INF 2310
Morfologi

- ❑ **Morfologiske operasjoner på binære bilder:**
  1. Basis-begreper
  2. Fundamentale operasjoner på binære bilder
  3. Sammensatte operasjoner
  4. Eksempler på anvendelser flettet inn
- ❑ GW, Kapittel 9.1-9.4 (mer i INF 4300)
- ❑ Merk: litt forenklet notasjon i forhold til boka.

```
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
```

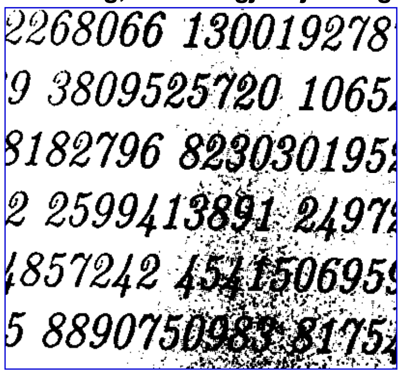
```
0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 1 0 0
0 0 1 0 0 0 1 1 0 0 0
0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 1 1 0 1 0 0 0
0 0 0 1 1 0 0 0 1 0 0
0 0 1 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0
```


Motivasjon

- ❑ Morfologisk bildebehandling dreier seg om en rekke ikke-lineære operatorer – diskrete varianter av matematisk morfologi.
- ❑ De fleste operatorene er beregnet på binære bilder
  - Interessert i objekters form (shape)
  - En morfologisk operasjon på et binært bilde gir et nytt binært bilde.
    - Ut-bildet har verdien 1 på et gitt sted bare hvis den utførte testen var positiv.
- ❑ Mange av dem brukes til å fjerne uønskede effekter etter segmentering
  - Fjerning av små objekter (støy-objekter)
  - Glatting av omrisset til større objekter
  - Fjerning av hull i objekter
  - Lenke sammen objekter
- ❑ Noen operatorer er nyttige til analyse og beskrivelse av objekter
  - Finne omriss av objekter
  - Finne fordelingen av størrelsen på objektene
  - Finne fordelingen av størrelsen på mellomrommene mellom objektene
  - Finne mønstre i bilder
- ❑ Operasjonene er ofte enkle og kan utføres svært raskt

Eksempel

- ❑ Problem: gjenkjenn alle tall i bildet automatisk.
- ❑ Her vil morfologiske operatorer være velegnet til å prosessere formen på de segmenterte tallene etter terskling, men før gjenkjenning.



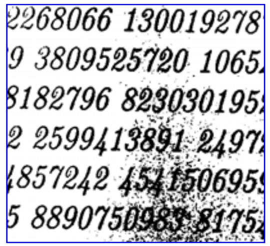
En del små støy-objekter.

Noen symboler er fragmentert i flere deler.

Kanten rundt noen symboler kan være rufsete.

Gjenkjenning av objekter – intro (mer i INF 4300)

OCR-gjennkjenning:



- ❑ Segmenter ved for eksempel terskling
- ❑ Fjern støy/lukk hull i segmentert bilde
- ❑ Finn tallene (sammenhengende objekter)
- ❑ Trekk ut egenskaper som beskriver objektenes form
- ❑ Tren på mange eksempler av hvert objekt.
- ❑ Gjenkjenn nye objekter ved å sammenligne med tidligere objekter i treningsdatabasen.

Litt sett-teori

- **La A være et sett i $\mathbb{Z}^2$. Dersom punktet a=(a_1,a_2) er et element i A skriver vi:** $a \in A$
- **Dersom a ikke er inneholdt i A skriver vi:** $a \notin A$
- **Settet uten noen elementer kalles det tomme settet og betegnes** $\varnothing$
- **Elementene som ikke er i A kalles komplementet til A, og betegnes A^c**
- **Dersom alle elementene i A er inneholdt i B sies A å være et subsett av B, dette skrives:** $A \subseteq B$
- **Med unionen av to sett A og B mener vi settet som består av alle elementer fra enten A eller B, dette betegnes:** $A \cup B$
- **Med snittet av A og B mener vi settet som består av elle elementer som finnes i både A og B, dette betegnes:** $A \cap B$

Litt sett-teori i praksis på binære bilder

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

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

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

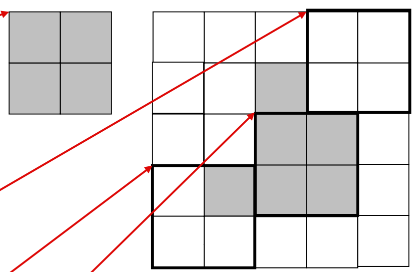
$$h = f \cap g \equiv h(x,y) = \begin{cases} 1 \text{ hvis } f(x,y) = 1 \text{ og } g(x,y) = 1 \\ 0 \text{ ellers} \end{cases}$$

