

Matematická morfologie

Počítačová grafika

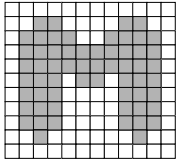
Mgr. Markéta Trnečková, Ph.D.

Proč matematická morfologie?

- v běžném zpracování obrazu často operujeme intenzitami (jasové operace, filtrování – konvoluce, . . .)
- ale co **tvar** objektu?
- mnoho úloh závisí právě na **geometrii**:
  - detekce a analýza okrajů, segmentace objektů
  - odstranění šumu při zachování tvaru
  - vyplnění drobných děr, čištění struktur
  - hledání konvexního obalu, kostry oblasti, její okraj
- **Morfologie** poskytuje **nástroje založené na teorii množin**, které pracují přímo s tvary (množinami bodů)
- **Aplikace:** průmyslová inspekce, biologie, zpracování signálů, lékařská zobrazovací technika a další oblasti

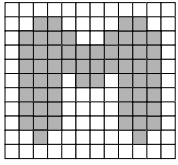
Obraz jako množina

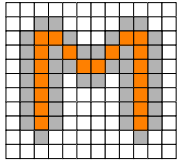
- binární obrázky – pixely jsou prvky $\mathbb{Z}^2$ (souřadnice x a y)
- šedotónové obrázky – pixely jsou prvky $\mathbb{Z}^3$ (prostorové souřadnice, intenzita)
- **dva typy množin** (v matematické morfologii):
  - objekty – pixely popředí a pozadí

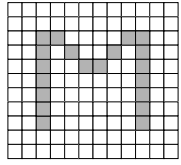


  - strukturní element (SE) – pixely popředí, pozadí a „don't care“









Strukturní element (SE)

- navíc zavádíme počátek (jádro) SE, protože se SE pracujeme obdobně jako s filtry



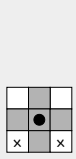
Základní operace se SE

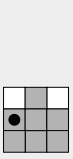
- Máme množinu B (body (x, y))
- **reflexe** $\hat{B}$
  $\hat{B} = \{(-x, -y) | (x, y) \in B\}$
  Otočení SE o 180° kolem počátku.
- **translace** $(B)_z$ o vektor $z = (x_z, y_z)$
  $(B)_z = \{(x + x_z, y + y_z) | (x, y) \in B\}$
  Také vztahujeme k počátku SE.

Strukturní element (SE)

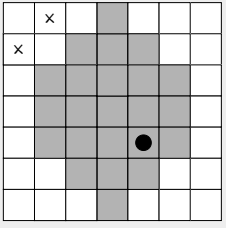
Příklad

K následujícím SE určete jejich reflexi.







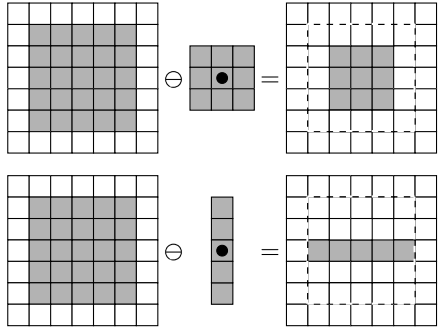


Základní morfologické operace

- SE posouváme po obrázku obdobně, jako v konvoluci
- provádíme množinové operace s pixely popředí SE (B) a pixely popředí obrázku (A)
- základní operace
  - **eroze**
  - **dilatace**
- jejich kombinace tvoří složitější operace
- výsledky operací vždy záleží na tvaru a velikosti SE

Eroze

- zmenšuje objekty v obraze, „ožírá“ jejich okraje
- $A \ominus B = \{z \mid (B)_z \subseteq A\}$
- bod z patří do výsledku, pokud je **celý strukturální element** B při posunutí do bodu z **obsažen uvnitř objektu** A
- **Výsledek:**
  - zmenšuje objekty
  - odstraňuje malé části a zužuje tenké spoje

