Számítógépes képfeldolgozás

Régió alapú szegmentálás

Tófalvi Tamás

Eötvös Loránd Tudományegyetem, Informatikai Kar
tofi@inf.elte.hu



ELTE
EÖTVÖS LORÁND
TUDOMÁNYEGYETEM

Tartalom

1. A régió alapú szegmentálás elvei
2. Régió-növesztés
   - Pixel-felhalmozás
   - Régió-egyesítés
3. Vágás-és-egyesítés
4. Példák és összefoglaló
   - Összehasonlító példák
   - Alkalmazások valós képeken
   - Összefoglalás

Tartalom

1. A régió alapú szegmentálás elvei
2. Régió-növesztés
   - Pixel-felhalmozás
   - Régió-egyesítés
3. Vágás-és-egyesítés
4. Példák és összefoglaló
   - Összehasonlító példák
   - Alkalmazások valós képeken
   - Összefoglalás

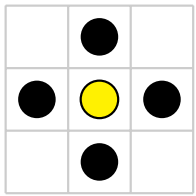
Küszöbölés és régió alapú szegmentálás

- A **küszöbölés** a pixeleket csak az intenzitásuk alapján osztályozza
  - a hisztogram nem tartalmaz térbeli információt

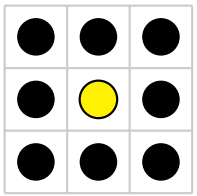
  $\Rightarrow$ nem garantál összefüggő régiókat
- A **régió alapú szegmentálás** az intenzitást és a térbeli szomszédságot együtt használja

  $\Rightarrow$ a kapott szegmensek, régiók garantáltan összefüggők
  - angolul: *region-based segmentation*
- Két alapkérdés
  - mikor összefüggő egy pixelhalmaz?
  - mikor homogén egy régió?

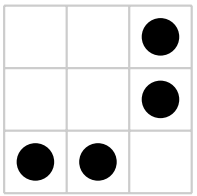
Szomszédság és összefüggőség



4-szomszédság



8-szomszédság



8-összefüggő, de nem 4-összefüggő

- Két $p, q \in S$ pixel **összefügg** az S halmazban, ha van olyan $p_0 = p, p_1, \ldots, p_n = q$, $p_i \in S$ pixelsorozat, amelyben p_i és p_{i-1} szomszédok
- S **összefüggő régió**, ha bármely két pixele összefügg S-ben
  - a 4-összefüggő régió 8-összefüggő is, fordítva nem mindig
- Az R régió **határpixele**: olyan R-beli pixel, amelynek van R-en kívüli szomszédja

A régió alapú szegmentálás célja

- Osszuk fel az I képet n darab $R_1, \ldots, R_n$ régióra, amelyek
  - összefüggők és
  - homogének
- Ehhez definiálunk egy $P(R)$ **homogénségi kritériumot**, amely minden $R \subset I$ régióra alkalmazható
- $P(R) = \mathrm{TRUE}$, ha minden R-beli képelemnek hasonló tulajdonságai vannak
  $\Rightarrow$ R homogén
- Egyébként $P(R) = \mathrm{FALSE}$
  $\Rightarrow$ R inhomogén (nem homogén)

Homogénségi kritériumok

- Egy R régió homogén, ha pl.
  - $|I_{max} - I_{min}|$ kicsi
  - minden $(x, y) \in R$ pixelre $|I(x, y) - I_{mean}|$ kicsi, ahol I_{mean} a régió átlaga
  - a σ_R intenzitásszórás az R régióban kicsi
- A „kicsi” egy T küszöbbel adható meg, pl. $|I_{max} - I_{min}| \leq T$
- A szegmentálás eredménye a következőktől függ:
  - milyen képi tulajdonságokat használunk
    $\Rightarrow$ intenzitás, szín, textúra
  - hogyan hasonlítjuk össze a tulajdonságokat
  - mekkora változásokat tolerálunk egy régión belül

A szegmentálás definíciója

Az I képet n darab $R_1, \ldots, R_n$ régióra bontjuk úgy, hogy

1. $\bigcup_{i=1}^{n} R_i = I$ — minden pixel valamely régióhoz tartozik
2. R_i összefüggő minden i-re — topológiai összefüggőség
3. $R_i \cap R_j = \emptyset$ minden $i \neq j$-re — nincs közös pixel
4. $P(R_i) = \mathrm{TRUE}$ minden i-re — minden régió homogén
5. $P(R_i \cup R_j) = \mathrm{FALSE}$ minden szomszédos R_i, R_j-re
   - bármely két szomszédos régió uniója inhomogén

- Az 5. feltétel nélkül az egyedi pixelekből álló, triviális szegmentálás is megfelelne
  $\Rightarrow$ a régiók száma minimális legyen

Régió alapú szegmentálási eljárások

- **Régió-növesztés** (*region growing*)
  - pixeleket vagy kisebb régiókat nagyobb régiókba csoportosítunk
  - addig növesztünk, amíg a homogénségi kritérium engedi
- **Pixel-felhalmozás** (*pixel aggregation*)
  - a régió-növesztés egyszerű formája: magpontokból indul
- **Régió-egyesítés** (*region merging*)
  - a régió-növesztés egyik formája; önálló műveletként vagy egy iteratív algoritmus részeként
- **Vágás-és-egyesítés** (*split-and-merge*)
  - a képet négyzetekre vágja, majd a hasonlókat egyesíti