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

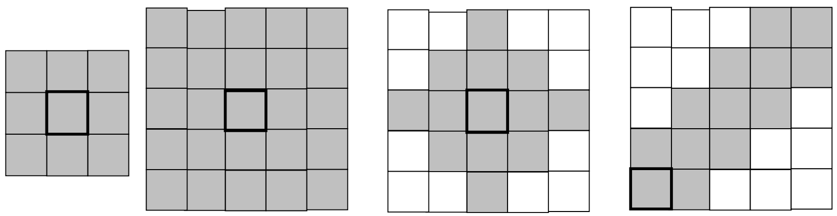
$$h = f \cup g \equiv h(x,y) = \begin{cases} 1 \text{ hvis } f(x,y)=1 \text{ eller } g(x,y)=1 \\ 0 \text{ ellers} \end{cases}$$

Tre sentrale begrep

- **Et strukturelement for et binært bilde er en liten matrise av piksler**
- **Når vi fører det binære strukturelementet over det binære bildet vil vi finne**
  - **Posisjoner der elementet ikke overlapper objektet**
  - **Posisjoner der elementet delvis overlapper objektet, dvs at elementet treffer objektet**
  - **Posisjoner der elementet ligger inni objektet, dvs at elementet passer i objektet**



Strukturelementenes form og origo



- **Strukturelementer kan ha ulik form og størrelse**
- **Et strukturelement har 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 elementet svarer til en pikselverdi ≠ 0 i bildet.
- Pikselverdier som faller under strukturelement-verdier som er lik 0 er irrelevante!

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 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 et sett X (det binære bildet) med strukturelementet S er definert som posisjonen til alle piksler x slik at S er inkludert i X når origo i S plasseres i x:

$$\varepsilon(X \mid S) = \{x \mid S_x \subseteq X\}$$

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

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 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
```

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

gir

```
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
```


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.

```
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 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 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
```


Iterativ erosjon

- Større strukturelement gir mer erosjon.
- Resultatet av erosjon med et stort strukturelement er 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
1 1 1
1 1 1
```

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 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
```

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

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 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
```

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, dvs. $g = f - (f \ominus s)$
- Strukturelementet avgjør 4- eller 8-naboskap

erodert med — gir => differanse

```
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
```

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

```
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 omriss som benytter 8-naboskap

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

```
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
```

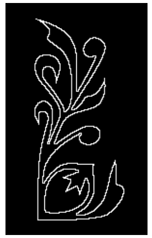
```
0 0 1 1 1 1 0 1 1 1 0
0 1 1 0 0 1 1 1 0 1 0
0 1 0 1 1 1 0 0 0 1 0
1 1 0 1 0 1 0 0 0 1 1
0 1 0 1 1 1 0 0 0 1 1
0 1 0 0 0 0 0 0 0 1 0
0 1 1 1 1 1 0 1 1 1 0
0 0 0 0 0 1 1 1 0 0 0
```

Sammenhengende ved 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
```



Treffer strukturelementet det binære bildet?

- Anta igjen at vi flytter strukturelementet rundt over et binært bilde.
- Strukturelementet treffer posisjonen (x,y) i bildet hvis et piksel ≠ 0 i elementet svarer til en pikselverdi ≠ 0 i bildet.
- Pikselverdier som faller under strukturelement-verdier som er lik 0 er irrelevante!
- Dette er en slags binær (logisk) 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 0 0 0 0 0 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 1 1 1 1 1 1 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 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 0 0 0 0 0 0 0 0 0 0
```

To forskjellige resultater

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


Dilasjon

- Plasser S 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 et sett X med strukturelementet S er definert som posisjonen til alle piksler x slik at S overlapper med X i minst ett piksel når origo i S plasseres i x:

$$\delta(X \mid S) = \{x \mid S_x \cap X \neq \emptyset\}$$

