INF2310 – 15.05.2012

Morfologiske operasjoner på binære bilder

- Repetisjon av grunnleggende mengdeteori
- Fundamentale operatorer
- Sammensatte operatorer
- Eksempler på anvendelser er flettet inn

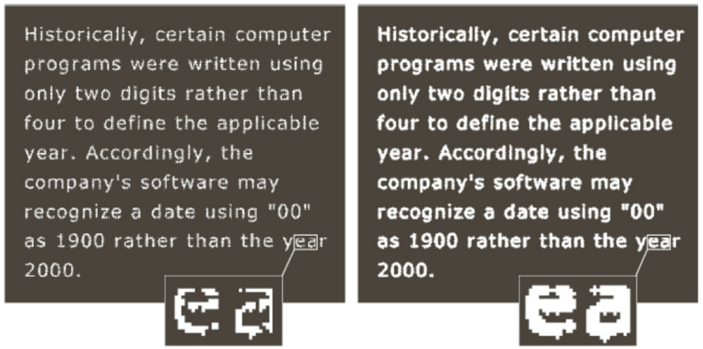
DIP: 9.1-9.4, 9.5.1, 9.5.2 og 9.5.5, og
deler av 2.6.4 og 9.5.9, sistnevnte bare kursorisk

Introduksjon

- Analyse av bilder og objekter basert på form (eng.: *shape*).
- Modifiserer former ved lokale operasjoner basert på mengdeteori (eng.: *set theory*).
- Mange operasjoner benyttes til å fjerne uønskede effekter etter segmentering:
  - Fjerning av små objekter (antas å være støy).
  - Glatting av omrisset til større objekter.
  - Fjerning av hull i objekter.
  - Lenke sammen objekter.
- Noen operasjoner er nyttige til analyse og beskrivelse av objekter:
  - Finne omriss av objekter.
  - Finne objektsskjeletter.
  - Finne objekter som inneholder en viss struktur.
  - Finne mønstre i bilder.
- Operasjonene er ofte enkle og kan utføres svært raskt.
- Kan generaliseres til gråtonebilder.

Eksempel: Lenke sammen objekter

- Morfologiske operasjoner er ofte velegnet til å forbedre en segmentering.
- Eks.: Defragmentere objekter:



0	1	0
1	1	1
0	1	0

a c
b

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.

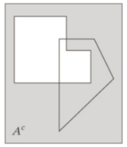
Litt mengdeteori

- En *mengde* består av *elementer*.
  - *Rekkefølgen* av elementene og *antallet* av hver element er *ubestemt*.
- Dersom a er inneholdt i A skriver vi: $a \in A$
- Dersom a ikke er inneholdt i 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 elle elementer som er i både A og B, dette betegnes:
  $$A \cap B$$

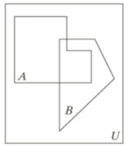
Mengder og binære bilder

- La A være en mengde i $\mathbb{Z}^2$.
  - Hver 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 koordinatene til sine hvite punkter. Mengden av disse punktene er en mengde i $\mathbb{Z}^2$.
- **Komplementet** til et binært bilde f:

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



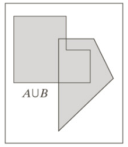
Komple-mentet til A



Konturen av to binære bilder, A og B

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

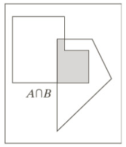
$$h = f \cup g \ \equiv \ h(x,y) = \begin{cases} 1 \text{ hvis } f(x,y)=1 \ \text{ eller } g(x,y)=1 \\ 0 \text{ ellers} \end{cases}$$