$\Rightarrow$ A lényegük egyszerű, de korrekt, hatékony megvalósításuk már nem

Tartalom

1 A régió alapú szegmentálás elvei

2 Régió-növesztés
- Pixel-felhalmozás
- Régió-egyesítés

3 Vágás-és-egyesítés

4 Példák és összefoglaló
- Összehasonlító példák
- Alkalmazások valós képeken
- Összefoglalás

Pixel-felhalmozás: az algoritmus

Algoritmus: pixel-felhalmozás

1. **Inicializálás**
   - kiválasztunk N darab s_i magpontot és egy T küszöböt
   - a régiók és átlagaik kezdőértéke: $R_i^{(0)} = \{s_i\}$, $M_i^{(0)} = I(s_i)$
2. **Iteráció**, k-adik lépés
   - megvizsgáljuk az összes $R_i^{(k)}$ minden határpixelének 8-szomszédjait
   - ha van olyan új (még szabad) p szomszéd, amelyre $|I(p) - M_i^{(k)}| \leq T$, akkor p-t hozzáadjuk $R_i^{(k)}$-hez
3. **Megállunk**, ha nem tudunk tovább növeszteni; különben felfrissítjük az összes $M_i^{(k)}$-t és iterálunk

Pixel-felhalmozás: numerikus példa

0	0	5	6	7
1	1	5	8	7
0	**1**	6	**7**	7
2	0	7	6	6
0	1	5	6	5

$k = 0$: a két mag
$M_1 = 1$, $M_2 = 7$

0	0	5	6	7
1	1	5	8	7
0	**1**	6	**7**	7
2	0	7	6	6
0	1	5	6	5

$k = 1$
$M_1 \approx 0{,}83$, $M_2 \approx 6{,}56$

0	0	5	6	7
1	1	5	8	7
0	**1**	6	**7**	7
2	0	7	6	6
0	1	5	6	5

$k = 2$: vége
$M_1 = 0{,}6$, $M_2 = 6{,}2$

- $T = 3$; kék: R_1, narancs: R_2, keretben: a magok
- $k = 0$: a magok 8-szomszédai közül a 0–2 értékűek R_1-be, az 5–8 értékűek R_2-be kerülnek
- $k = 1$: az új átlagokkal a felső és az alsó sor is besorolódik
  - pl. a felső sor 5-öse: $|5 - 0{,}83| > 3$, de $|5 - 6{,}56| \leq 3 \Rightarrow R_2$
- $k = 2$: nincs új szomszéd $\Rightarrow$ megállunk

A küszöb és a sorrend hatása

0	0	5	6	7
1	1	5	8	7
0	**1**	6	**7**	7
2	0	7	6	6
0	1	5	6	5

$T = 4$
előbb R_1-et bővítjük

0	0	5	6	7
1	1	5	8	7
0	**1**	6	**7**	7
2	0	7	6	6
0	1	5	6	5

$T = 4$
előbb R_2-t bővítjük

0	0	5	6	7
1	1	5	8	7
0	**1**	6	**7**	7
2	0	7	6	6
0	1	5	6	5

$T = 8$
egyesítés után

- $T = 4$: a 2. sor 5-öse mindkét maghoz csatolható: $|5 - 1| = 4$ és $|5 - 7| = 2$
  - $\Rightarrow$ az eredmény a feldolgozási sorrendtől függ
- $T = 8$: a növesztés után a két régió uniója is homogén
  - minden pixel legfeljebb 4,04-dal tér el a 3,96-os átlagtól
  - $\Rightarrow$ régió-egyesítés után egyetlen régió marad
- Az eredmény a magpontoktól is függ

