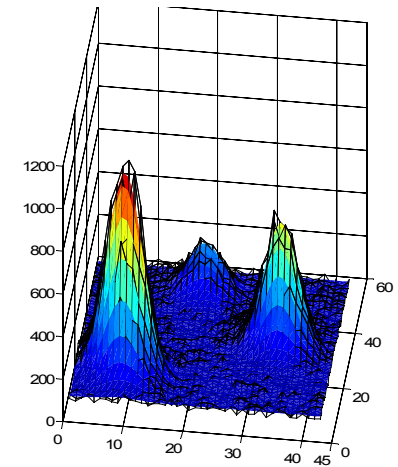
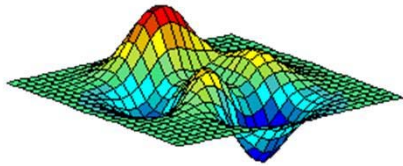
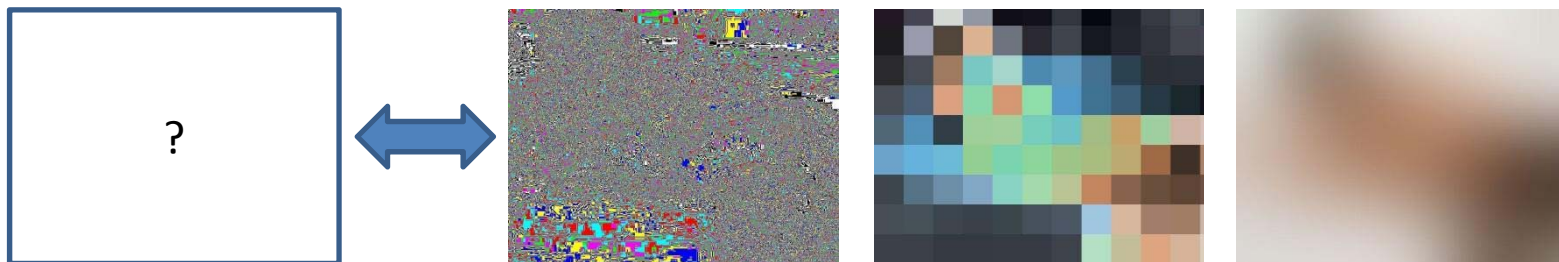




Digitális képanalízis

Nagy Péter

DEOEC Biofizikai és Sejtbiológiai Intézet



Digitális képfeldolgozás

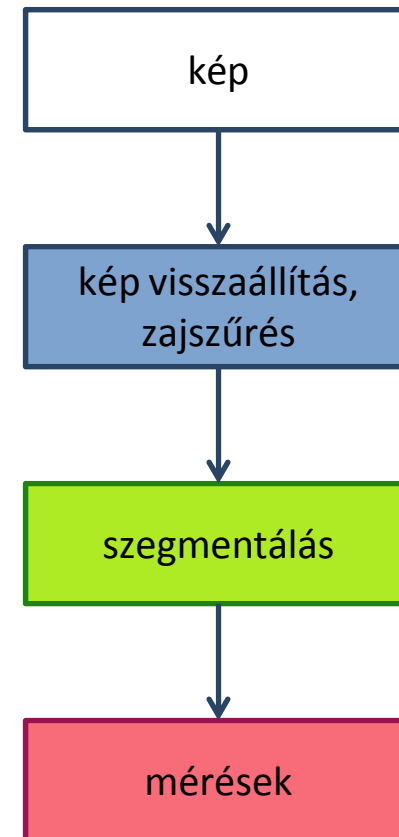
- Képek tárolása, képtömörítés (veszteséges és veszteség mentes), 3D képhalmaz, ortogonális metszet
- Digitális és optikai feloldóképesség
- Hisztogram, LUT

- Filterek (a tér és frekvencia doménekből)
- Képet degradáló tényezők
- Kép visszaállítás („image restoration”), zajszűrés, simítás
- Dekonvolúció
- Színes képek tárolása, színes LUT, spektrális szűrés („spectral unmixing”)
- Matematikai morfológia

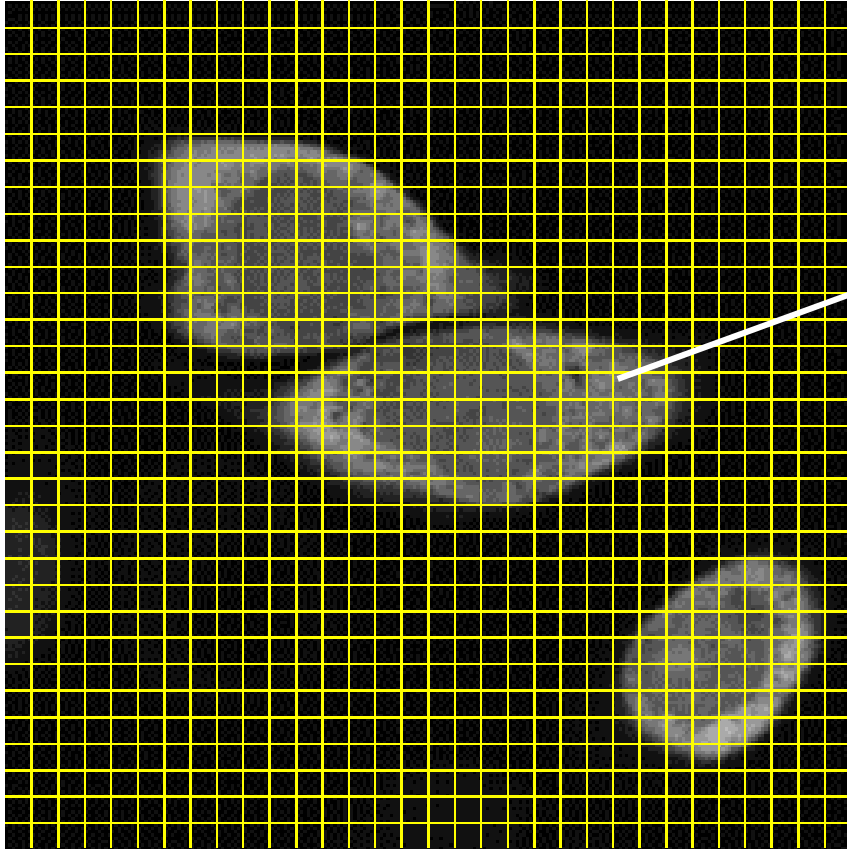
- Képszegmentálás

- Mérések

- Tudományos képanalizáló programok



Képek digitális reprezentációja

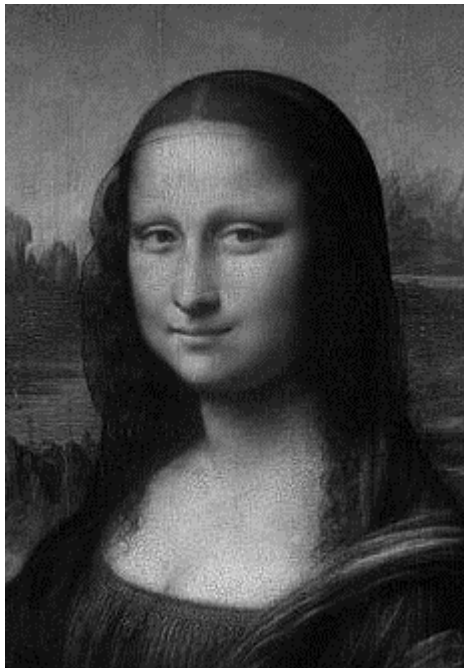


- Pixel („picture element”): a legkisebb egység a digitális képben.
- Értéke egy szám, amely a detektor megfelelő helyén detektált fotonok számával arányos (szürkeskálájú kép esetében).
- Annak megfelelően, hogy hány értéket vehet fel a pixel, a kép lehet:

intenzitás skála	lehetséges értékek	tárigény
bináris	0,1	1 bit/pixel
8-bites	0,1,...,255	1 byte/pixel
12-bites	0...4095	2 byte/pixel
16-bites	0...65535	2 byte/pixel
lebegőpontos	praktikusan tetszőleges	32 bit/pixel, 4 byte/pixel
kettős pontosságú		64 bit/pixel, 8 byte/pixel

Miért van szükség digitális képanalízisre?

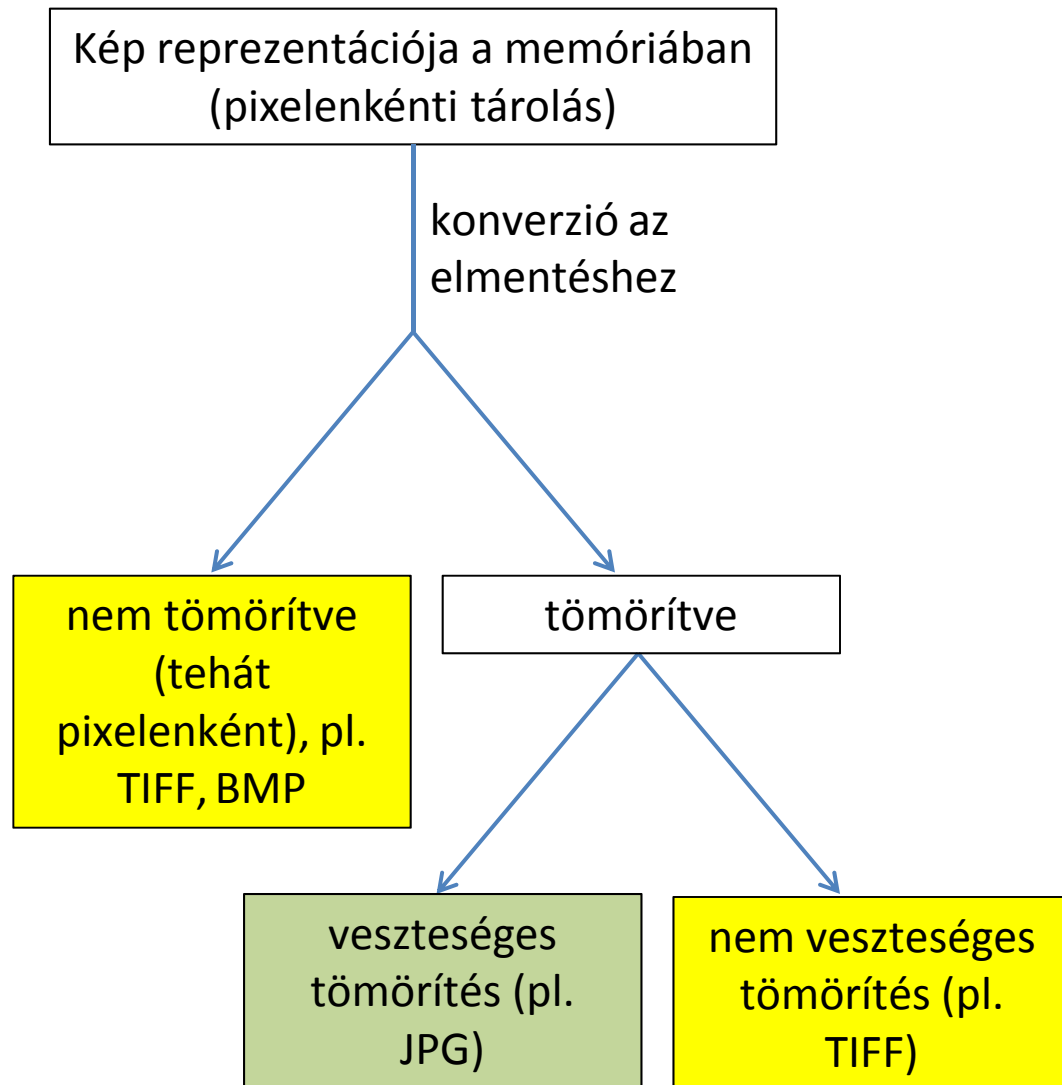
Mi ez?



96	95	94	96	99	101	101	99	95	87	92	100	94	90	96	100	99	97	97	99	103	103	97	92	92	90	91	96	100	100	101	104	100	99	96	94	
99	98	97	99	101	103	102	101	103	105	103	94	93	109	110	92	99	96	94	95	99	103	105	105	100	100	93	87	92	103	102	94	102	100	98	95	
99	98	97	98	100	101	100	99	95	92	101	110	104	96	87	76	56	53	49	47	49	55	62	66	80	92	101	98	94	96	100	102	105	103	99	96	
102	101	101	101	101	102	101	101	109	102	108	108	81	50	38	38	30	27	24	19	18	20	25	30	42	54	76	96	103	99	100	107	106	104	101	97	
110	110	110	110	109	108	109	109	117	111	94	62	36	40	47	40	34	34	32	28	23	21	20	21	21	19	26	51	86	110	109	98	107	105	102	100	
112	113	114	113	112	111	112	113	110	83	56	39	38	53	65	61	37	37	34	31	27	21	17	14	13	17	12	11	40	87	110	106	109	107	105	104	
112	113	115	114	112	112	113	115	95	49	43	55	63	62	60	57	51	47	40	34	30	25	19	15	11	19	19	8	8	36	81	115	113	111	109	108	
115	117	118	118	116	115	117	119	48	44	47	56	71	91	92	74	66	56	44	35	30	26	21	16	17	5	8	17	7	2	44	102	116	114	112	110	
118	112	116	122	119	128	120	73	38	65	91	114	135	140	134	133	120	105	88	65	39	27	25	20	15	16	8	10	15	7	15	47	113	109	112	116	
119	119	122	126	125	122	97	50	65	98	126	143	159	163	157	155	142	128	113	90	60	42	32	21	15	16	10	11	16	6	5	23	85	117	114	117	
122	125	126	127	129	108	67	36	91	130	159	167	175	179	174	170	162	149	136	115	83	58	40	23	18	17	10	9	14	8	1	6	40	110	114	115	
124	125	126	129	127	86	42	45	106	148	173	172	174	179	177	173	169	157	146	127	98	72	50	31	23	19	11	7	9	10	8	9	8	86	115	115	
123	122	131	133	119	63	23	60	116	154	174	168	167	173	173	171	168	155	143	127	102	81	61	41	28	21	15	11	8	11	13	11	2	56	112	114	
125	123	139	131	101	43	14	70	121	151	165	160	161	165	165	165	161	146	134	121	101	85	69	51	37	23	19	18	11	9	10	5	5	26	92	102	
132	131	144	111	69	32	20	79	122	143	154	155	159	161	158	159	157	142	131	122	108	99	87	71	56	27	17	19	11	8	9	4	7	8	62	85	
141	138	143	89	41	27	32	90	125	141	151	156	164	163	157	158	160	146	137	132	123	118	109	95	74	32	13	15	9	9	14	10	9	6	44	79	
149	150	128	85	24	16	36	99	130	158	146	155	154	161	167	162	157	140	134	138	133	123	109	96	69	41	16	10	11	7	7	12	5	11	31	68	
150	147	123	74	26	19	42	98	131	133	120	133	160	156	152	132	97	100	102	106	106	99	88	82	59	37	18	14	13	8	7	13	8	11	26	58	
145	133	105	51	30	22	36	74	98	95	84	89	138	144	130	86	45	73	83	76	73	60	44	42	55	36	20	18	16	9	8	13	11	10	20	49	
138	120	90	35	35	24	28	51	58	77	66	59	101	141	116	57	54	90	88	60	50	42	35	46	68	43	22	17	17	11	8	12	11	9	17	46	
123	117	93	37	35	22	34	72	72	94	71	73	95	154	109	57	90	120	105	65	58	68	83	107	89	54	22	14	15	12	9	10	12	9	17	46	
93	107	92	41	24	13	41	114	122	123	92	111	118	163	103	72	103	132	129	104	101	113	123	135	96	57	20	11	15	13	9	9	14	9	16	44	
60	84	73	40	17	10	39	130	146	145	129	140	137	161	106	91	111	138	155	151	145	142	129	113	85	50	17	10	15	13	10	10	14	8	12	38	
43	66	53	39	23	19	33	122	139	158	165	151	142	157	119	108	132	150	170	171	161	152	130	100	71	41	15	12	15	13	10	11	13	5	8	33	
51	45	47	28	29	22	23	101	148	156	158	139	152	160	119	103	123	145	158	165	156	140	118	77	57	35	16	13	15	13	13	15	14	11	18	34	
40	35	38	24	27	21	19	88	135	147	153	139	148	148	113	103	124	143	145	148	141	121	101	73	50	30	14	13	16	14	12	13	12	8	12	25	
34	29	31	20	25	19	14	67	126	146	147	127	141	151	117	92	93	139	148	140	130	108	84	61	42	26	13	13	16	14	11	11	11	7	9	17	
36	31	28	19	24	18	11	47	106	124	130	126	122	93	52	38	71	141	153	128	114	97	75	53	39	25	13	12	14	13	11	11	12	9	10	16	
39	33	25	17	21	18	13	33	81	97	123	153	137	68	34	54	83	128	125	108	102	84	67	56	40	27	15	11	12	13	12	12	12	11	13	17	
37	33	23	17	20	19	18	25	73	100	111	130	135	107	86	81	81	90	86	104	114	85	64	58	38	28	16	11	12	14	14	13	11	12	14	16	
37	36	26	22	22	21	21	18	59	104	101	94	107	109	91	54	70	78	87	114	116	86	65	50	32	25	16	12	13	16	15	12	9	11	14	16	
40	42	32	28	25	22	21	11	36	94	121	141	141	112	100	90	78	103	116	116	94	76	67	43	26	22	16	13	15	17	15	11	7	11	15	17	
51	38	35	29	13	12	20	16	15	43	90	128	131	107	87	83	84	88	94	92	78	58	45	40	28	20	15	15	15	12	11	13	9	11	14	17	
61	50	42	33	20	17	18	16	11	20	49	95	137	153	143	128	109	100	88	77	63	48	38	35	24	18	13	14	14	11	10	11	9	11	14	17	
58	55	40	28	25	22	18	21	18	12	17	52	111	155	155	133	103	85	65	53	46	38	33	30	21	17	14	14	13	11	10	11	9	11	14	16	
63	62	41	23	25	22	17	24	21	19	14	21	56	94	99	83	67	55	43	39	38	33	27	24	20	18	16	15	14	13	13	13	10	12	14	15	
79	77	51	26	23	19	14	21	11	18	16	9	17	37	45	40	42	37	33	34	32	26	21	20	23	22	20	16	14	14	14	14	11	12	12	13	14
79	76	55	29	19	17	17	19	10	14	15	12	12	20	30	36	34	30	28	27	25	22	24	28	31	31	26	18	13	12	13	12	12	12	13	13	
71	67	57	36	18	18	23	19	18	14	15	19	17	19	37	60	49	42	35	31	30	30	37	44	46	45	38	25	15	13	13	11	12	12	13	13	
75	71	67	48	21	19	25	15	19	12	14	21	16	15	46	86	77	66	54	47	44	43	47	52	59	58	48	31	19	15	14	12	13	13	12	12	
69	70	61	52	21	22	19	20	17	18	18	17	22	16	32	98	104	86	69	63	59	55	57	62	71	69	57	38	22	15	13	13	10	14	15	12	

- Azért, hogy az emberi szem felismerő képességét helyettesítsük.
- Kvantitatívan és reprodukálhatóan elemezzük ki a képeket.

Képek tárolása



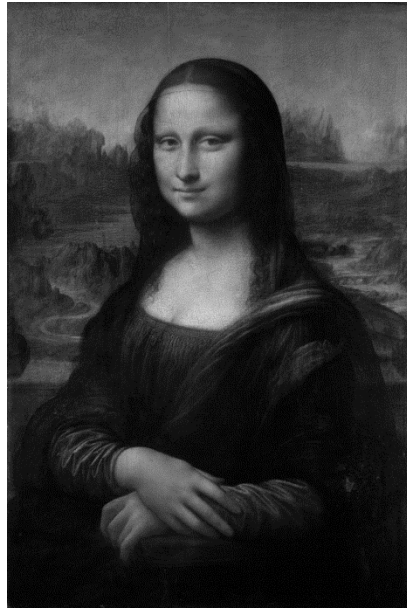
-  prezentáció céljából
-  képanalízis és tudományos publikáció céljából

Kép file-ok részei:

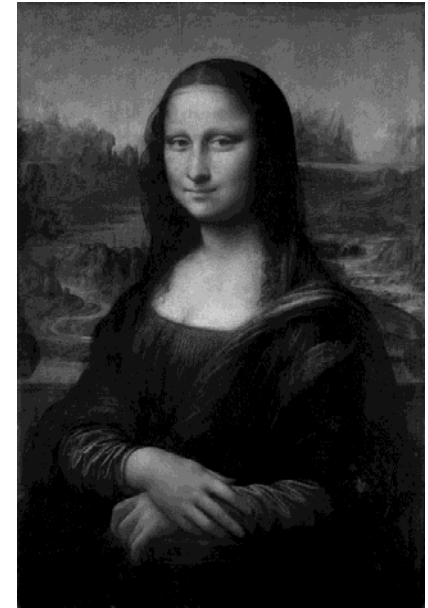
- leíró rész: megadja, hogy az adatokat hogyan kell értelmezni (hány bites feloldás, X-Y méret, stb.)
- a tulajdonképpeni adatok

Veszteséges tömörítés: a tömörített adatokból rekonstruált kép nem egyezik meg teljesen az eredeti képpel.

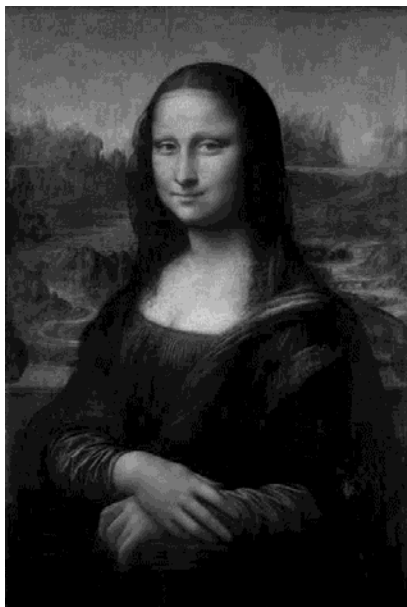
Veszteséges
képtömörítés



TIFF, 388 KB



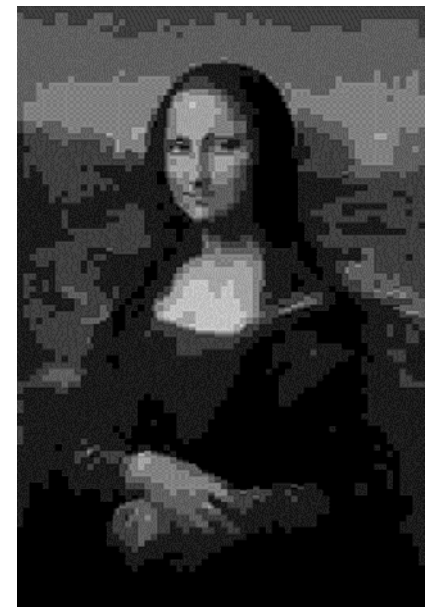
JPG, 59KB



JPG, 26KB



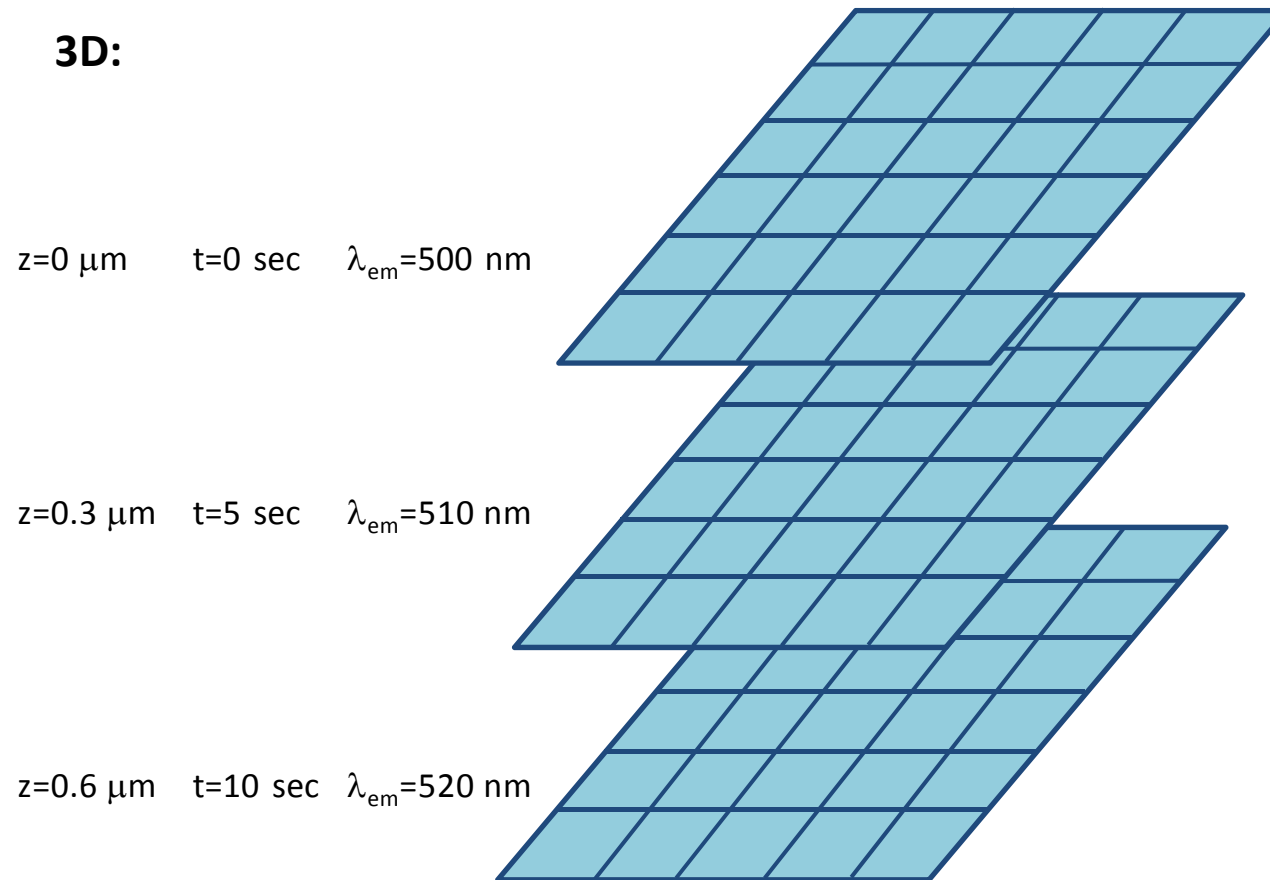
JPG, 20 KB



JPG, 18 KB

Kettőnél több dimenziós képek: 3D, 4D, 5D, ...

3D:



A harmadik dimenzió lehet:

- mélység (pl. konfokális mikroszkópos 3D szeletsorozat esetében)
- idő
- hullámhossz
- ...

>3D:

Elképzelhető háromnál több dimenziós kép is, pl. 1-3. dimenziók: x, y, z ; 4. dimenzió: idő; 5. dimenzió: hullámhossz.

3D-s képek megjelenítése 1.

