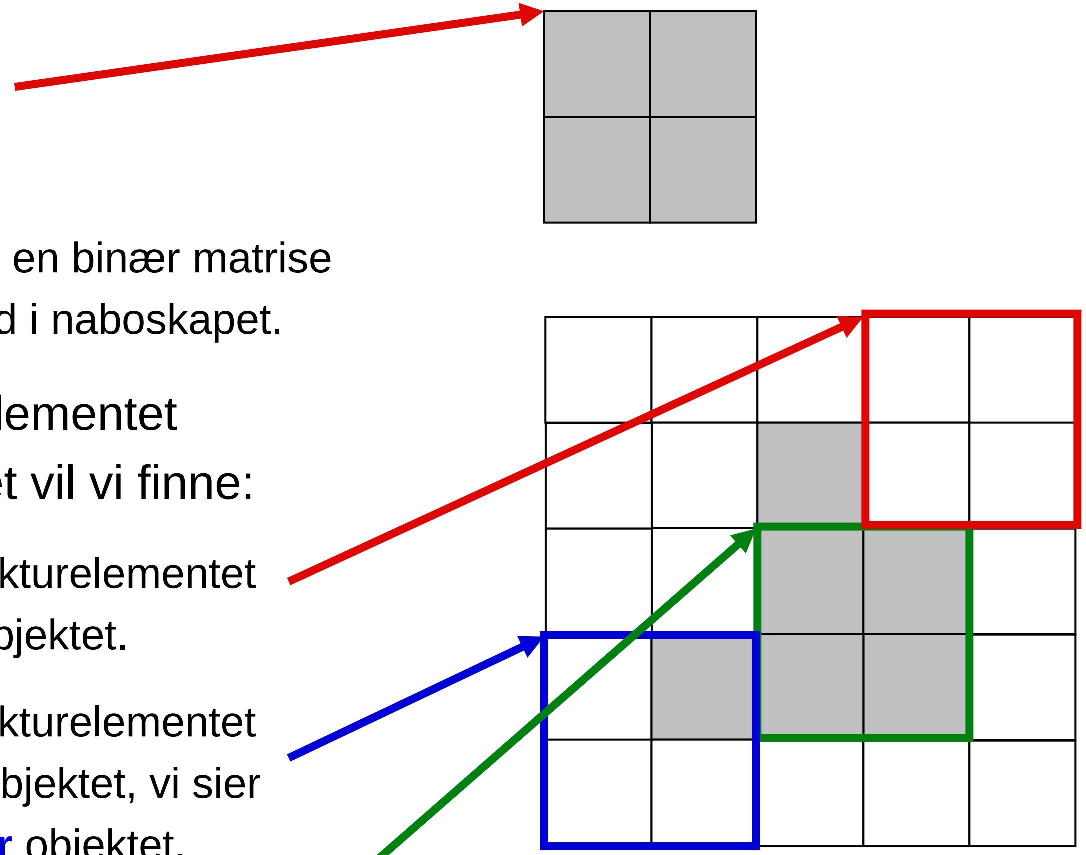
$$h = f \cap g \;\equiv h(x,y) = \begin{cases} 1 \text{ hvis } f(x,y) = 1 \text{ og } g(x,y) = 1 \\ 0 \text{ ellers} \end{cases}$$

Snittet av A og B



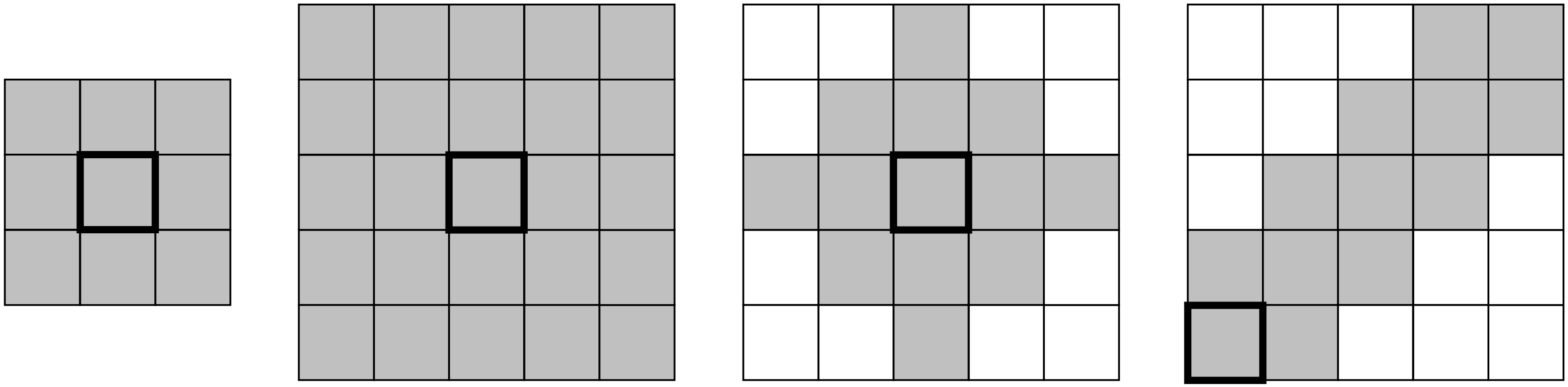
# Tre sentrale begrep

- 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** i objektet.



I figurene markerer grått med i mengden (forgrunnspiksel), og hvitt ikke med.

# Strukturelementenes form og origo



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

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

# <u>Passer</u> strukturelementet til det binære bildet?

- Vi flytter strukturelementet rundt over et binært bilde.
- Strukturelementet **passer** i posisjonen (x,y) i bildet hvis hvert element ≠ 0 i strukturelementet overlapper en pikselverdi ≠ 0 i bildet.
- Kan tenkes på som en **«minimums-korrelasjon»**.
- I denne sammenhengen vil vi alltid:
  - Ignorere pikselverdier som overlapper 0 i strukturelementet.
    - 0 markerer jo «ikke er med i naboskapet».
  - Anta at piksler utenfor bildet er 0.