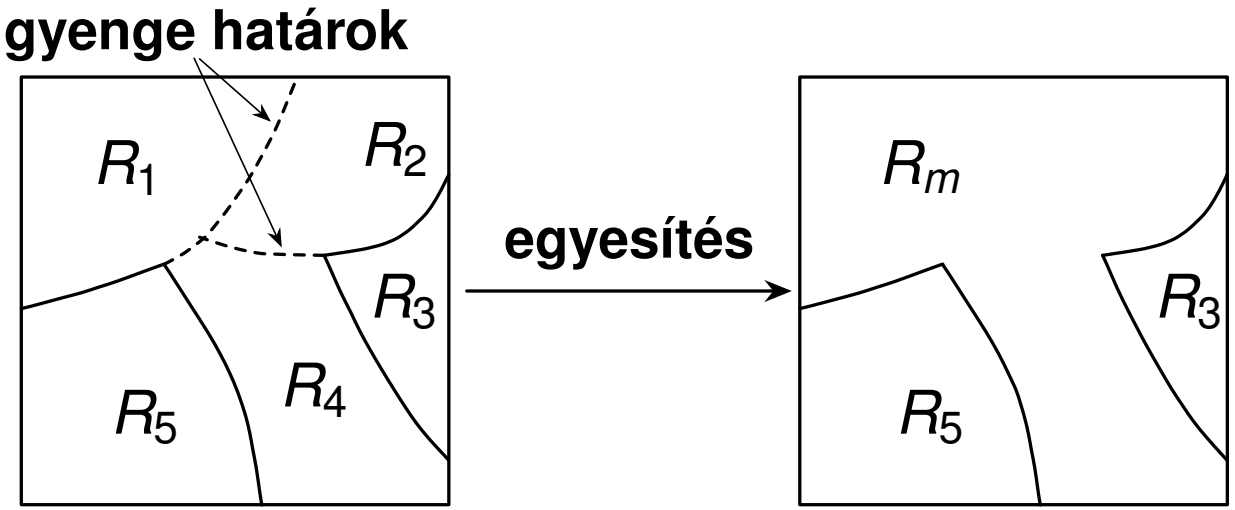
A magpontok kiválasztása

- A magpontok kiválasztása nem egyszerű feladat
  - $\Rightarrow$ az eredmény függ a magpontoktól
- Egyes (pl. orvosi) alkalmazásokban kézzel is megadhatók
- Ha nem tudjuk, hány szegmens van a képen, és honnan érdemes kiindulni:
  - sok magpontot szórunk szét véletlenszerűen
  - végrehajtjuk a pixel-felhalmozást
  - $\Rightarrow$ **túlszegmentálás** (*oversegmentation*)
  - $\Rightarrow$ utólagos régió-egyesítéssel kompenzáljuk

Régió-egyesítés

- A régió-növesztés egyik formája
- Akkor használjuk, ha szomszédos régiók hasonló tulajdonságúak
  - $\Rightarrow$ feltehetően egy szegmens részei
  - $\Rightarrow$ egybeolvaszthatók
- Tipikus példa: pixel-felhalmozáskor több mag került egy szegmensbe
  - $\Rightarrow$ a szegmens több régióra bomlott
  - $\Rightarrow$ ezeket egybeolvasztjuk
- Felhasználása
  - önálló műveletként, pl. pixel-felhalmozás után
  - egy iteratív szegmentáló algoritmus részeként

Egyesítési kritériumok



- Két szomszédos régió, R_i és R_j egyesíthető, ha
  - az uniójuk homogén: $P(R_i \cup R_j) = \mathrm{TRUE}$, **vagy**
  - a közös határuk „gyenge”: nincs rajta erős él (nagy gradiensű pixel), vagy sok rajta a kis gradiensű pixel
- Az ábrán R_1, R_2 és R_4 közös határai gyengék $\Rightarrow$ $R_m = R_1 \cup R_2 \cup R_4$

Régió-növesztés egyesítéssel

- A képet sok kisebb, homogén **magrégióra** bontjuk
  $\Rightarrow$ közel állandó intenzitás vagy más lokális tulajdonság
- Ezek után iteratívan egyesítjük a szomszédos régiókat

Algoritmus: régió-növesztés egyesítéssel

1. Bontsuk a képet sok kis, homogén magrégióra
2. Keressünk két szomszédos, egyesíthető R_i, R_j régiót: $P(R_i \cup R_j) = \mathrm{TRUE}$, vagy gyenge a közös határuk
3. Egyesítsük őket: $R_i \leftarrow R_i \cup R_j$
4. Ismételjük a 2–3. lépést, amíg van egyesíthető régiópár

Honnan lesznek a magrégiók?

- A magpixelek kézi megadását többnyire el akarjuk kerülni
- Sűrű, véletlenszerű vagy szabályos **magpixel-halmaz**
  - ebből felhalmozással sok magrégiót kapunk
  - az esetleg be nem járt pixeleket külön kell feldolgozni
  - $\Rightarrow$ túlszegmentált eredmény, jöhet az iteratív egyesítés
- **vagy** végigpásztázzuk a képet
  - minden fel nem dolgozott pontból felhalmozást indítunk
  - a kapott magrégiókat iteratívan egyesítjük
  - $\Rightarrow$ az eredmény függ a pásztázás sorrendjétől

A régió-növesztés összefoglalása

Előnyök

- Garantáltan összefüggő régiók
- Beépíthetők az elvárások (a priori tudás): alak, méret, textúra
  - $\Rightarrow$ kívánt tulajdonságú régiókat kapunk

Hátrányok

- A magpixelek kijelölése nem egyszerű
- Sok a heurisztika: egyesítési szabályok, gyenge határok
- Soros módszer: nem könnyen párhuzamosítható
- Sorrendfüggő: az eredmény függ a feldolgozási sorrendtől
  - $\Rightarrow$ erre vannak részmegoldások, de nem tökéletesek