Unionen av A og B

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

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

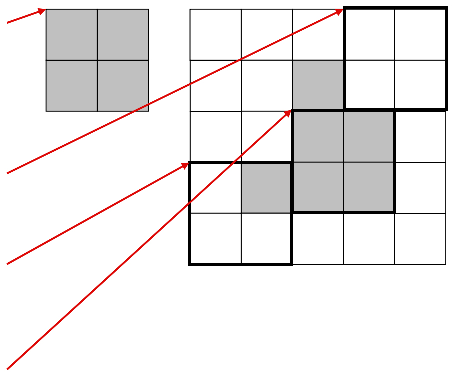


Snittet av A og B

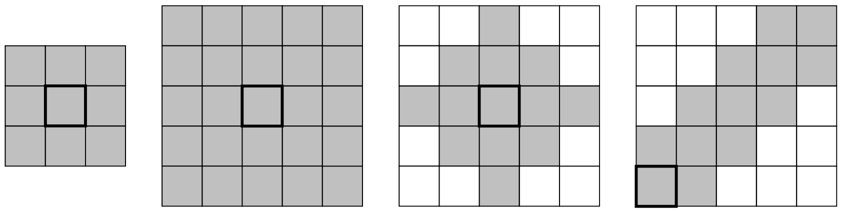
(Deler av figur 2.31 i DIP)

Tre sentrale begrep

- Et **strukturelement** for et binært bilde er en liten matrise av piksler
- 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.



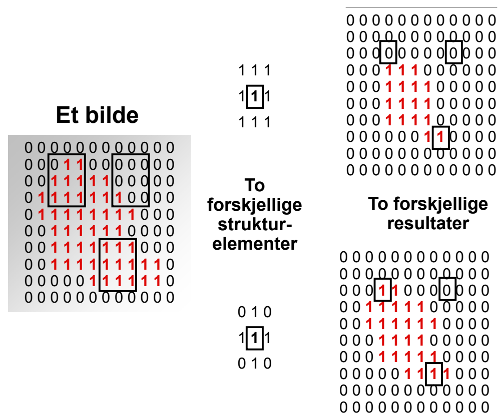
Strukturelementenes form og origo



- 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 □

Passer strukturelementet til det binære bildet?

- Anta at vi flytter strukturelementet rundt over et binært bilde.
- Strukturelementet **passer** i posisjonen (x,y) i bildet hvis hvert piksel ≠ 0 i struktur-elementet svarer til en pikselverdi ≠ 0 i bildet.
- Dette er en slags binær (logisk) korrelasjon.
- Pikselverdier som faller under strukturelement-verdier som er lik 0 er irrelevante!
- Piksler utenfor bildet antas å være 0.



Erosjon

- Plasser strukturelementet S slik at origo ligger over bildet f med koordinatene (x,y), og bruk regelen:

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

- Erosjon av et bilde f med strukturelementet S betegnes:

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

- 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:

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

```
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

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

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


Effekter av erosjon

- Erodering krymper objekter.
- Piksler fjernes også innenfra, hvis objektet har hull.
- Erosjon fjerner *små* utstikk på objektets omriss.
- Erosjon utvider innbuktninger i objektets omriss.
- Resultatet er avhengig av strukturelementets form.
- 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        0 1 0
1 1 1        1 1 1
1 1 1        0 1 0
```