**Et bilde**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 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 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**To forskjellige struktur-elementer**

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**To forskjellige resultater**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 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 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 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 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 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 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# Erosjon

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

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

$$\begin{matrix}0&1&0&0&0&0&0&1&1&0&0\\1&1&1&0&0&0&1&1&1&1&0\\0&1&1&1&0&1&1&1&1&0&0\\0&0&1&1&1&1&1&1&0&0&0\\0&0&0&1&1&1&1&1&1&0&0\\0&0&1&1&1&1&0&1&1&1&0\\0&1&1&1&1&0&0&0&1&1&1\\0&0&1&1&0&0&0&0&0&1&0\end{matrix}$$

**erodert med**

- Erosjon av et bilde f med strukturelementet S betegnes:

$$f \;\theta\; S$$

$$\begin{matrix}1&1&1\\1&\boxed{1}&1\\1&1&1\end{matrix} \qquad\qquad \begin{matrix}0&1&0\\1&\boxed{1}&1\\0&1&0\end{matrix}$$

**gir** **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:

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

$$\begin{matrix}0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0\end{matrix} \qquad \begin{matrix}0&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&1&1&0&0\\0&0&1&0&0&0&1&1&0&0&0\\0&0&0&1&0&1&1&0&0&0&0\\0&0&0&0&1&1&0&1&0&0&0\\0&0&0&1&1&0&0&0&1&0&0\\0&0&1&1&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0&0\end{matrix}$$

# 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**

$$\begin{matrix}1&1&1\\1&\boxed{1}&1\\1&1&1\end{matrix} \qquad\qquad \begin{matrix}0&1&0\\1&\boxed{1}&1\\0&1&0\end{matrix}$$

**gir** | **gir**

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

# 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:

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

$$\begin{matrix}
0&0&1&1&1&1&0&1&1&1&0\\
0&1&1&1&1&1&1&1&1&1&0\\
0&1&1&1&1&1&1&1&1&1&0\\
1&1&1&1&0&1&1&1&1&1&1\\
0&1&1&1&1&1&1&1&1&1&1\\
0&1&1&1&1&1&1&1&1&1&0\\
0&1&1&1&1&1&1&1&1&1&0\\
0&0&0&0&0&1&1&1&0&0&0
\end{matrix}$$

**erodert 2 ganger med**

$$\begin{matrix}
1&1&1\\
1&\boxed{1}&1\\
1&1&1
\end{matrix}
\qquad\qquad
\begin{matrix}
0&1&0\\
1&\boxed{1}&1\\
0&1&0
\end{matrix}$$

**gir** / **gir**

$$\begin{matrix}
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0
\end{matrix}
\qquad\qquad
\begin{matrix}
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&1&1&1&0&0\\
0&0&0&0&0&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0
\end{matrix}$$

# 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: $\mathbf{g = f - (f \ominus S)}$
- Strukturelementet avgjør 4- eller 8-tilkoblet:

**Et bilde**

```
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**

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

**gir**

```
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
```

**=>** **differanse**

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

**Sammenhengende kanter dersom man bruker 8-tilkobling**

**erodert med**

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

**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
```

**=>** **differanse**

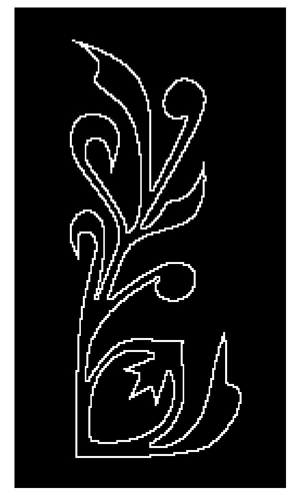
```
0 0 1 1 1 1 0 1 1 1 0
0 1 1 0 0 1 1 1 0 1 0
0 1 0 1 1 1 0 0 0 1 0
1 1 0 1 0 1 0 0 0 1 1
0 1 0 1 1 1 0 0 0 1 1
0 1 0 0 0 0 0 0 0 1 0
0 1 1 1 1 1 0 1 1 1 0
0 0 0 0 0 1 1 1 0 0 0
```

**Sammenhengende kanter ved bruk av 4-tilkobling**

# Eksempel: Kantdeteksjon ved erosjon

$$\begin{matrix}0 & 1 & 0\\ 1 & \boxed{1} & 1\\ 0 & 1 & 0\end{matrix}$$

$$\begin{matrix}1 & 1 & 1\\ 1 & \boxed{1} & 1\\ 1 & 1 & 1\end{matrix}$$



**Et bilde** **erodert med** **=>** **differanse**

# Treffer strukturelementet det binære bildet?

- Vi flytter strukturelementet rundt over et binært bilde.
- Strukturelementet **treffer** i posisjonen (x,y) i bildet hvis et element ≠ 0 i strukturelementet overlapper en pikselverdi ≠ 0 i bildet.
- Her reflekterer vi (roterer 180°) strukturelementet før vi flytter det rundt.
- Kan tenkes på som en **«maximums-konvolusjon»**.

**Et bilde**

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

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

**To forskjellige struktur-elementer**

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

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

**To forskjellige resultater**

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

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:

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

- Dilasjon av et bilde f med strukturelementet S betegnes:

$$f \oplus S$$

- 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:

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

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

**dilatert med**

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

**gir**

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

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

**gir**

```
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
```

# Effekter av dilasjon

- Dilasjon **utvider** objekter.
- Dilasjon fyller i hull i objektet.
  - Fyller igjen hullet hvis strukturelementet er stort nok i forhold til hullet.
- Dilasjon glatter ut innbuktninger i objektets omriss.
- Resultatet er avhengig av strukturelementet.
- Større strukturelement gir større dilasjons-effekt.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 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**

$$\begin{matrix} 1 & 1 & 1 \\ 1 & \boxed{1} & 1 \\ 1 & 1 & 1 \end{matrix}$$

**gir**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 | 0 | 0 | 0 | 0 | 0 |

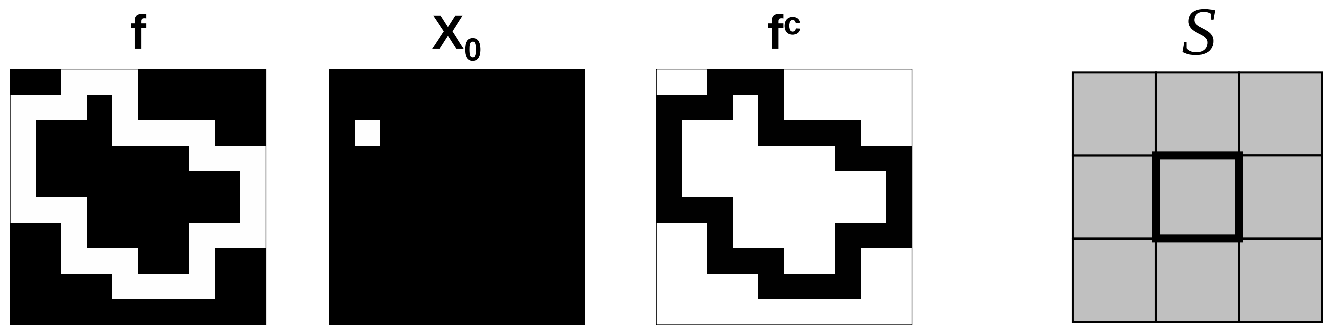
$$\begin{matrix} 0 & 1 & 0 \\ 1 & \boxed{1} & 1 \\ 0 & 1 & 0 \end{matrix}$$