Tartalom

1 A régió alapú szegmentálás elvei

2 Régió-növesztés
- Pixel-felhalmozás
- Régió-egyesítés

3 Vágás-és-egyesítés

4 Példák és összefoglaló
- Összehasonlító példák
- Alkalmazások valós képeken
- Összefoglalás

Vágás-és-egyesítés: az algoritmus

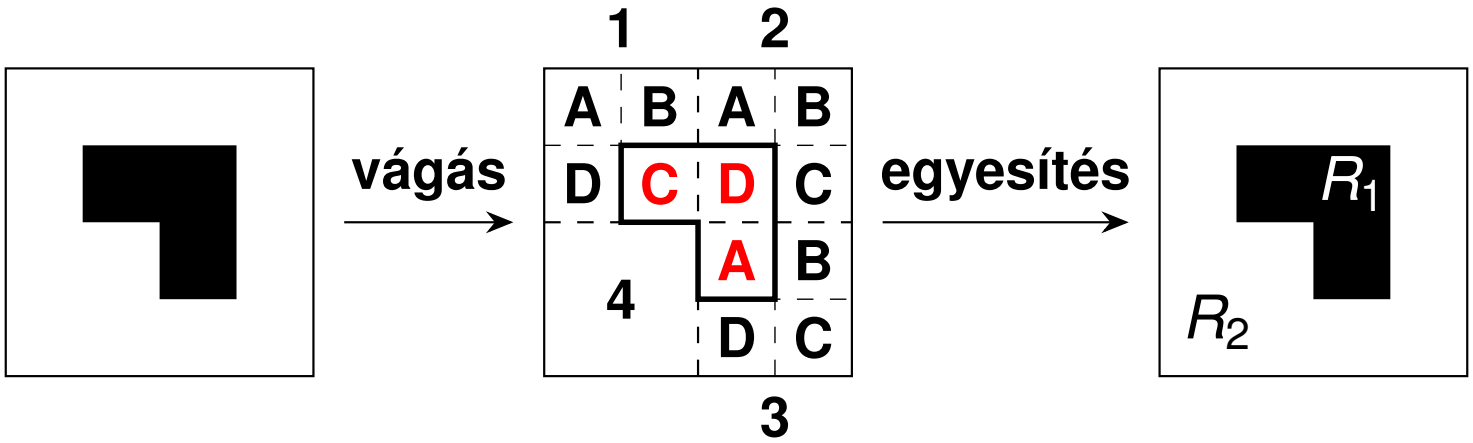
Algoritmus: vágás-és-egyesítés

1. **Fentről lefelé** (*top-down*)
   - minden inhomogén R_i négyzetet négy egyenlő részre vágunk
   - megállunk, ha minden négyzet homogén: $P(R_i) = \mathrm{TRUE}$

   $\Rightarrow$ az eredmény egy **négyesfa**
2. **Lentről felfelé** (*bottom-up*)
   - minden szinten egyesítünk két szomszédos R_i és R_j régiót, ha $P(R_i \cup R_j) = \mathrm{TRUE}$
3. Iteráljuk a két fázist, amíg van új vágás vagy egyesítés

- Vágás-és-egyesítés: *split-and-merge*; négyesfa: *quadtree*

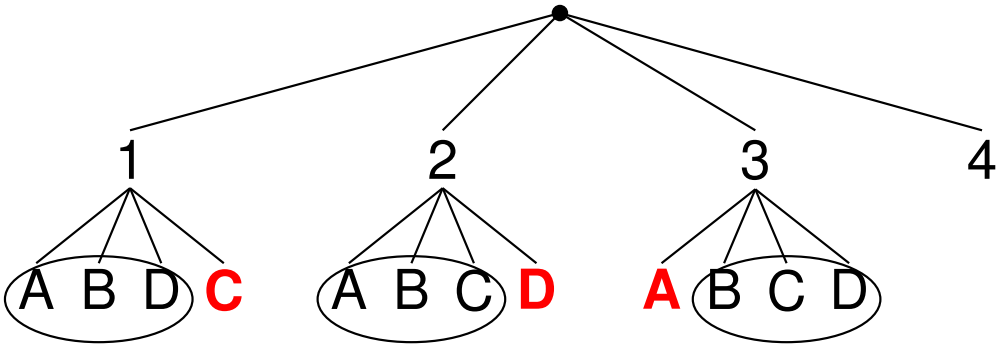
Szegmentálás vágás-és-egyesítéssel



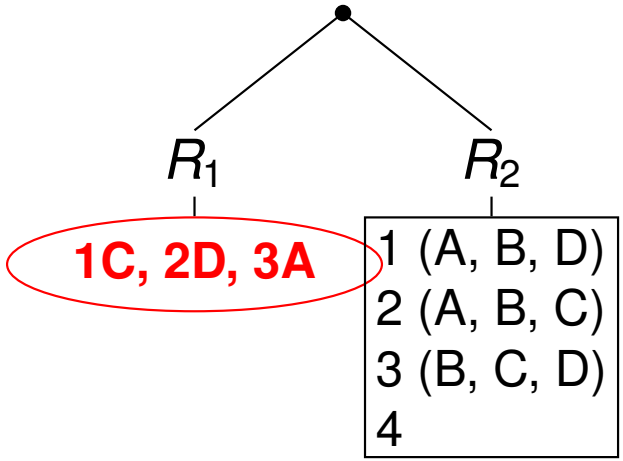
- **Vágás**: az 1., 2. és 3. negyed inhomogén $\Rightarrow$ négy részre (A, B, C, D) vágjuk
  - a 4. negyed és a részek homogének $\Rightarrow$ a vágás véget ér
- **Egyesítés**: a homogén uniójú szomszédos részeket összevonjuk