```
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

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

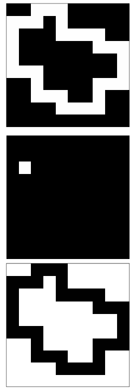

Anvendelse - Region-fylling med dilasjon

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

f

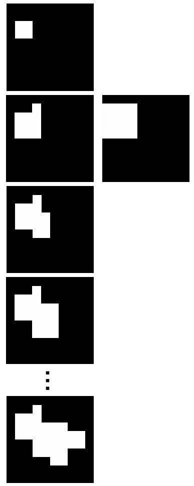
X_0

f^c



X_1

X_2

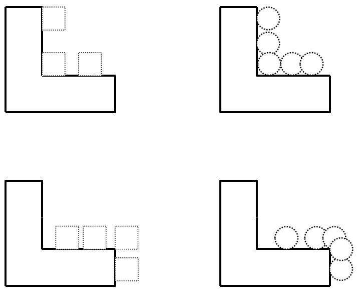


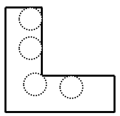
$(X_1 \oplus S)$

(Bilder 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

- Dilasjon og erosjon er duale,dvs operasjonene har motsatt effekt:

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

- Der f^C er komplementet til f, dvs at vi erstatter 1 med 0 og 0 med 1, og Ŝ er strukturelementet rotert 180°.
- For å dilatere f med <u>symmetrisk</u> S kan vi erodere komplementet til f med S, og ta komplementet av resultatet.
- Både dilasjon og erosjon kan utføres av samme prosedyre, hvis vi kan rotere et strukturelement og finne komplementet til 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
```

dilatert 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
```

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
```

erodert med

```
0 1 0
1 1 1
0 1 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 transponerte) 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.

Dilasjon: Andre egenskaper

- Dilasjon er kommutativ. Det er en konvensjon i faget å la første operand være bildet og andre være strukturelementet (som er mindre), men

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

- Dilasjon er assosiativ.

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

- Dette er veldig nyttig hvis vi ser at strukturelementet S kan dekomponeres, dvs at S er S1 dilatert med S2. Da kan vi spare en del regnetid. 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.

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

- Erosjon av snittet av to mengder er gitt ved snittet av de to erosjonene:

$$(X \cap Y) \ \theta \ S = (X \ \theta \ S) \cap (Y \ \theta \ S)$$

- 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

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

Åpning

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

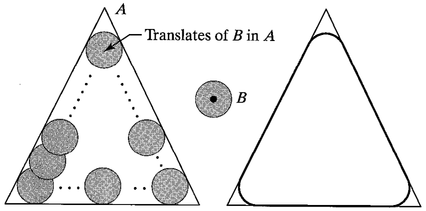
$$f \circ S = \left(f \ \theta \ \mathrm{S}\right) \oplus S$$

- Navnet skyldes at operasjonen kan skape en åpning mellom to strukturer som bare henger sammen i en tynn bro, uten å krympe de to strukturene, slik som en erosjon alene ville gjort.

Åpning

- Vi kan visualisere morfologisk åpning ved å tenke oss at strukturelementet føres langs kanten av objektet.
- Først langs innsiden av objektet. Dermed krymper objektet. Konkave hjørner kan bli avrundet hvis strukturelementet er rundt.
- Deretter fører vi strukturelementet rundt utsiden av resultatobjektet ovenfor. Objektet vokser igjen, men små utstikkere som forsvant i forrige steg blir ikke gjenopprettet.
- For runde strukturelementer: De konvekse hjørnene som beholdt formen ved erosjon blir nå avrundet ved dilasjon, mens de konkave som ble avrundet ved erosjon, blir til rette hjørner igjen ved dilasjon.
- Idempotens: Gjentatte anvendelser av åpning med samme strukturelement gir ingen endringer i resultatet.

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



(Figur fra Gonzales&Woods)

Lukking

- En dilasjon av et bilde utvider objektet, fyller i hull og innbuktninger i omrisset.
- Hvis vi eroderer resultatet av en dilasjon med det roterte strukturelementet, vil objektene stort sett beholde sin størrelse og form, men innbuktninger som er fylt igjen ved dilasjonen vil ikke gjenoppstå. Objekter som er smeltet sammen ved dilasjonen vil ikke bli adskilt igjen.
- Dette er en morfologisk lukking.

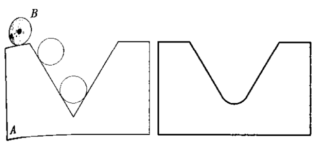
$$f \bullet S = \left(f \oplus \hat{S}\right) \theta \hat{S}$$

- Navnet skyldes at operasjonen kan lukke en åpning mellom to strukturer som bare er adskilt med et lite gap, uten at de to regionene vokser, slik som en dilasjon ville ha gjort.

Lukking II

- Vi kan visualisere morfologisk lukking ved å tenke oss at strukturelementet føres langs kanten av objektet.
- Først langs utsiden av objektet. Dermed vokser objektet. Konvekse hjørner kan bli avrundet.
- Deretter fører vi strukturelementet rundt innsiden av resultatobjektet ovenfor. Objektet krymper igjen, men små "viker" som forsvant i forrige steg blir ikke gjenopprettet.
- Idempotens: Gjentatte anvendelser av lukking med samme strukturelement gir ingen endringer i resultatet.