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

- Større strukturelement gir mer erosjon/fjerning.
- Resultatet av erosjon med et stort strukturelement er (nesten) lik resultatet av gjentatt erosjon med et mindre element 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
```

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 0 0 0 0 0 0 0 0        0 0 0 0 0 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 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 - Kantdeteksjon ved erosjon

- Erodering fjerner piksler langs omrisset av en region.
- Vi kan finne kanten av regionene i bildet ved å subtrahere et erodert bilde fra originalbildet: **g = f - (f $\ominus$ S)**
- Strukturelementet avgjør 4- eller 8-naboskap:

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

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-naboskap

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:

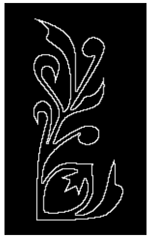
```
0 0 1 1 1 1 0 1 1 1 0
0 1 1 0 0 1 1 1 0 1 0
0 1 0 1 1 1 0 0 0 1 0
1 1 0 1 0 1 0 0 0 1 1
0 1 0 1 1 1 0 0 0 1 1
0 1 0 0 0 0 0 0 0 1 0
0 1 1 1 1 1 0 1 1 1 0
0 0 0 0 0 1 1 1 0 0 0
```

Sammenhengende kanter ved bruk av 4-naboskap

Eksempel – Kantdeteksjon ved erosjon / omriss

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

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



Et bilde **erodert med => differanse**

Treffer strukturelementet det binære bildet?

- Anta at vi igjen flytter strukturelementet rundt over et binært bilde.
- Strukturelementet **treffer** i posisjonen (x,y) i bildet hvis et piksel ≠ 0 i struktur-elementet svarer til en pikselverdi ≠ 0 i bildet.
- Her reflekterer vi (roterer 180°) strukturelementet før vi flytter det rundt.
- Dette er en slags binær (logisk) konvolusjon.
- Merk at hovedforskjellen er **treffer**, **ikke passer**.

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 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
```

```
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 er:
- pikselverdier som faller under strukturelement-verdier som er lik 0 er irrelevante, og
- piksler utenfor bildet antas å være 0.

Dilasjon (dilatasjon)

- Roter S og plasser den slik at origo ligger i (x,y), og bruk regelen:

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

- 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 overlapper med A i minst ett piksel 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
```

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

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


Effekter av dilasjon

- Dilasjon utvider objekter.
- Dette gjelder både indre og ytre omriss til objektet.
- Dilasjon fyller i hull i objektet.
- Dilasjon glatter ut innbuktninger i objektets omriss.
- Resultatet er avhengig av strukturelementets form.
- Større strukturelement gir mer effekt av dilasjon.

```
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

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

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

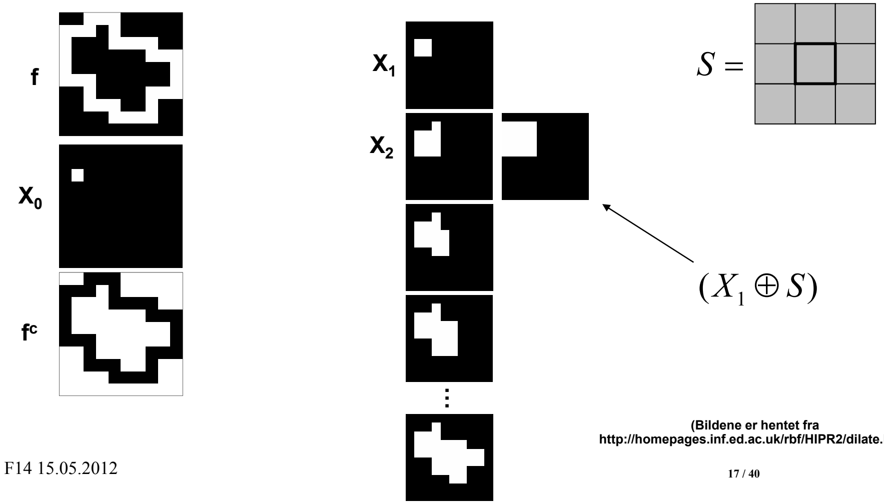
gir **gir**

```
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
```

```
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 0 1 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 - Region-fylling med dilasjon

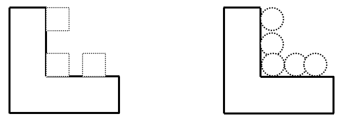
- La X_0 inneholde et punkt i regionen som skal fylles.
- Deretter iterer over følgende: $X_k = (X_{k-1} \oplus S) \cap f^c$



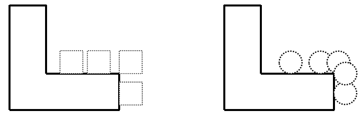
(Bildene er hentet fra http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm)

Effekter av kvadratiske og runde strukturelementer

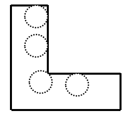
- Ved dilatering av konkave rettvinklede hjørner vil kvadratiske og runde strukturelementer gi samme effekt.