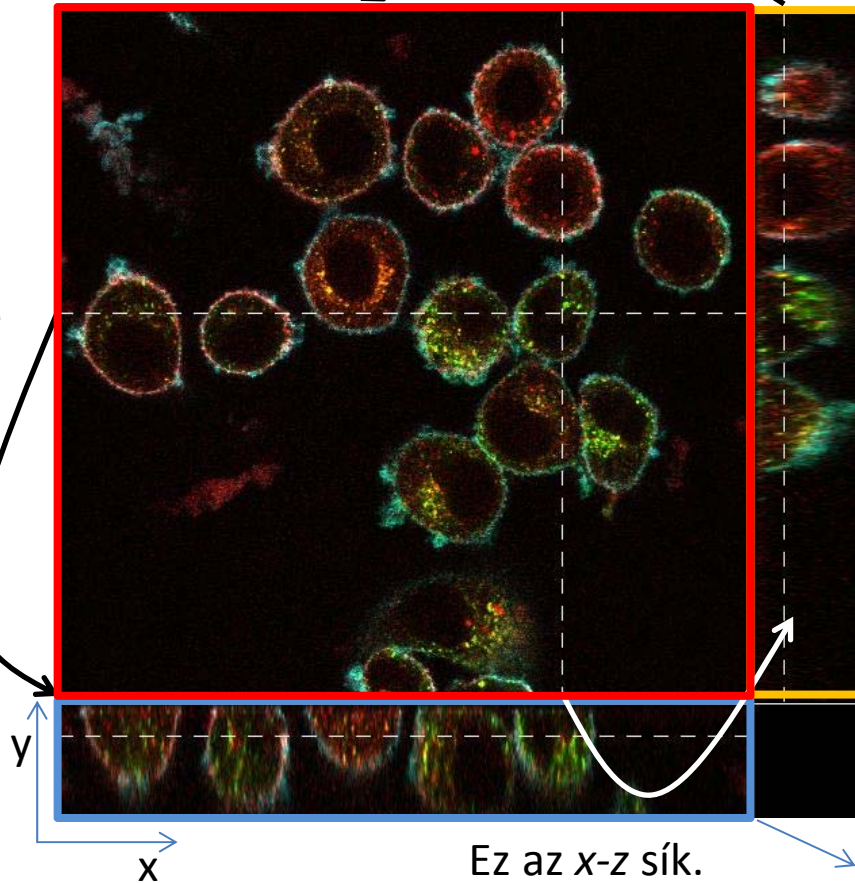
Ortogonalis metszet:

Ez az x-y sík.

x-y sík

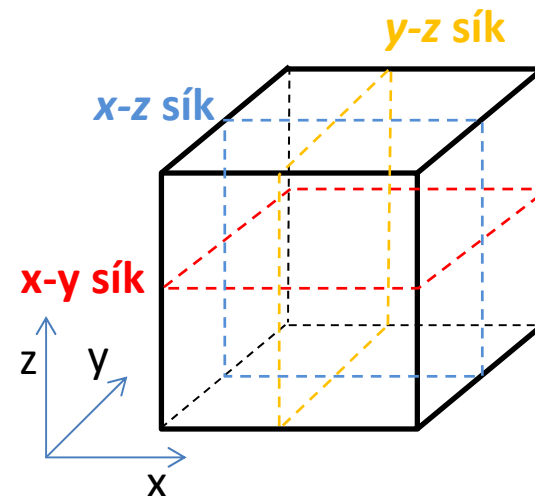
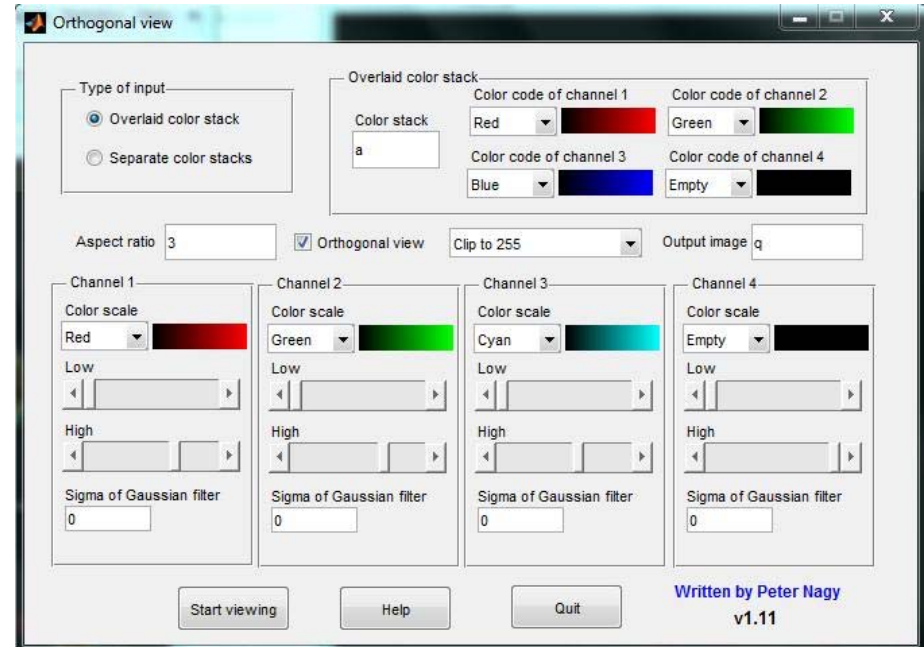
y-z sík

Ez az y-z sík.



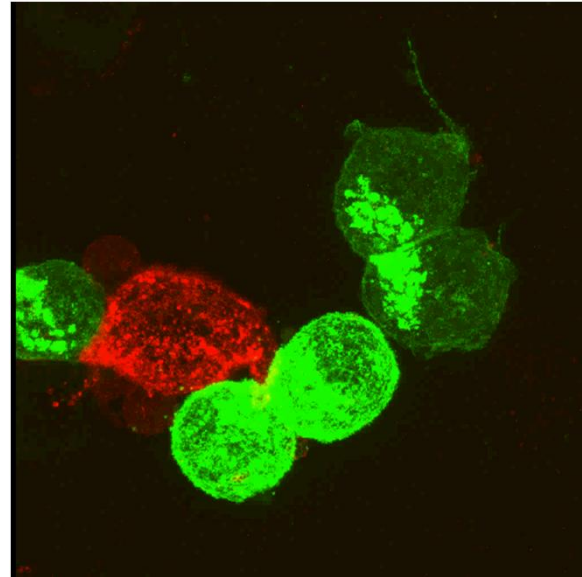
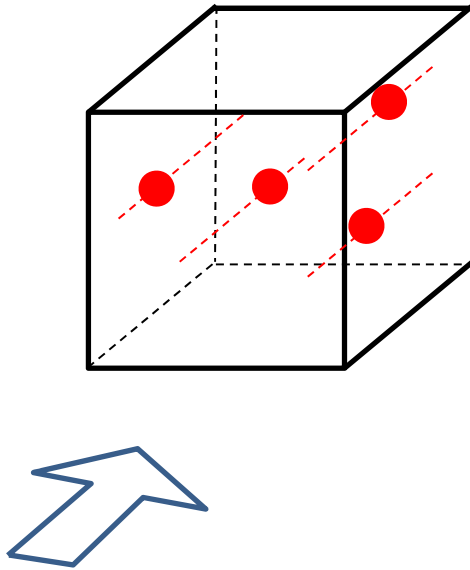
Ez az x-z sík.

x-z sík



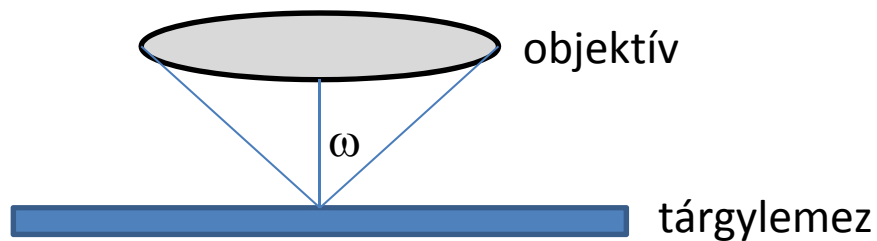
3D-s képek megjelenítése 2.

Maximum intenzitás projekció:



- Ebből az irányból nézzük.
- Minden projekciós vonal mentén megkeressük, hogy melyik pont a legfényesebb, és azt tüntetjük fel a 2D projekción.
- Előny: gyors kiszámolni.
- Hátrány: gyenge 3D-s hatás
- 3D-s hatás javítható, ha különböző irányokból is feltüntetjük a maximum intenzitás projekciót: „forgatás”.

Optikai és digitális feloldóképesség



Optikai feloldóképesség:

$$d = \frac{\lambda}{2NA} = \frac{\lambda}{2n \sin \omega}$$

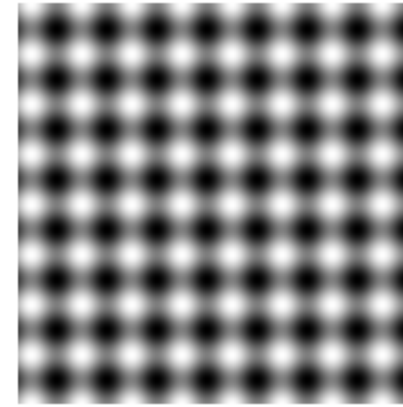
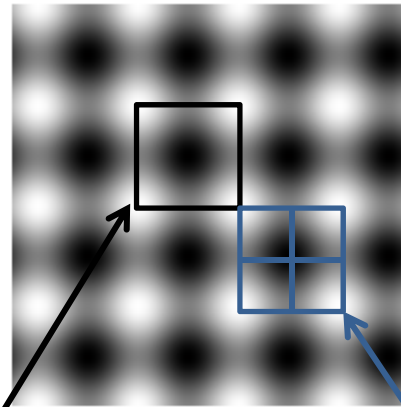
λ – hullámhossz

n – tárgy és objektív közötti közeg törésmutatója

Digitális feloldóképesség:

- Nincs értelme, ha a digitális feloldóképesség jobb, mint az optikai.
- Közelítő szabály: a pixelméretnek kb. ugyanakkorának kell lenni, mint az optikai feloldóképesség.
- Pontos szabály: **Nyquist mintavételezési tétel:** ha egy függvény legmagasabb frekvenciájú komponensének frekvenciája f , akkor a függvényt rekonstruálni lehet $2f$ frekvenciájú mintavételezéssel $\Rightarrow$ ha az optikailag feloldható legkisebb távolság d , akkor a pixelméretnek $d/2$ -nek kell lenni.

d – ezzel az optikai feloldóképességgel a bal oldali periodikus struktúra feloldható, a jobb oldali nem.

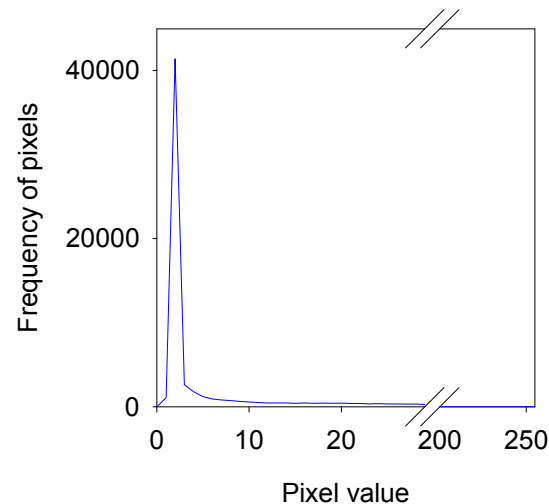
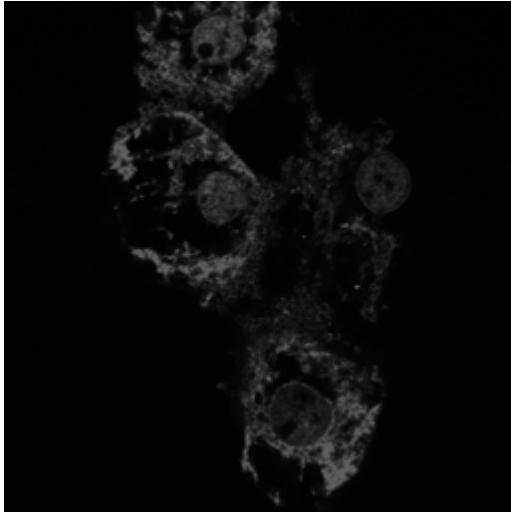


Ha az optikai feloldóképesség d :

Pixelméret a közelítő szabály szerint.

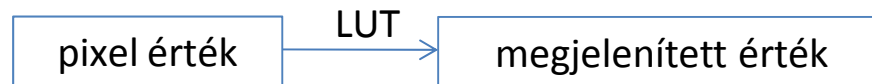
Pixelméret a Nyquist kritérium alapján.

Hisztogram, LUT



A bal oldali hisztogramon a pixelekben **tárolt** intenzitás értékek eloszlása látható.

- A pixelek megjelenített színét (ill. szürkescálás értékét) az ún. LUT (look-up table) határozza meg. A LUT minden tárolt pixel értékhez egy megjelenítendő színt vagy szürkescála értéket rendel.
- A LUT a képben tárolt információt nem befolyásolja, csak a megjelenítés módját.



Szürkescálás (gray scale) képek esetében a megjelenítést általában 256 szintű intenzitás skálán végzik (0 – fekete, 255 – fehér), ui. az emberi szem ennél több intenzitásszintet nem tud megkülönböztetni.

Tárolt pixel értékek:

1	2	3
4	5	6
7	8	9

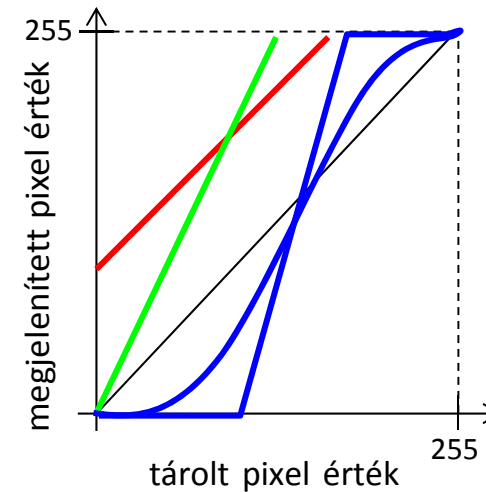
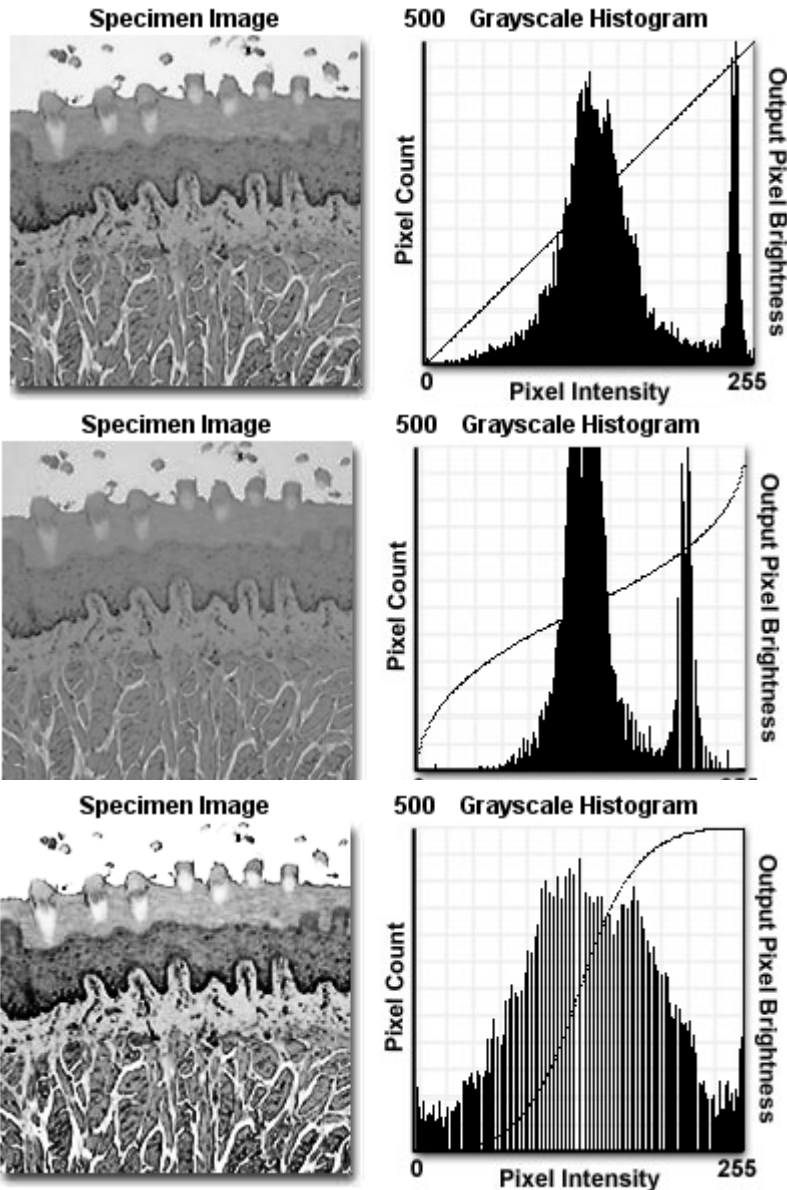


	1	2	3	4	5	6	7	8	9
LUT1	1	2	3	4	5	6	7	8	9
LUT2	20	40	60	80	100	120	140	160	180
LUT3	20	40	60	80	100	255	255	255	255

A LUT hatása

A LUT megadható az ábrázolt pixel értékek és a tárolt pixel intenzitások közötti összefüggés grafikonoszerű megadásával:

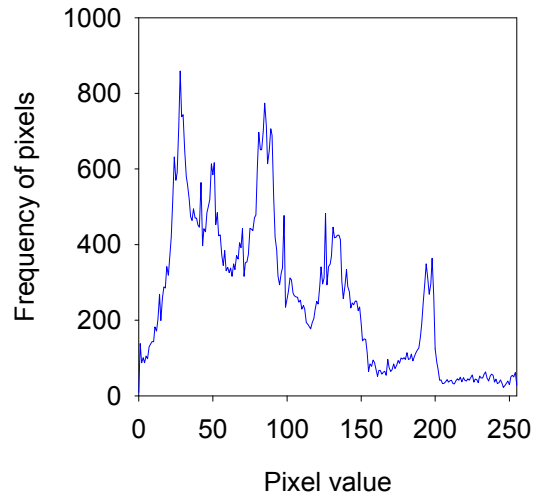
fokozott kontraszt csökkentett kontraszt



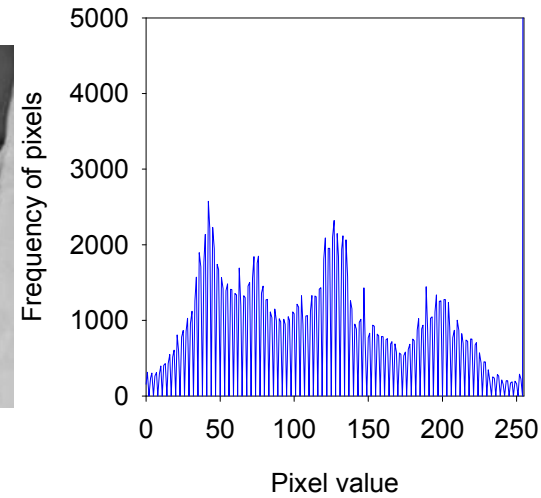
- fokozott fényesség („brightness”): minden pixel ábrázolt értékét növeli.
- fokozott intenzitás: meredekebb összefüggés a megjelenített és a tárolt pixel értékek között. A magas tárolt értékű pixelek ábrázolt értéke jobban nő.
- fokozott kontraszt: a skála közepén levő pixelek megjelenített intenzitáskülönbségét fokozza.

A LUT módosítása Corel PhotoPaint-tel

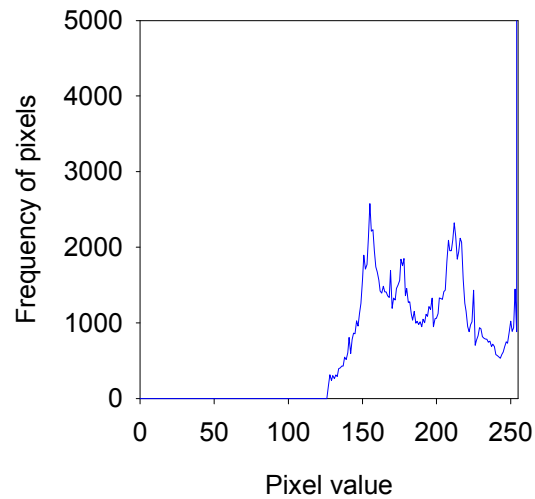
Eredeti



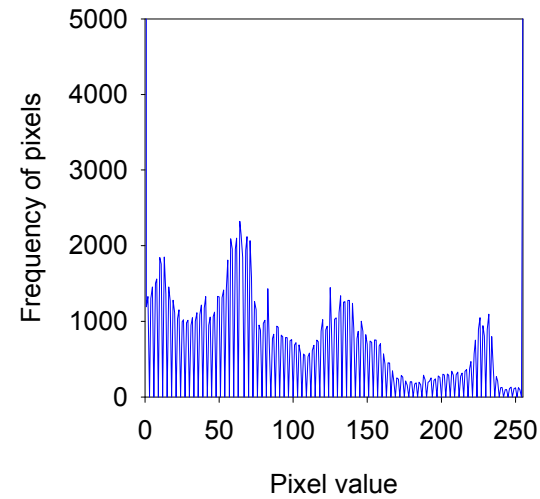
Intensity=50



Brightness=50

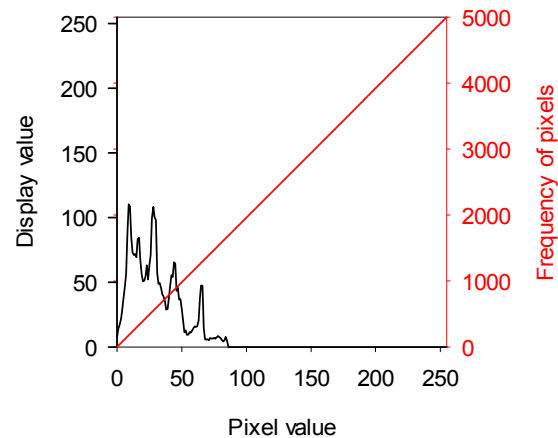
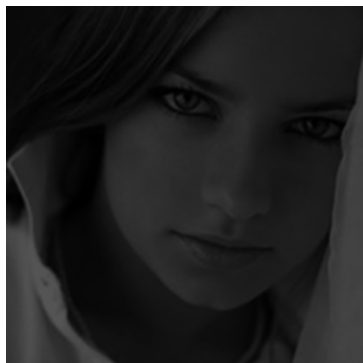


Contrast=50

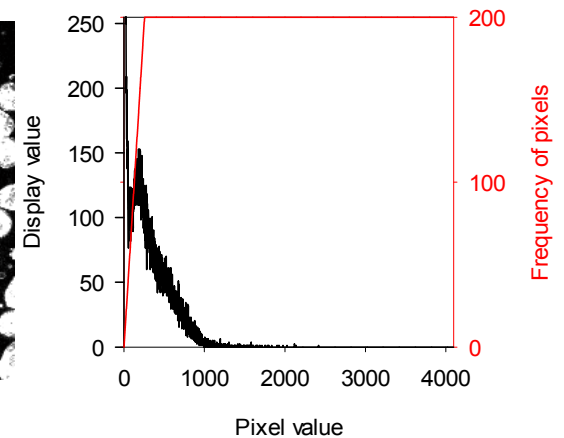
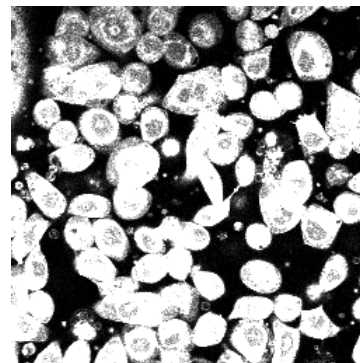


A LUT hatása

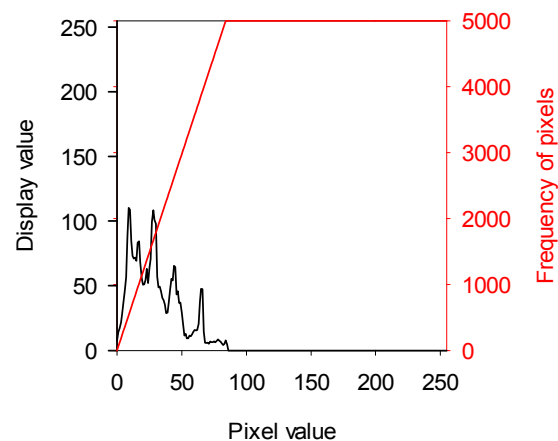
Eredeti:



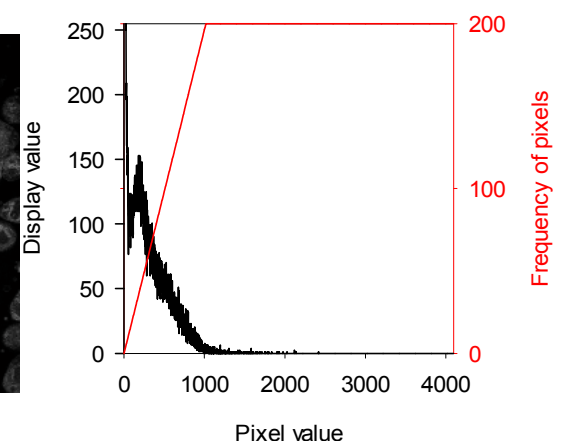
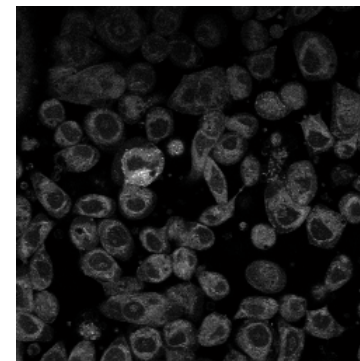
Eredeti (12 biten tárolt kép):



Linear stretch:



Linear stretch:



„Linear stretch” esetében a legalacsonyabb és legmagasabb ábrázolt értékeket a legalacsonyabb és legmagasabb tárolt pixel értékhez rendeljük, és közöttük az ábrázolt értékeket lineárisan szét húzzuk.

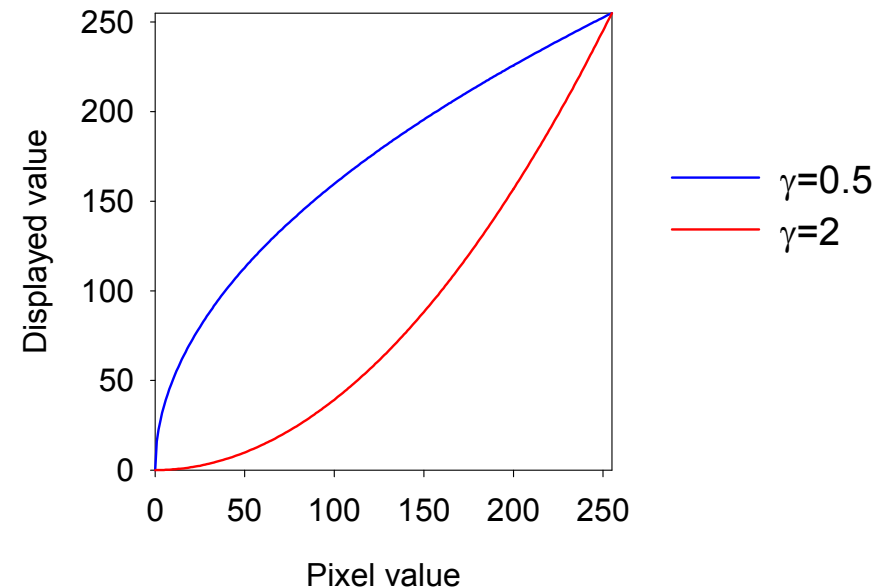
Gamma korrekció

$$Pixel_{display} = const \cdot Pixel_{stored}^{\gamma}$$

Hatvány függvény szerint rendeli egymáshoz a tárolt és ábrázolt pixel értékeket.