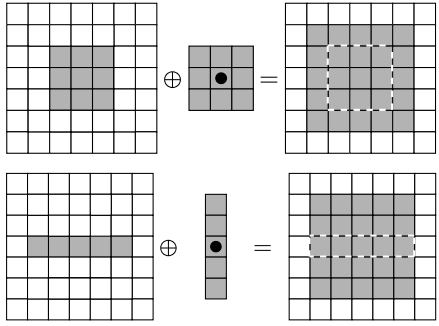


Eroze



Dilatace

- rozšiřuje objekty v obraze, „přidává“ pixely k jejich hranám
- $A \oplus B = \{z \in E \mid (\hat{B})_z \cap A \neq \emptyset\}$
- bod z patří do výsledku, pokud se **překrývá alespoň jeden bod reflektovaného strukturálního elementu** $\hat{B}$ posunutého do bodu z s objektem A
  Často se volí symetrický SE. V takovém případě reflexe není nutná.
- **Výsledek:**
  - rozšiřuje objekty
  - vyplňuje malé díry a spojuje blízké objekty



Dilatace







Dualita eroze a dilatace

- **Eroze** a **dilatace** jsou **duální operace** vůči doplňku množiny.
- Platí:

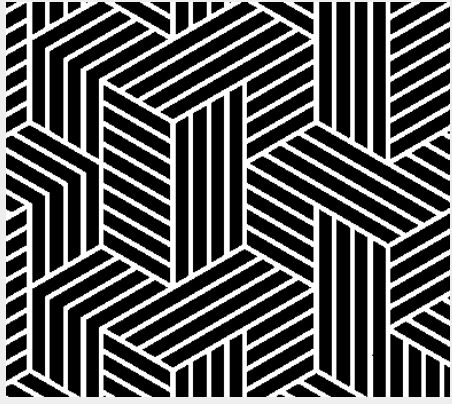
$$(A \ominus B)^c = A^c \oplus \hat{B}, \qquad (A \oplus B)^c = A^c \ominus \hat{B}$$

- **Význam:**
  - eroze objektu A odpovídá dilataci jeho doplňku A^c reflexí B
  - dilatace objektu A odpovídá erozi jeho doplňku
- **Důsledek:**
  - stačí implementovat jen jednu z operací — druhou lze odvodit dualitou
  - zejména se hodí, když je SE symetrický $B = \hat{B}$
  - dualita se zachovává i pro složitější morfologické operace (např. otevření a uzavření)
- Důkaz je možné najít např. v knize Digital image processing (Gonzales a Woods)

Eroze a dilatace

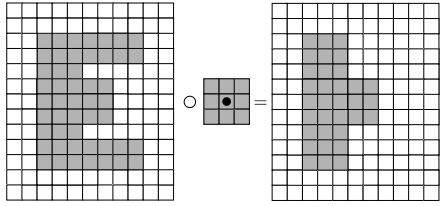
Příklad

Jak bychom mohli v následujícím obrázku odstranit pouze svislé čáry?



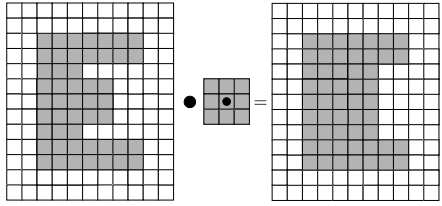
Otevření

- odstranění malých objektů nebo výstupků, zachování tvaru větších oblastí
- $A \circ B = (A \ominus B) \oplus B$
- nejprve erodujeme A – odstraníme malé detaily, zmenšíme objekty,
- poté dilatujeme – obnovíme přibližný tvar původních objektů bez odstraněných částí
- **Výsledek:**
  - odstraňuje malé izolované objekty
  - „vyhlazuje“ obrys objektů
  - zachovává topologii větších tvarů



Uzavření

- vyplňuje malé mezery a otvory, spojuje blízké objekty, zachovává větší tvary
- $A \bullet B = (A \oplus B) \ominus B$
- nejprve dilatujeme A – rozšíříme objekty a vyplníme malé mezery
- poté erodujeme – zmenšíme zpět na původní velikost, ale mezery zůstanou zaplněny
- **Výsledek:**
  - spojuje blízké objekty
  - vyhlazuje obrysy objektů z vnitřní strany
  - zachovává celkový tvar a topologii větších struktur



Otevření a uzavření



Původní obraz



Otevření



Uzavření

Otevření a uzavření – Dualita

Příklad

Už dříve bylo řečeno, že i pro uzavření a otevření platí dualita. Jak by byla definovaná?