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

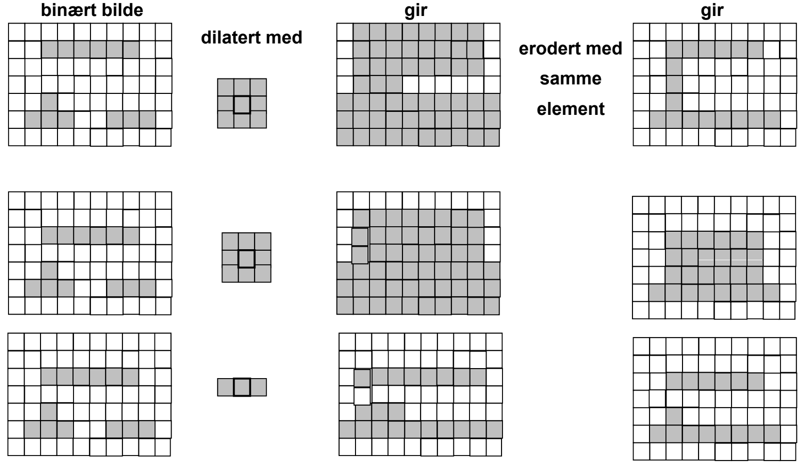


Lukking –"connecting pixels"

binært bilde

gir

gir



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

Dualitet mellom åpning og lukking

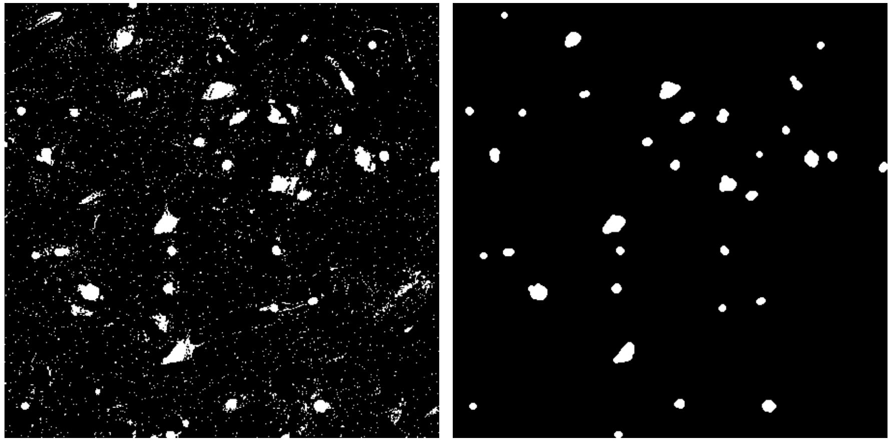
- **Lukking er en dual operasjon til åpning, og omvendt.**

$$f \bullet S = (f^C \circ S)^C \qquad f \circ S = (f^C \bullet S)^C$$

- **Lukking av et binært bilde med et strukturelement kan gjøres ved å ta komplementet til bildet, åpne det med strukturelementet, og ta komplementet av resultatet.**
- **Og omvendt: Åpning av et binært bilde med strukturelementet kan gjøres ved å ta komplementet til bildet, lukke det med strukturelementet, og ta komplementet av resultatet.**
- **Vi kan altså utføre begge operasjonene med kode bare for den ene, hvis vi har kode for å produsere komplementet til et binært bilde.**
- **Lukking er en *ekstensiv* transformasjon (piksler legges til).**
- **Åpning er en *antiekstensiv* transformasjon (piksler fjernes).**

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

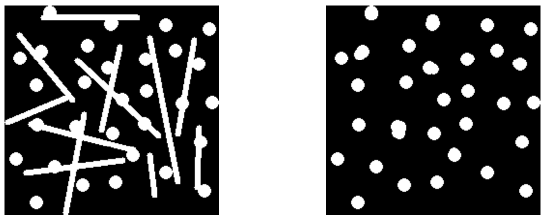
Eksempel - Åpning som støyfjerner



Åpning med 7x7 sirkulært strukturelement

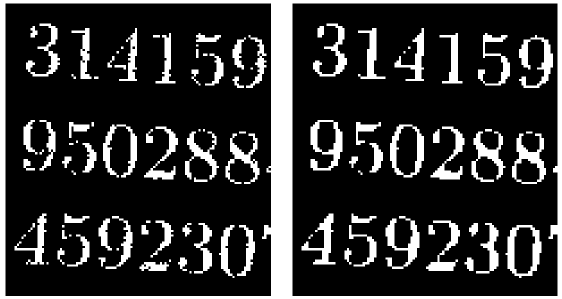
(Bilder hentet fra http://homepages.inf.ed.ac.uk/rbf/HIPR2/)