- Ved dilatering av konvekse rettvinklede hjørner vil runde strukturelementer gi avrundede hjørner.



- Runde strukturelementer gir avrundede hjørner ved erosjon av rettvinklede konkave hjørner.



Dualitet – I

- Dilasjon og erosjon er *duale* med hensyn til komplementering og reflektering (180° rotering), dvs. 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.
- => Både dilasjon og erosjon kan utføres av samme prosedyre, hvis 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 – II

- Betrakt et binært bilde som en samling av sammenhengende regioner av piksler med verdi 1 på en bakgrunn av piksler med verdi 0.
- Erosjon er å finne de posisjonene der strukturelementet passer inni regionene.
- Dilasjon er å finne de posisjonene der (det 180° roterte) strukturelementet passer inni bakgrunnen, og så komplementere resultatet.
- Dermed er det lett å se at erosjon med et sirkelformet element runder av konkave objekt-hjørner, mens dilasjon med samme element runder av konvekse objekt-hjørner.
  - … fordi konvekse objekt-hjørner har konkave bakgrunns-hjørner.

Dilasjon: Andre egenskaper

- **Dilasjon er kommutativ.**

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

  - Det er en konvensjon i faget at første operand er bildet og andre er strukturelementet (som er mindre), men dette har altså ingen betydning.

- **Dilasjon er assosiativ.**

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

  - Hvis strukturelementet S kan dekomponeres, dvs. at S er S1 dilatert med S2, kan vi spare en del regnetid, spesielt hvis S1 og S2 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 **<u>IKKE</u>** kommutativ:

$$f \ \theta \ S \neq S \ \theta \ f$$

- Suksessiv erosjon (eller dilasjon) av bildet f med A og så med B er ekvivalent med erosjon (dilasjon) av bildet f med A **dilatert** med B:

$$(f \ \theta \ \mathrm{A}) \ \theta \ \mathrm{B} = f \ \theta \ (\mathrm{A} \oplus \mathrm{B})$$

  - Passer det med denne tidligere påstanden? :
    «Hvis s_2 er formlik s_1, men dobbelt så stort, så er f ɵ s_2 ≈ (f ɵ s_1) ɵ s_1»

Åpning

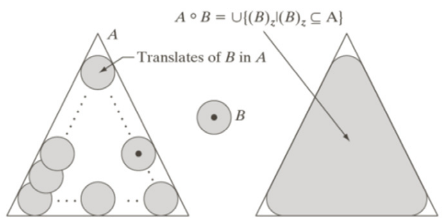
- En erosjon av et bilde fjerner alle strukturer som ikke kan inneholde strukturelementet, og "krymper" alle andre strukturer.
- Hvis vi dilaterer resultatet av en erosjon med samme strukturelement, vil de strukturene som "overlevde" erosjonen bli omtrentlig gjenskapt.
- Dette er en **morfologisk åpning**.

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

- Navnet skyldes 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 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 DIP)

- 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 en 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

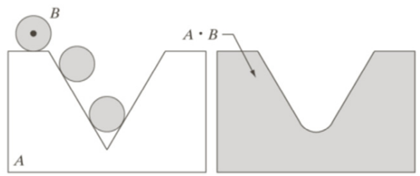
- En dilasjon av et bilde utvider objektet, fyller i hull og innbuktninger i omrisset.
- Hvis vi eroderer resultatet av en dilasjon med samme strukturelement, vil objektene stort sett beholde sin opprinnelige størrelse og form, men hull&innbuktninger som ble fylt igjen ved dilasjonen vil ikke gjenoppstå.
- Dette er en **morfologisk lukking**.

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

- Navnet skyldes 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 lykke 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 en lukking avsluttes med en erosjon).