Otevření a uzavření – Dualita

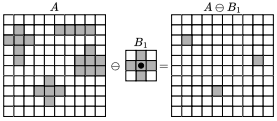
Příklad

Už dříve bylo řečeno, že i pro uzavření a otevření platí dualita. Jak by byla definovaná?

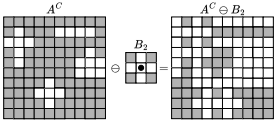
$$(A \circ B)^c = A^c \bullet \hat{B}, \qquad (A \bullet B)^c = A^c \circ \hat{B}$$

Hit-or-Miss transformace

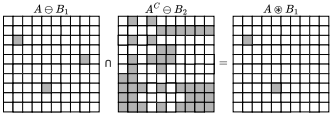
- hledá v obraze přesné tvary nebo vzory (šablony)
- $A \circledast B_{1,2} = (A \ominus B_1) \cap (A^c \ominus B_2)$
- $(A \ominus B_1)$, identifikuje místa, kde shluky pixelů odpovídají prvnímu SE



- $(A^c \ominus B_2)$ identifikuje, pixely, u kterých pixely dané druhým SE tvoří pozadí



- výsledek

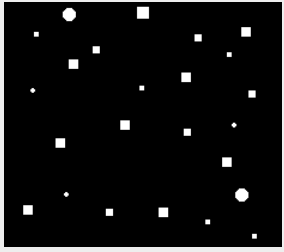


Hit-or-Miss transformace

- SE neobsahují pixely popředí na stejných místech (mnohdy je $B_2 = B_1^C$), může se stát, že na nějakém místě není ani jeden pixel popředí, v takovém případě se jedná o „don't care“ prvky
- můžeme se setkat i s definicí $A \circledast B = (A \ominus B) \cap (A^c \ominus B^c)$
  kde komplement „don't care“ prvku je opět „don't care“ prvek

Příklad

Jak bychom nalezli všechny horní levé rohy čtverců např. v následujícím obrázku?

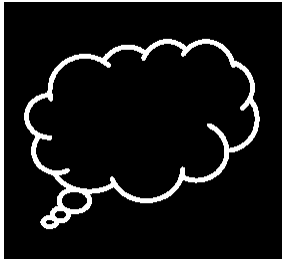


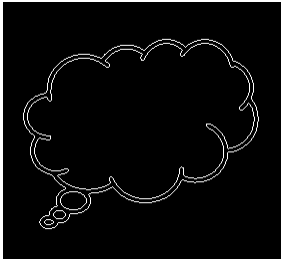
Praktické aplikace matematické morfologie

- Matematická morfologie není jen teorie – umožňuje řešit konkrétní úlohy při zpracování obrazů
- Typické problémy, které řešíme pomocí morfologických operací:
  - Zacelování děr v objektech
  - Hledání okrajů a obrysů
  - Detekce spojitých komponent (clustering pixelů)
  - Výpočet konvexního obalu objektů
  - Ztenčení a skeletonizace
  - ...
- Ukážeme si **principy algoritmů**, matematickou interpretaci a příklady na obrazech

Hledání obrysu objektu

- obrys objektu lze získat pomocí **eroze**
- obrys$(A) = A - (A \ominus B)$
- eroze odstraní okrajové pixely,
- odečtením od původního obrazu získáme právě **obrys objektu**.
- Vnější obrys lze získat pomocí dilatace obrys$(A) = (A \oplus B) - A$





Vyplňování děr v obraze

- chceme doplnit vnitřní **díry** uvnitř objektů, aniž bychom změnili jejich vnější tvar
- **Algoritmus**:
  - zvolíme bod p uvnitř díry,
  - vytvoříme množinu $X_0 = \{p\}$,
  - iterativně definujeme $X_k = (X_{k-1} \oplus B) \cap A^c$
  - až do ustálení $X_k = X_{k-1}$.
- **Výsledek** $A_{\text{vyplněné}} = A \cup X_k$



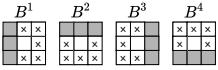
Extrakce spojitých komponent

- chceme **oddělit jednotlivé objekty** (komponenty) v binárním obraze
- **Algoritmus**:
  - zvolíme bod p ležící uvnitř objektu,
  - postupně bod rozšiřujeme pomocí souvislosti dané SE,
  - získáme celou **spojitou komponentu**, která obsahuje bod p.
- $X_k = (X_{k-1} \oplus B) \cap A$
  $X_0 = \{p\}, \quad X_k = X_{k-1} \Rightarrow$ komponenta nalezena.
- výsledkem je množina všech pixelů spojených s bodem p
- lze opakovat pro nalezení všech komponent v obraze
- automaticky – vybereme libovolný bílý pixel, který nepatří do žádné dosud nalezené komponenty