  $\Rightarrow$ R_1: az objektum (1C, 2D, 3A), R_2: a háttér

A négyesfa



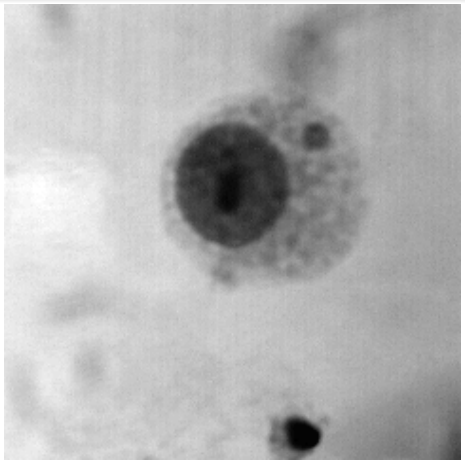
vágás-és-egyesítés

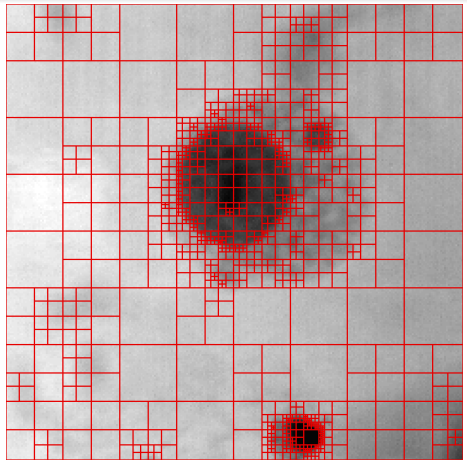


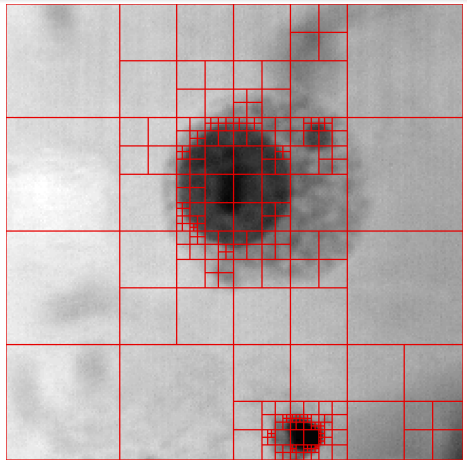
szegmentált régiók

- A gyökér a teljes kép, a levelek homogén négyzetek; ellipszisben az egyesíthető testvérlevelek
- 1C, 2D és 3A a képen szomszédos, a fában viszont csak a gyökérnél találkoznak
  - $\Rightarrow$ csak **vertikális** adatforgalom (a fa ágai mentén)

A vágási fázis egy valós képen


sejtkép


$T = 40$: 1045 levél


$T = 80$: 268 levél

- Homogénségi kritérium: $|I_{max} - I_{min}| \leq T$
- Homogén területeken nagy négyzetek, az élek mentén egyre kisebbek
- Nagyobb $T \Rightarrow$ kevesebb vágás, nagyobb négyzetek
- Az egyesítés ezekből a négyzetekből épít régiókat
  - $\Rightarrow$ a négyzetes struktúra az eredményen is látszik

A vágás-és-egyesítés összefoglalása

Előnyök

- Garantáltan összefüggő régiók
- Beépíthetők az elvárások, bár kisebb mértékben, mint a régió-növesztésnél

Hátrányok

- Ha a képet elforgatva vagy eltolva digitalizáljuk, lényegesen más eredményt kaphatunk
- A négyzetes struktúra látszik az eredményképen
- Egyszerű formában csak vertikális adatforgalmat támogat
  - $\Rightarrow$ két szomszédos régió messze lehet egymástól a négyesfában, és csak ott egyesül, ahol a két ág találkozik

Tipikus alkalmazás: Földrajzi Információs Rendszerek (GIS)

Tartalom

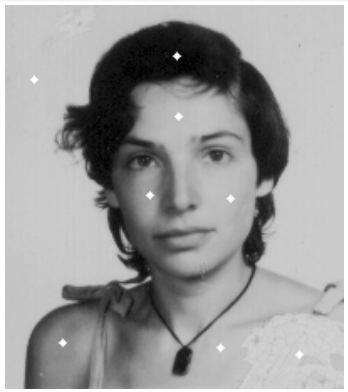
1 A régió alapú szegmentálás elvei

2 Régió-növesztés
- Pixel-felhalmozás
- Régió-egyesítés

3 Vágás-és-egyesítés

4 Példák és összefoglaló
- Összehasonlító példák
- Alkalmazások valós képeken
- Összefoglalás

Pixel-felhalmozás növekvő küszöbbel


kép magokkal


$T = 20$


$T = 30$


$T = 40$

- A magpontokat kézzel helyeztük el (fehér pontok)
- $T = 20$: kevés mozgásszabadság
  - a változó intenzitású arcrégió nem tud nőni
  - $\Rightarrow$ a kis változású hajrégió nagy területet foglal el
- Növekvő T-re az arcrégió tovább terjed, a hajrégió zsugorodik
- Az eredmény a magpixelektől is függ