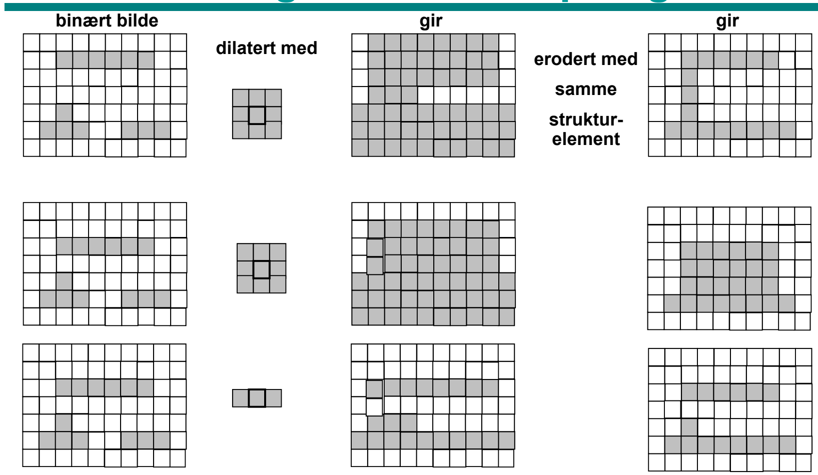


(Deler av figur 9.9 i DIP)

Også lukking er idempotent:

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

Lukking lukker små åpninger



- Strukturelementets form, og objektenes form og avstand er avgjørende for resultatet.

Dualitet mellom åpning og lukking

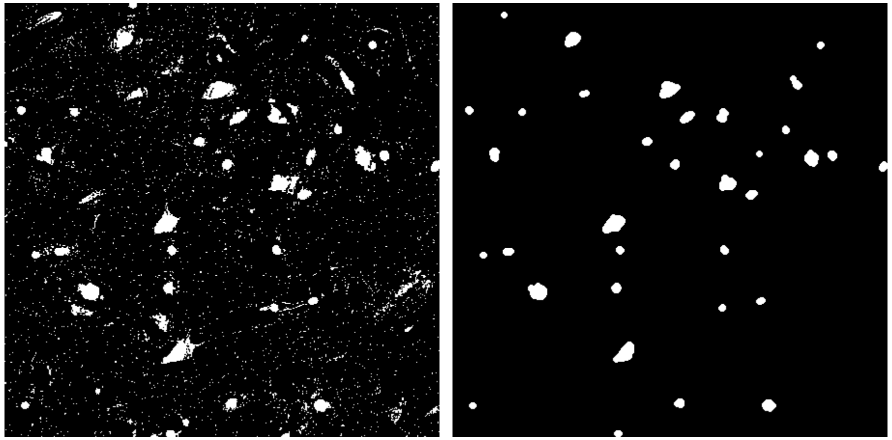
- Lukking er en dual operasjon til åpning med hensyn til komplementering og reflektering (180° rotering), og omvendt:

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

- **Lukking** av et binært bilde med et strukturelement kan gjøres ved å komplementere bildet, åpne det med det speilvendte 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 (180° rotere) og komplementere et binært bilde.
- Lukking er en *ekstensiv* transformasjon (piksler legges til).
- Åpning er en *antiekstensiv* transformasjon (piksler fjernes).