Konvexní obal objektu

- **konvexní obal** je nejmenší konvexní množina, která obsahuje všechny pixely objektu
- uvažujme následující SE



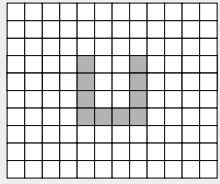
- $X_k^i = (X_{k-1}^i \circledast B^i) \cup X_{k-1}^i$
  $X_0^i = A$
- **Algoritmus**:
  - nastavíme $X_0^i = A$
  - dokud $X_k^i = (X_{k-1}^i$ provádíme iteraci, výsledek označíme D^i
  - konvexní obal $= \cup_{i=1}^4 D^i$
- **Důležitá poznámky:**
  - výsledkem není konvexní obal, jak ho známe
  - výsledek je dán volbou SE
  - pro lepší výsledek bychom museli použít i SE jiných směrů.

Konvexní obal objektu

- **Algoritmus**:
  - nastavíme $X_0^i = A$
  - dokud $X_k^i = (X_{k-1}^i$ provádíme iteraci, výsledek označíme D^i
  - konvexní obal $= \cup_{i=1}^4 D^i$

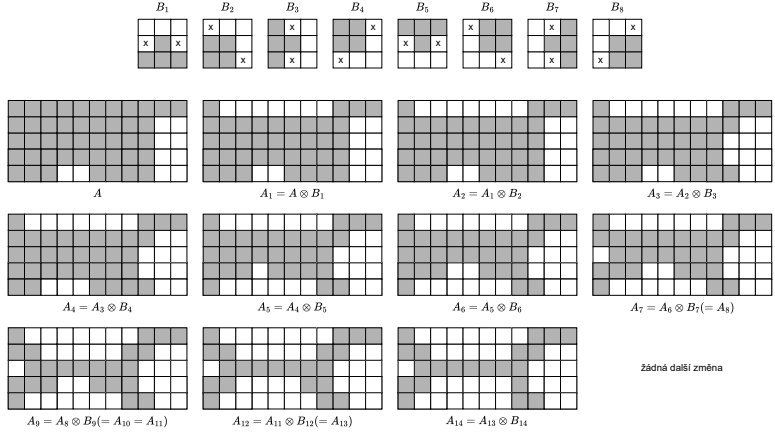
Příklad

Odsimulujte algoritmus na následujícím obrázku.



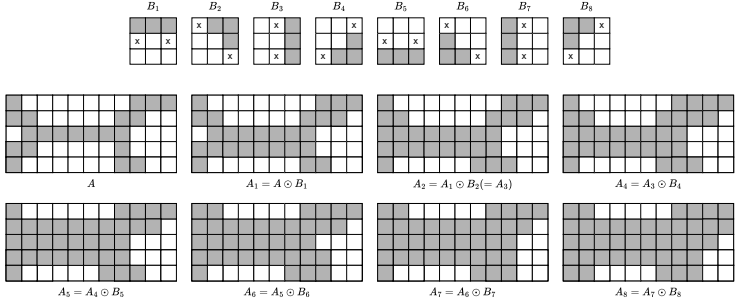
Ztenčení objektů (Thinning)

- chceme zmenšit tloušťku objektů na minimum, aniž by se změnila topologie
- máme sekvenci SE B_i
- $A \otimes B_i = ((\ldots((A \otimes B_1) \otimes B_2)\ldots) \otimes B_n)$
  kde $A \otimes B = A - (A \circledast B)$
- Ztenčování provádíme dokud se výsledek nemění (po celém průchodu přes všechny SE)



Zesílení objektů (Thickening)

- chceme zesílit nebo rozšířit objekty bez změny jejich topologie
- máme sekvenci SE B_i
- $A \odot B_i = ((\ldots((A \odot B_1) \odot B_2)\ldots) \odot B_n)$, kde $A \odot B = A \cup (A \circledast B)$
- mohou se objevovat body, které jsou od objektu odděleny (odstraňují se dodatečně)