$\gamma < 1 \rightarrow$ gamma kompresszió

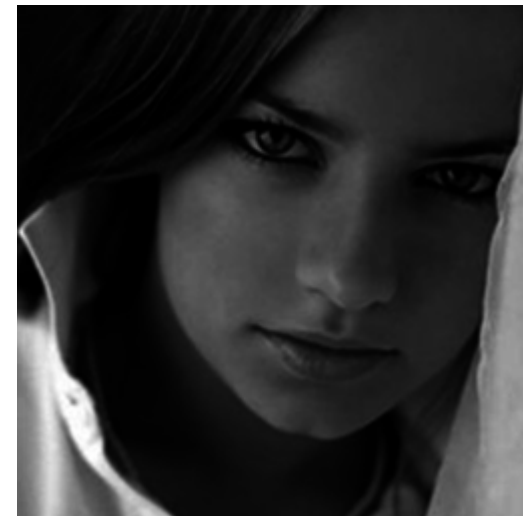
$\gamma > 1 \rightarrow$ gamma expanzió



eredeti

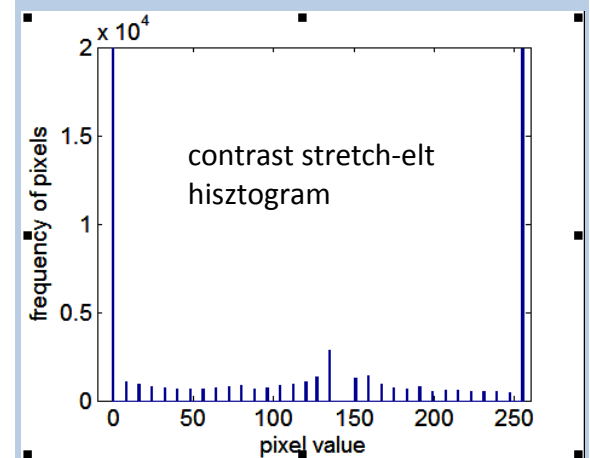
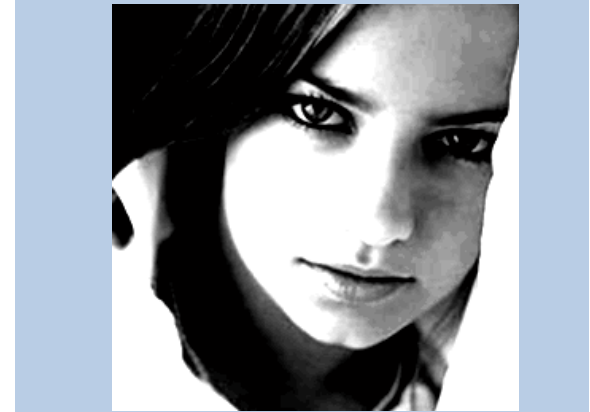
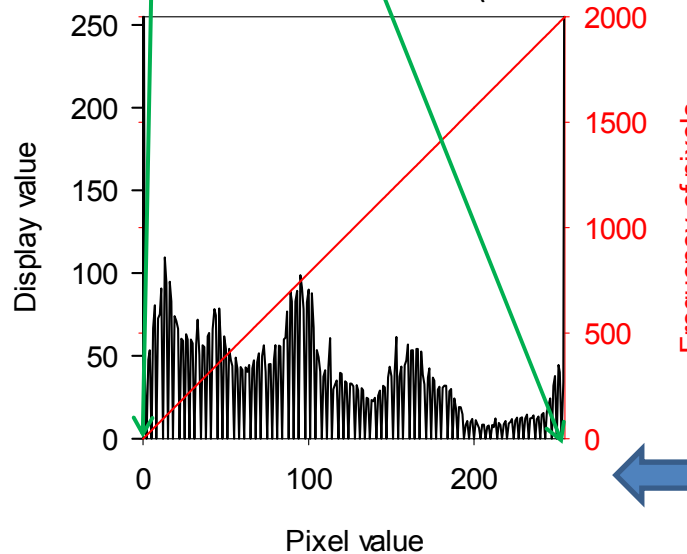
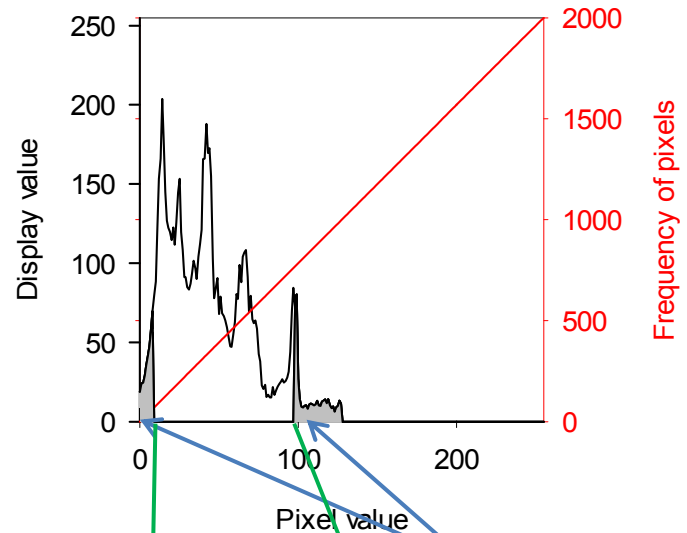
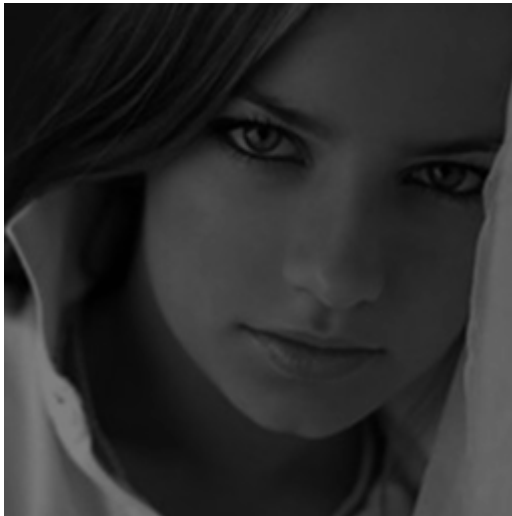


$\gamma=0.5$



$\gamma=2$

Contrast stretching: a kontraszt fokozása



Contrast stretch az alsó és felső 30% 0-hoz és 255-höz való kihúzásával → a pixelek jelentős része fekete vagy fehér lesz.

A 5%-os és 95%-os percentilis értékekhez 0-t és 255-t rendel, a köztes értékeket egyenletesen elosztja.

Contrast stretch-elni lehet a megjelenített értékeket és a tárolt pixel értékeket (pl. a fenti esetben). A tárolt pixel értékek módosítása csak a kép kiértékelése után célszerű a megjelenítés optimalizálásához.

Hisztogram kiegyenlítés (histogram equalization)

Egy olyan algoritmus, amely biztosítja, hogy a minden intenzitás a képen ugyanolyan mértékben reprezentált (ugyanolyan relatív frekvenciával fordul elő).

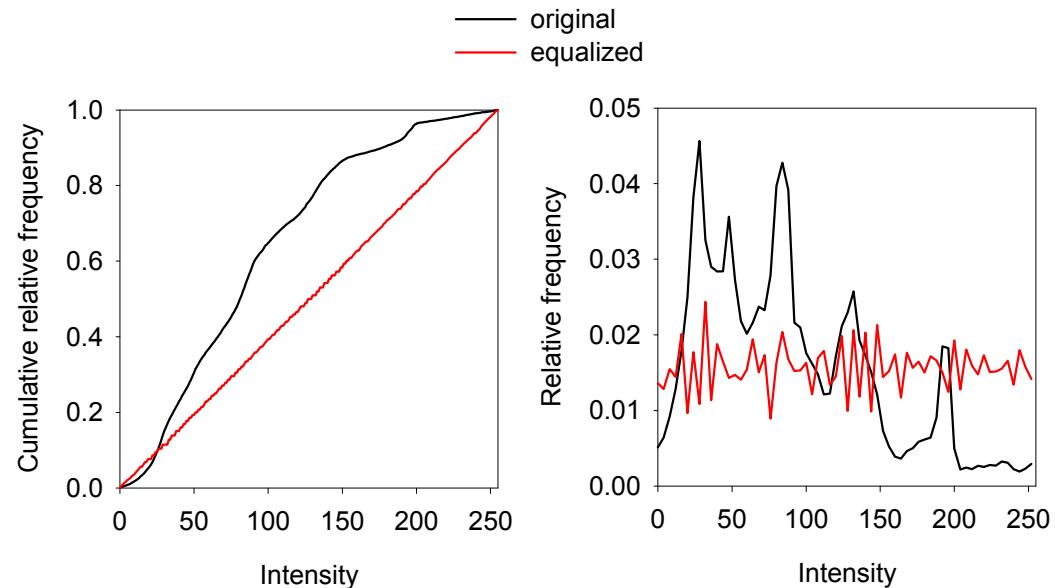
A képen $0, 1, \dots, I_{\max}$ intenzitások fordulnak elő, és $p(f)$ az intenzitás eloszlásfüggvénye:

$$p(f) = P(I \leq f)$$

Ha az intenzitásokat a

$$I = I_{\max} p(f)$$

tüntetjük fel, akkor az I -nél kisebb-egyenlő intenzitások aránya $I/I_{\max}$ lesz.



Filterek

- Képek kiértékelése és zajszűrése során alkalmazott olyan eszköz, amely egy adott pixel új értékét a pixel és környezetének intenzitás értékeiből egy adott függvény szerint számolja ki.

- Filterek szemléltetése:

- o képlettel:

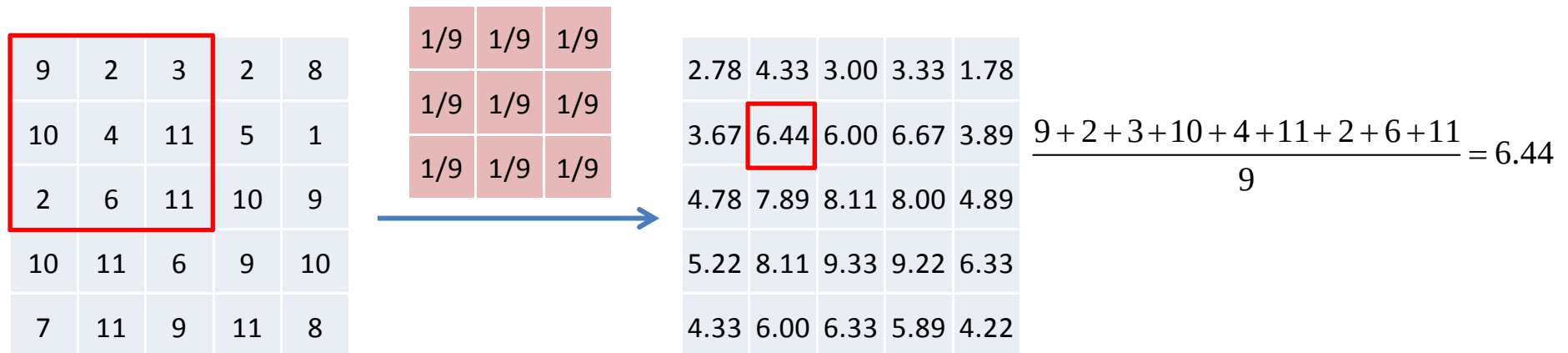
$$g_{x,y} = \sum_{k=-m}^m \sum_{l=-m}^m w_{k,l} f_{x+k,y+l}$$

$g_{x,y}$ – a pixel számított új értéke

$f_{x+k,y+l}$ – a pixel és környezetének eredeti értékei

$w_{k,l}$ – súlyfaktor

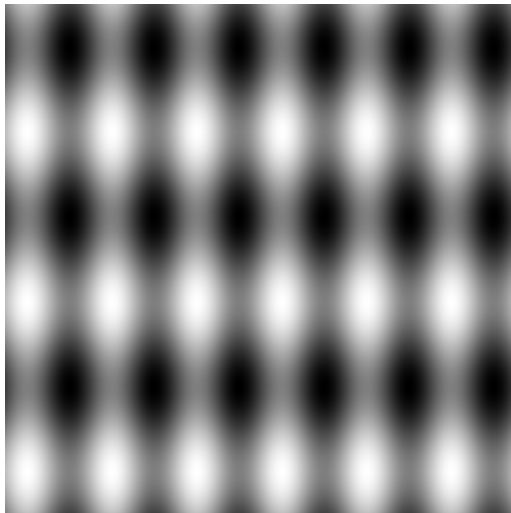
- o a filter kirajzolásával:



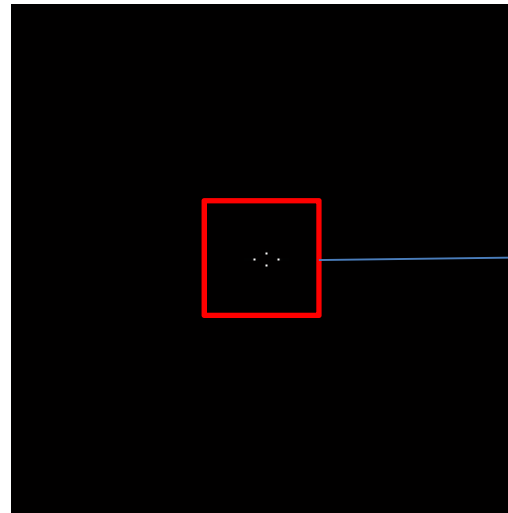
- A fenti filter mérete 3×3, de lehet 5×5, 7×7, stb.
- Léteznek filterek simításra (pl. a fenti), határvonal detektálására, gradiens meghatározására, stb.

Fourier és inverz Fourier transzformáció

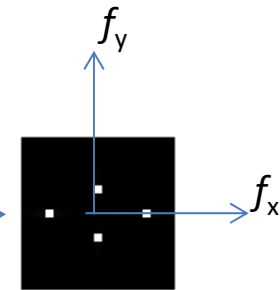
- Az előző ábrán a filterek térbeli doménben történő alkalmazása került bemutatásra.
- A képek hagyományos, térbeli doménben történő ábrázolása azt mutatja, hogy a pixelek a tér adott pontjában milyen intenzitásúak.
- Megadható, hogy a képen található struktúrák milyen térbeli frekvenciával jellemezhetők, tehát a kép átkonvertálható a frekvencia doménbe.
- Ezt az eljárást Fourier transzformációnak nevezik:



Eredeti kép (f)



Fourier transzformált kép (F)



$$F_{k,l} = \sum_{x=1}^n \sum_{y=1}^n f_{x,y} e^{-2\pi i \frac{kx+ly}{n}} \quad a_{k,l} = \sum_{x=1}^n \sum_{y=1}^n f_{x,y} \cos\left(2\pi \frac{kx+ly}{n}\right), \quad b_{k,l} = -\sum_{x=1}^n \sum_{y=1}^n f_{x,y} \sin\left(2\pi \frac{kx+ly}{n}\right)$$

- Az eredeti kép inverz Fourier transzformációval előállítható a frekvenciákból:

$$f_{x,y} = \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n F_{k,l} e^{2\pi i \frac{kx+ly}{n}} \quad f_{x,y} = \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n \left(a_{k,l} \cos\left(2\pi \frac{kx+ly}{n}\right) - b_{k,l} \sin\left(2\pi \frac{kx+ly}{n}\right) \right)$$

Filterek a térbeli és frekvencia doménben

A filterek alkalmazhatók a frekvencia doménben is:

$$b = f * a$$

b – a filter által adott kép
 f – a filter
 a – az eredeti kép
 $*$ - a filter alkalmazása (konvolúció).

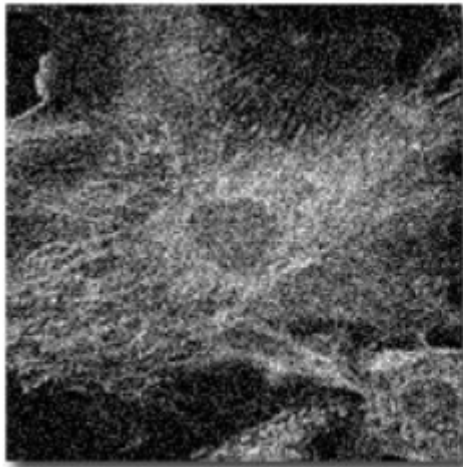
$$B = F \cdot A$$

B – b Fourier transzformáltja
 F – a filter Fourier transzformáltja
 A – az eredeti kép Fourier transzformáltja
 $\cdot$ - szorzás

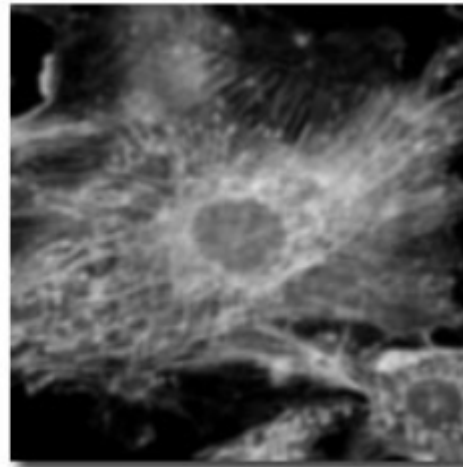
$$b = \text{invFT}(B)$$

Sok filter esetében a frekvencia doménben történő alkalmazás a számítás gyorsasága szempontjából előnyösebb.

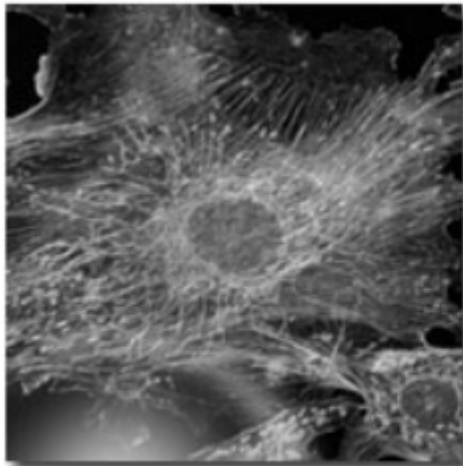
A képet degradáló tényezők digitális képalkotás során



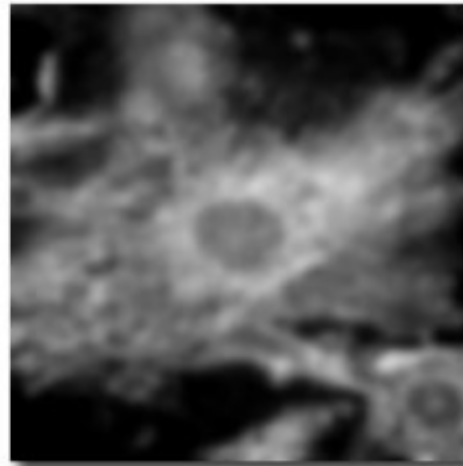
(a)



(b)



(c)



(d)

- a. zaj (noise)
- b. fényszórás (scatter)
- c. „ragyogás” (glare)
- d. elkenődés (blur)

ad a:

A zajnak két forrása van:

- foton detektálás statisztikus volta (Poisson eloszlás)
- a képalkotó rendszer (normális eloszlás)

ad b:

- A mintában jön létre.
- Minél vastagabb a minta, annál jelentősebb.

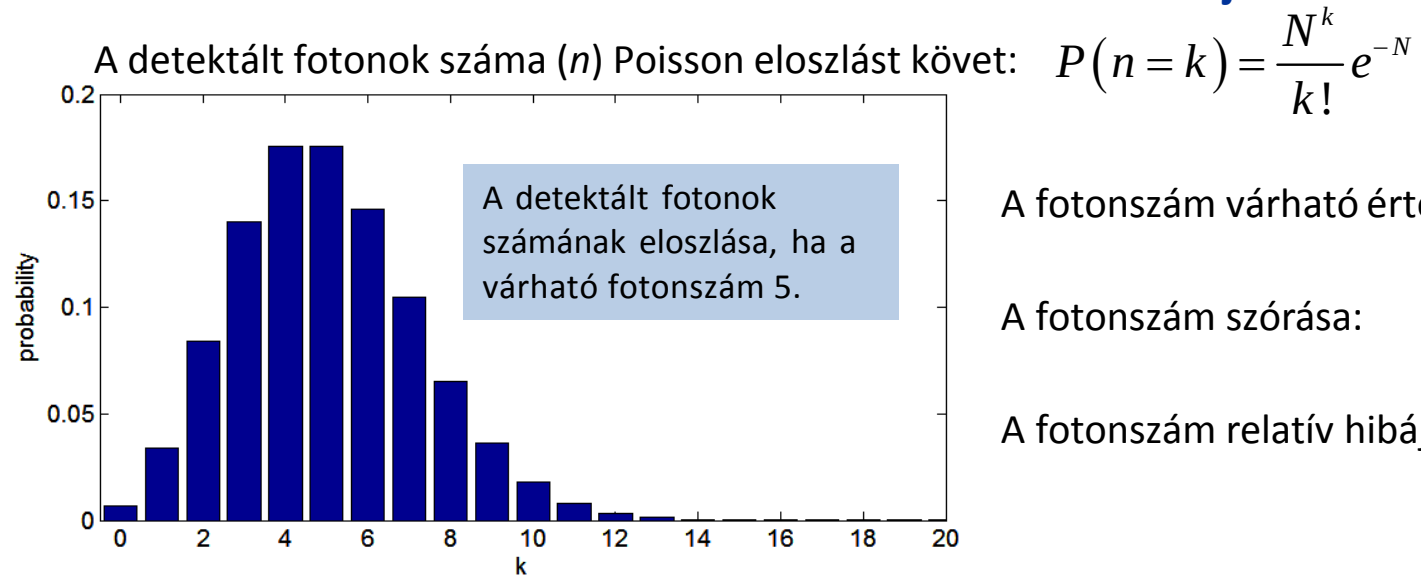
ad c:

- A mikroszkóp optikai rendszerében bekövetkező fényszórás és reflexió következménye.
- Modern mikroszkópokban általában minimális (ha jól be vannak állítva).

ad d:

- A fény hullámtermészete miatt a képalkotás során bekövetkező diffrakció következménye.
- A PSF-fel („point spread function”) lehet jellemezni.

Foton detektálás miatt keletkező zaj



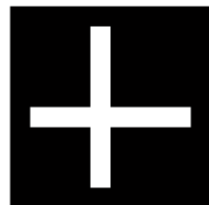
A fotonszám várható értéke: N

A fotonszám szórása: $\sqrt{N}$

A fotonszám relatív hibája: $\frac{\sqrt{N}}{N} = \frac{1}{\sqrt{N}}$

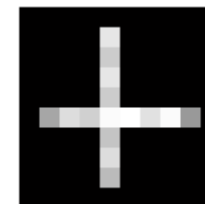
Minél nagyobb a várt fotonszám, annál kisebb a relatív hiba.

$N=5$



A képek relatív skálán vannak ábrázolva („linear stretch”), ezért tűnik a két bal oldali kép egyenlő fényességűnek.

$N=50$



Eredeti kép (zaj nélkül)

Poisson zaj hozzáadásával

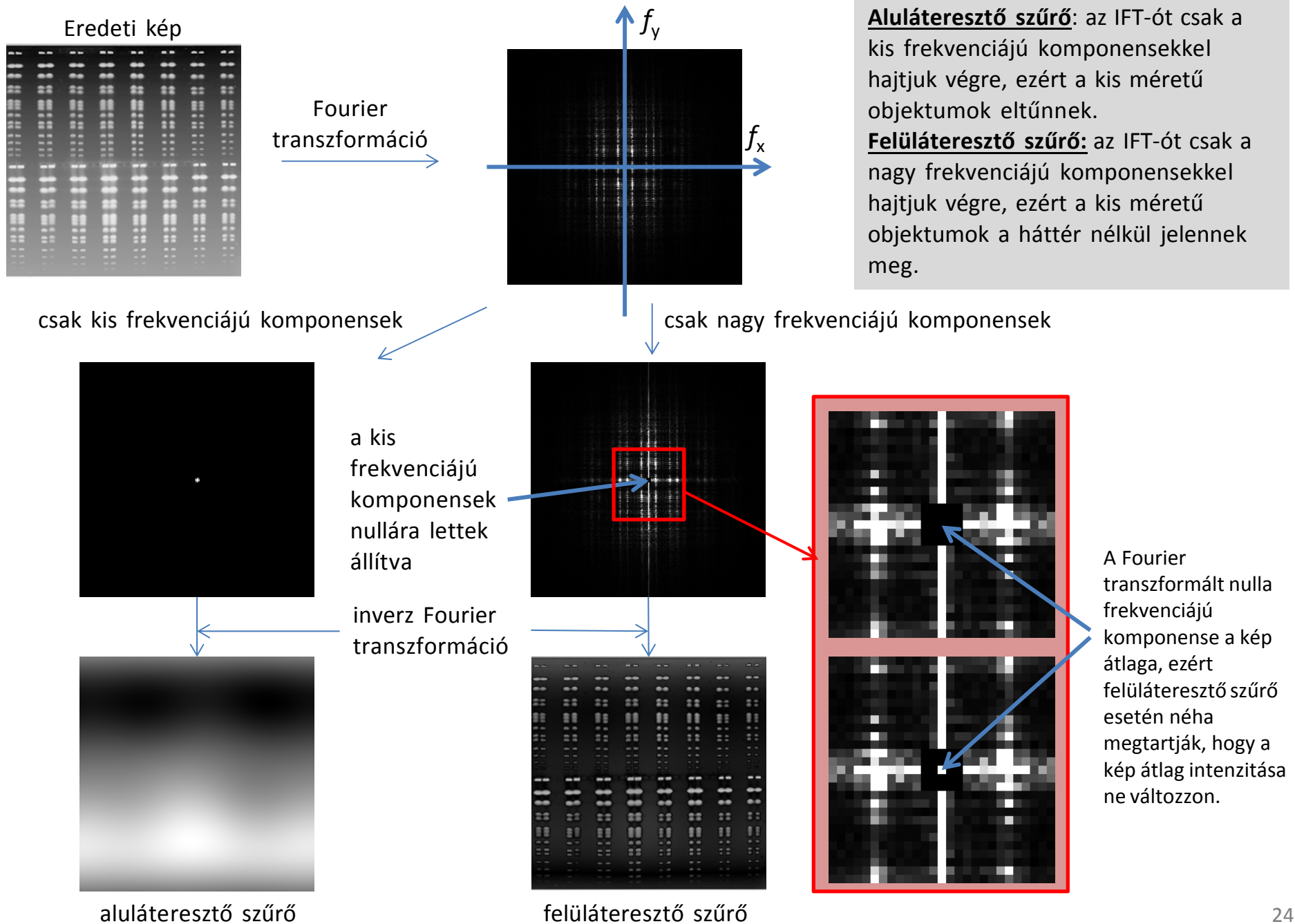
Kép visszaállítás („image restoration”)

- Mikroszkópos képek felvétele során gyakran jelentős mennyiségű zaj adódik a jelhez.
- A kép visszaállítás célja a zaj elnyomása
 - a tárgy pontosabb reprezentációja
 - a kép szebb megjelenése érdekében.
- A kép minőségét leggyakrabban a következő két tényező rontja:
 - zaj (noise) → hatásának csökkentése a kép visszaállító algoritmusokkal (zajszűrés)
 - elkenődés (blur) → hatásának csökkentése dekonvolúcióval

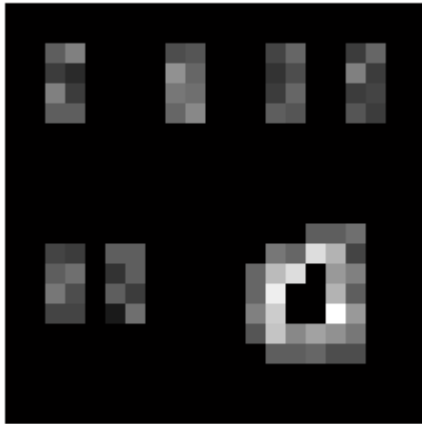
Kép visszaállítás (zajszűrés)

- A zaj statisztikus modellezésével elérhető a zaj eltávolítása, de ezt ritkábban alkalmazzák.
- A leggyakrabban alkalmazott algoritmusok alapelve:
 - *zaj miatti relatív hiba* $\sim \frac{1}{\sqrt{N}}$, N - átlagos fotonszám
 - Tehát ha az új pixel értékeket több pixel felhasználásával számoljuk ki
 - csökken a zaj relatív mértéke, mert nő azon fotonok száma, melyekből az intenzitást kiszámoljuk.
 - de csökken az effektív feloldóképesség is, hiszen egy pixel intenzitását több, a környezetében levő pixel befolyásolja.