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

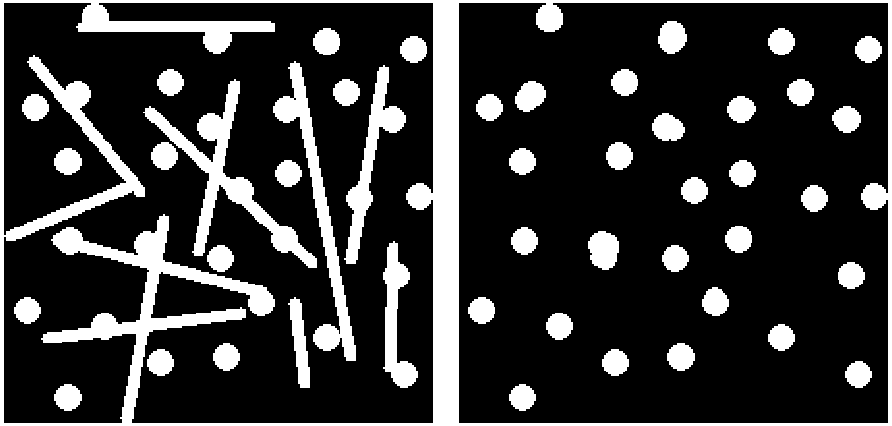
Eksempel - Åpning som støyfjerner



Åpning med et (7x7) sirkulært strukturelement

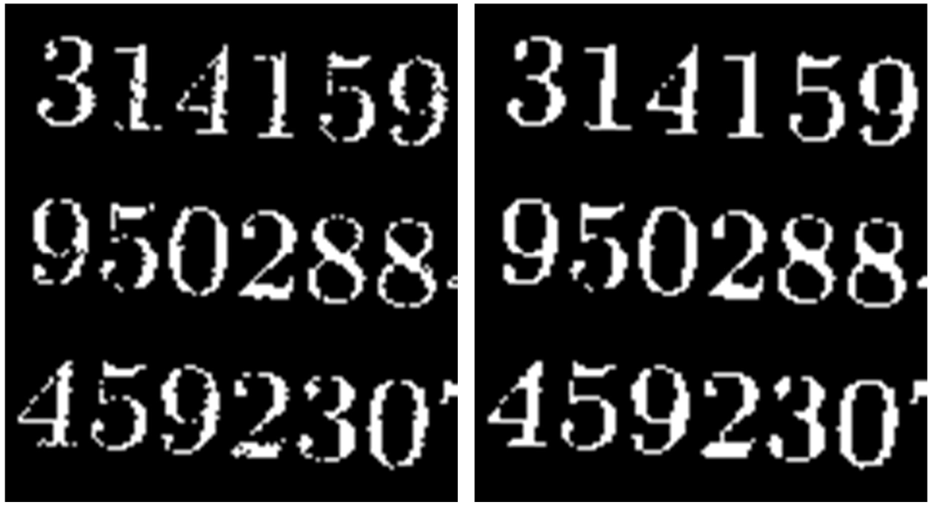
(Bildene er hentet fra http://homepages.inf.ed.ac.uk/rbf/HIPR2/)

Eksempel – Formseparering ved åpning



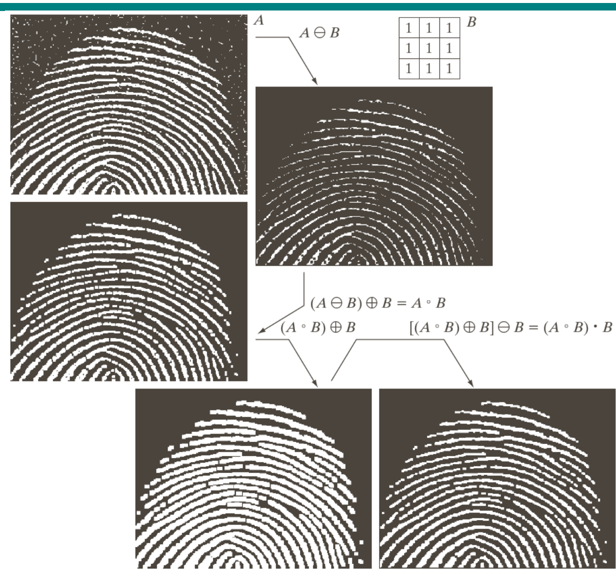
Åpning med et sirkulært strukturelement

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 (kursorisk)

- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G
  og et strukturelement B som definerer naboskapet
- *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.