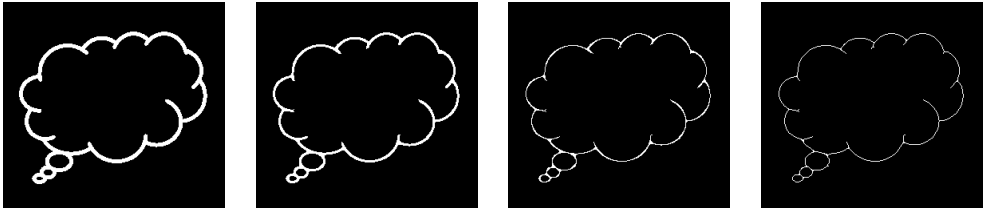


Příklad

Jak bude vypadat konečný výsledek?

Kostra objektu (Skeleton)

- skeletonizace zjednodušuje objekt na **jednopixelovou kostru**, která zachovává **topologii a tvar**
- $S(A) = \bigcup_{k=0}^{K} [(A \ominus kB) - (A \ominus kB) \bullet B]$ kde:
  - $A \ominus kB = ((\dots((A \ominus B) \ominus B)\dots) \ominus B)$, k krát
  - K je maximální počet iterací, dokud eroze nevede k prázdné množině

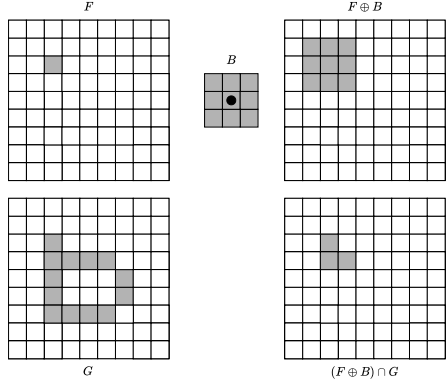


- existují i algoritmy na ořezávání krátkých větví kostry – **Pruning**

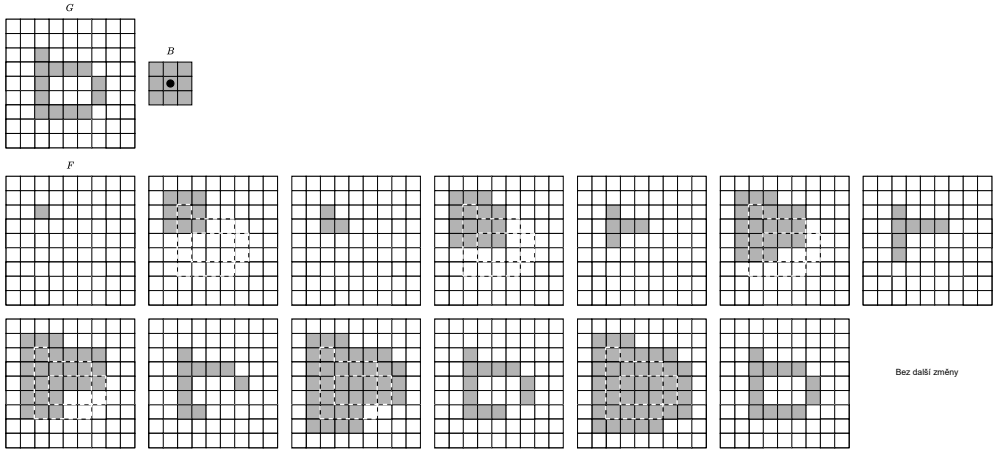
Morfologická rekonstrukce

- dosud jsme pracovali s 1 obrázkem a několika SE
- teď budeme mít 2 obrázky
  - F **marker** – obsahuje startovací body rekonstrukce
  - G **maska** – podmínky rekonstrukce
- strukturní element používáme k definici sousednosti

- **geodetická dilatace** – $D_G^{(1)}(F) = (F \oplus B) \cap G$
- $D_G^{(n)}(F) = D_G^{(1)}(D_G^{(n-1)}(F))$
- opakujeme, dokud se výsledek mění

