

# Ipari képfeldolgozás

## Alakzatelemzés

Megyesi Zoltán

[megyesi.zoltan@gamf.kefo.hu](mailto:megyesi.zoltan@gamf.kefo.hu)

KF GAMF Kar AAI Szakcsoport



## Alakzatok

- 2D objektumok
- 3D objektumok vetületei
- Szegmentált bináris képek
- Sík alakzatok
- Görbék, zárt kontúrok

## Feladatok

- Alakzatileírások létrehozása
- Alakzatok felbontása részekre
- Alakzat illesztés
- Alakzat felismerés
- Eltolás és elforgatás meghatározása

## Kimenet és bemenet

- Input: Egy alakzat képe
- Output: 2 típus – a módszerhez kötődik
  - Jellemző vektor (scalar transform methods)
    - Osztályozás, statisztikai elemzés
  - Kép (spatial domain methods)
    - Struktúra elemzés
    - Pl.: Medial axis, morphological processing

## Módszerek tulajdonságai – Információ veszteség

- Minden módszer alakzataból készít leírást, vagy alakzatot
- Veszteséges vagy veszteségmentes módszerek
  - A leírásból kimenet alakzataból visszaállítható-e az eredeti alakzat?
  - Információ megőrző módszerek
    - Az információ veszteség kontrolálható: Az eredeti alakzat visszaállítható egy jól meghatározható pontosságig
    - Tökéletes visszaállíthatóság (pl.: MA)
  - Információ veszteséssel járó módszerek
    - Az információ veszteség nem kontrolálható, nem határozható meg
    - Az alakzat nem visszaállítható (pl: terület, kerület információ)

## Módszerek tulajdonságai – alak reprezentációk

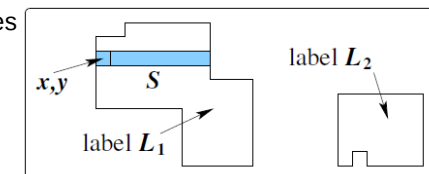
- Az alakzatok leírhatók
  - Kontúrral
  - Területtel
- Terület alapú módszerek
  - Az alak teljes területével számolunk
  - 2D rendezés, szomszédság
- Kontúr alapú módszerek
  - Az alakok körvonalával számolunk
  - Kontúr menti rendezés (1D), nincs szomszédság
- Más tudunk elemezni a két reprezentációval

## Adat struktúrák

- Az alakzat reprezentációk más-más adat struktúrát igényelnek
  - Bináris képek tárolására laklamzhatók
  - Eltérő tömörítési képesség
  - Eltérő műveletek
- Terület alapú struktúrák
  - Kép mátrix
  - Negyedelő fa (quadtree)
  - Futamhossz kód (run-length code)
- Kontúr alapú struktúrák
  - Lánckód (chain code)
  - Lánckód változatok

## Futamhossz kód (Run Length Code, RLC)

- A képet vízszintes futamok halmazával írjuk le
  - Futam: maximális összefüggő pixelsorozat
- Futamok leírása:
  - Első pixel pozíciója:  $(x, y)$
  - Hossz (S): a futamot alkotó sorozat elemszáma
  - Címke (L): Azt az objektumot jelöli, amihez a futam tartozik
- Az RLC meghatározása:
  - Címkézetlenül: a futamok megállapítása
  - Címkézetten: komponens analízis, megállapítani mely futamok tartoznak egybe
- Az RLC veszteségmentes



## Összefüggő komponens analízis (Connected component analysis)

- Feladat: meghatározni bináris képeken az összefüggő pixeleket
  - Reprezentáció függő
- Rekurzív terjedés: a kép mátrixon rekurzívan keressük és kapcsoljuk az összefüggő pixeleket
- Lánc kód:
  - Az összefüggő objektum kontúrján járunk végig
- RLC alapú:
  - Címkezetlen futamhossz kód
  - A futamok felcímkézése

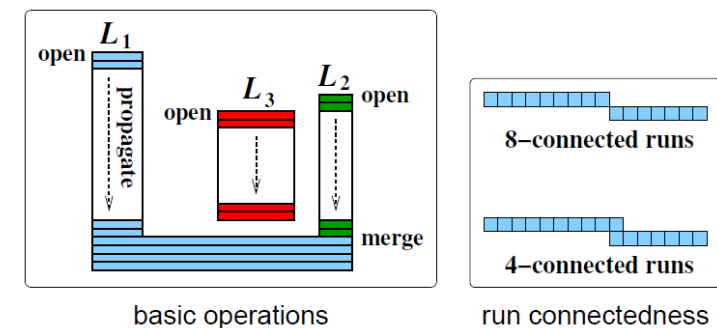
## Összefüggő komponens analízis – egyszerű megoldás

- Működés:
  - Keressük meg a következő címkezetlen elemet a kép mátrixban
  - Terjeszkedünk a szomszédokra
  - A terjeszkedés során érintett pixeleket címkézzük az aktuális címkével
  - Ha megáll a terjeszkedés vegyük a következő címkezetlen elemet
- Tulajdonságok:
  - Rekurzív implementáció: pár sor a kód
  - Nagy komponensek esetén lassú
  - Mély lehet a rekúzió
  - Stack elfogyhat
  - A pixelek látogatásának sorrendje nem befolyásolható