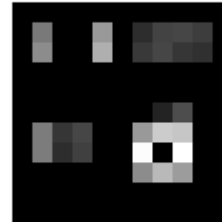
Alul- és felüláteresztő szűrők



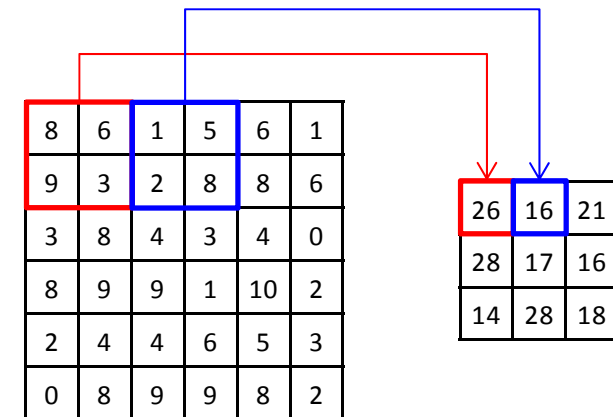
Kép visszaállítás, zajszűrés: pixel pooling



Eredeti kép

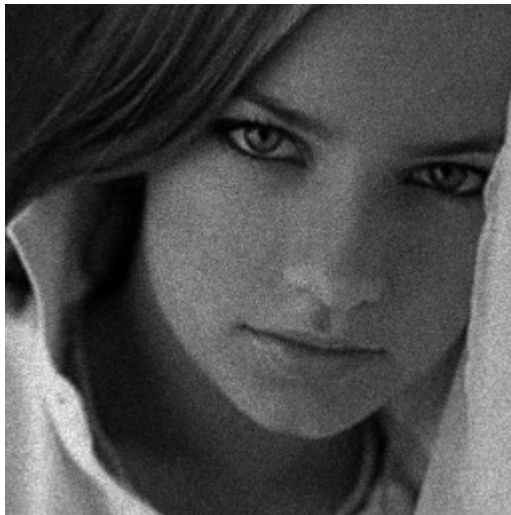


Simított kép



Eredeti kép

Simított kép



Eredeti kép



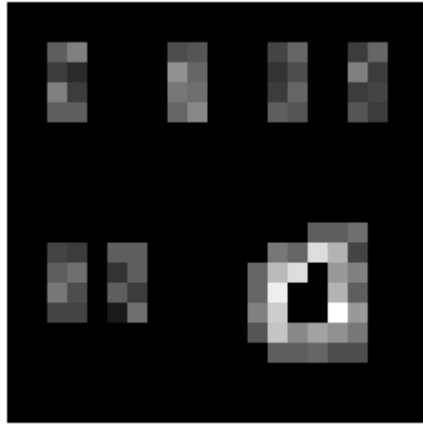
Simított kép

- Az eljárás során az eredeti kép minden 2×2 -es blokkját összegezzük és az összeget írjuk a simított képbe.
- A simított kép ennek megfelelően fele akkora lesz, mint az eredeti.
- A blokkméret természetesen lehet más is.

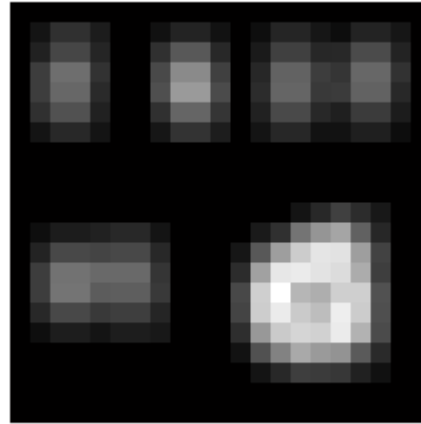
Kép visszaállítás, zajszűrés: átlag szűrő

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

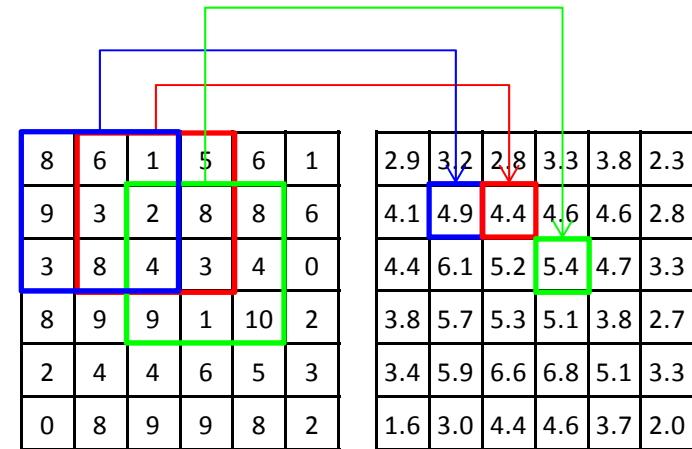
A 3×3-as átlagolást megvalósító szűrő.



Eredeti kép

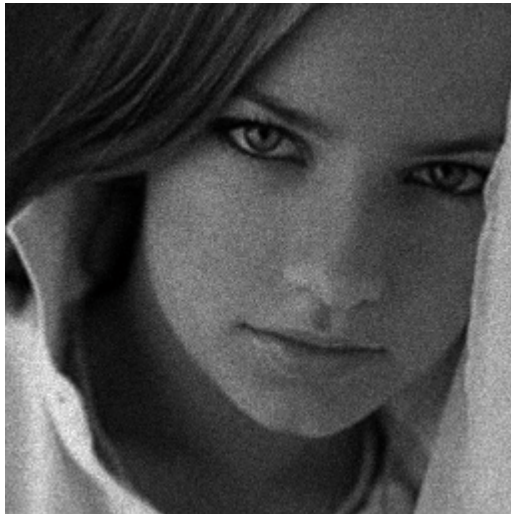


Simított kép

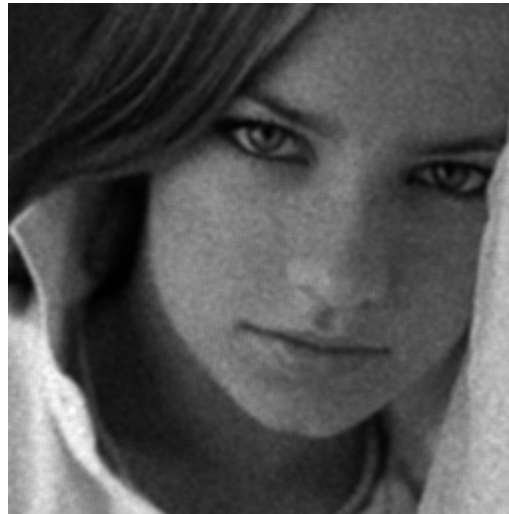


Eredeti kép

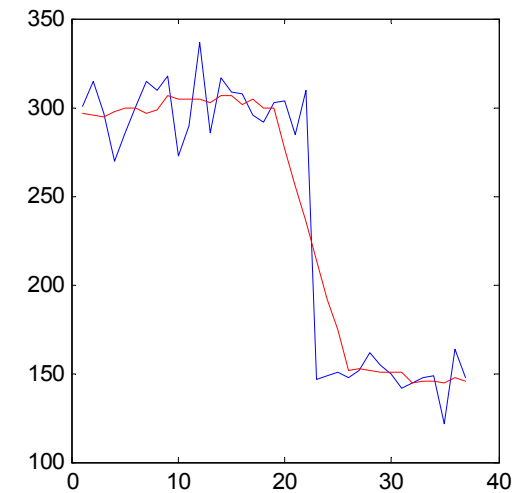
Simított kép



Eredeti kép



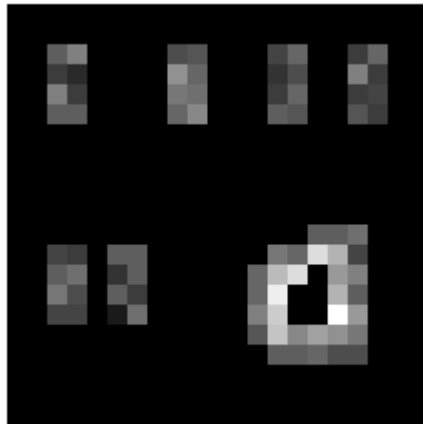
Simított kép



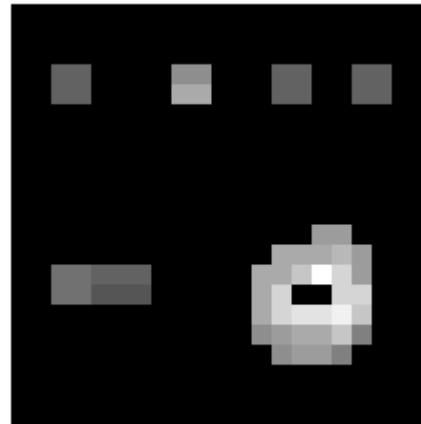
Eredeti, átlagolt (7-es ablak méret)

- Minden pixelt helyettesít a környező pixelek (az ábrákon 3×3 pixel) átlagával.
- A simított kép mérete (a pixel pooling-gal ellentétben) megegyezik az eredeti kép méretével.
- Hátránya: a határvonalakat is kisimítja, elnyomja.

Kép visszaállítás, zajszűrés: medián szűrő

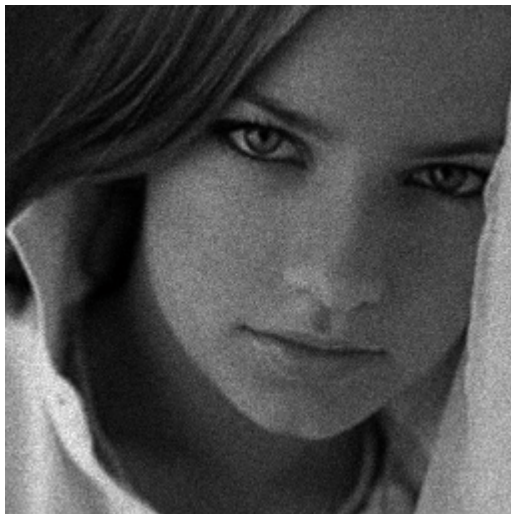
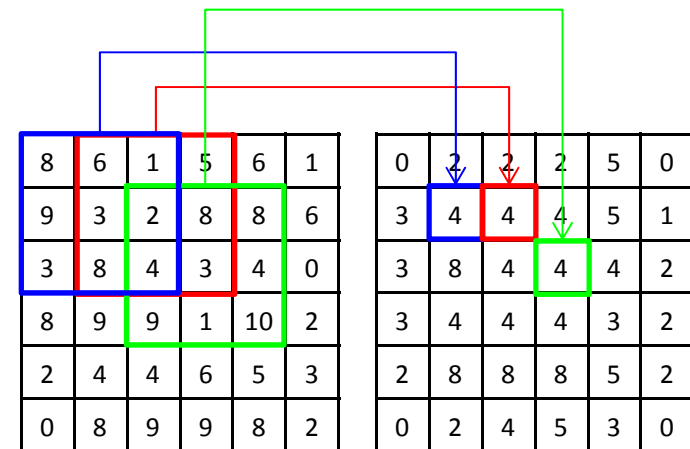


Eredeti kép



Simított kép

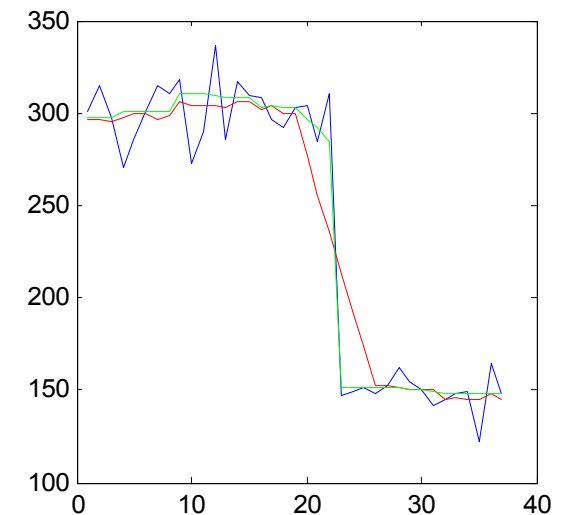
Az adott 9 elemből álló minta mediánja.



Eredeti kép



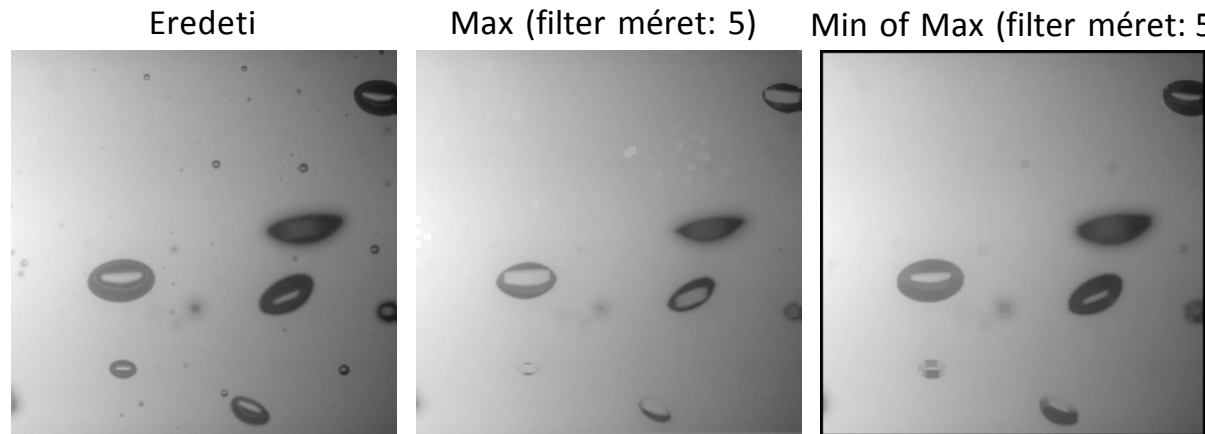
Simított kép



Eredeti, átlag szűrő (7-es ablakméret),
medián szűrő (7-es ablakméret)

- Minden pixelt helyettesít a környező pixelek (az ábrákon 3×3 pixel) mediánjával.
- A határvonalakat sokkal jobban megőrzi, mint az átlag szűrő.

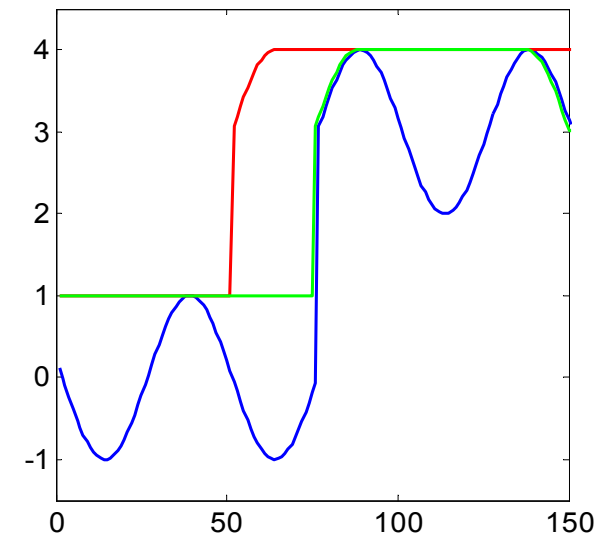
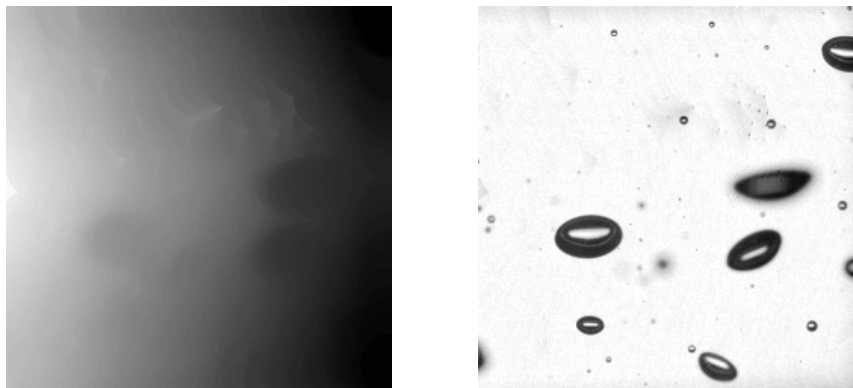
Kép visszaállítás, zajszűrés: max-min (min of max) szűrő



1. Maximum szűrés: egy adott nagyságú környezet maximumát rakja minden pixelbe (ezzel eltávolítja az adott nagyságnál kisebb méretű objektumokat).
2. Az előbb lépés kimenetén egy minimum szűrést hajt végre (ezzel visszahelyezi a határvonalakat az eredeti pozíciójukba).

A max-min szűrő háttérkivonásra is alkalmazható:

Min of Max (filter méret: 50) Eredeti – minOfMax(50)



Eredeti szignál

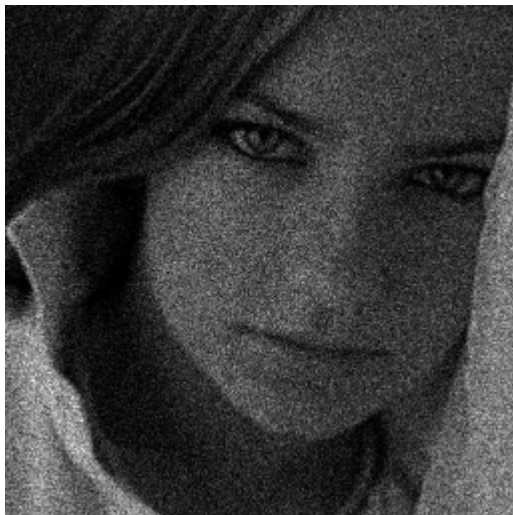
Max

Min of Max

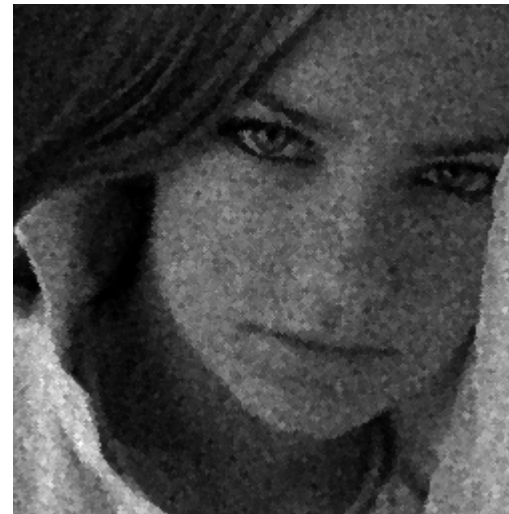
Kép visszaállítás, zajszűrés: Kuwahara filter

20	23	27	19	20
18	25	23	21	27
167	161	23	19	28
171	165	170	31	55
174	167	163	25	23

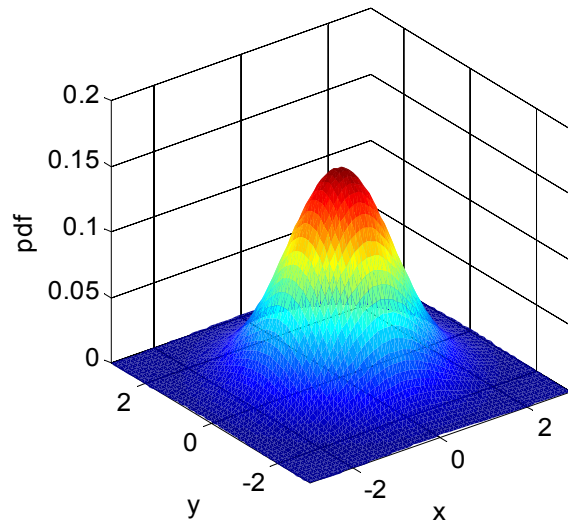
A legkisebb varianciával
rendelkező al-ablak átlaga.



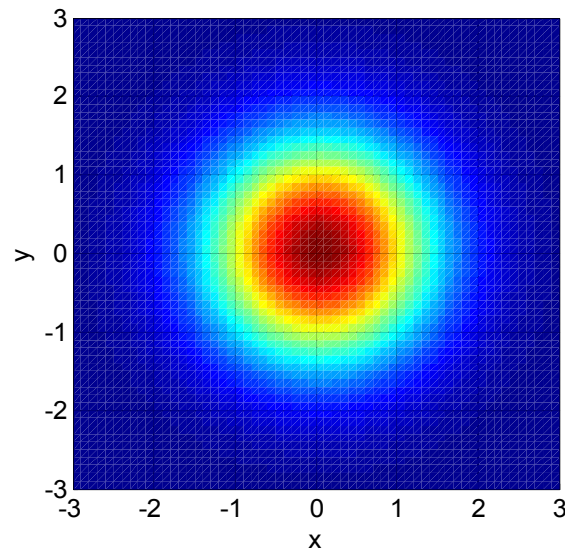
Kuwahara



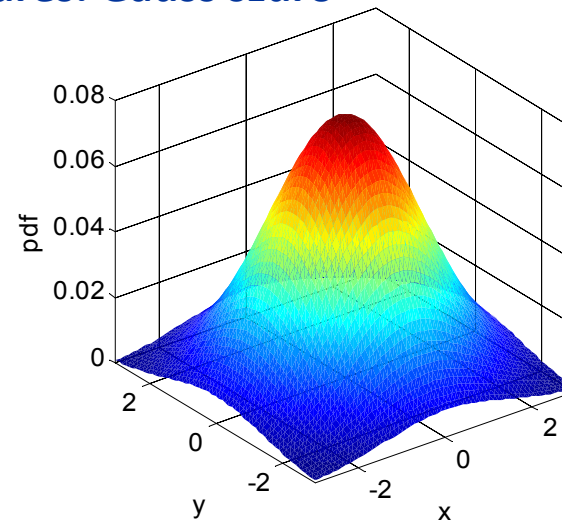
Kép visszaállítás, zajszűrés: Gauss szűrő



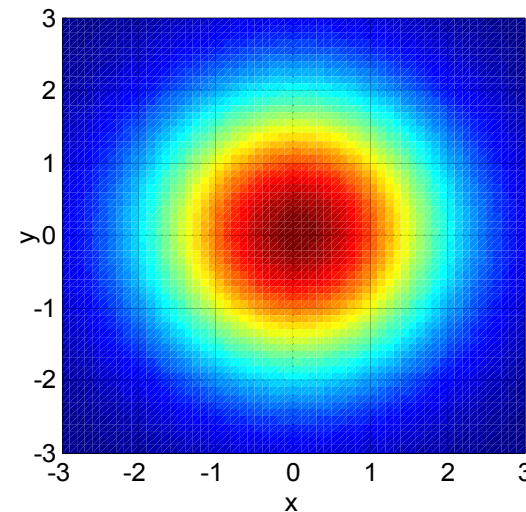
$$\text{COV} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



0.075	0.124	0.075
0.124	0.204	0.124
0.075	0.124	0.075



$$\text{COV} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

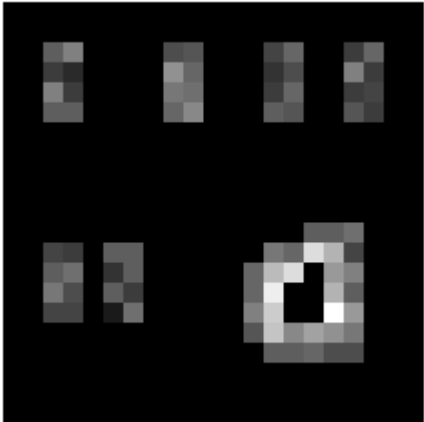


0.102	0.115	0.102
0.115	0.131	0.115
0.102	0.115	0.102

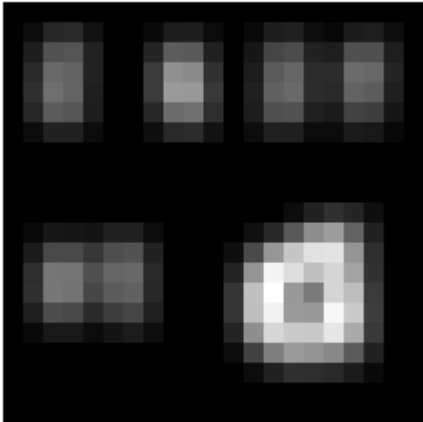
A Gauss függvények digitalizált változata, amelyet filterként használunk.

- A Gauss filtereknek két választható paraméterük van:
 - a filter mérete (hány pixel széles)
 - a normál eloszlás szórása
- Mivel a Gauss függvény Fourier transzformáltja is Gauss függvény, a Gauss szűrés aluláteresztő szűrőnek felel meg.

Kép visszaállítás, zajszűrés: Gauss szűrő



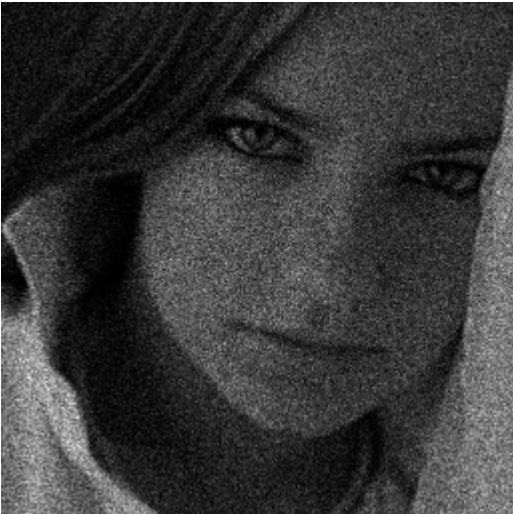
Eredeti kép



Simított kép

$$\text{COV} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

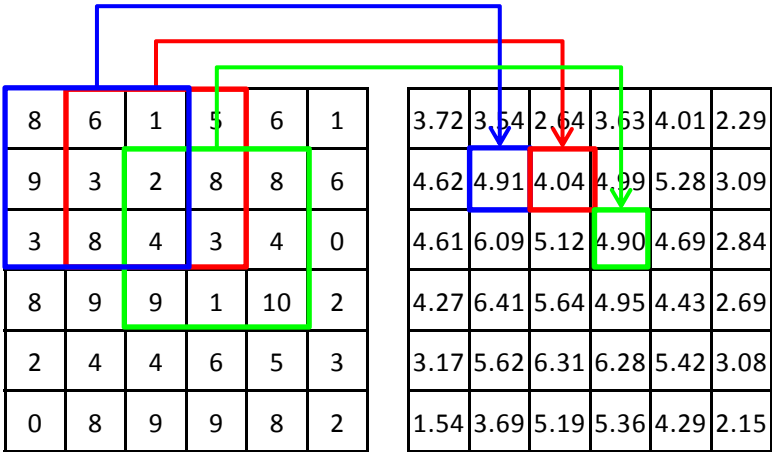
0.075	0.124	0.075
0.124	0.204	0.124
0.075	0.124	0.075



Eredeti kép

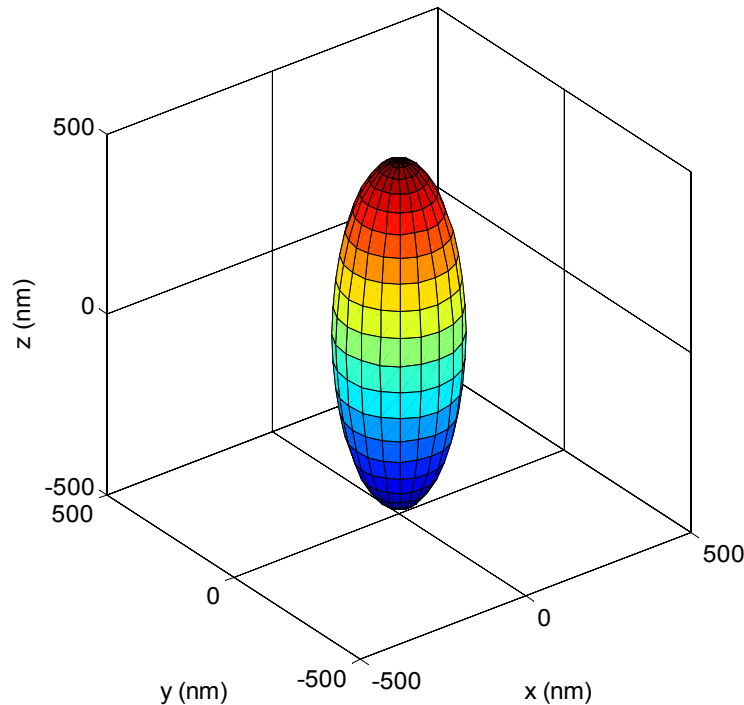


Simított kép



A „point spread function” (PSF)

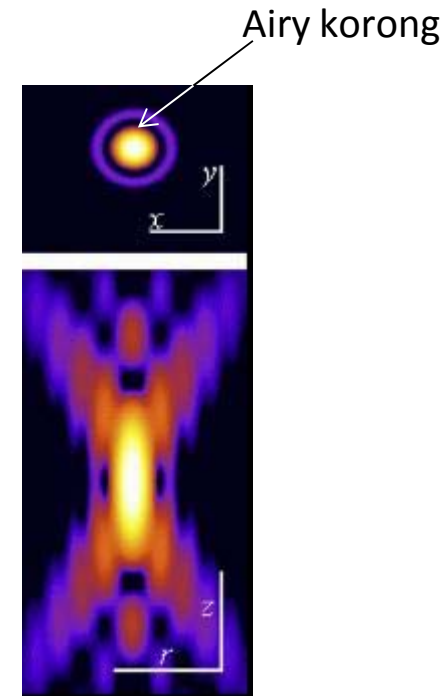
- Egy pontszerű fényforrás képe a fény hullámtermészete és az emittált fotonok csak egy részének begyűjtése miatt tökéletesen működő optikai mikroszkóp esetében sem egy pont.
- A pontszerű fényforrás fentiek miatt kialakuló kiterjedt képét PSF-nek („point spread function”) nevezzük.