Régió-növesztés egyesítéssel növekvő küszöbbel



$T = 20$



$T = 30$



$T = 40$

- Végigpásztáztuk a képet, és minden új pontból magrégiót indítottunk
- A küszöb növelésének itt ellenkező hatása van: a régiók száma csökken
  - $\Rightarrow$ a szegmentálás kevésbé finom lesz
  - $\Rightarrow$ a hajrégió egyre nő

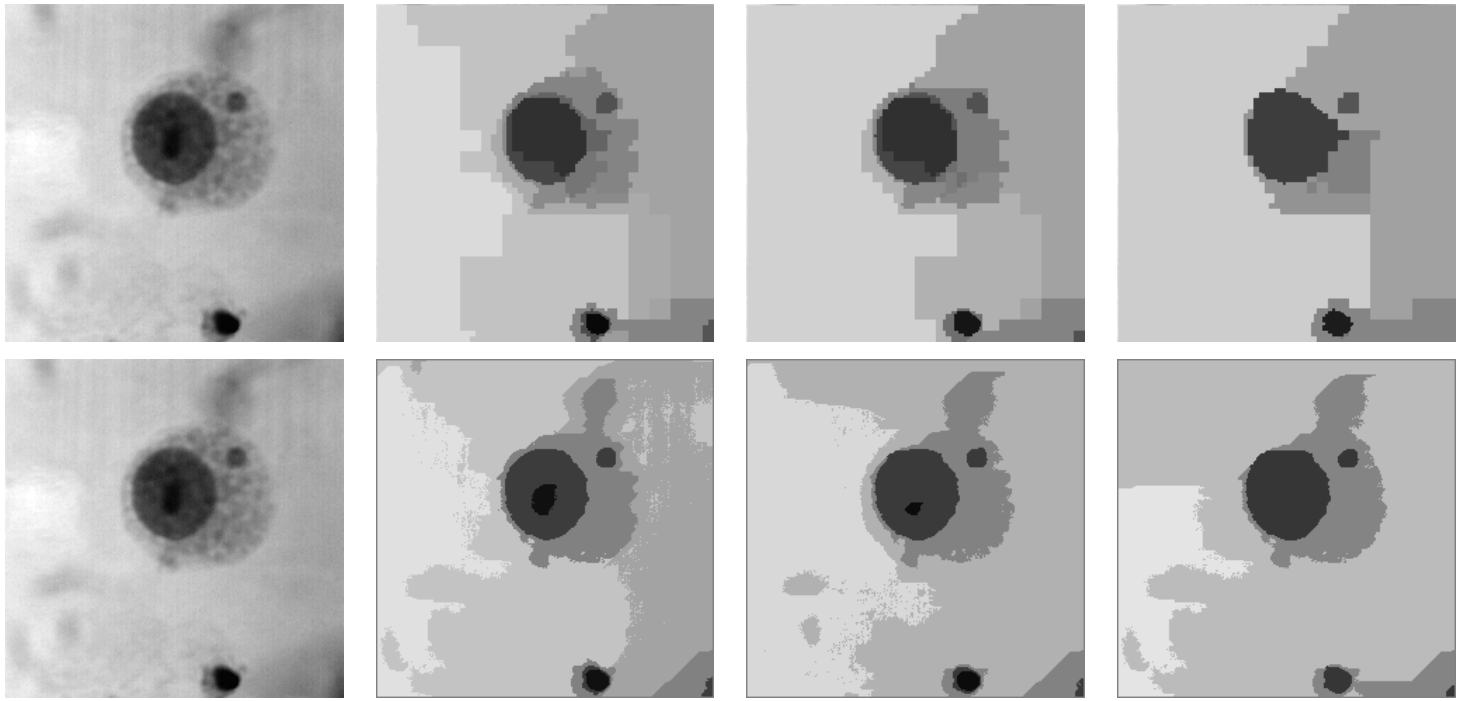
Egyesítés és felhalmozás összehasonlítása



egyesítés $T = 20$ | egyesítés $T = 40$ | felhalmozás $T = 20$ | felhalmozás $T = 40$

- Zöld: a hajrégió
- Egyesítés növekvő küszöbbel: a hajrégió nő
- Felhalmozás növekvő küszöbbel: a hajrégió csökken

Vágás-és-egyesítés kontra egyesítés



sejtkép $T = 40$ $T = 50$ $T = 60$

- Felső sor, vágás-és-egyesítés: látszik a négyesfára jellemző négyzetes struktúra
- Alsó sor, egyesítés: finomabb eredmény