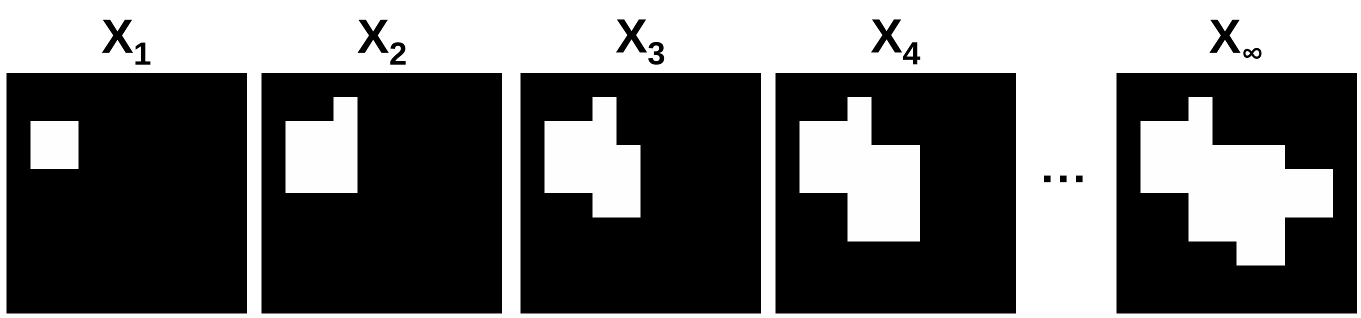
**gir**

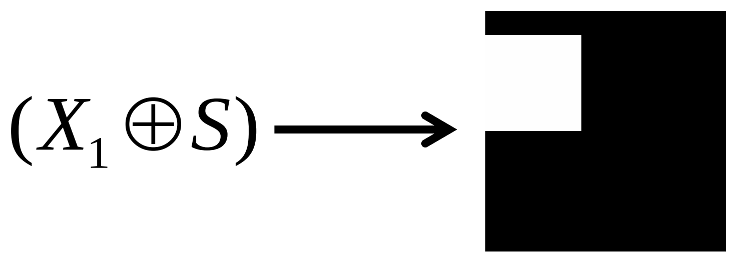
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 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.
- Deretter iterer $X_k = (X_{k-1} \oplus S) \cap f^c$ inntil konvergens:



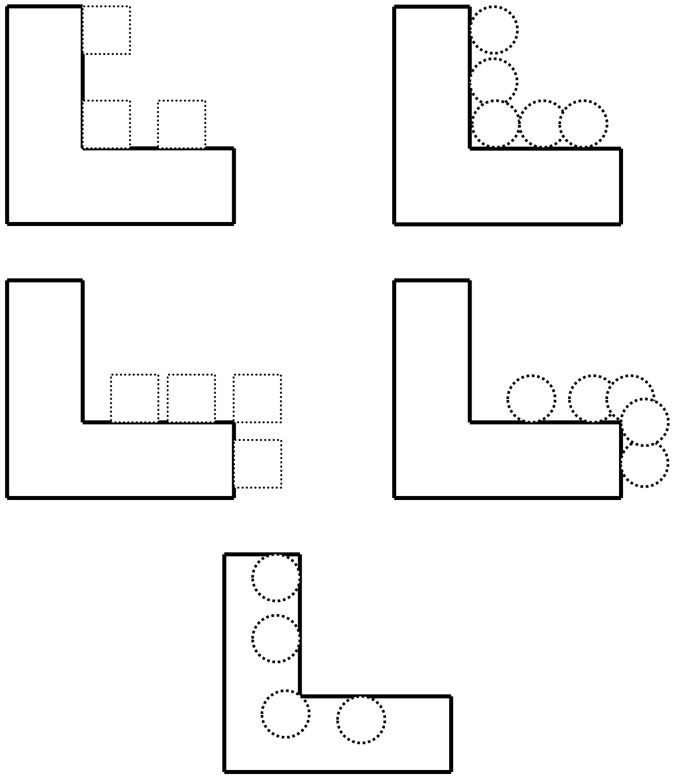




(Bildene er hentet fra
http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm)

# Effekter av sirkulære strukturelementer på hjørner

- Både dilatering og erodering med **kvadratiske strukturelementer bevarer formen til hjørner**.
- **Dilatering** av **konkave** hjørner med sirkulære strukturelementer **bevare** hjørnenes form.
- **Dilatering** av **konvekse** hjørner med sirkulære strukturelementer **avrunde** hjørnene.
- Sirkulære strukturelementer gir **avrundede** hjørner ved **erosjon** av **konkave** hjørner.
  - Konvekse hjørner bevares.



# Dualitet

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

$$f \oplus S = \left(f^c \,\theta\, \hat{S}\right)^c$$

$$f \;\theta\; S = \left(f^c \oplus \hat{S}\right)^c$$

- For å dilatere f med <u>symmetrisk</u> 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 180° rotere et strukturelement og finne komplementet til et binært bilde.

**et bilde**

```
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
```

**komplementet**

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

**dilatert med**

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

**erodert med**

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

**gir**

```
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
```

**gir**

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

**og disse bildene er komplementære**

# 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, d.v.s. 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:

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

# 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?
    «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

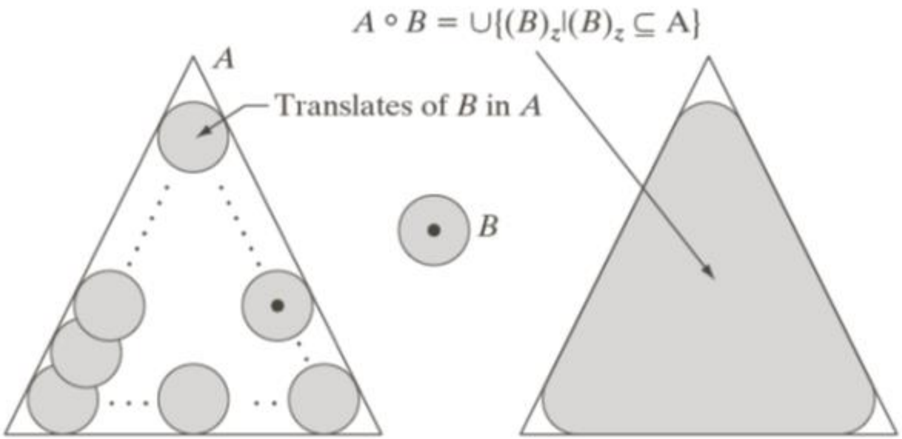
- **Erosjon** av et bilde **fjerner** alle strukturer som ikke kan inneholde strukturelementet, og «**krymper**» alle andre strukturer.
- Hvis vi **dilaterer** resultatet av en erosjon med samme strukturelement, vil de strukturene som «**overlevde**» erosjonen bli omtrentlig **gjenskapt**.
- Dette er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

$$f \circ S = \left(f \;\theta\; \mathrm{S}\right) \oplus S$$

- 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 spisser av en tusjpenn**.
- Det er bare tillatt å **fargelegge innenfor objekter**.
  - Et par detaljer: Man må holde tusjen vinkelrett på tegneflaten og med samme rotasjon som strukturelementet.