A PSF három dimenzióban

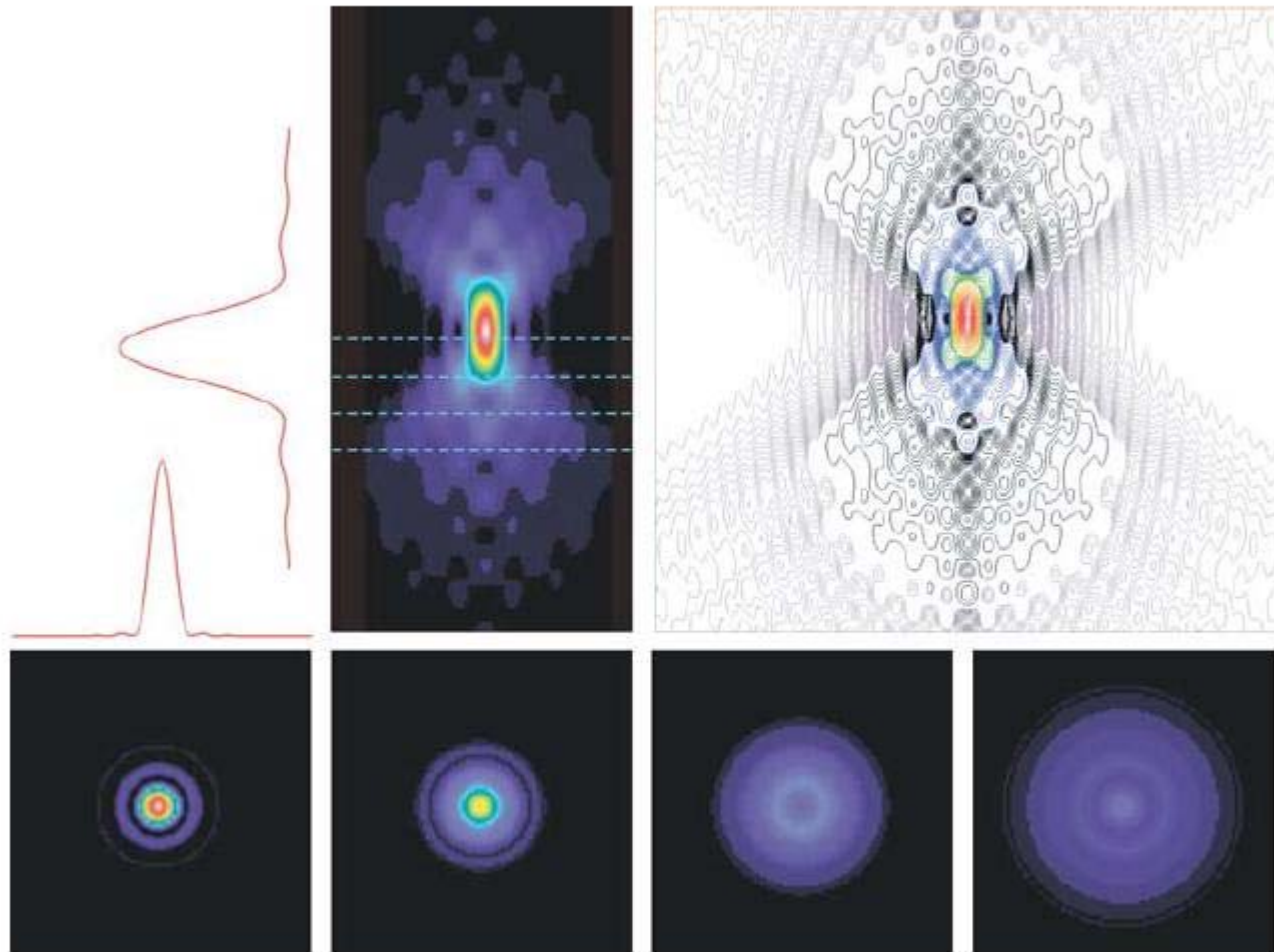
$$d_{lateral} = \frac{0.4\lambda}{NA} \quad d_{axial} = \frac{1.4\lambda n}{NA^2}$$

λ – hullámhossz, NA – numerikus apertúra, n – a tárgy és az objektív közötti közeg törésmutatója



A PSF két dimenziós vetületei

A PSF három dimenzióban

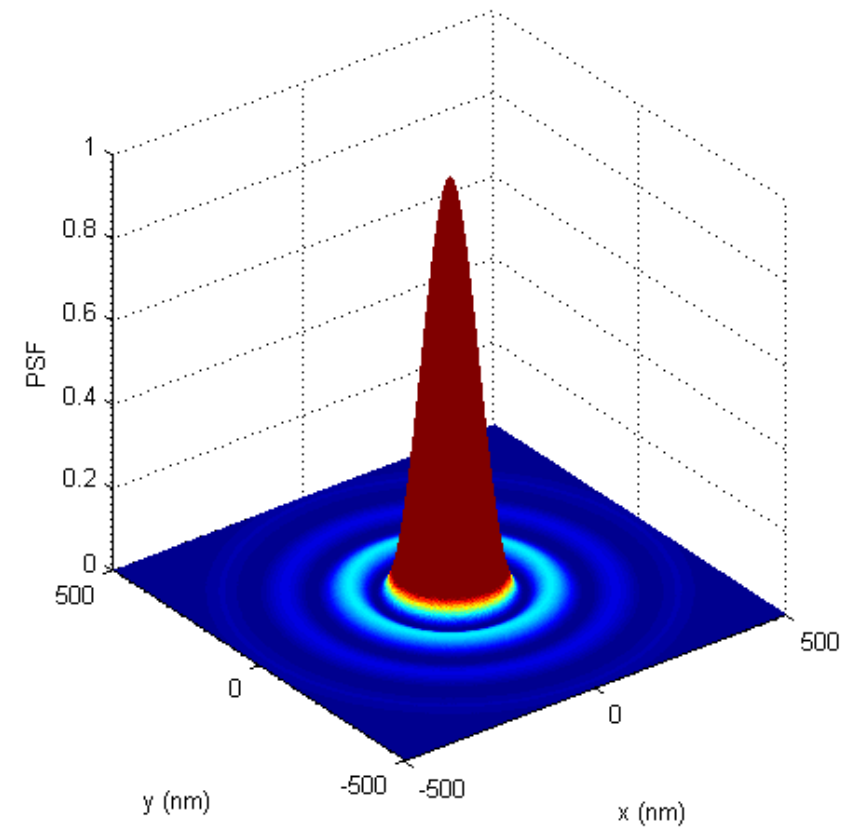
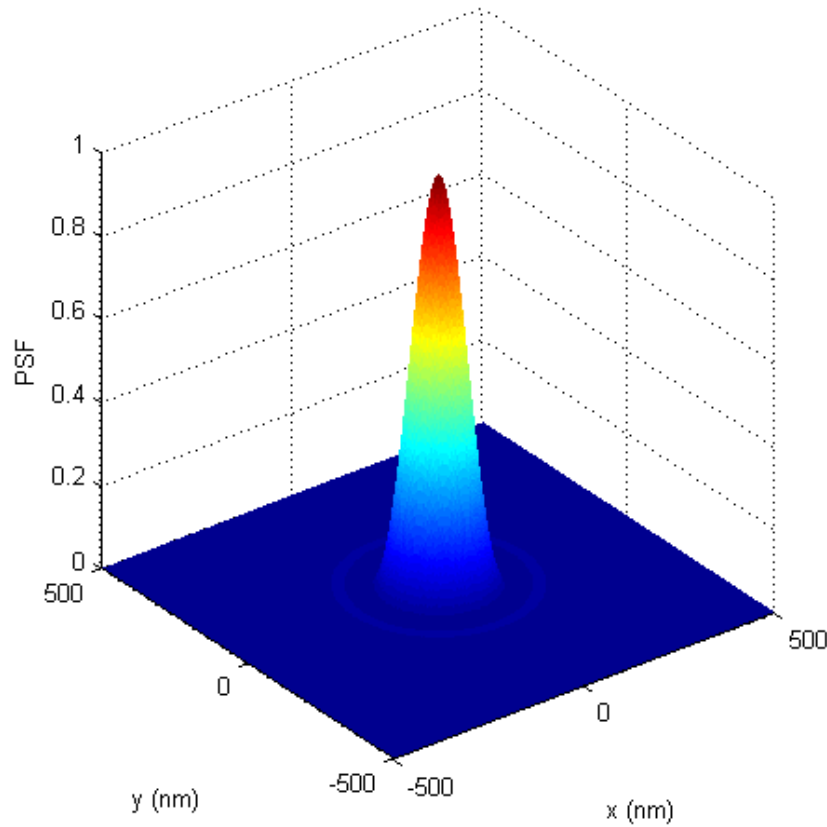


A PSF

A PSF egy x-y síkban a következő Bessel függvénnyel írható le:

$$PSF = \left(\frac{2J_1\left(\frac{2\pi}{\lambda} r \tan \omega\right)}{\frac{2\pi}{\lambda} r \tan \omega} \right)^2$$

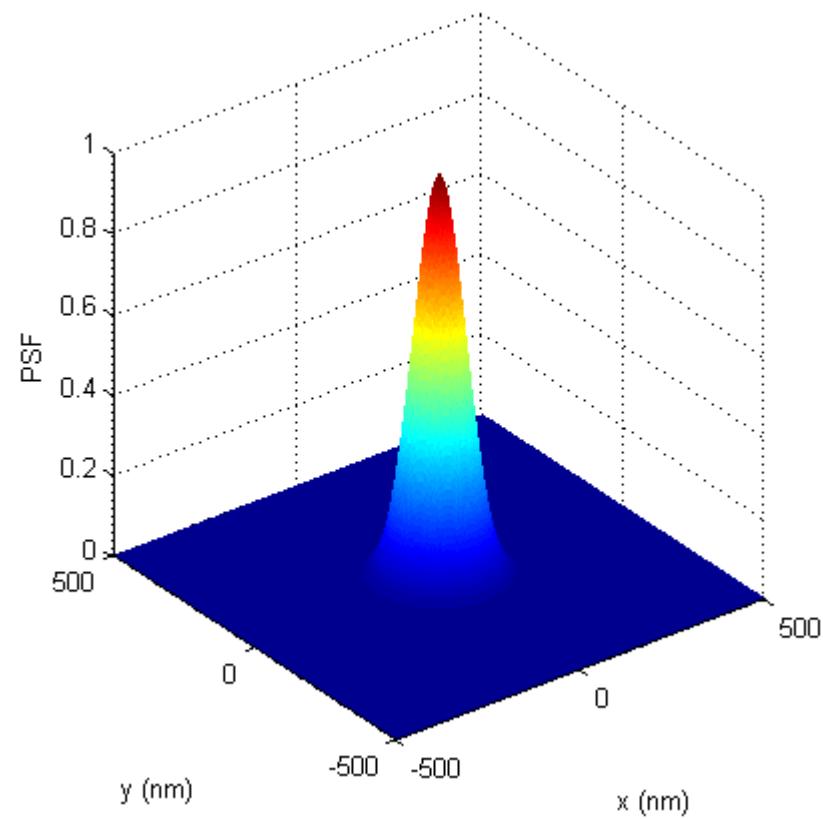
λ – hullámhossz
 r – távolság az optikai tengelytől
 ω – az objektív fél nyílásszöge
 J_1 – Bessel függvény (első fajú, első rendű)



A PSF

A PSF centrális maximuma (Airy korong) egy x-y síkban jól közelíthető Gauss függvénnyel:

$$PSF = e^{-2.5 \left(\frac{2}{\lambda} r \tan \omega \right)^2}$$



Az optikai feloldás értelmezése konvolúcióval

Tárgy								PSF (konvolúciós kernel)			Kép									
3	9	4	9	8	6	6	7	0.000	0.167	0.000	0.000	0.500	1.500	0.667	1.500	1.333	1.000	1.000	1.167	0.000
8	6	1	9	4	1	7	2	0.167	0.333	0.167	0.500	3.833	5.167	4.500	6.500	5.833	4.500	5.333	3.667	1.167
4	5	2	5	2	2	6	4	0.000	0.167	0.000	1.333	4.833	5.833	3.833	6.167	4.667	3.500	4.833	3.667	0.333
9	1	1	5	4	4	5	6				0.667	5.000	3.833	2.667	4.667	3.167	2.833	5.000	3.667	0.667
2	9	2	3	1	8	5	8				1.500	4.167	4.333	2.000	3.833	3.333	4.500	5.167	4.833	1.000
3	6	2	9	1	0	3	1				0.333	4.167	4.833	3.167	3.833	3.000	4.333	5.667	4.667	1.333
1	4	4	4	9	0	7	9				0.500	2.500	5.000	4.167	4.667	3.500	2.000	3.167	3.667	0.167
1	5	0	1	10	2	2	8				0.167	1.667	4.000	3.000	5.167	5.500	3.000	4.667	5.667	1.500
											0.167	1.333	2.500	1.667	2.667	5.333	2.667	3.500	4.500	1.333
											0.000	0.167	0.833	0.000	0.167	1.667	0.333	0.333	1.333	0.000

- A kép a tárgy PSF-fel való konvolúciójaként jön létre.
- Az 5 értékű pixel képe a saját és szomszéd pixeleinek hozzájárulásával jön létre:

<table border="1"> <tr><td>9</td><td>4</td><td>9</td><td>8</td><td>6</td></tr> <tr><td>6</td><td>1</td><td>9</td><td>4</td><td>1</td></tr> <tr><td>5</td><td>2</td><td style="background-color: #f8d7da;"></td><td>2</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>5</td><td>4</td><td>4</td></tr> <tr><td>9</td><td>2</td><td>3</td><td>1</td><td>8</td></tr> </table>	9	4	9	8	6	6	1	9	4	1	5	2		2	2	1	1	5	4	4	9	2	3	1	8	<table border="1"> <tr><td>9</td><td>4</td><td>9</td><td>8</td><td>6</td></tr> <tr><td>6</td><td>1</td><td>9</td><td>4</td><td>1</td></tr> <tr><td>5</td><td>2</td><td style="background-color: #f8d7da;"></td><td>2</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>5</td><td>4</td><td>4</td></tr> <tr><td>9</td><td>2</td><td>3</td><td>1</td><td>8</td></tr> </table>	9	4	9	8	6	6	1	9	4	1	5	2		2	2	1	1	5	4	4	9	2	3	1	8	<table border="1"> <tr><td>9</td><td>4</td><td>9</td><td>8</td><td>6</td></tr> <tr><td>6</td><td>1</td><td>9</td><td>4</td><td>1</td></tr> <tr><td>5</td><td>2</td><td style="background-color: #f8d7da;"></td><td>2</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>5</td><td>4</td><td>4</td></tr> <tr><td>9</td><td>2</td><td>3</td><td>1</td><td>8</td></tr> </table>	9	4	9	8	6	6	1	9	4	1	5	2		2	2	1	1	5	4	4	9	2	3	1	8	<table border="1"> <tr><td>9</td><td>4</td><td>9</td><td>8</td><td>6</td></tr> <tr><td>6</td><td>1</td><td>9</td><td>4</td><td>1</td></tr> <tr><td>5</td><td>2</td><td style="background-color: #f8d7da;"></td><td>2</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>5</td><td>4</td><td>4</td></tr> <tr><td>9</td><td>2</td><td>3</td><td>1</td><td>8</td></tr> </table>	9	4	9	8	6	6	1	9	4	1	5	2		2	2	1	1	5	4	4	9	2	3	1	8	<table border="1"> <tr><td>9</td><td>4</td><td>9</td><td>8</td><td>6</td></tr> <tr><td>6</td><td>1</td><td>9</td><td>4</td><td>1</td></tr> <tr><td>5</td><td>2</td><td style="background-color: #f8d7da;"></td><td>2</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>5</td><td>4</td><td>4</td></tr> <tr><td>9</td><td>2</td><td>3</td><td>1</td><td>8</td></tr> </table>	9	4	9	8	6	6	1	9	4	1	5	2		2	2	1	1	5	4	4	9	2	3	1	8
9	4	9	8	6																																																																																																																													
6	1	9	4	1																																																																																																																													
5	2		2	2																																																																																																																													
1	1	5	4	4																																																																																																																													
9	2	3	1	8																																																																																																																													
9	4	9	8	6																																																																																																																													
6	1	9	4	1																																																																																																																													
5	2		2	2																																																																																																																													
1	1	5	4	4																																																																																																																													
9	2	3	1	8																																																																																																																													
9	4	9	8	6																																																																																																																													
6	1	9	4	1																																																																																																																													
5	2		2	2																																																																																																																													
1	1	5	4	4																																																																																																																													
9	2	3	1	8																																																																																																																													
9	4	9	8	6																																																																																																																													
6	1	9	4	1																																																																																																																													
5	2		2	2																																																																																																																													
1	1	5	4	4																																																																																																																													
9	2	3	1	8																																																																																																																													
9	4	9	8	6																																																																																																																													
6	1	9	4	1																																																																																																																													
5	2		2	2																																																																																																																													
1	1	5	4	4																																																																																																																													
9	2	3	1	8																																																																																																																													

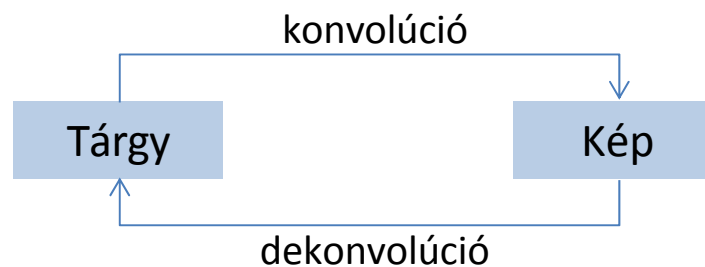
$$5 \cdot 0.33 + 2 \cdot 0.167 + 2 \cdot 0.167 + 9 \cdot 0.167 + 5 \cdot 0.167 = 4.67 \quad \text{kép} = \text{tárgy} * \text{PSF}$$

$$\text{kép}(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \text{tárgy}(i, j) \text{PSF}(x-i, y-j)$$

Konvolúció-dekonvolúció

Tárgy								PSF (konvolúciós kernel)			Kép									
0	0	0	0	0	0	0	0	0.000	0.167	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0	0.167	0.333	0.167										
0	0	0	0	0	0	0	0	0.000	0.167	0.000										
0	0	0	1	0	0	0	0				0.000	0.167	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.333	0.167	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.167	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
0	0	0	0	0	0	0	0				0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	

- Így értelmezhető, hogy egy pontszerű tárgy képe szétterül, és megfelel a PSF-nek.
- A konvolúció fordított műveletével, a dekonvolúcióval, a képből a PSF ismeretében a tárgy előállítható.



Konvolúció-dekonvolúció

- A dekonvolúció nem adható meg képlettel a térbeli doménben.
- Mind a konvolúció, mind a dekonvolúció elvégezhető a frekvencia doménben:

	konvolúció	dekonvolúció
térbeli domén	$tárgy * PSF = kép$	–
frekvencia domén	$FT(tárgy) \cdot FT(PSF) = FT(kép)$	$FT(kép) ./ FT(PSF) = FT(tárgy)$
	IFT ↓ $kép$	IFT ↓ $tárgy$

FT – Fourier transzformáció

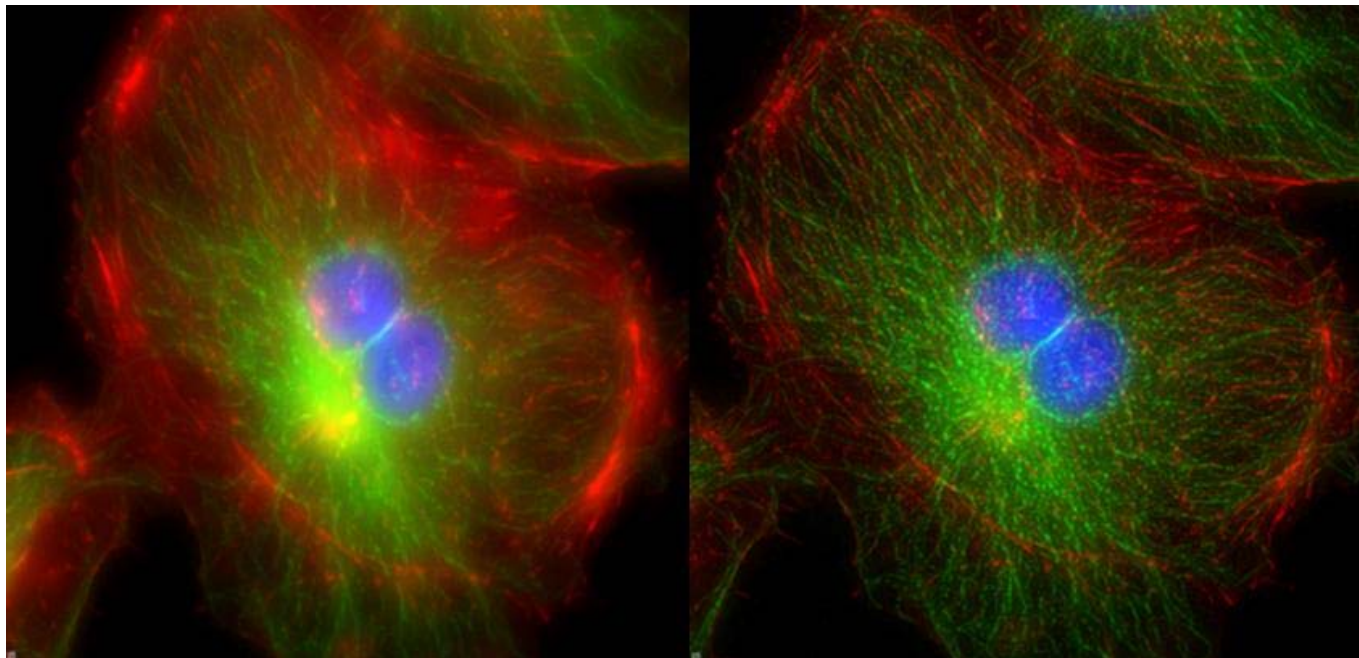
IFT – inverz Fourier transzformáció

./ – elemenkénti osztás

. – elemenkénti szorzás („*” Matlab-ban)

Dekonvolúció három dimenzióban

- Mivel egy adott képponthoz nemcsak az azonos síkban levő tárgypontok járulnak hozzá, hanem a felette és alatta levő síkokban levők is, a dekonvolúciónak csak 3D-ben van igazán értelme.
- Ebben az esetben a kép a 3D tárgy 3D PSF-fel való konvolúciójával jön létre.
- Ezért a tárgyat a 3D kép 3D PSF-fel történő dekonvolúciójával lehet kiszámolni.
- Ez az eljárás a hagyományos (nem konfokális) mikroszkóppal készített 3D stack-ekre alkalmazható → a kontrasztot jelentősen, a feloldóképességet enyhén növeli.
- A 3D PSF meghatározása kulcsfontosságú. Általában egy pontszerűnek tekinthető fluoreszcens gyöngy 3D képének meghatározásával történik.



Eredeti konfokális kép

dekonvolúció után

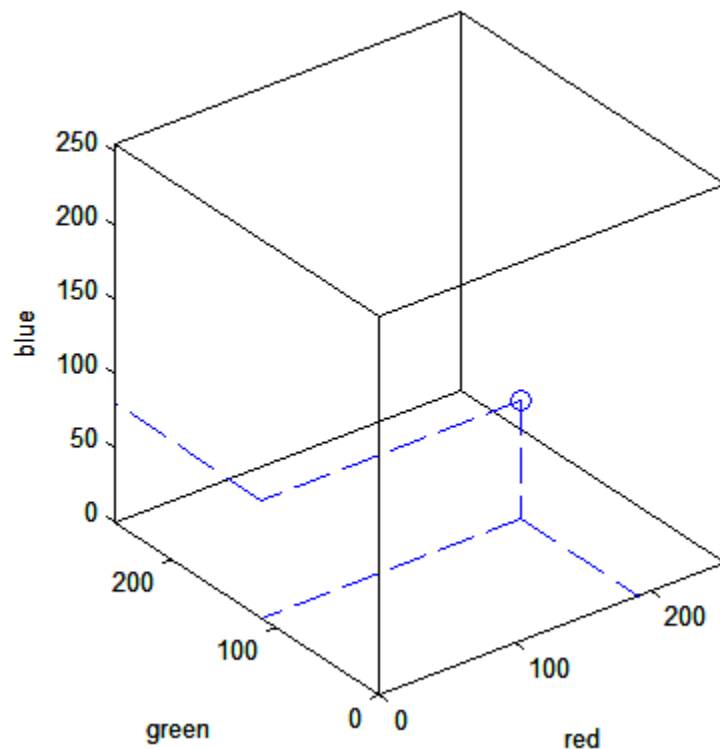
Színes képek

- Az emberi szemben háromféle színérzékeny receptorsejt van (csapok), amelyek a spektrum vörös, zöld és kék tartományára érzékenyek.
- Ezért a színt egy háromváltozós függvénynek vagy egy háromdimenziós térnek lehet elképzelni:

RGB színtér:

- R: vörös szín intenzitása [0-255] vagy [0-1]
- G: zöld szín intenzitása [0-255] vagy [0-1]
- B: kék szín intenzitása [0-255] vagy [0-1]

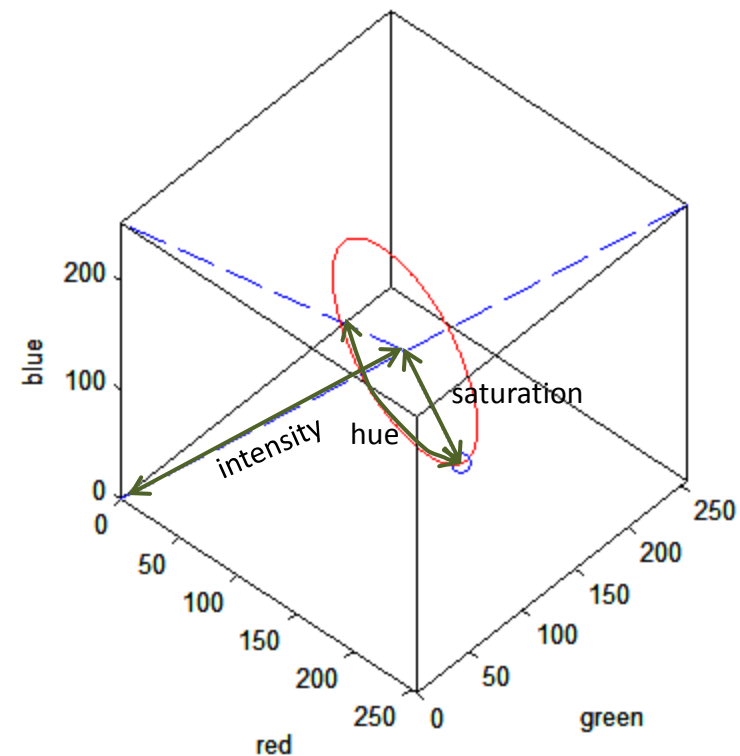
A színes képek értelmezésével kapcsolatban az RGB színmodellt használják a legtöbbször.