## RLC alapú összefüggő komponens analízis

- Bemenet: címkezetlen futamok
- Kimenet: címkézett futamok
- Két menetes algoritmus
- Első menet: Három lépés
  - Új címke létrehozása
  - Aktuális címke terjesztése
  - Ekvivalens címkék összefűzése (ekvivalencia táblázat frissítés)
- Második menet:
  - Az ekvivalencia táblázat alapján újra címkézni
  - A valódi komponens szám meghatározása

## RLC alapú összefüggő komponens analízis – első menet



- Három művelet:
  - Új címke létrehozása
  - Aktuális címke terjesztése
  - Ekvivalens címkék összefűzése (ekvivalencia táblázat frissítés)

## RLC alapú összefüggő komponens analízis – első menet

- Inicializálás:
  - Szomszédság kiválasztása
  - Ekvivalencia táblázat inicializál  $E_{ij} = 1; E_{ij} = 0, i \neq j$
  - Címke számláló inicializálása:  $L=0$
- Ha címkézetlen futamot találunk keressünk címkézett futamot a megelőző sorban
  - Ha nem találunk: új címke, a futam címkéje legyen  $L, L+=1$
  - Egyetlen kapcsolódó futamot találunk ahol  $L' \leq L$ : legyen a futam címkéje is  $L'$
  - Több kapcsolódó futam ahol  $L_k \leq L$ :
    - vegyük a legkisebb indexű címkét  $L'_m < L'_k, k \neq m$
    - legyen a futam címkéje  $L_m$
    - Állítsuk be minden  $k$ -ra:  $E_{km}=1$

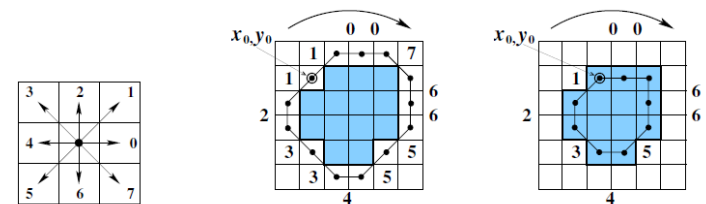
## Kerület alapú leírások

- Lánckód (chain code)
  - Összefüggő alakzatokra
  - Bináris képre
- Kontúr meredekség sorozat (contour slope sequence)
  - Alakzatokra
- Szög függvény (radial function)
  - Alakzatokra

## Lánc kód

- 1 Összefüggő alakzatra:
  - Vegyünk egy kiinduló kontúr pixelt
  - Kövessük az alakzat kontúriját
  - Tároljuk le az utat, mint a következő kontúr irányát: raszteren 8 irány lehetséges
- A Kód:
  - Kezdőpont  $(x,y)$
  - Irány sorozat  $\{c_1, \dots, c_n\}$  a következő pixelekhez
- Bináris képre (több összefüggő alakzat):
  - A komponenseket megkeressük
  - Minden komponens lánc kódját meghatározzuk
  - Belső kontúrok meghatározása (lyukak)
  - Külön figyelem a beágyazott objektumokra

## Lánc kód



eight chain codes 8-connected ground 8-connected figure

- 8 szomszédság, a háttéren 4 szomszédság az előtéren
  - Az első olyan háttérből induljunk, amittől jobbra objektum van
  - A komponens külsejét járjuk körbe
  - PI: 1, 0, 0, 7, 6, 6, 5, 5, 4, 3, 3, 2, 1
- 4 szomszédság, a háttéren 8 szomszédság az előtéren
  - Az első előtér pixelből induljunk, aminek van szomszédja tőle balra
  - A komponens belsejét járjuk körbe
  - PI: 0, 0, 6, 6, 5, 4, 3, 2, 1

## Lánc kód – belső és külső kontúrok

- A belső és külső kontúrok egymásba lehetnek ágyazva (kisebb objektum a lyukon belül)
- Válasszuk ki a külső kontúr körüljárás irányát
  - PI: óramutató járásával egyező
  - A terület pozitív, ha az y tengely felfele mutat
- Válasszuk ki a belső kontúr körüljárás irányát
  - Mindig ellenkező a külső kontúréhoz képest
  - A terület negatív
  - A terület előjelével döntsük el, hogy lyuk vagy objektum

## Lánc kód – alakzat visszaállítása

- Az eredeti alakzat visszaillesztése lánc kódból:
  - A kezdő pontból kezdve rajzoljuk újra a kontúr
  - Ha nincs kezdő pont, az újrarajzolás eltolás erejéig egyértelmű: az alakzat megmarad
- Bináris kép visszaállítása
  - Probléma: kontúr egybeágyazás lehetséges
  - Belső pontok meghatározására van szükség
  - Kontúrok környezetét vizsgáljuk
  - Megfelelő sorrendben töltjük fel a területeket
  - Beágyazás elemzés szükséges
  - Ellenőrizni kell, hogy egy kontúr pont egy másikon belül van-e

## Lánc kód - Megvalósítás

- Nem triviális
  - Jelöljük meg a látogatott pontokat
  - Amikor dönteni kell forduljunk jobbra
  - Ha kezdeti pontba értünk, keressünk új objektumot
- RLC alapján is lehet lánc kódot építeni
  - Bonyolultabb mint a kép mátrix használatával
- Lánc kód és az objektumok változása:
  - Eltolás: kezdeti pont megváltozik, a lánc nem
  - Forgatás: nem kiszámítható, újra kell számolni a láncot