(Deler av figur 9.8 i G&W)

- **Åpningen er resultatet av å fargelegge så mye man har lov til.**
- For runde strukturelementer: Konvekse hjørner blir avrundet, konkave hjørner beholdes rette.
  - Akkurat som ved dilasjon (skyldes at enhver åpning avsluttes med en dilasjon).

Åpning er **idempotent**:

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

d.v.s. at gjentatte anvendelser med samme strukturelement gir ingen endring.

# Lukking

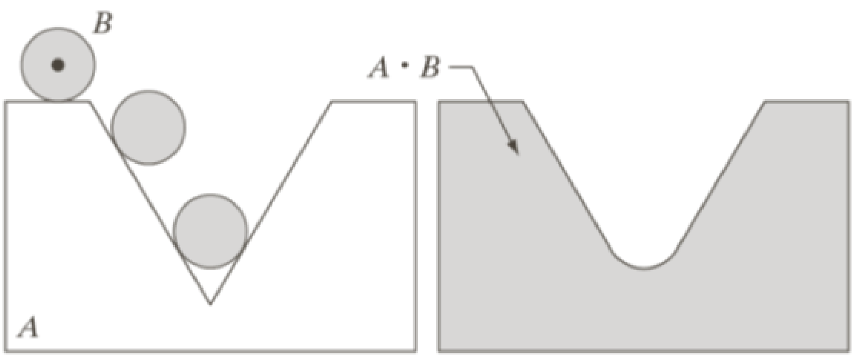
- **Dilasjon** av et bilde **utvider** strukturer,
  **fyller** i **hull og innbuktninger** i omrisset.
- Hvis vi **eroderer** resultatet av en dilasjon med samme strukturelement, vil strukturene **stort sett** få **gjenskapt** sin opprinnelige størrelse og form, men **hull og innbuktninger** som ble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

$$f \bullet S = \left(f \oplus S\right) \theta \; S$$

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

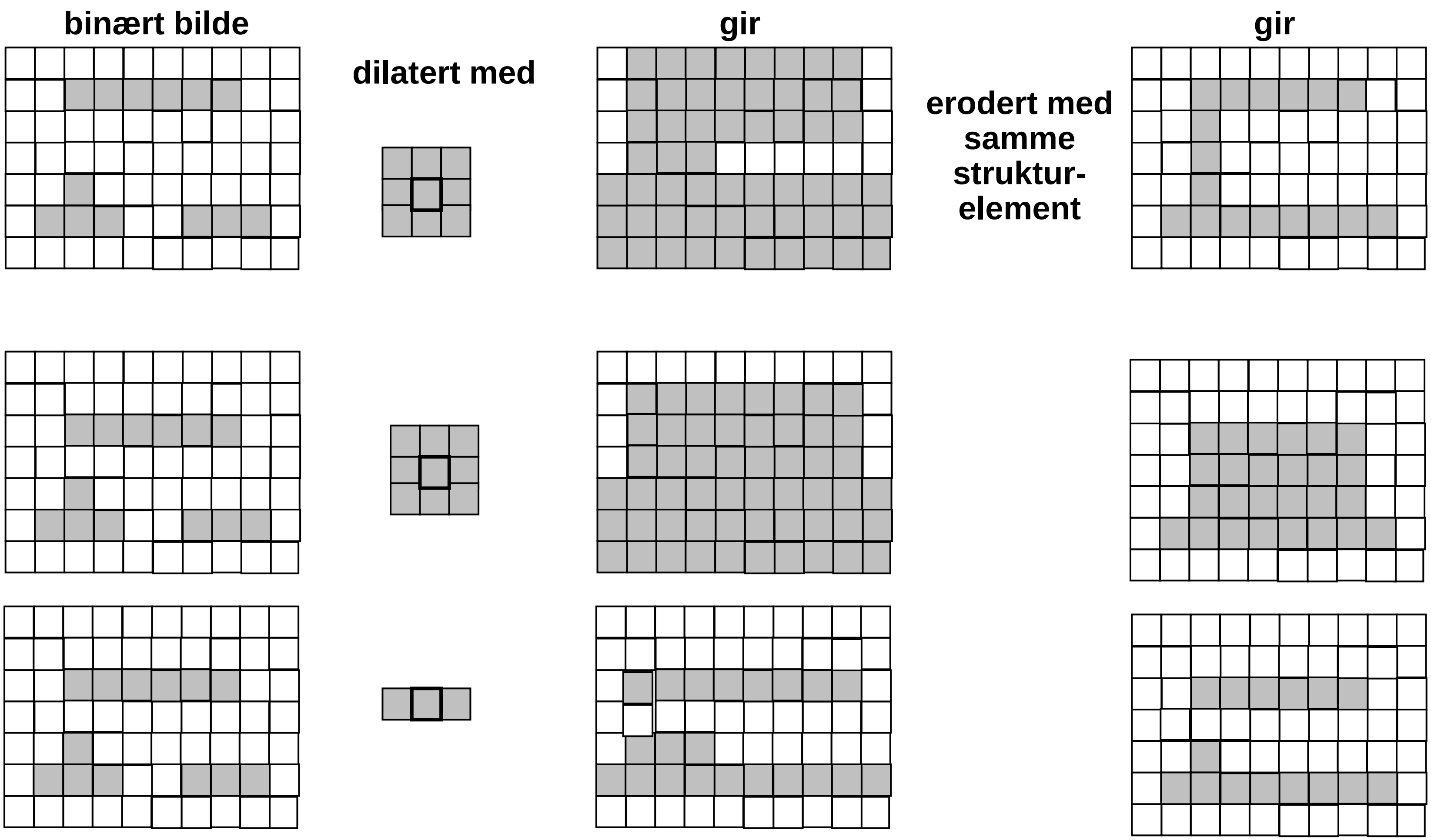



(Deler av figur 9.9 i G&W)

Også lukking er **idempotent**:

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

# Lukking lukker små åpninger




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

# Dualitet mellom åpning og lukking

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

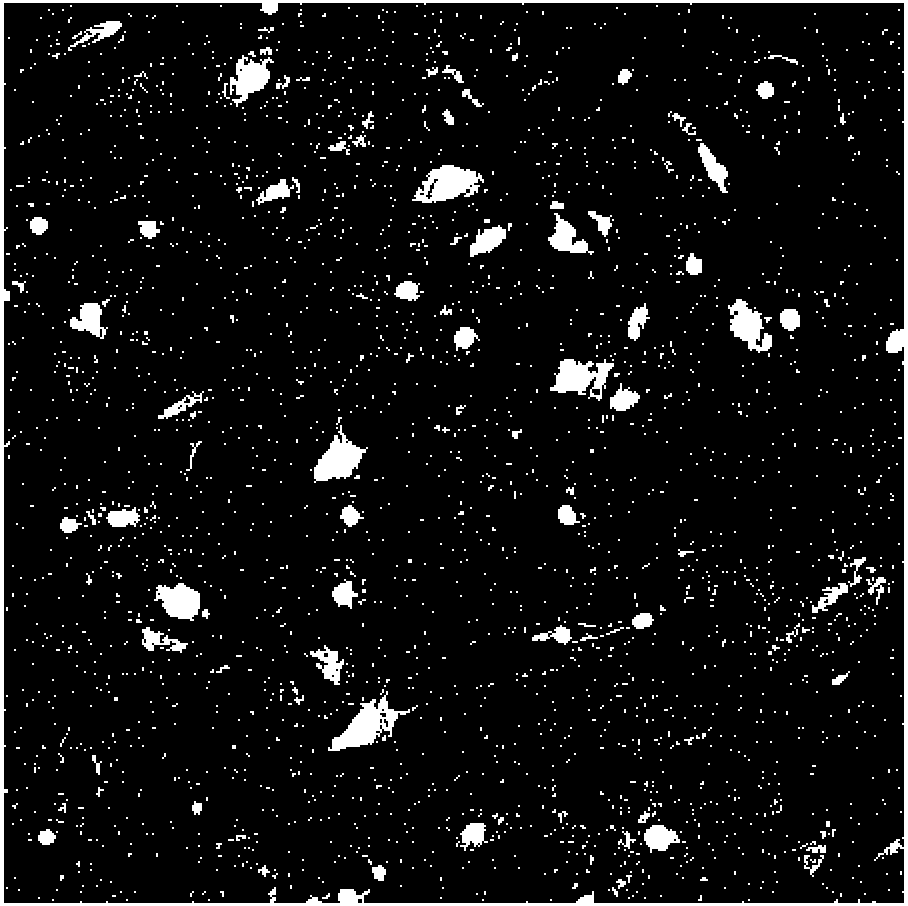
$$f \bullet S = (f^c \circ \hat{S})^c \qquad f \circ S = (f^c \bullet \hat{S})^c$$

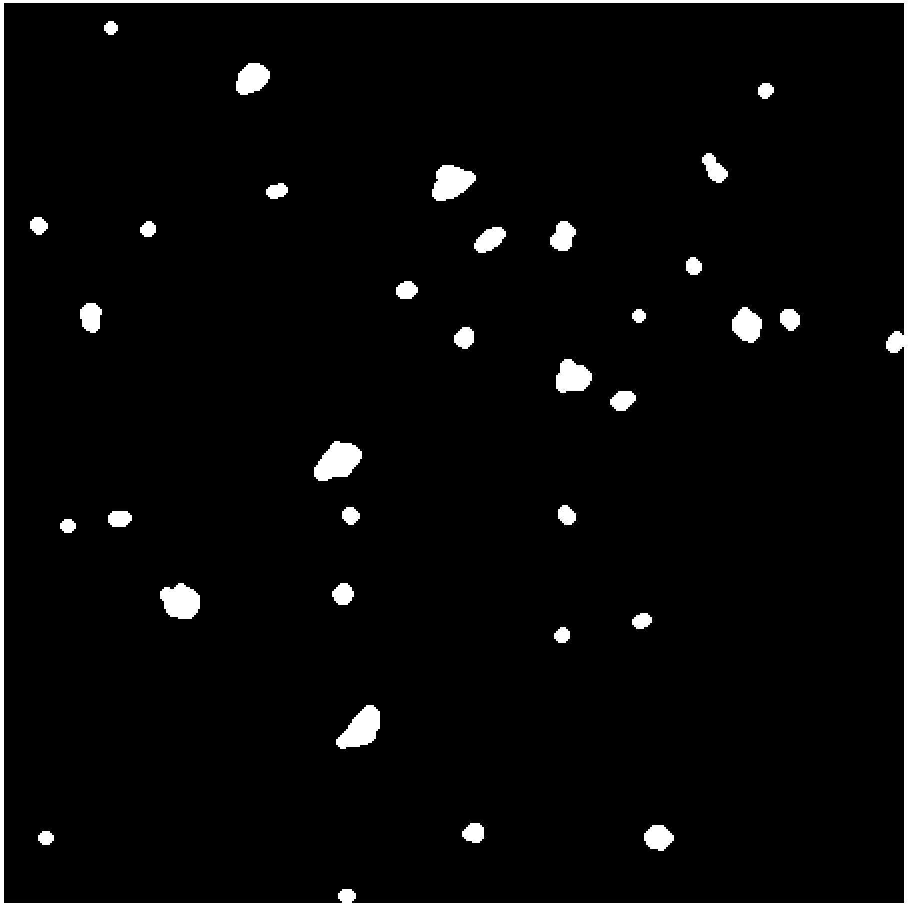
- Lukking kan utføres ved å komplementere bildet, åpne det med det speilvendte (180° rotere) strukturelementet, og ta komplementet av resultatet.
  - Tilsvarende for åpning.
- Vi kan altså utføre begge operasjonene med kode bare for den ene, hvis vi har kode å speilvende og komplementere et binært bilde.

- **Lukking** er en **ekstensiv** transformasjon (piksler legges til).
- **Åpning** er en **antiekstensiv** transformasjon (piksler fjernes).