Åpning ved rekonstruksjon

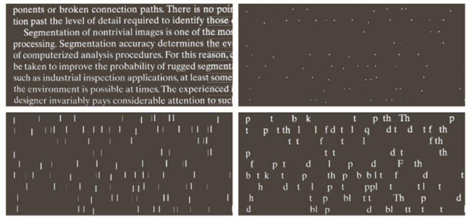
- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G
  og et strukturelement B som definerer naboskapet
- *Morfologisk rekonstruksjon ved dilasjon*:

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

  der $F^0 = F$.
- Med ord: Iterativt dilater F med B,
  og begrens resultatet med G,
  inntil ingen endring.

a b
c d

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 parameter i tillegg til f og S)
  - Utfør morfologisk rekonstruksjon ved dilasjon der G = f.
- Tilsvarer vanlig åpning, bare at de gjenværende objektene blir *nøyaktig bevart*.

Helautomatisk hullfylling

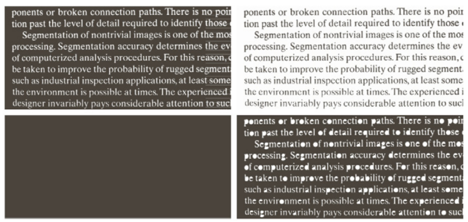
- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G
  og et strukturelement B som definerer naboskapet
- *Morfologisk rekonstruksjon ved dilasjon*:

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

  der $F^0 = F$.
- Med ord: Iterativt dilater F med B,
  og begrens resultatet med G,
  inntil ingen endring.

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

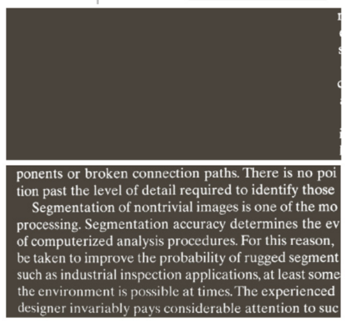
- Tidligere: Bilde f og strukturelement S
- Nå: Markørbilde F (startpunktene), maskebilde G
  og et strukturelement B som definerer naboskapet
- *Morfologisk rekonstruksjon ved dilasjon*:

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

  der $F^0 = F$.
- Med ord: Iterativt dilater F med B,
  og begrens resultatet med G,
  inntil ingen endring.

a b

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).



- Kantrydding (av bildet f) (er også helautomatisk):
  - La markørbildet F være lik f langs bildekanten, og ellers 0;

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

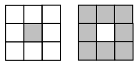
  - Utfør morfologisk rekonstruksjon ved dilasjon der G = f og subtraher resultatet fra f.

“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 da:

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

- Et objektpiksel bevares kun hvis:
  - S_1 passer innenfor objektet og
  - S_2 passer utenfor objektet.
- Kan brukes til å finne/behandle et bestemt mønster i et bilde, f.eks. til å:
  - Finne bestemte strukturer.
  - Fjerne enkeltpiksler.
  - Benyttet i “tynning” (om to slides).



“Hit-or-miss” – eksempel

```
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
```

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 mellom de to delresultatene

Morfologisk tynning

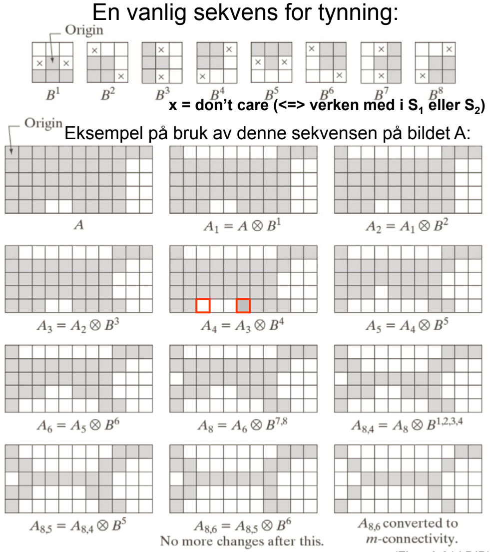
- Morfologisk tynning:

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

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

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

En vanlig sekvens for tynning:



(Figur 9.21 i DIP)

□ og ■ markerer to korreksjoner

Oppsummering

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