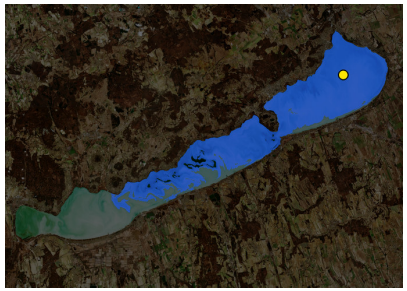
Alkalmazások valós képeken

kép	**terület**	**módszer**
a Balaton műholdképen	távérzékelés, környezetfigyelés	pixel-felhalmozás, egy mag
tüdő CT-szeleten	orvosi képfeldolgozás	pixel-felhalmozás, két mag
a Hold	bolygókutatás	régió-növesztés egyesítéssel
könyvoldal	dokumentumelemzés	vágás-és-egyesítés

- Mindegyik képen az előző diák algoritmusait futtattuk
- A régió alapú módszerek akkor működnek jól, ha a keresett régió
  - összefüggő,
  - belül homogén a választott $P(R)$ szerint, és
  - a határán $P(R)$ már nem teljesül

Pixel-felhalmozás: a Balaton műholdképen






kép a maggal $T = 20$ $T = 40$ $T = 80$

- Sentinel-2 műholdkép (2019. február 27.); $I = G - R$ (zöld mínusz vörös csatorna): a vízen nagy, a szárazföldön kicsi
- Egyetlen magpont a keleti medencében
- $T = 40$: átjut a Tihanyi-szoroson, de a nyugati medence még kimarad
- $T = 80$: a teljes tó (1130 iteráció)
- $T = 100$: a régió kifolyik a szárazföldre

Kép: ESA, módosított Copernicus Sentinel-2 adatok (2019), CC BY-SA 3.0 IGO

Küszöbölés kontra pixel-felhalmozás



küszöbölés: $I > 30$

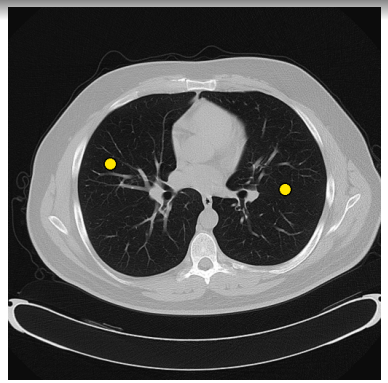


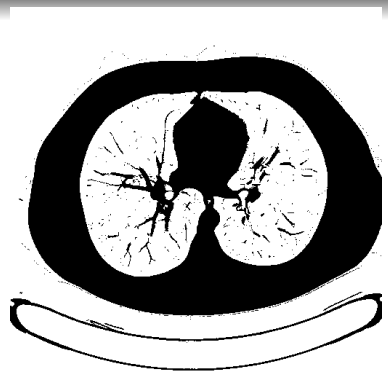
pixel-felhalmozás: $T = 80$

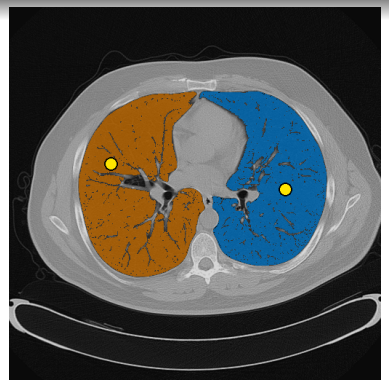
- Küszöbölés: 1152 összefüggő komponens; a zöld vetések is bekerülnek (piros), a nyugati medence egy része kimarad
- Pixel-felhalmozás: egyetlen összefüggő régió, a tó
- Hasonló a képszerkesztők „varázspálca” eszköze: a kattintás a magpont, a tolerancia a T küszöb

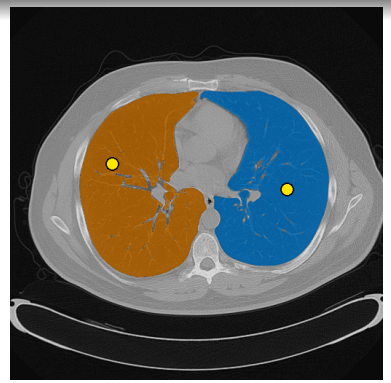
Kép: ESA, módosított Copernicus Sentinel-2 adatok (2019), CC BY-SA 3.0 IGO

Pixel-felhalmozás: a tüdő CT-szeleten


két mag


küszöbölés

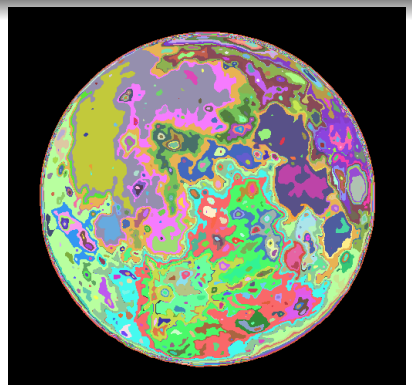
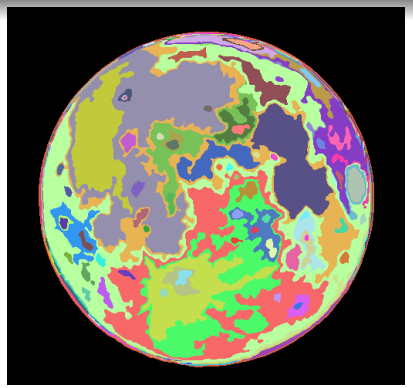
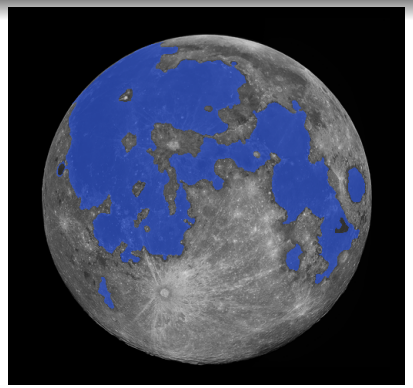