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

# Eksempel: Støyfjerning med åpning

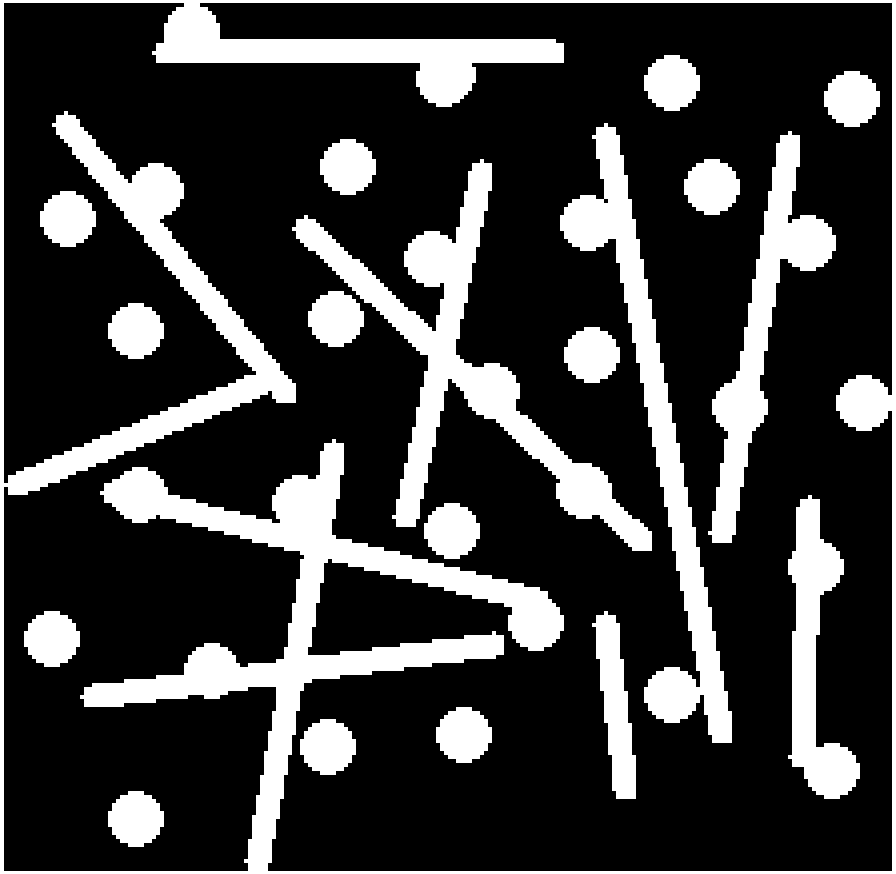


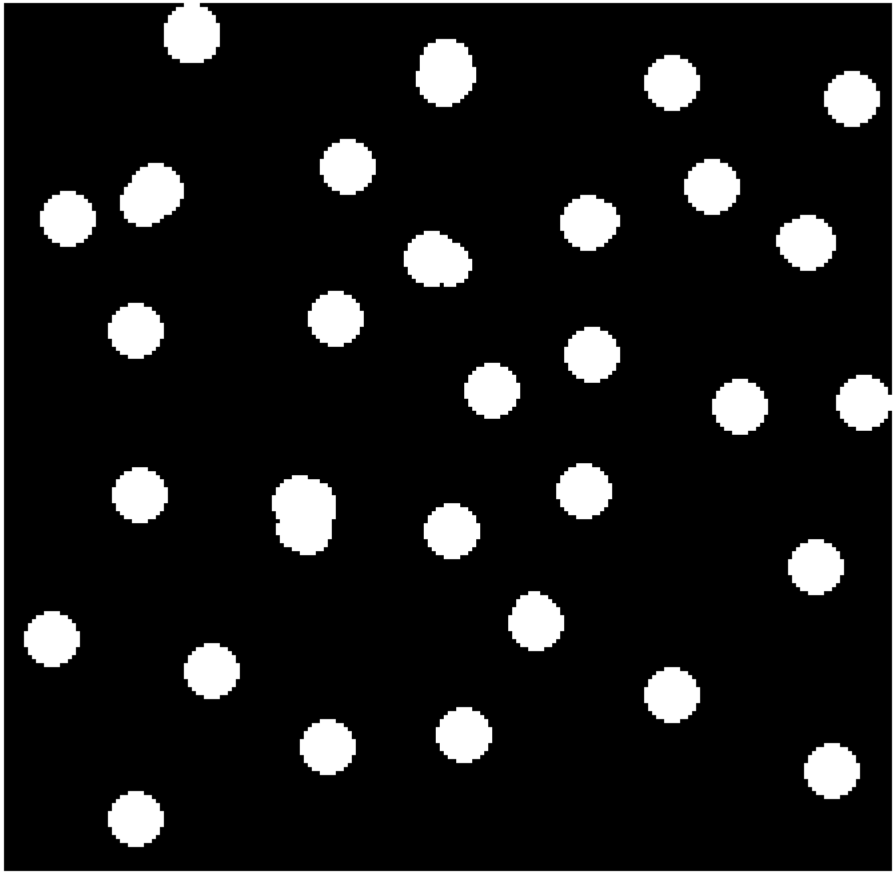


**Åpning med et (7x7) sirkulært strukturelement**

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

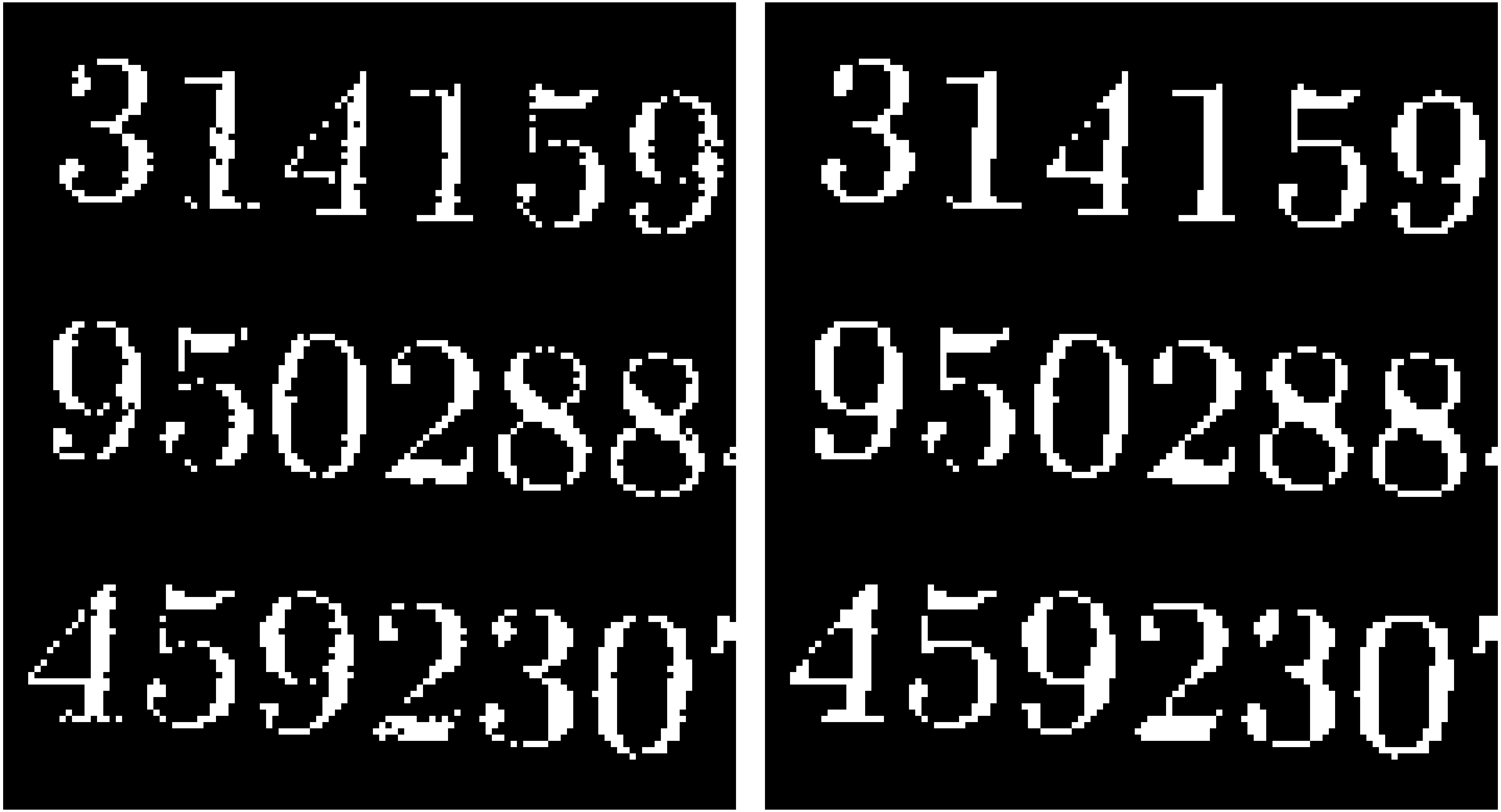
# Eksempel: Formseparering ved åpning





**Åpning med et sirkulært strukturelement**

# Eksempel: Filtrering ved lukking




**Lukking med et (3x3) kvadratisk strukturelement**

# 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.)

# 9.5.9 Morfologisk rekonstruksjon

- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G og et strukturelement B som definerer tilkoblingstypen
- **Morfologisk rekonstruksjon ved dilasjon**:

$$F^k = (F^{k-1} \oplus B) \bigcap G$$

der $F^0 = F$.

- Med ord: Iterativt dilater F med B,
  og begrens resultatet med G,
  inntil ingen endring.

- B = $\begin{matrix} 0 & 1 & 0 \\ 1 & \boxed{1} & 1 \\ 0 & 1 & 0 \end{matrix}$ $\qquad$ $\begin{matrix} 1 & 1 & 1 \\ 1 & \boxed{1} & 1 \\ 1 & 1 & 1 \end{matrix}$

definerer 4-tilkobling 8-tilkobling i markørbildet

  - Merk: Omvendt som for kantdeteksjonen!
    … fordi ut-bildet da var det som ble erodert v.b.a. strukturelementet.

# Åpning ved rekonstruksjon

- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G og et strukturelement B som definerer tilkoblingstypen
- **Morfologisk rekonstruksjon ved dilasjon**:

$$F^k = (F^{k-1} \oplus B) \bigcap G$$

der $F^0 = F$.

- Med ord: Iterativt dilater F med B, og begrens resultatet med G, inntil ingen endring.

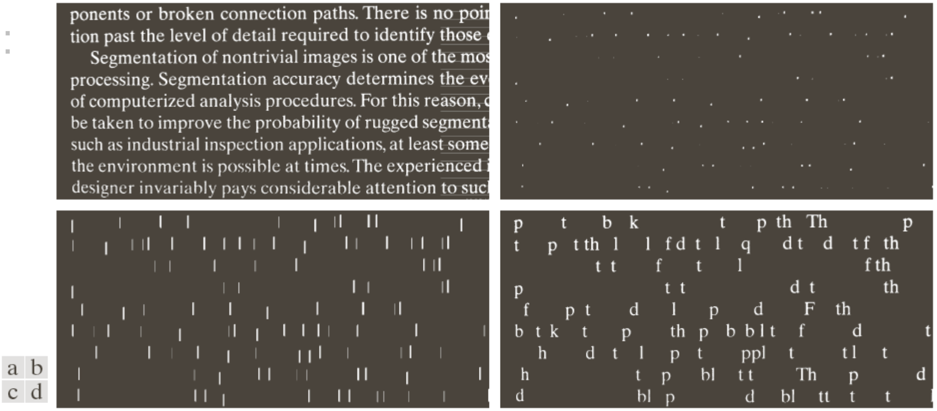



**FIGURE 9.29** (a) Text image of size 918 × 2018 pixels. The approximate average height of the tall characters is 50 pixels. (b) Erosion of (a) with a structuring element of size 51 × 1 pixels. (c) Opening of (a) with the same structuring element, shown for reference. (d) Result of opening by reconstruction.

- Åpning ved rekonstruksjon (f med S):
  - Lag markørbildet F ved å n ganger erodere f med S (n er en tilleggsparameter).
    - Kunne vært definert uten parameteren n. Da må S definerer hele eroderingen.
  - Utfør morfologisk rekonstruksjon ved dilasjon der G = f.
- Tilsvarer vanlig åpning, bare at de gjenværende objektene blir *nøyaktig bevart*.

# Helautomatisk hullfylling

- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G og et strukturelement B som definerer tilkoblingstypen
- **Morfologisk rekonstruksjon ved dilasjon**:

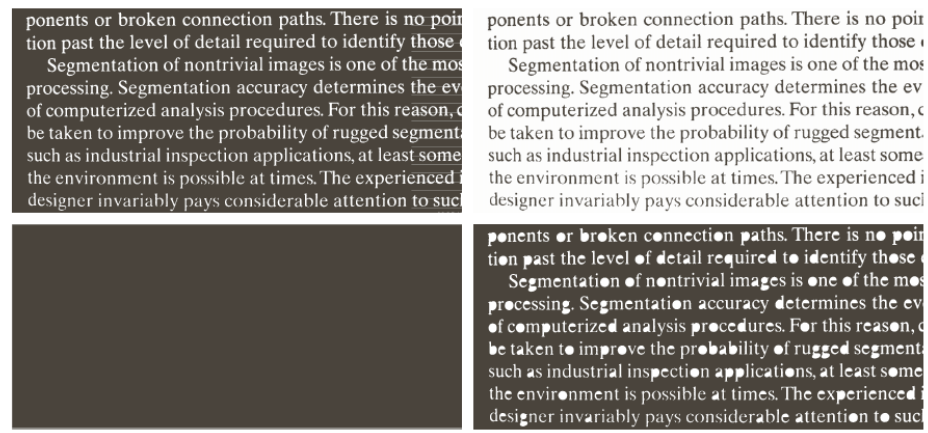
$$F^k = (F^{k-1} \oplus B) \cap G$$

der $F^0 = F$.

- Med ord: Iterativt dilater F med B, og begrens resultatet med G, inntil ingen endring.

a b
c d

**FIGURE 9.31** (a) Text image of size 918 × 2018 pixels. (b) Complement of (a) for use as a mask image. (c) Marker image. (d) Result of hole-filling using Eq. (9.5-29).