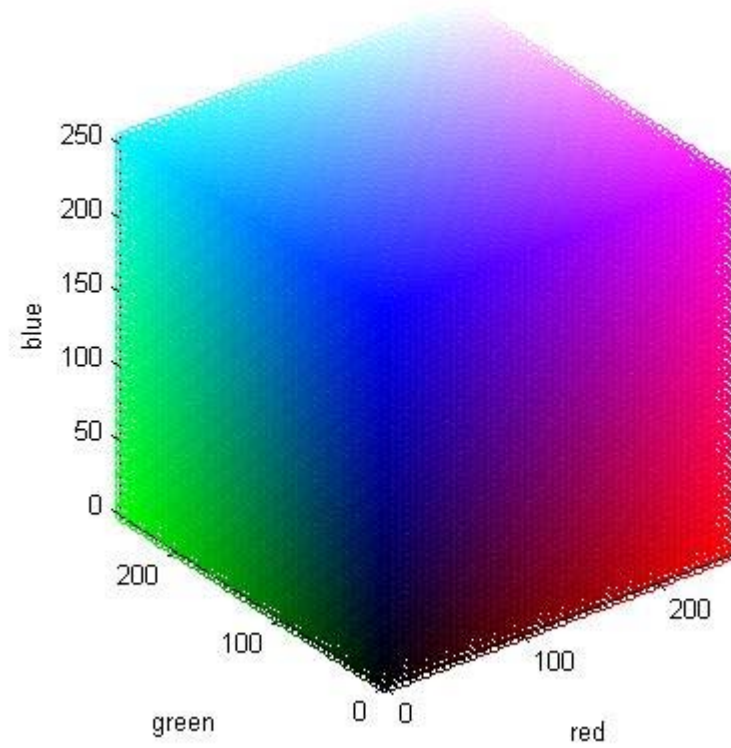
HSI színtér (hue (színárnyalat)-saturation (telítettség)-intensity):

a kocka átlójára merőlegesen rajzolunk egy kört, amin rajta van az (R,G,B) pont

- H: a körön az (R,G,B) pontba húzott sugár és a kék sarokba húzott egyenes közti szög
- S: a kör sugarának aránya a kör középpontjából az (R,G,B) ponton át a kocka széléig húzott vonalhoz képest (milyen messze van az (R,G,B) pont az átlótól)
- I: a kör középpontjának távolsága (0,0,0)-tól

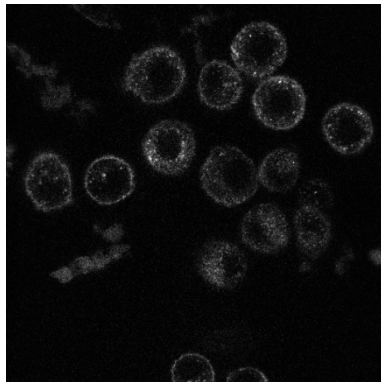


Az RGB színtér

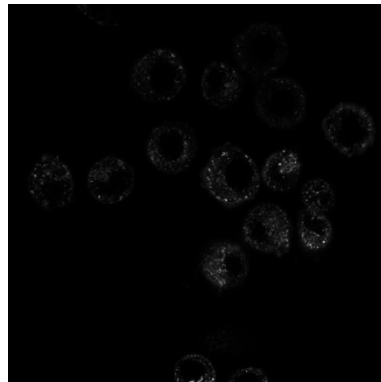


Színes képek, színes kompozit képek létrehozása

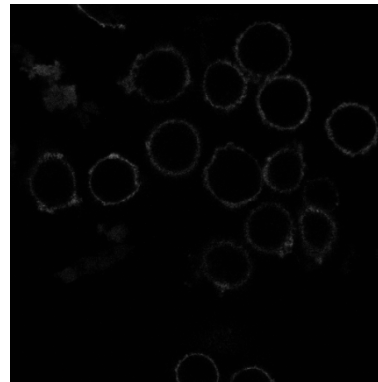
- A biológiai minták vizsgálata során a felvett képek általában szürkeskálás képek, ezért színes képek akkor keletkeznek, ha több csatornában (hullámhossztartományban) felvett képet egymásra rétegzünk.
- Eredeti szürkeskálás képek:
 - első hullámhossztartomány → vörös csatorna
 - második hullámhossztartomány → zöld csatorna
 - harmadik hullámhossztartomány → kék csatorna
- A jobb láthatóság kedvéért más színeket is lehet használni, pl. a kék fekete háttéren való rossz láthatósága miatt a kék csatorna helyett gyakran cián (cyan) csatornát használnak.
- Az egyes csatornákat érdemes „contrast stretch”-elni (a láttatni kívánt intenzitások értékét 255-re növelni)



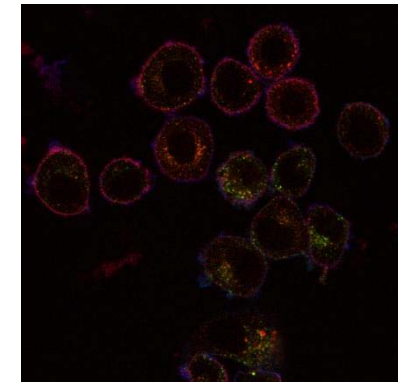
Első csatorna → vörös



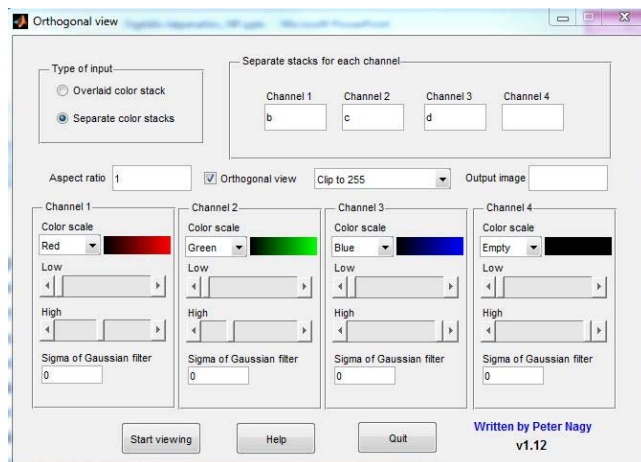
Második csatorna → zöld



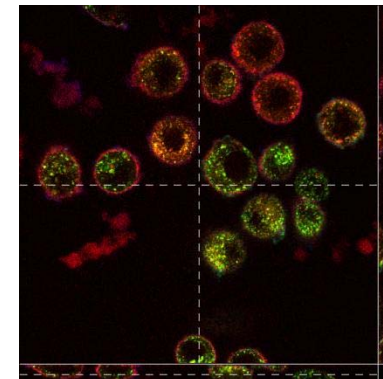
Harmadik csatorna → kék



Színes kompozit kép
kontraszt módosítás nélkül

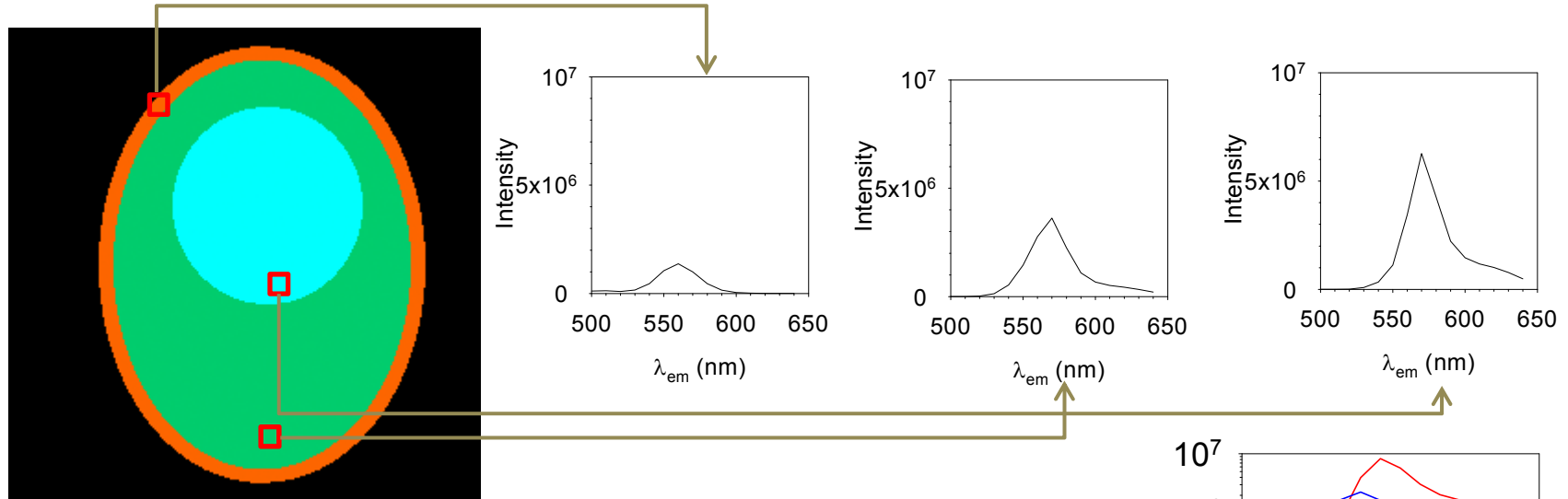


Színes kompozit kép a vörös
és a zöld csatorna „contrast
stretch”-elése után:



Spectral unmixing

- Előfordul, hogy a fluorofórok eloszlását az egyes csatornában felvett képek alapján nem lehet megmondani, ha
 - a minta egymást átfedő spektrumú fluorofórokkal van megjelölve
 - az egyes pixeleken többféle fluorofór keveredik

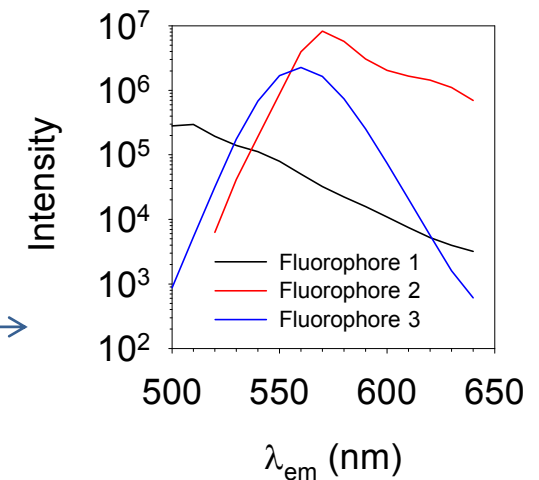


- A minta minden egyes pixelében felvesszük a spektrumot (általában az emissziós spektrumot).
- A mintát három fluorofórral festettük meg, melyek spektrumát meg lehet mérni:
- A fluoreszcencia intenzitást minden pixelben a következő egyenlet írja le:

$$I(\lambda_1) = F(1)s(1, \lambda_1) + F(2)s(2, \lambda_1) + F(3)s(3, \lambda_1)$$

$F(1), F(2), F(3)$ – az egyes fluorofórok relatív mennyisége az adott pixelben

$s(1, \lambda_1), s(2, \lambda_1), s(3, \lambda_1)$ – az egyes fluorofórok spektruma az adott hullámhosszon



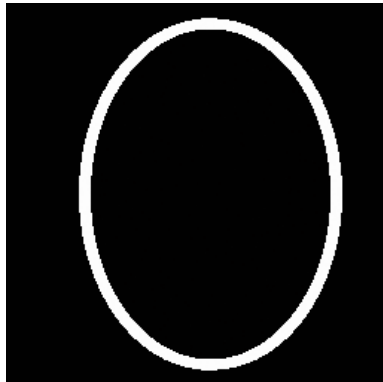
Spectral unmixing

- Az előző egyenlet mátrix formában:

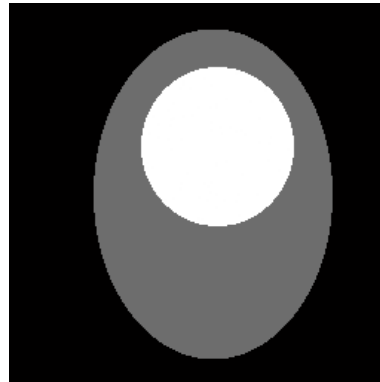
$$\begin{bmatrix} s(1, \lambda_1) & s(2, \lambda_1) & \dots & s(k, \lambda_1) \\ s(1, \lambda_2) & s(2, \lambda_2) & \dots & s(k, \lambda_2) \\ \dots & \dots & \dots & \dots \\ s(1, \lambda_n) & s(2, \lambda_n) & \dots & s(k, \lambda_n) \end{bmatrix} \begin{bmatrix} F(1) \\ F(2) \\ \dots \\ F(k) \end{bmatrix} = \begin{bmatrix} I(\lambda_1) \\ I(\lambda_2) \\ \dots \\ I(\lambda_n) \end{bmatrix}$$

- A fenti egyenletrendszer túlhatározott, hiszen k db (példánkban 3) független változóra n db (példánkban 15) egyenlet jut, ezért csak közelítéssel oldható meg illesztés segítségével.
- Ennek eredményeképpen minden fluorofór eloszlása megadható:

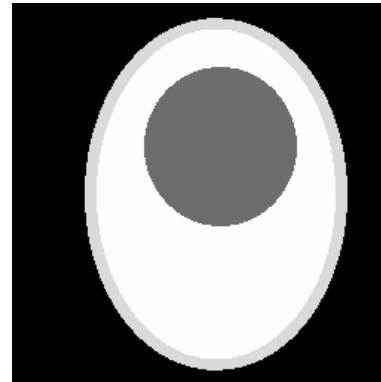
Az 1. fluorofór eloszlása



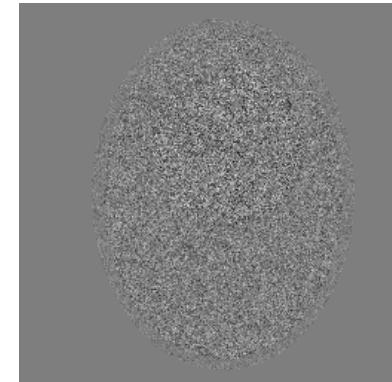
A 2. fluorofór eloszlása



A 3. fluorofór eloszlása

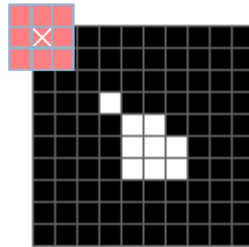


Az illesztés hibája („residual”)



Matematikai morfológia, morfológiai képanalízis: alapelvek

- Morfológia: Olyan eszközök összessége, melyek a képen levő objektumok alakjának és határvonalainak (morfológiájának) leírására és reprezentációjára használhatók.
- Minden esetben definiálni kell egy másik képet (halmazt), amivel a műveletet végre fogjuk hajtani. Ezt strukturális elemnek („structuring element”, S.E., kernel) nevezzük, pl.: 
- A S.E.-nek van egy kitüntetett referencia pontja (fehér x), amivel a pozícióját definiáljuk.
- A S.E.-t a kiértékelendő kép minden pixelére elhelyezzük, és minden egyes pixelben az adott operátornak megfelelő műveletet hajtunk végre.

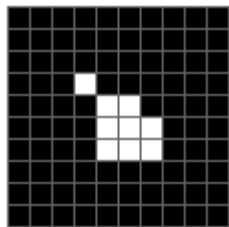


Morfológiai képanalízis: dilatació

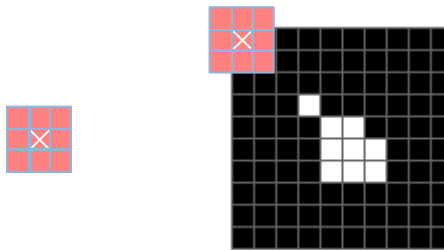
Bináris dilatació ($\oplus$): Azon pixelek összessége, amelyre a strukturális elem referencia pixelét helyezve a strukturális elem és az eredeti objektum között lesz átfedés.

$$A \oplus B = \bigcup_{(x,y) \in A} \hat{B}_{x,y} = \{(x,y) \mid \hat{B}_{x,y} \cap A \neq \emptyset\}$$

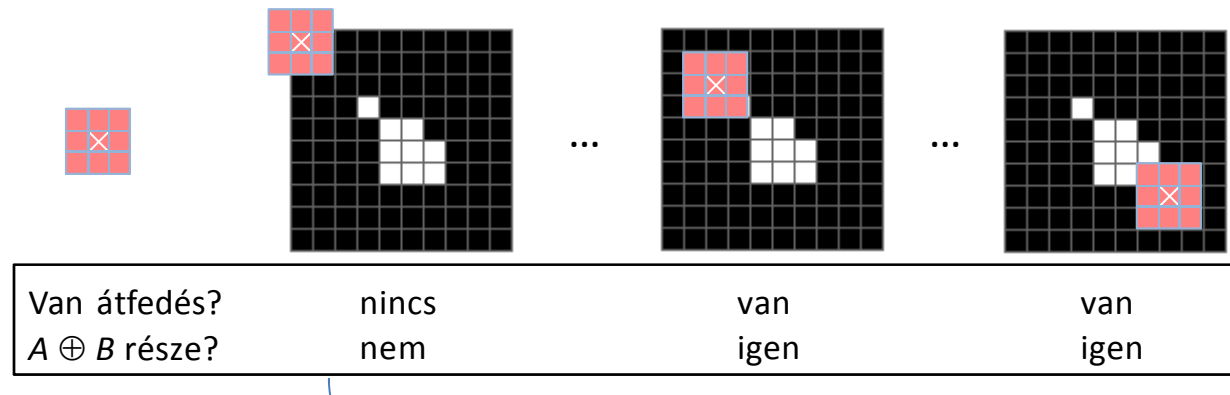
Eredeti kép (A):



S.E. (B):



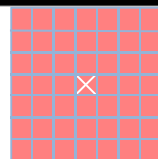
A dilatació lépésenként:



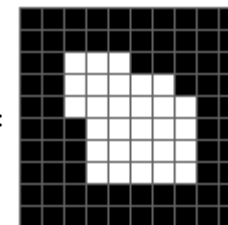
Eltávolítja azokat a lyukakat, melyek kisebbek a strukturális elemnél:



Strukturális elem (nem arányos a képpel):



$A \oplus B =$

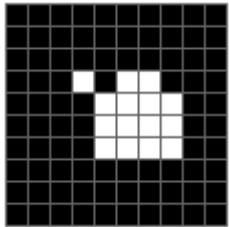


Morfológiai képanalízis: erózió

Bináris erózió ($\ominus$): Azon pixelek összessége, amelyre a strukturális elem referencia pixelét helyezve a strukturális elem teljes egészében az eredeti objektumon belül található.

$$A \ominus B = \{(x, y) \mid B_{x,y} \subset A\}$$

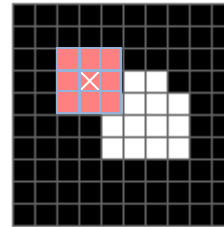
Eredeti kép (A):



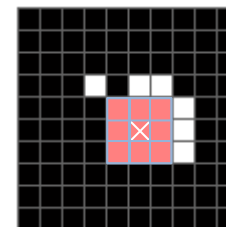
S.E. (B):



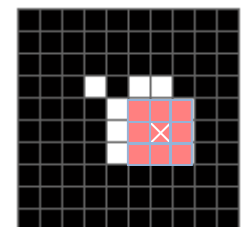
Az erózió lépésenként:



...



...



$B \subset A$?

nem

igen

igen

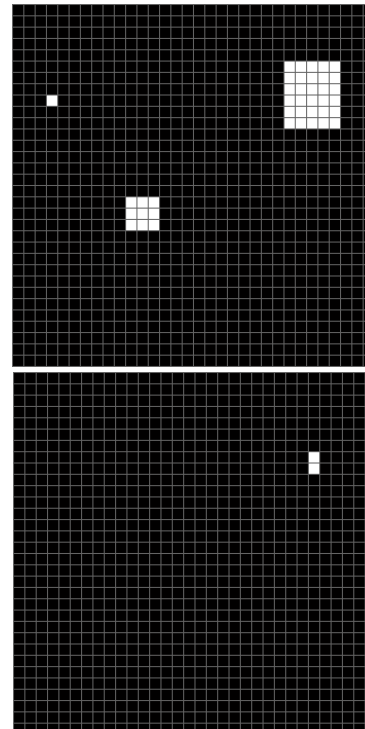
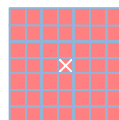
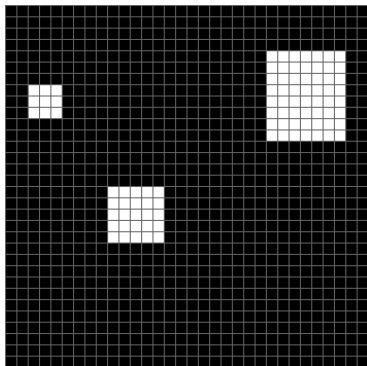
$A \ominus B$ része?

nem

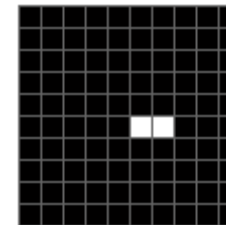
igen

igen

Eltávolítja azokat az objektumokat, melyek kisebbek a strukturális elemnél:



$A \ominus B =$



Morfológiai képanalízis: nyitás és zárás

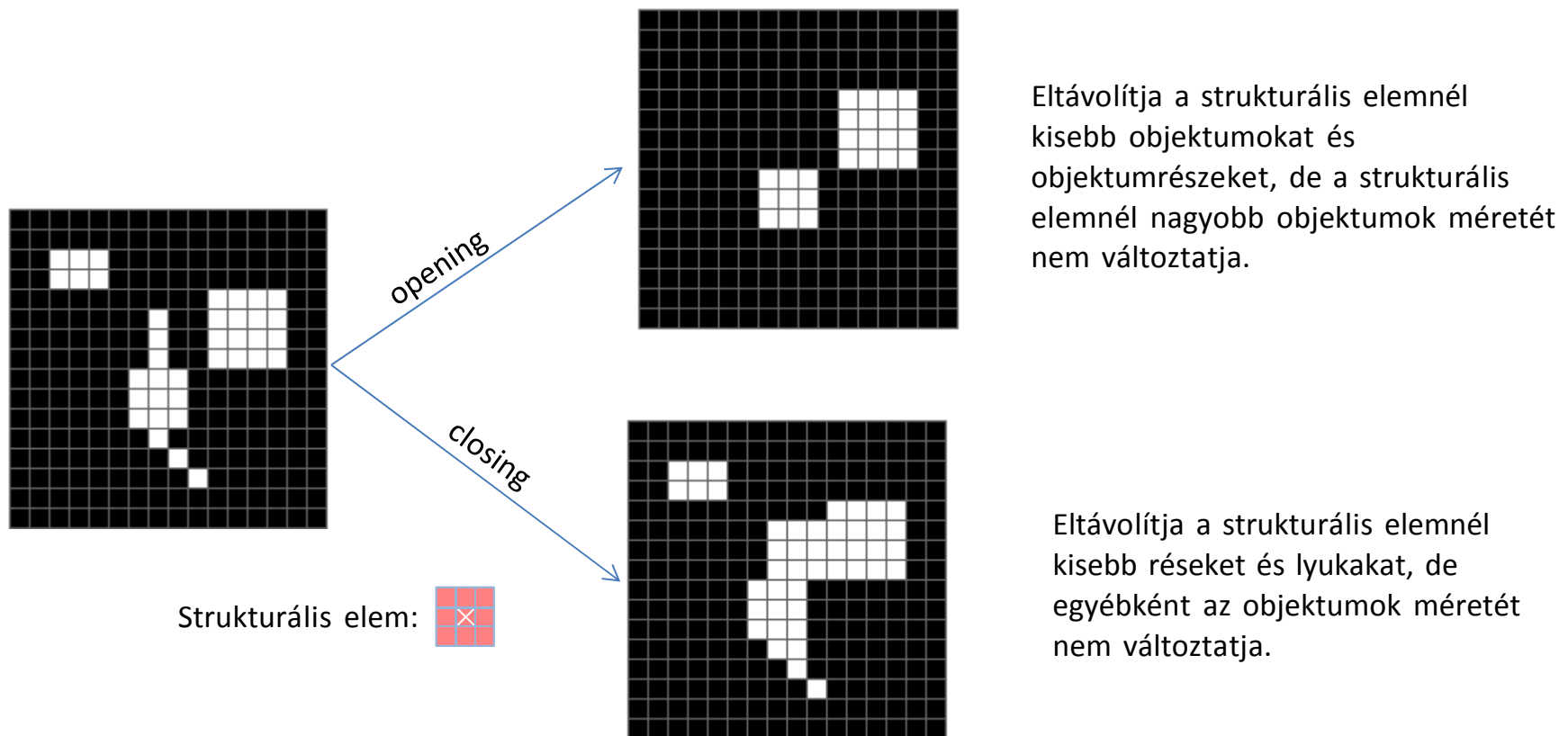
Cél: olyan operátor, amely egy adott méretnél kisebb lyukakat vagy objektumokat eltávolít a képről az objektumok alakjának jelentős torzítása nélkül.