$T = 20$


$T = 60$

- Mellkas-CT, tüdőablak; mindkét tüdőfélben egy mag $\Rightarrow$ két régió: R_1 (narancs) és R_2 (kék)
- Küszöbölés ($I < 60$): a testen kívüli levegő és az asztal is bekerül
- $T = 20$: lyukacsos régiók (erek, zaj); $T = 60$: tömör tüdőmaszk
- Nagy a kontraszt: $T = 100$-nál sem folyik ki a régió

Kép: Ptrump16, Wikimedia Commons, CC BY-SA 4.0 (módosítva)

Régió-növesztés egyesítéssel: a holdtengerek

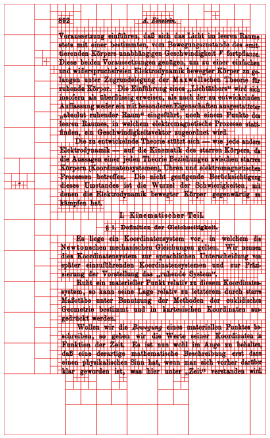





a Hold magrégiók egyesítés után holdtengerek

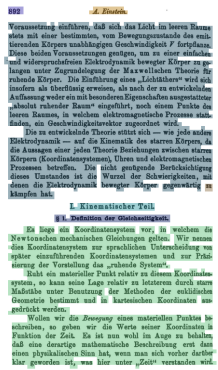
- Gauss-simítás után végigpásztázzuk a képet, felhalmozás $T_0 = 5$-tel
  - $\Rightarrow$ túlszegmentálás: 4086 magrégió
- Egyesítés, ha $P(R_i \cup R_j)$: minden pixelre $|I(x,y) - I_{mean}| \leq 20$
  - $\Rightarrow$ 256 régió, ebből 35 nagy
- A nagy, sötét régiók a holdtengerek: a korong 37%-a
- Nem kellett előre tudni, hány tenger van, és hol

Kép: Gregory H. Revera, Wikimedia Commons, CC BY-SA 3.0 (módosítva)

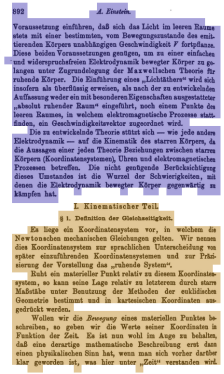
Vágás-és-egyesítés: egy könyvoldal tagolása



négyesfa



$m = 8$



$m = 16$

- $P(R)$: $|I_{max} - I_{min}| \leq 80$; a legkisebb négyzet $m \times m$-es
- Egyesítés után az üres papír egyetlen háttérrégió
- A többi régió csoportjai a szövegblokkok
- $m = 8$: 7 blokk; oldalszám, élőfej, címek, bekezdések
- $m = 16$: 2 blokk; a fejezetcím előtt és után
- Alkalmazás: oldalelemzés, pl. karakterfelismerés (OCR) előtt

Kép: A. Einstein, Annalen der Physik 17 (1905), 892. o.; közkincs

Más szegmentálási módszerek

- **Variációs**: aktív kontúrok, *level set* módszerek
- **Statisztikai**: pl. Markov véletlen mezők (*Markov Random Field*, MRF)
- **Modell alapú**: adott alakú és más tulajdonságú régiók keresése
- **Él alapú**: az objektumok behatárolása élkövetéssel
- **Textúra, szín alapú**: a régiók textúráját, színét használják
- **Mozgás alapú**: objektumok szegmentálása mozgás alapján

Összefoglalás

	pixel-felhalmozás	**növesztés egyesítéssel**	**vágás-és-egyesítés**
kiindulás	N magpont	sok kis magrégió	a teljes kép
irány	lentről felfelé	lentről felfelé	fentről lefelé, majd lentről felfelé
növekvő T	a régiók tovább terjednek	kevesebb, nagyobb régió	kevesebb vágás, nagyobb régiók
fő gond	a magpontok kiválasztása, sorrendfüggés	sorrendfüggés, heurisztikus szabályok	négyzetes struktúra, eltolás- és forgatásfüggés

- Mindhárom módszer garantáltan összefüggő régiókat ad
  $\Rightarrow$ szemben a küszöböléssel
- A $P(R)$ kritérium és a T küszöb szabja meg, mekkora változást tolerálunk egy régión belül