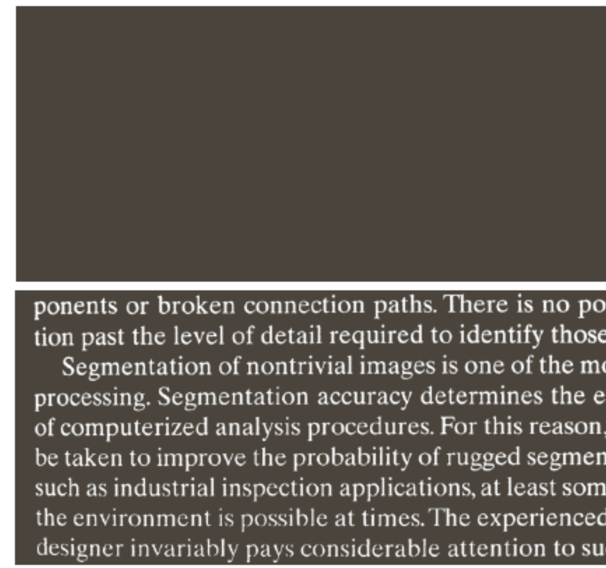

- Helautomatisk hullfylling (av bildet f):
  - La markørbildet F(x,y) være 1-f(x,y) når (x,y) er på bildekanten, og ellers 0;

$$F(x,y) = \begin{cases} 1 - f(x,y) & \text{hvis } (x,y) \text{ er på bildekanten} \\ 0 & \text{ellers} \end{cases}$$

  - Utfør morfologisk rekonstruksjon ved dilasjon der $G = f^c$ og komplementer resultatet.

# Kantrydding

FIGURE 9.32 Border clearing. (a) Reconstruction-by-dilation of marker image. (b) Image with no objects touching the border. The original image is Fig. 9.29(a).




- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G og et strukturelement B som definerer tilkoblingstypen
- **Morfologisk rekonstruksjon ved dilasjon**:

$$F^k = (F^{k-1} \oplus B) \cap G$$

der $F^0 = F$.

- Med ord: Iterativt dilater F med B, og begrens resultatet med G, inntil ingen endring.
- Kantrydding (av bildet f) (er også helautomatisk):
  - La markørbildet F være lik f på bildekanten, og ellers 0;

$$F(x,y) = \begin{cases} f(x,y) & \text{hvis } (x,y) \text{ er på bildekanten} \\ 0 & \text{ellers} \end{cases}$$

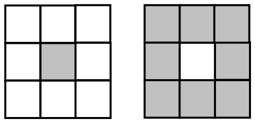
  - Utfør morfologisk rekonstruksjon ved dilasjon der G = f og subtraher resultatet fra f.

# «Hit-or-miss»-transformasjonen

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

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

- Et **forgrunnspiksel** i ut-bildet **oppnås** kun hvis:
  - **$S_1$ passer forgrunnen** rundt pikselet og
  - **$S_2$ passer bakgrunnen** rundt pikselet.
- Kan brukes til å finne/behandle bestemte mønstre i et bilde, f.eks. til å:
  - Finne bestemte strukturer.
  - Fjerne enkeltpiksler.
  - 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 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
```

**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**

```
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 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:

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

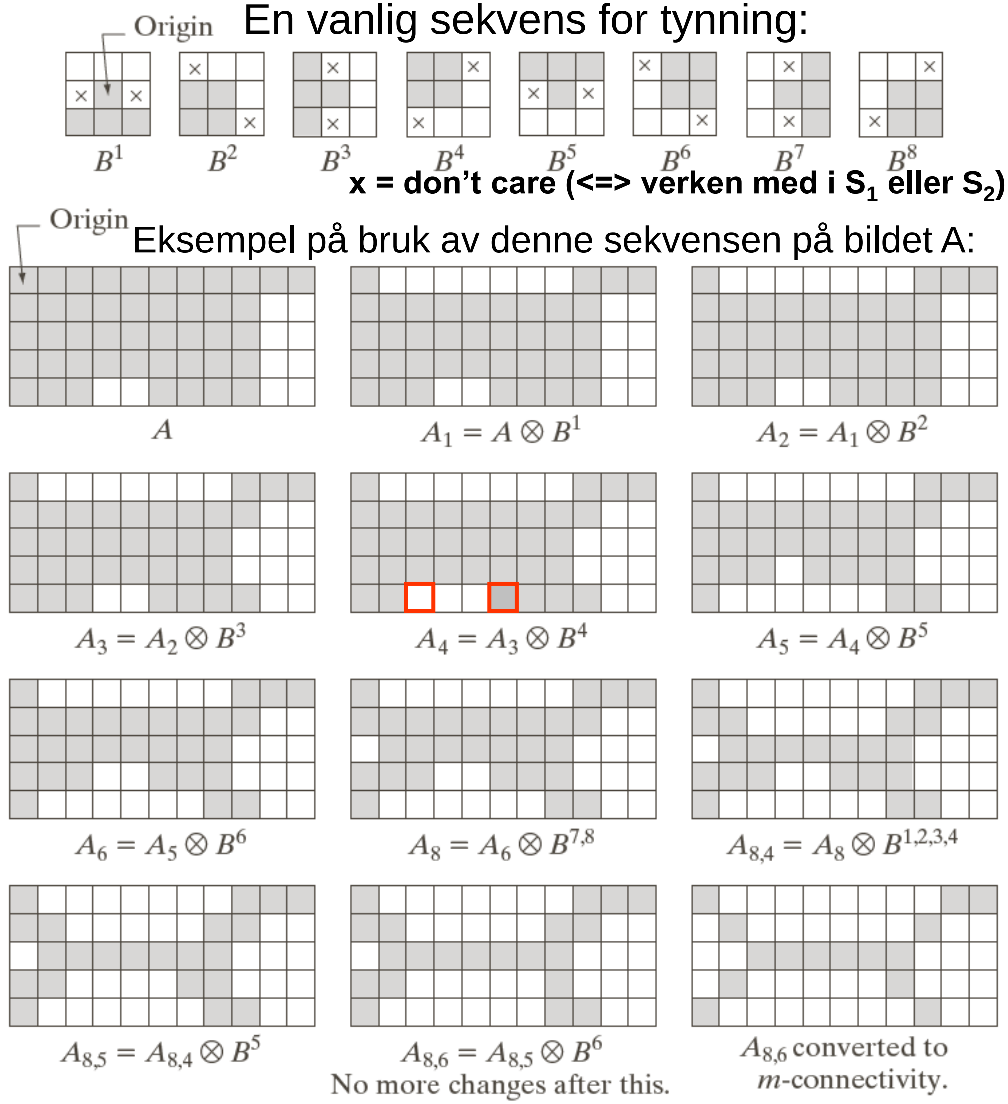
**gjentatt** med en sekvens av strukturelementer, $S^k$ for k=1,..,n, inntil **ingen** av strukturelementene **skaper noen endring**.

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

En vanlig sekvens for tynning:

**x = don't care (<=> verken med i $S_1$ eller $S_2$)**

Eksempel på bruk av denne sekvensen på bildet A:




(Figur 9.21 i G&W)

□ og ■ markerer to korreksjoner.

# Oppsummering

- Grunnleggende begreper:
  - Strukturelement (med origo)
  - **Erosjon**
  - **Dilasjon**
  - Dualitet
  - Åpning (erosjon etterfulgt av dilasjon)
  - Lukking (dilasjon etterfulgt av erosjon)
  - Morfologisk rekonstruksjon
  - Hit-or-miss
- Eksempler på anvendelser.