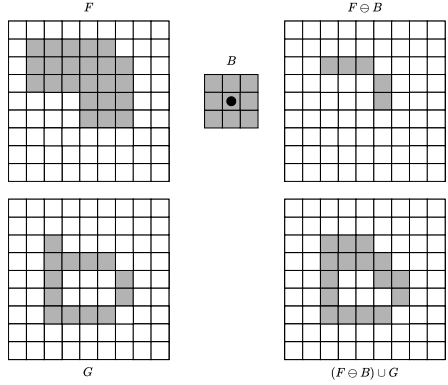


Morfologická rekonstrukce



Morfologická rekonstrukce

- **geodetická eroze** – $E_G^{(1)}(F) = (F \ominus B) \cup G$
- $E_G^{(n)}(F) = E_G^{(1)}(E_G^{(n-1)}(F))$
- opakujeme, dokud se výsledek mění (**rekonstrukce dilatací**)

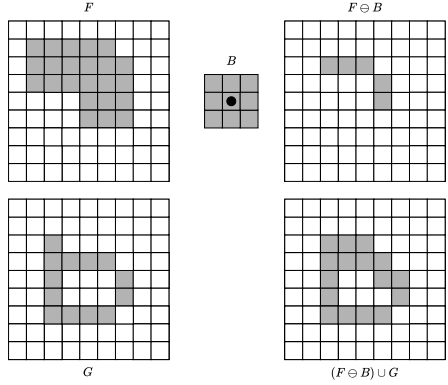


Příklad

Udělejte zbytek kroků rekonstrukce, obdobně jako na předchozím slide.

Morfologická rekonstrukce

- **geodetická eroze** – $E_G^{(1)}(F) = (F \ominus B) \cup G$
- $E_G^{(n)}(F) = E_G^{(1)}(E_G^{(n-1)}(F))$
- opakujeme, dokud se výsledek mění (**rekonstrukce erozí**)



Příklad

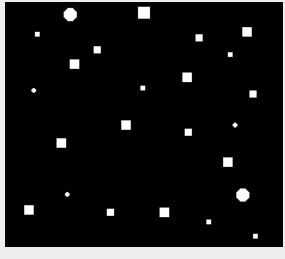
Udělejte zbytek kroků rekonstrukce, obdobně jako na předchozím slide.

Morfologická rekonstrukce – Aplikace

- existuje velké množství aplikací morfologické rekonstrukce
- liší se tím, co považujeme za masku a jak zvolíme marker

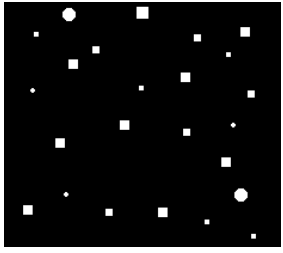
Příklad

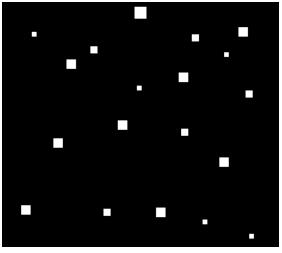
- uvažujme nalezené levé horní rohy z dřívějšího příkladu a ty použijme jako markery
- původní obrázek použijeme jako masku
- provedeme rekonstrukci dilatací
- Co dostaneme jako výsledek?





Morfologická rekonstrukce – Aplikace





Morfologická rekonstrukce – Aplikace

Otevření rekonstrukcí

- klasické otevření – odstraněny malé objekty v obraze a následnou dilatací se snažíme vrátit do původního stavu, výsledek ale závisí na tvaru SE a původního tvaru
- otevření rekonstrukcí odstraňuje malé objekty, ale zachovává přesné tvary těch větších
- F – erodovaný obrázek
- G – původní obrázek
- rekostruujeme dilatací



Původní obraz



Otevření



Rozdíl



Původní obraz



Otevření rekonstrukcí



Rozdíl

Morfologická rekonstrukce – Aplikace

Zacelení děr

- zacelení děr v objektech
- F – negativ okrajových pixelů obrázku, jinde 0
- G – negativ původního obrázku
- rekostruujeme dilatací
- výsledkem je negativ vyplněného obrázku





Morfologická rekonstrukce – Aplikace

Vyčištění okrajů

- odstanění objektů, které leží na okraji obrázku
- F – okrajové pixely obrázku, jinde 0
- G – negativ původního obrázku
- rekostruujeme dilatací
- výsledkem je to, co chceme z obrázku odstranit