Eksempel – Formseparering ved åpning



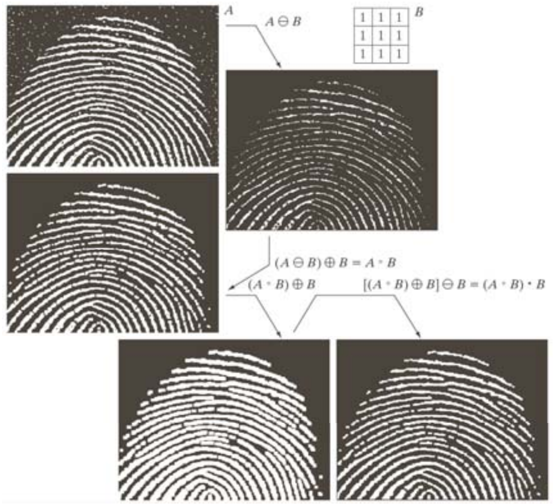
Åpning med sirkulært strukturelement

Eksempel – Filtrering ved lukking



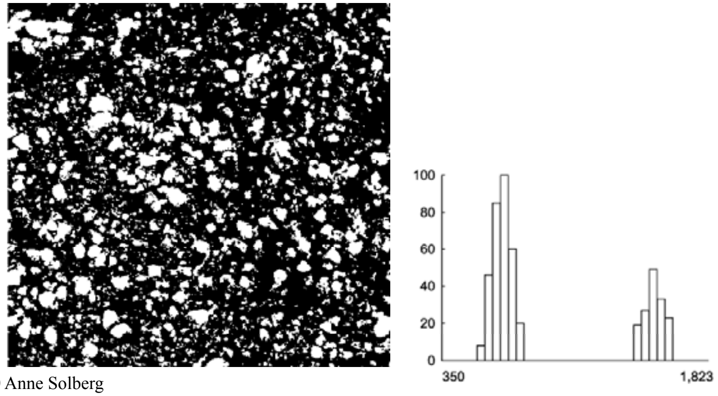
Lukking med 3x3 strukturelement

Eksempel – Filtrering ved åpning og lukking



Eksempel – Granulometri ved åpning

- Gjentatte åpninger av originalbildet med gradvis større strukturelementer
- Benytter at effekten av åpning er størst i regioner hvor elementenes størrelser er tilnærmet lik strukturelementets



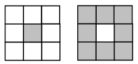
“Hit and miss”-transformasjonen

- Dette er en transformasjon som brukes til å finne et bestemt mønster i et bilde.
- Strukturelementet er et par $[s_1,s_2]$ som er disjunkte mengder.

$$f(*)\{S_1,S_2\} = (f\,\theta\,S_1) \cap (f^C\theta\,S_2)$$

- Et objektpiksel bevares kun hvis s_1 passer innenfor og s_2 passer utenfor objektet.

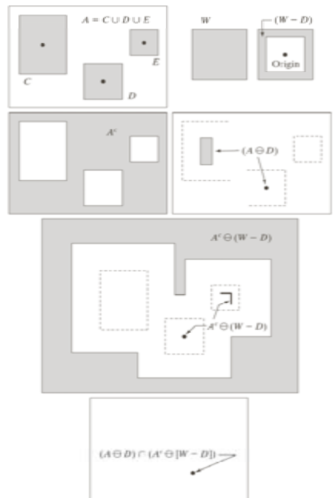
- Eksempel på bruk:
  - Finne bestemte strukturer.
  - Fjerne enkeltpiksler.
  - Benyttet i ”tynning”



“Hit or miss”- transformen

- Finn et eksakt mønster – “template matching”
- Mål: finn eksakt formen gitt ved objekt D.
- D kan passe inni mange objekter, så vi trenger å se på lokal bakgrunn W-D.
- Beregn først erosjonen av A med D, AθD (alle piksler der D passer inni A)
- Sjekk også om bakgrunnen passer: beregn A^C, komplementet til A. Kooridinatene det D passer til snittet mellom AθD og erosjonen av A^C med W-D, A^C θ(W-D).
- Hit-or-miss uttrykkes $A \circledast D$:

$$(A\,\theta D) \cap [A_c\,\theta(W-D)]$$



”Hit and 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 B1

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 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 B1

```
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 B2

```
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 B2

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 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-and-miss resultatet
Logisk AND mellom de to delresultatene

Oppsummering

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