Dilatáció: eltávolítja azokat a lyukakat, melyek kisebbek a strukturális elemnél, **de felnagyítja az objektumokat.**

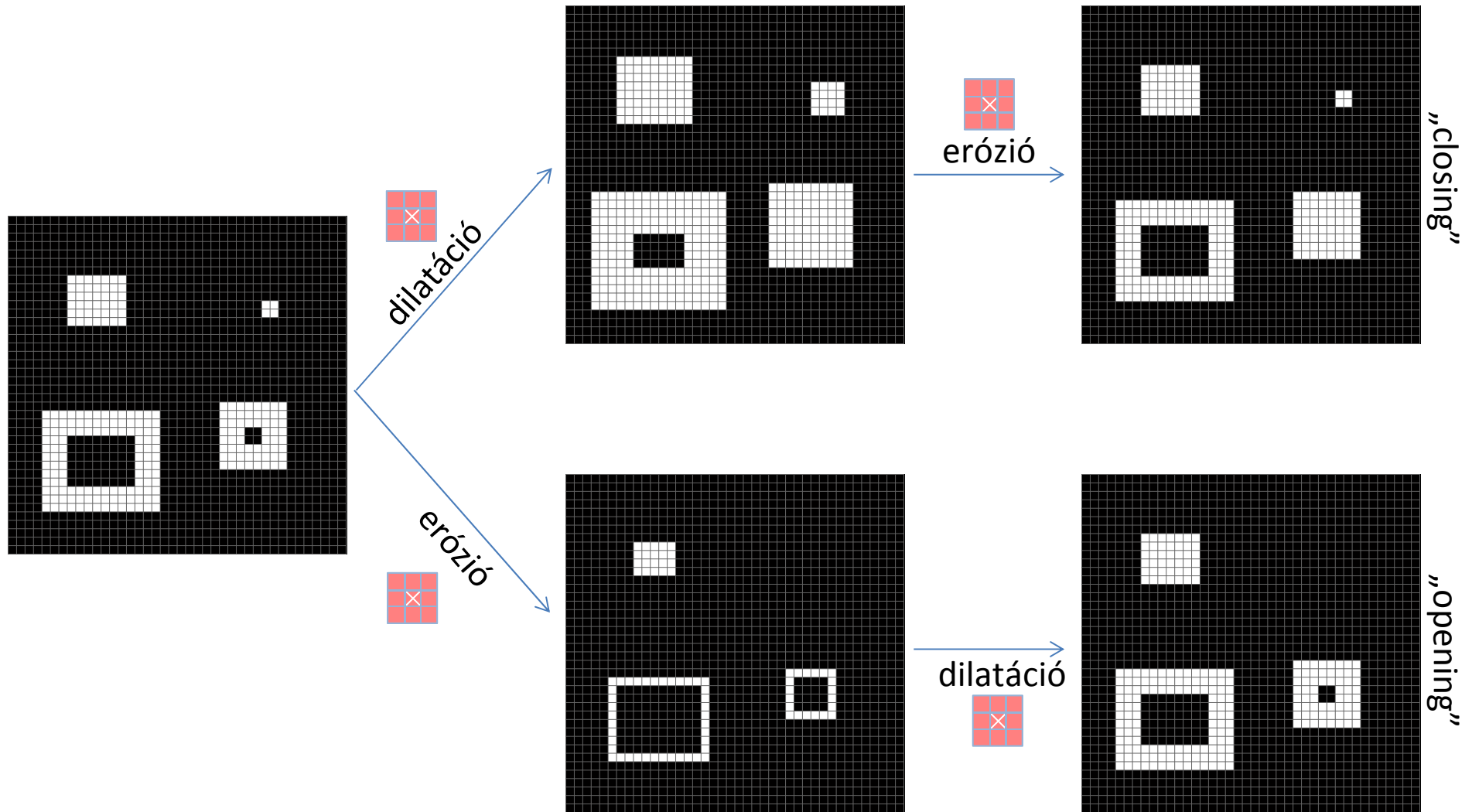
Erózió: eltávolítja azokat az objektumokat, melyek kisebbek a strukturális elemnél, **de összenyomja az objektumokat.**

Nyitás („opening”, $\circ$): az erózió dilatációja: $A \circ B = (A \ominus B) \oplus B$

Zárás („closing”, $\bullet$): a dilatáció eróziója: $A \bullet B = (A \oplus B) \ominus B$



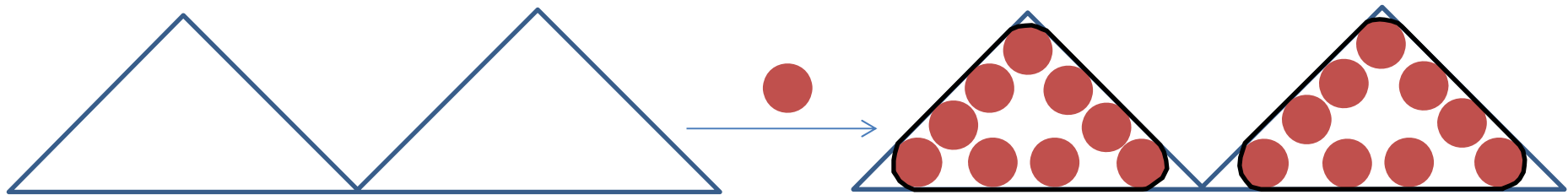
Morfológiai képanalízis: erózió, dilatáció, „opening” és „closing”



Morfológiai képanalízis: a nyitás és zárás szemléltetése

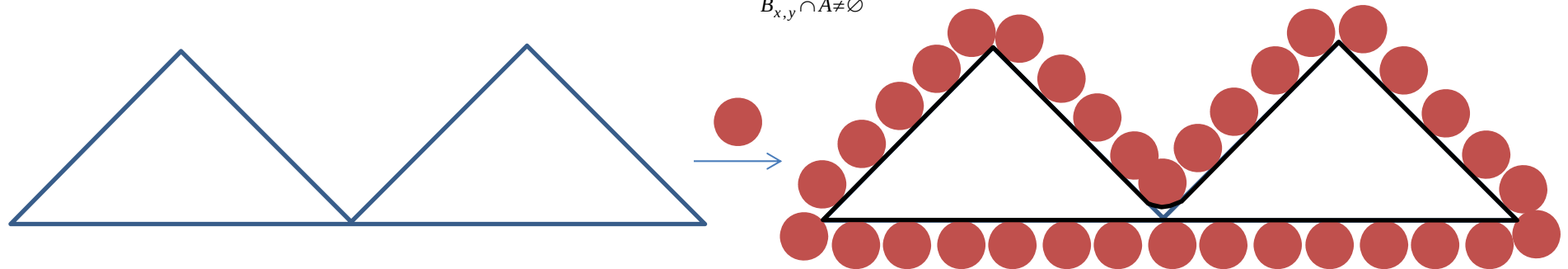
Nyitás: a S.E. uniója minden olyan pozícióban, ahol az egész S.E. az objektumon belül van. A nyitás eltávolítja azokat az objektumokat, ill. objektum részeket, melyek nem képesek a S.E.-et magukba foglalni, kisimítja az objektum kontúrokat, a vékony összeköttetéseket és a protrúziókat eltávolítja.

$$A \circ B = \bigcup_{B_{x,y} \subset A} B_{x,y}$$



Zárás: komplementere a S.E. uniójának minden olyan pozícióban, ahol a S.E. egyáltalán nem fed át az objektummal. Szintén simítja az objektum kontúrját, de kitölti a S.E.-nél kisebb lyukakat vagy réseket.

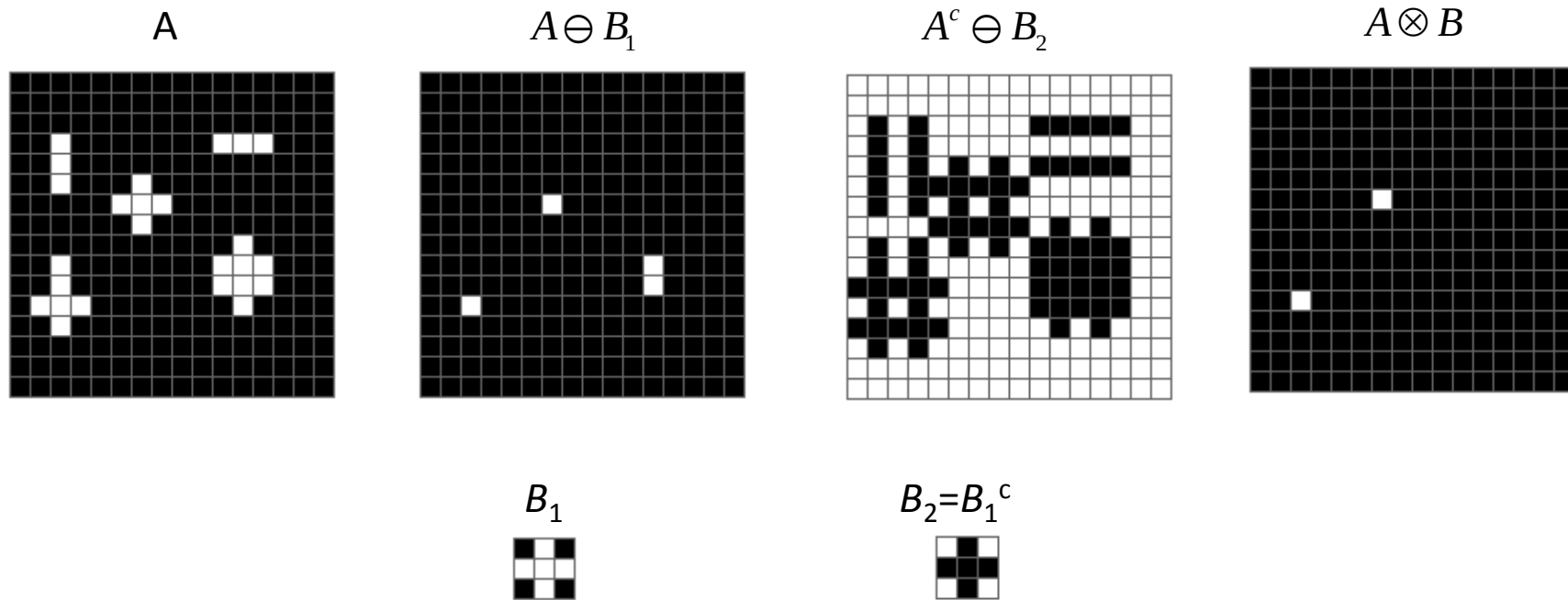
$$A \bullet B = \bigcup_{B_{x,y} \cap A \neq \emptyset} B_{x,y}$$



Morfológiai képanalízis: „hit-or-miss” művelet

Olyan művelet, amely felhasználható egy bizonyos alakú objektum jelenlétének igazolására.

$$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$$



A „hit-or-miss” művelet a B_1 strukturális elemmel azonos alakú objektumokat keresi meg.
A 'c' felsőindex a halmaz komplementerét jelenti.

Morfológiai képanalízis: határvonal azonosítás

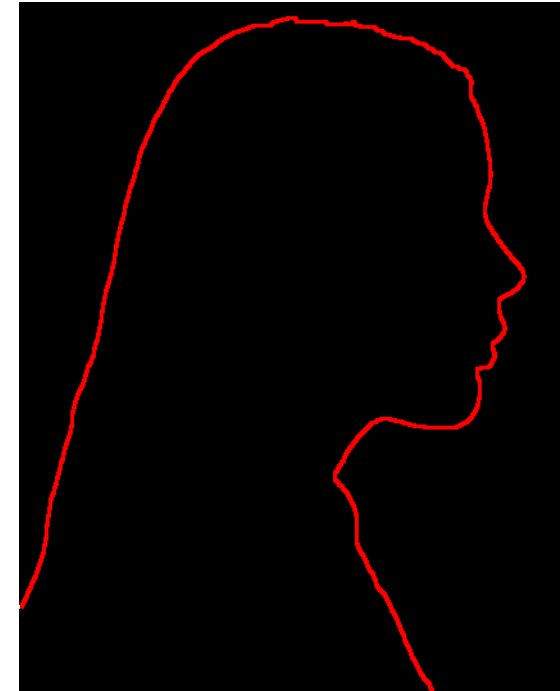
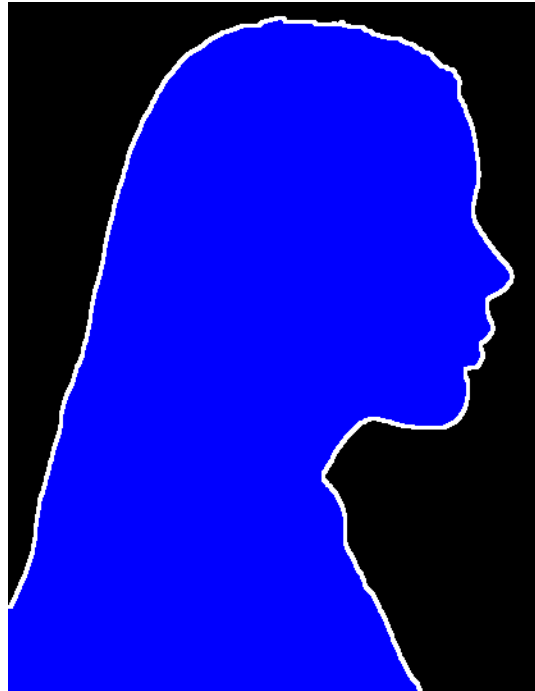
$$\text{boundary}(A) = A - (A \ominus B)$$

kék – $A \ominus B$

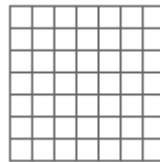
fehér – eredeti kép széle

határvonal

A



strukturális elem:
(vastag határvonal)



strukturális elem:
(vékony határvonal)

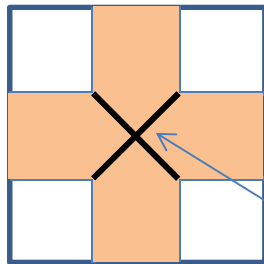


Morfológiai képanalízis: morfológiai operációk nem bináris („gray-scale”) képeken

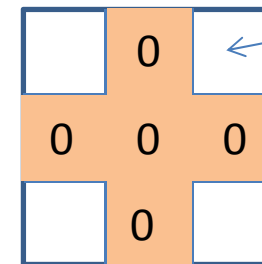
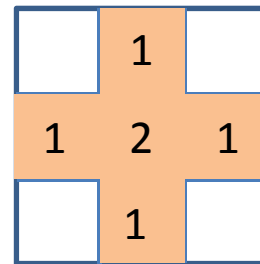
Művelet	Bináris képen	Szürkeskálájú képen
Dilatáció	$A \oplus B = \bigcup_{(x,y) \in A} \hat{B}_{x,y} = \{(x,y) \mid \hat{B}_{x,y} \cap A \neq \emptyset\}$	$(A \oplus B)(x,y) = \max_{(k,l) \in B} (a(x+k, y+l) + b(k,l))$
Erózió	$A \ominus B = \{(x,y) \mid B_{x,y} \subset A\}$	$(A \ominus B)(x,y) = \min_{(k,l) \in B} (a(x+k, y+l) + b(k,l))$
Nyitás	$A \circ B = \bigcup_{B_{x,y} \subset A} B_{x,y}$	$A \circ B = (A \ominus B) \oplus B$
Zárás	$A \bullet B = \bigcup_{B_{x,y} \cap A \neq \emptyset} B_{x,y}$	$A \bullet B = (A \oplus B) \ominus B$

a – a pixelek intenzitása a feldolgozandó képen
 b – a strukturális elem értékei

Szürkeskálájú strukturális elem:



Referencia pixel



Ha minden érték nulla,
akkor egyszerű
lokális maximum
operátorról van
szó.

Definiálja, hogy mely pixeleket vegye figyelembe.

... és hogy mennyit adjon hozzá a kép pixel értékeihez.

Morfológiai képanalízis: szűrkeskálájú morfológiai operátorok szemléltetése

1	2	3	3	1	2	3	5	7	4
4	2		2	2	1	1	4	8	9
5				3	3	2	3	4	2
6	4		4	4	4	4	4	4	4
5	2	5	5	5	5	5	4	5	6
4	5	4	5	6	6	6	4	5	4
5	4	4	5	7	7	7		6	4
4	5	5	4	8	8				6
2	4	4	4	9	9	9		6	5
2	2	3	6	10	10	10	5	7	6

dilatáció (max)

erózió (min)

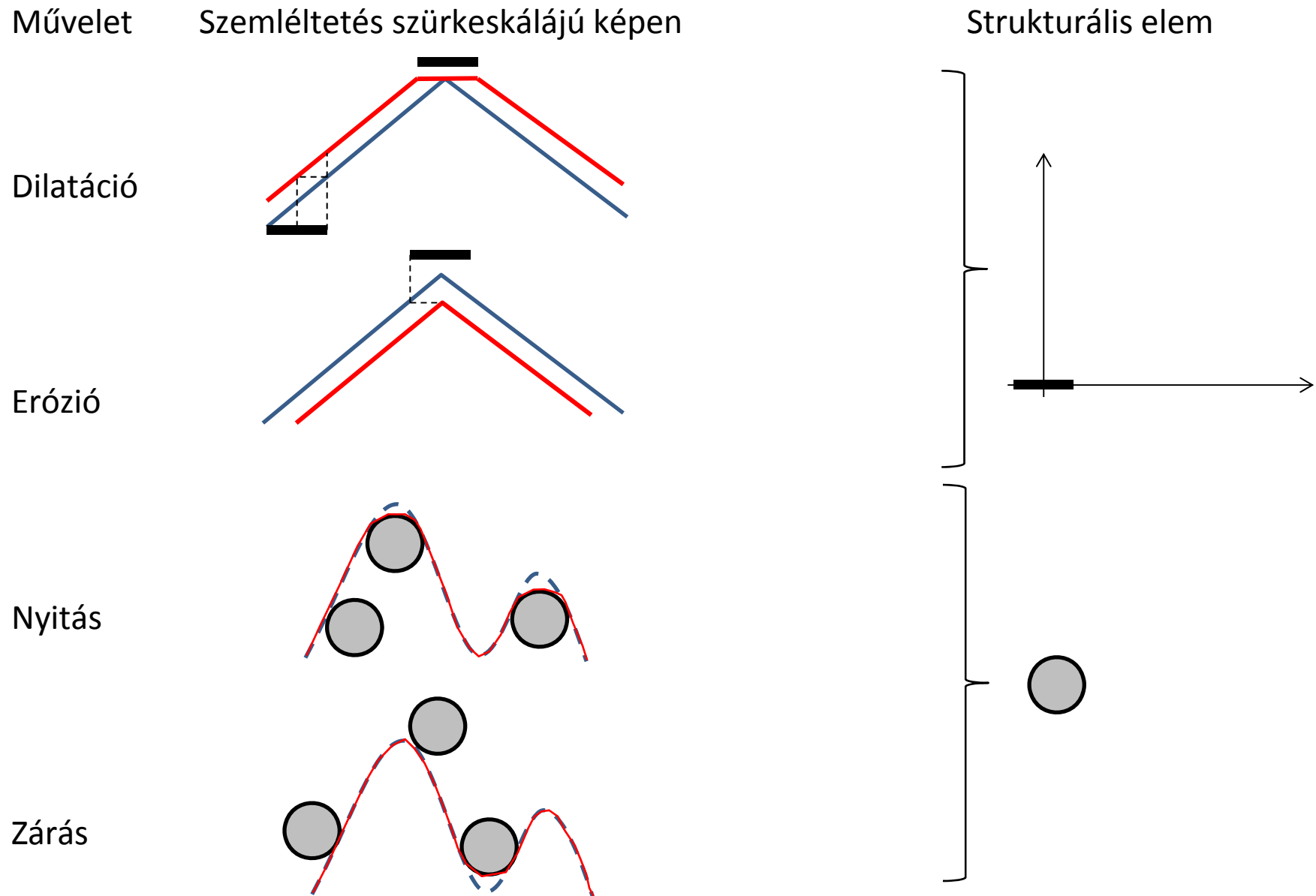
4	3	3	3	3	3	5	7	8	9
5	4	3	3	3	3	4	8	9	9
6	5	4	4	4	4	4	4	8	9
6	6	5	5	5	5	5	4	5	6
6	5	5	5	6	6	6	5	6	6
5	5	5	6	7	7	7	6	6	6
5	5	5	7	8	8	8	7	6	6
5	5	5	8	9	9	9	8	6	6
4	5	5	9	10	10	10	9	7	6
2	4	6	10	10	10	10	10	7	7

1	1	2	1	1	1	1	3	4	4
1	2	2	2	1	1	1	1	4	2
3	2	3	2	2	1	1	2	2	2
4	2	3	3	3	3	2	3	4	2
2	2	2	4	4	4	4	4	4	4
4	2	4	4	5	5	4	4	4	4
4	4	4	4	5	6	4	4	4	4
2	4	4	4	4	7	5	4	5	4
2	2	3	4	4	8	6	5	5	5
2	2	2	3	6	9	5	5	5	5

Strukturális elem:

	0	
0	0	0
	0	

Morfológiai képanalízis: morfológiai operációk nem bináris („gray-scale”) képeken



Morfológiai képanalízis: morfológiai operációk nem bináris („gray-scale”) képeken

Eredeti kép



Dilatáció



Zárás



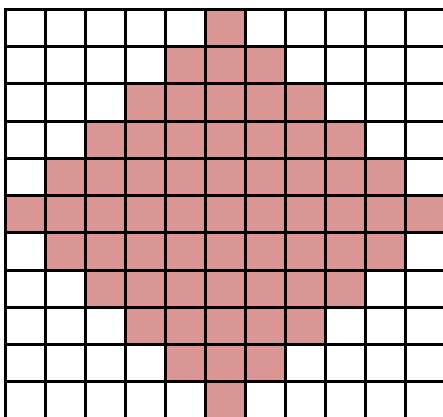
Erózió



Nyitás



Strukturális elem:



Morfológiai képanalízis: „tophat” transzformáció szürkeskálájú képen

Olyan művelet, amely a strukturális elemnél nagyobb objektumokat (intenzitás trendeket) távolít el a képről.

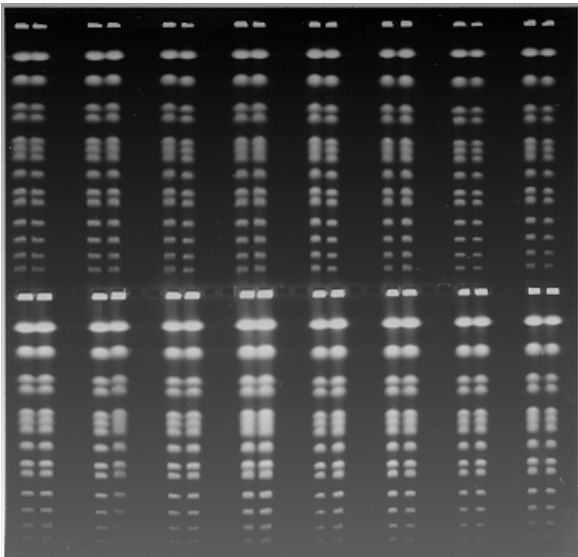
$$\text{tophat}(A) = A - (A \bullet B)$$

világos alapon sötét objektumok kiemelése

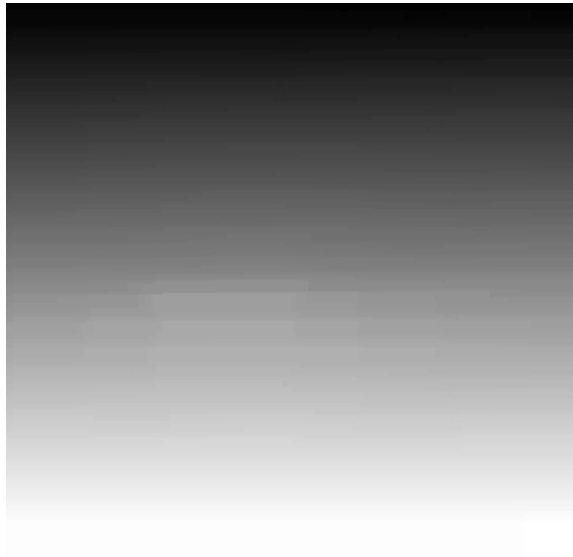
$$\text{tophat}(A) = A - (A \circ B)$$

sötét alapon világos objektumok kiemelése

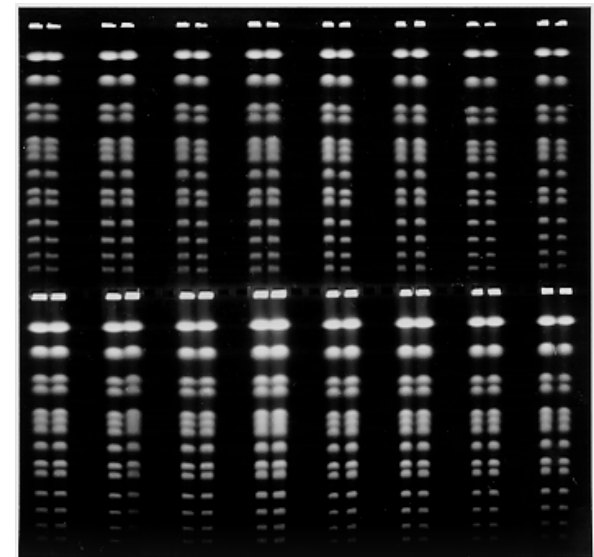
A



$A \circ B$



$\text{tophat}(A)$

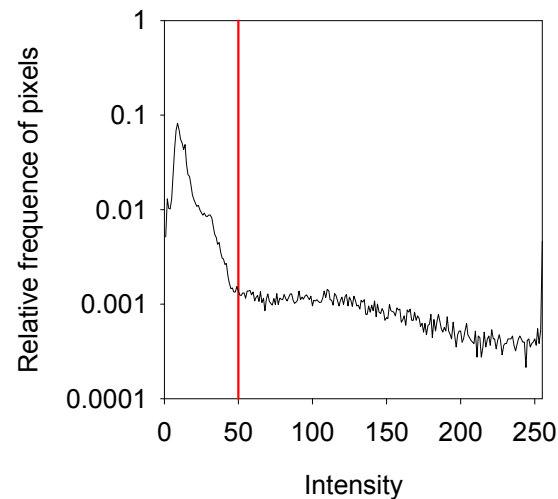
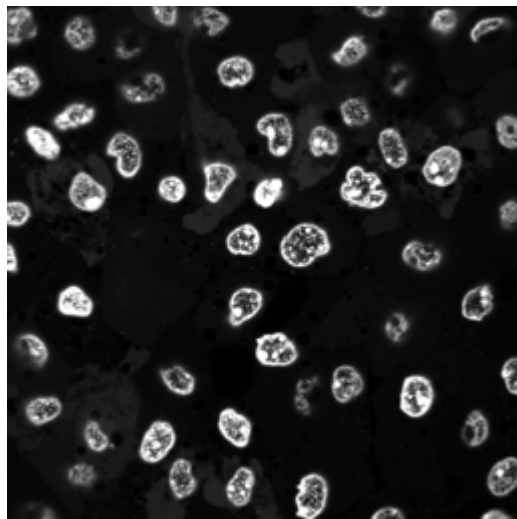


Szegmentálás

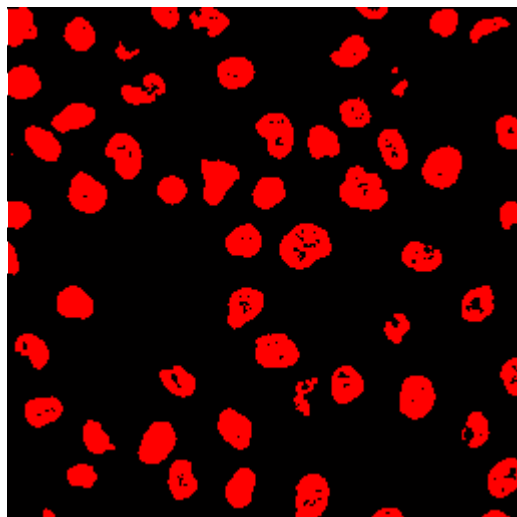
- A kép pixeleinek különböző szegmensekre (csoportokra, halmazokra) bontása.
- Legtöbbször két szegmensre kell a képet osztani:
 - előtér („foreground”) vagy objektumok: ami a kiértékelőt érdekli
 - háttér („background”): ami a kiértékelőt nem érdekli
- A szegmentálás eredménye:
 - annyi pixelhalmaz (legtöbbször kettő), amennyire a képet szegmentáltuk
 - a halmazokat elválasztó határvonalak.
- A szegmentálást gyakran zajszűrés és háttérlevonás előzi meg.
- A szegmentálásra azért van szükség, mert a számítógépnek nincs veleszületett alakfelismerő képessége (l. Mona Lisa kép és számok formájában).
- A szegmentálás lehetséges módszerei:
 - hisztogram alapú szegmentálás, küszöbölés (manuális, intermeans, maximum entrópia, Otsu)
 - határdetektálás (LoG, Canny)
 - régió alapú szegmentálás (watershed, split and merge)

Szegmentálás: manuális küszöbölés

- A küszöbölési eljárások első lépése a kép hisztogramjának előállítása:



- Manuális küszöbölés esetén a hisztogram alapján vagy a kép szemlélése alapján kiválasztunk egy olyan intenzitás értéket, ami a legjobban elválasztja egymástól a háttérrel az objektumoktól:



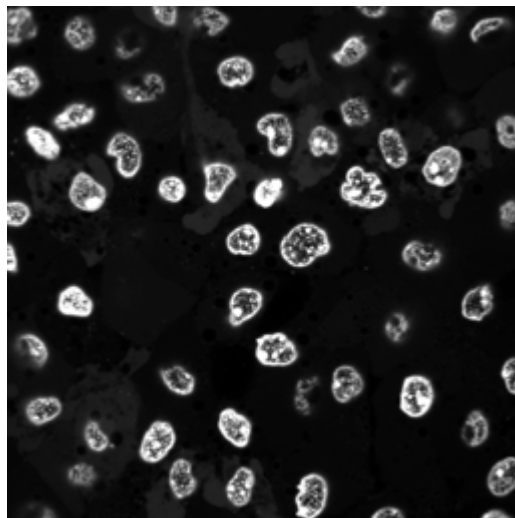
`segmented=original>50;`

Szegmentálás: intermeans algoritmus

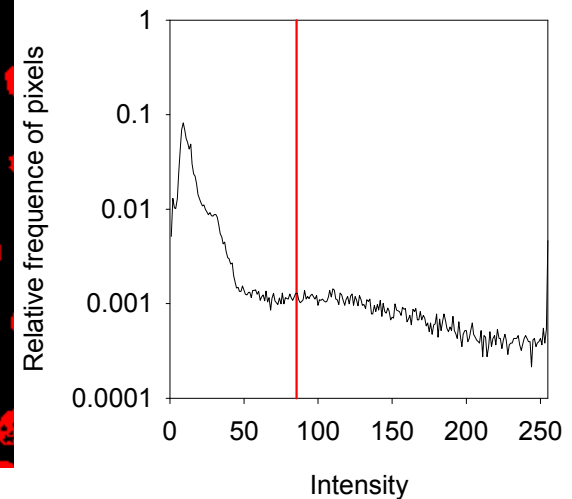
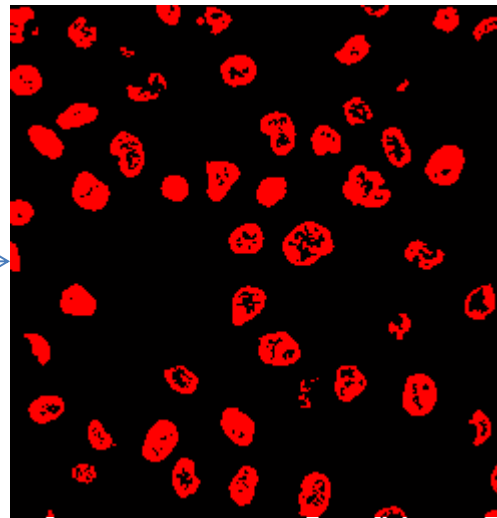
1. Kezdeti küszöb=a kép mediánja
2. μ_1 =a küszöb alatti pixelek átlaga, μ_2 =a küszöb feletti pixelek átlaga
3. A küszöb újrabecslése:

$$thr_{new} = \frac{\mu_1 + \mu_2}{2}$$

4. A 2. és 3. lépéseket addig ismételni, amíg nincs konvergencia.



küszöb:
85.5



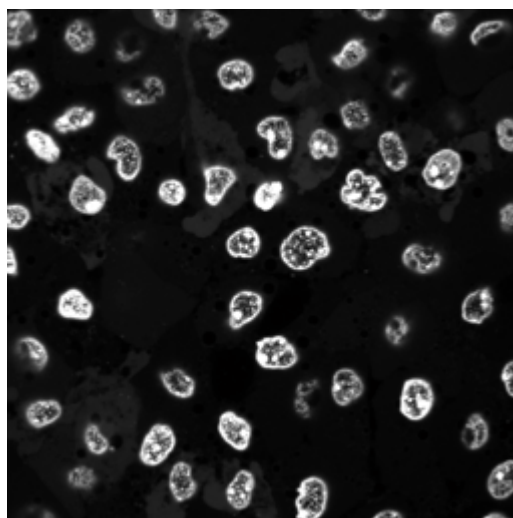
Szegmentálás: maximum entrópia algoritmus

Az információelméletben az entrópia egy valószínűségi változó vagy halmaz bizonytalanságát (és ezzel információtartalmát) jellemzi. A Shannon képlet szerint egy k valószínűségi változó entrópiája:

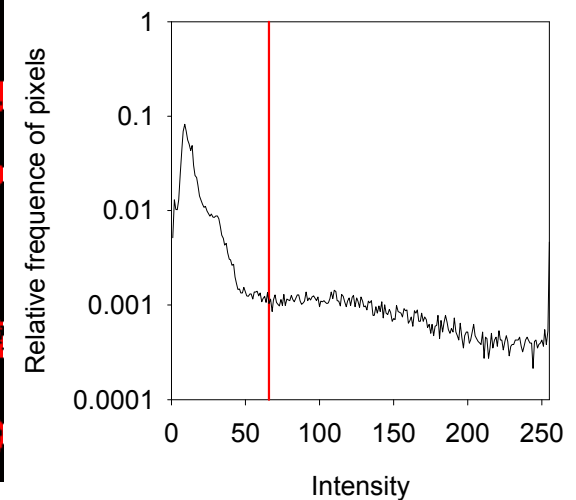
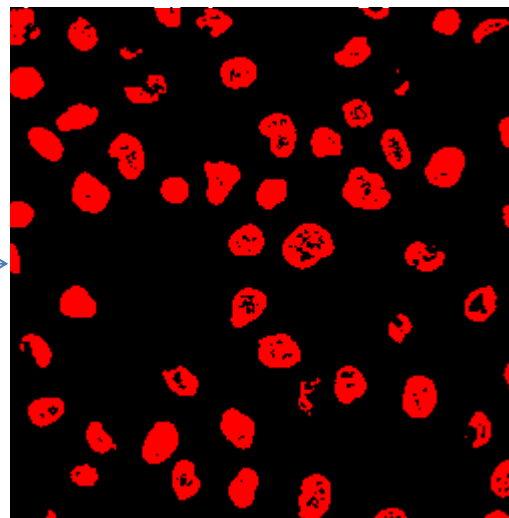
$$H(k) = -\sum_i p_i \log_2 p_i$$

Az algoritmus:

1. Minden lehetséges küszöbre (0-tól a maximum intenzitásig) ki kell számolni a küszöb alatti pixelek entrópiáját.
2. Az összes küszöb közül azt kell kiválasztani, amihez a legnagyobb entrópia tartozik.



küszöb:
66



Szegmentálás: Otsu módszere

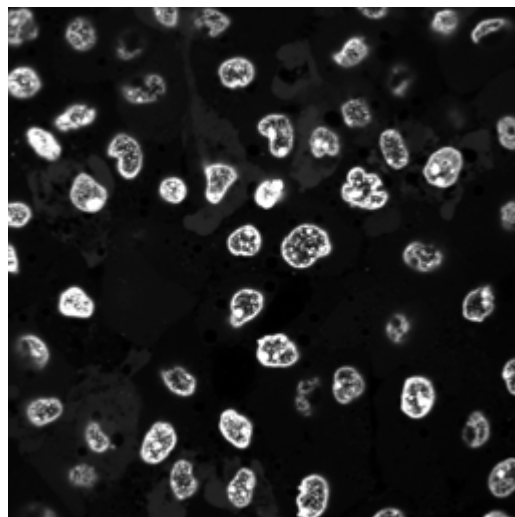
$$\begin{aligned}
 SS_{tot} &= \sum_{i \in \text{all pixels}} (x_i - \bar{x})^2 = \sum_{j \in \text{all groups}} \sum_{k \in \text{all pixels in group } j} (x_{j,k} - \bar{x}_j + \bar{x}_j - \bar{x})^2 = \\
 &= \sum_j \sum_k (x_{j,k} - \bar{x}_j)^2 + 2 \sum_j \sum_k (x_{j,k} - \bar{x}_j)(\bar{x}_j - \bar{x}) + \sum_j \sum_k (\bar{x}_j - \bar{x})^2 = \\
 &= \sum_j SS_{intra,j} + \sum_j n_j (\bar{x}_j - \bar{x})^2 = \sum_j n_j S_j^2 + \sum_j n_j (\bar{x}_j - \bar{x})^2 \\
 \sum_j \sum_k (x_{j,k} - \bar{x}_j)(\bar{x}_j - \bar{x}) &= \sum_j (\bar{x}_j - \bar{x}) \underbrace{\sum_k (x_{j,k} - \bar{x}_j)}_0 = 0
 \end{aligned}$$

csoportokon
belüli négyzetes
eltérések

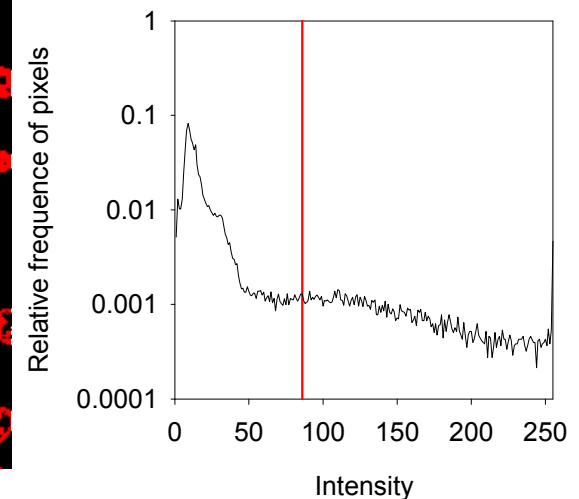
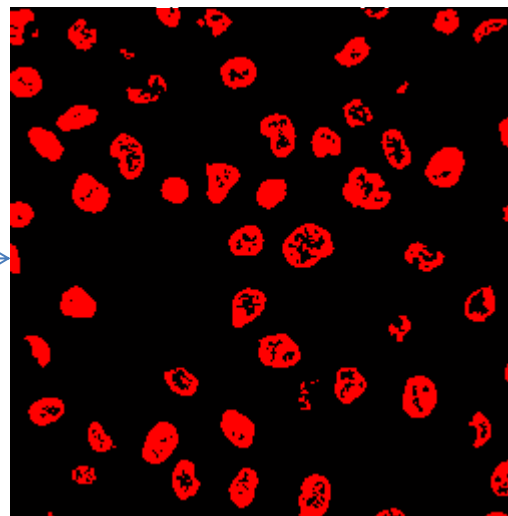
csoportok közötti
négyzetes eltérés

nulla minden csoportban
SS – sum of squares

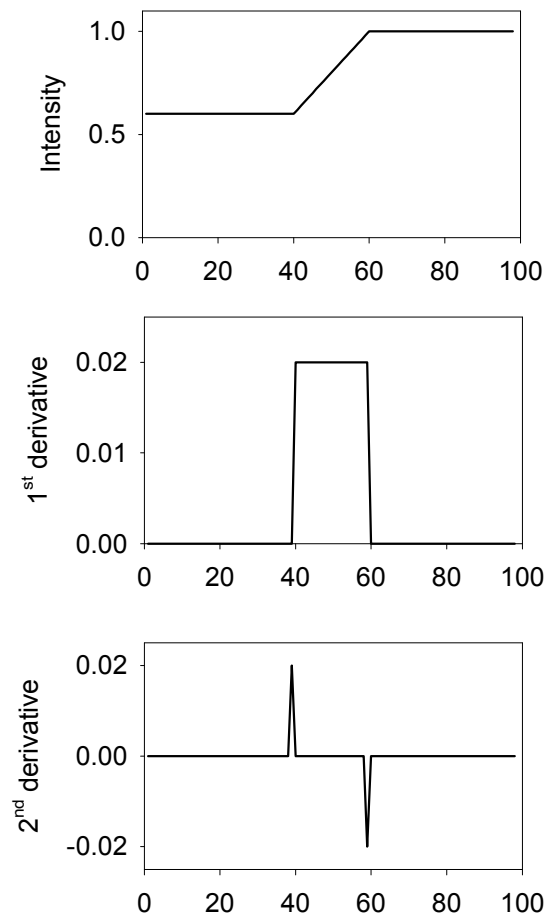
- Otsu módszerének lényege az csoportok közötti variancia (vagy SS) maximalizálása, ami ekvivalens a csoportokon belüli variancia (vagy SS) minimalizálásával (tehát minden csoportba olyan pixelek kerüljenek, amelyek hasonlóak egymáshoz intenzitásban).
- Algoritmus:
 - Minden lehetséges küszöbre ki kell számolni a csoportokon belüli SS értéket (kék téglalap).
 - Azt a küszöböt kell választani, amelyiknél a csoportokon belüli SS minimális.



küszöb:
66



Szegmentálás: Laplace operátor



- Egy függvény első és második deriváltja alkalmas a határvonalak detektálására.
- Egy kétváltozós függvény második deriváltja (Laplace operátor):

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- A második derivált szemléltetése:

Függvény	$f(1)$	$f(2)$	$f(3)$
Első derivált	$f(2)-f(1)$	$f(3)-f(2)$	
Második derivált	$f(3)-f(2)-(f(2)-f(1)) = f(3)+f(1)-2f(2)$		

- A fentiek értelmében a Laplace operátor x és y komponensei:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

- A Laplace operátort megvalósító digitális filter:

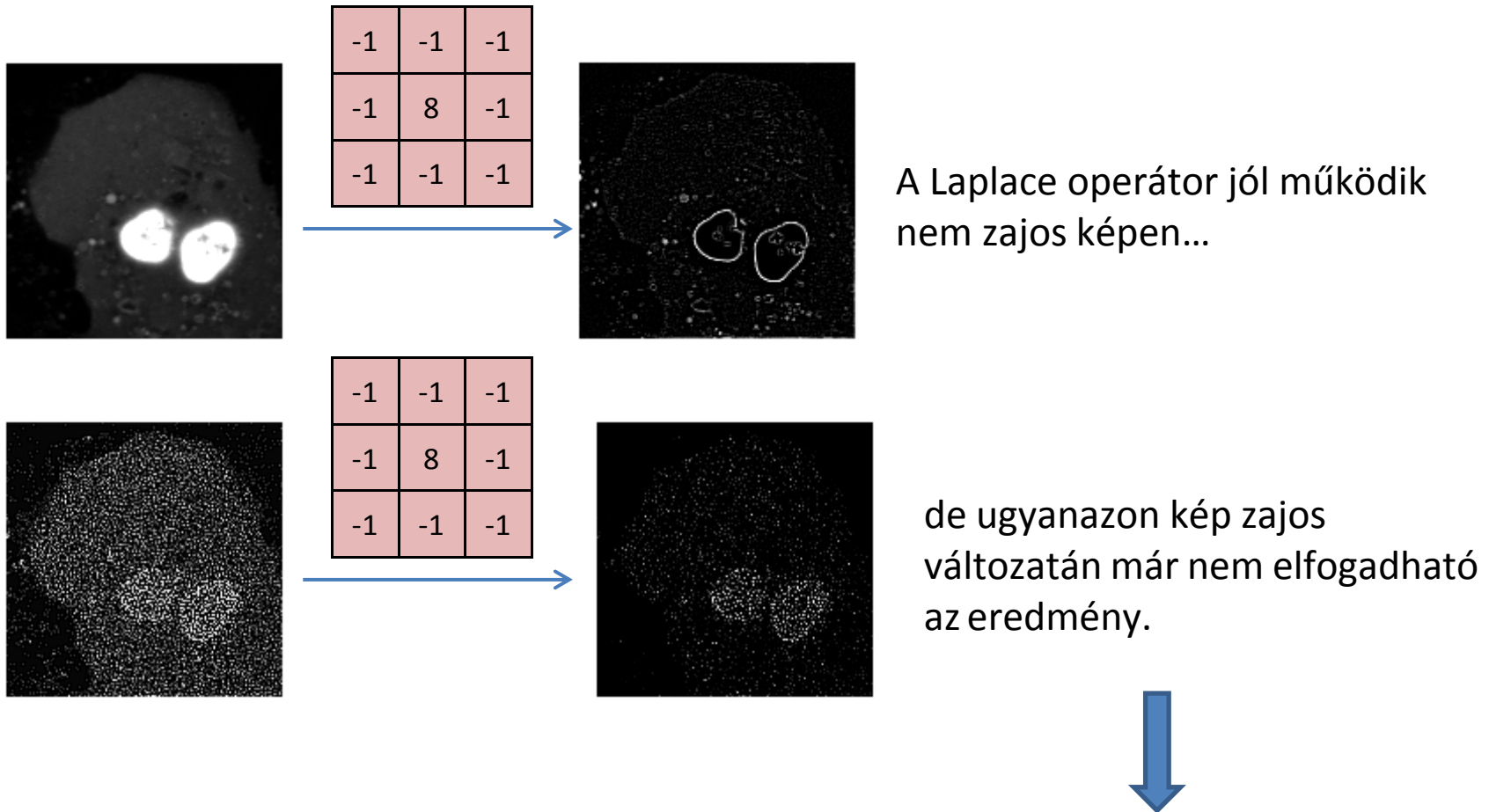
csak az x és y irányt figyelembe véve:

0	-1	0
-1	4	-1
0	-1	0

a diagonális irányokat is figyelembe véve:

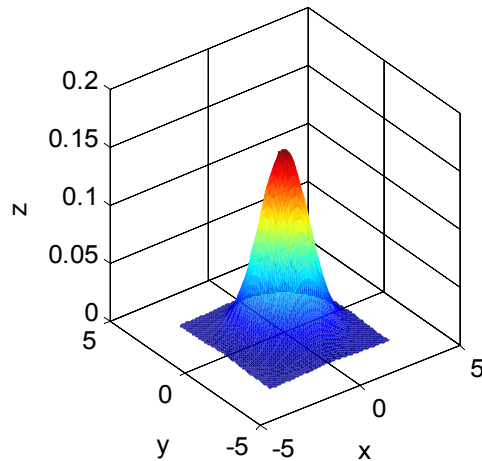
-1	-1	-1
-1	8	-1
-1	-1	-1

Szegmentálás: Laplace vs. LoG (Laplacian of Gaussian)

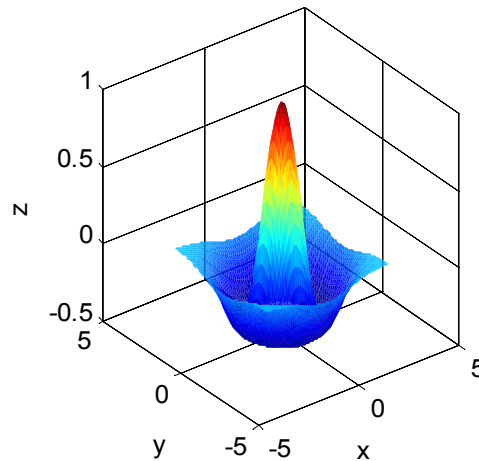


A Laplace operátor nagyon zajérzékeny, ezért alkalmazását mindig a kép simításával (aluláteresztő szűrő) kell összekötni → Gauss simítás majd Laplace operátor = LoG

Szegmentálás: LoG (Laplacian of Gaussian)



Gauss függvény



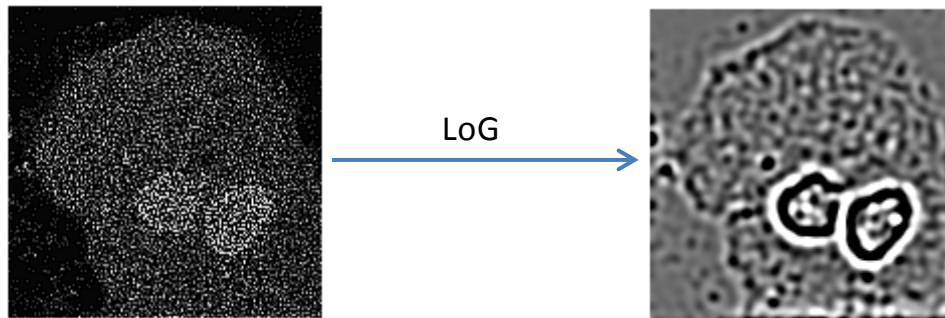
Gauss függvény második deriváltja=LoG

Ha a Gauss függvény második deriváltjának (LoG) megfelelő digitális filter segítségével dolgozzuk fel a képet, a zajszűrést és a határvonal detektálást egyszerre végezzük el.

A LoG operátor egyik digitális megfelelője:

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

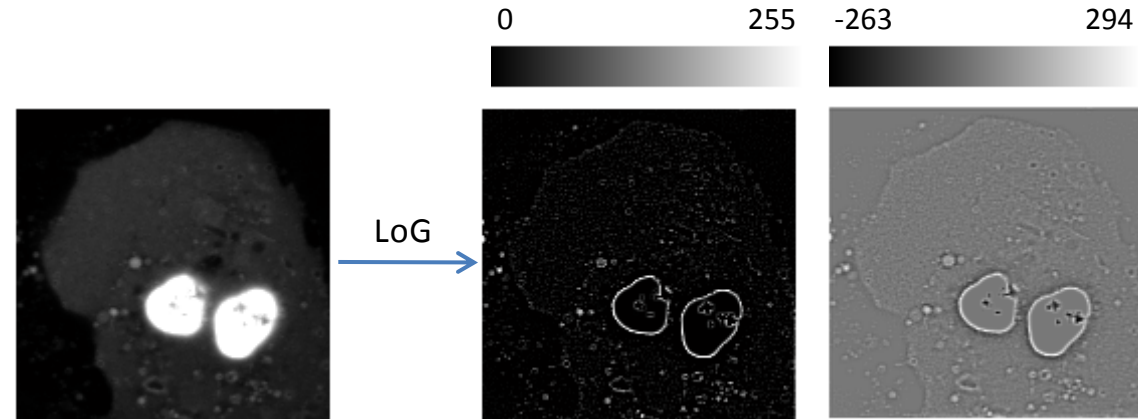
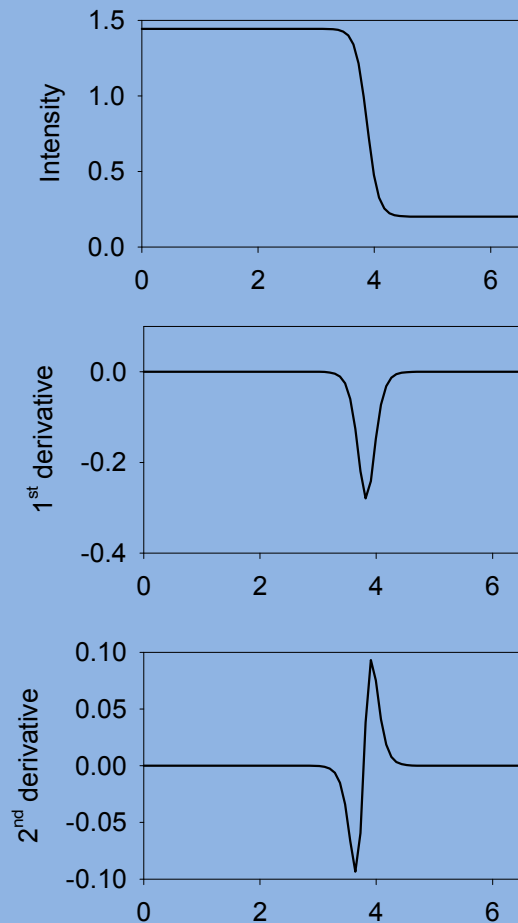
- A kép zajosságától függően lehet változtatni:
 - a kernel (filter méretét). A jobb oldali esetben 5×5.
 - a Gauss függvény szórását (minél nagyobb, annál jelentősebb a zajszűrő hatás)



Szegmentálás: LoG transzformált előjelváltása („zero-crossing”)

Bár a LoG filter által készített kép intenzitása jelzi a határt, nem klasszifikálja a pixeleket határvonal-nem határvonal osztályokba (pl. határvonal pixelekhez egyet rendel, a többihez nullát).

A második derivált előjelváltása (zero-crossing) jelzi a határvonal helyét.



előjelváltás helyeinek
megkeresése (zero-
crossing detection)

Az LoG transzformált előjelváltó
helyeinek megkeresésével a
határvonalak egyértelműen
detektálhatók
zero-crossing pixel – 1
minden egyéb pixel – 0



Szegmentálás az első derivált (gradiens) alapján

- Egy kétváltozós függvény (pl. kétdimenziós kép) első deriváltját x és y irányban a következő képletekkel kell meghatározni:

$$G_x = \frac{\partial f}{\partial x}, G_y = \frac{\partial f}{\partial y}$$

- G_x és G_y értékét a következő digitális operátorokkal lehet meghatározni Prewitt és Sobel szerint:

Prewitt

G_x

z_1

z_2

z_3

z_4

z_5

z_6

z_7

z_8

z_9

-1

0

1

-1

0

1

-1

0

1

-1

-1

-1

0

0

0

1

1

1

$z_3 + z_6 + z_9 - z_1 - z_4 - z_7$

$z_6 + z_7 + z_8 - z_1 - z_2 - z_3$

Sobel

G_x

-1

0

1

-2

0

2

-1

0

1

-1

-2

-1

0

0

0

1

2

1

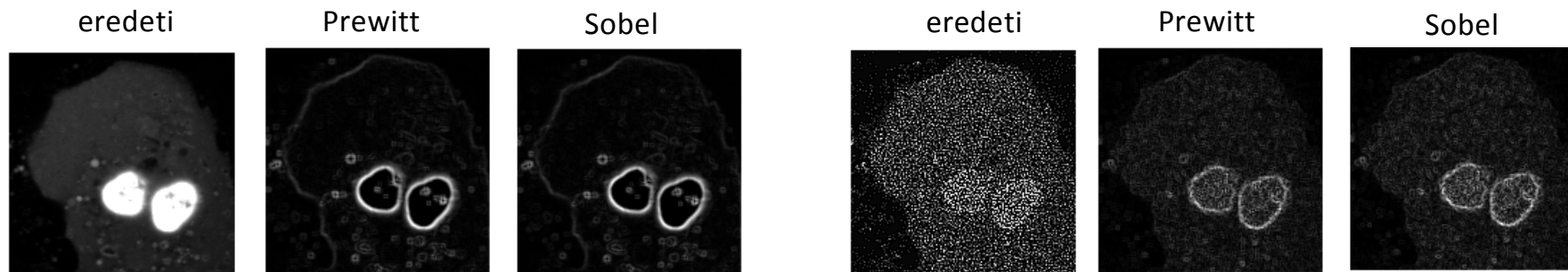
$z_3 + 2z_6 + z_9 - z_1 - 2z_4 - z_7$

$z_6 + 2z_7 + z_8 - z_1 - 2z_2 - z_3$

- A gradiens abszolút értékét pontosan és közelítőleg a következő képlet szerint kell kiszámolni:

$$|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

Azért használják az egyszerűsítést, mert sokkal kevesebb számítást igényel, de manapság ez egyre kevésbé szempont a gyors számítógépek elérhetősége miatt.



Szegmentálás: Canny határvonal detektálás

1. Simítás egy Gauss szűrő felhasználásával, pl. $\sigma=2$, 5×5 Gauss szűrő: $\longrightarrow$

2. Gradiens kiszámítása a Sobel operátor alapján.

3. Minden pixelben a gradiens irányának kiszámítása: $\Theta = \arctan\left(\frac{G_y}{G_x}\right)$

0.015	0.021	0.024	0.021	0.015
0.021	0.031	0.035	0.031	0.021
0.024	0.035	0.040	0.035	0.024
0.021	0.031	0.035	0.031	0.021
0.015	0.021	0.024	0.021	0.015

4. Nem-maximum elnyomás („non-maximum suppression”):

- kerekítsük a gradiens irányát 45° egész számú többszörösére (vízszintes, függőleges, átlós)
- hasonlítsuk össze a gradiens nagyságát azon szomszédos pixelben található értékkel, amely a gradiens irányában fekszik
- ha a vizsgált pixel gradiens értéke nagyobb, mint a másiké, akkor tartsuk meg az értéket; ha nem, akkor nullázzuk ki.

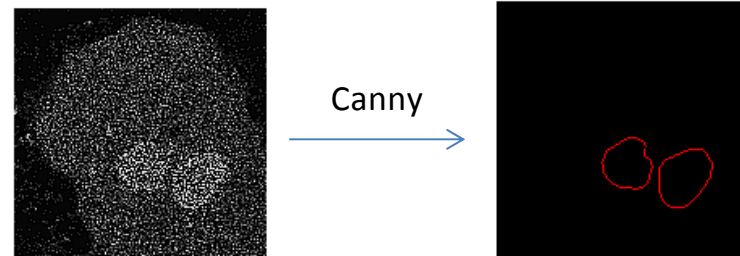
Ezáltal a határvonalat egy pixel szélességűre csökkentettük.

5. Kettős küszöbölés:

- a gradiens pixelek közül azok, amelyek a felső küszöbnél nagyobbak, megtartjuk (erős határvonal)
- az alsó küszöbnél kisebb gradiens értékeket nullázzuk
- a két küszöb közötti gradiens pixeleket (gyenge határvonal) megjelöljük további analízisre

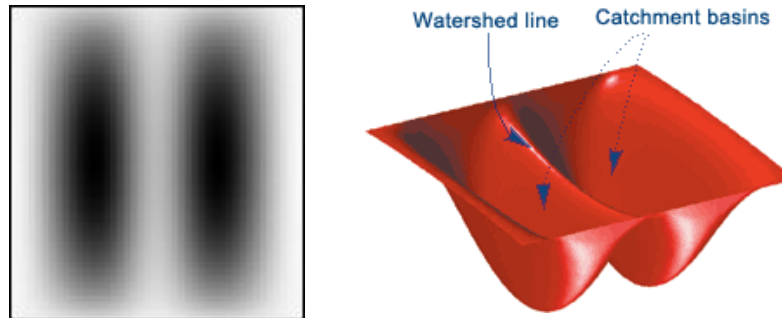
2↑	3↑	5↑	4↑	6↑
4↑	5↑	7↑	6↑	7↑
5↑	6↑	4↑	3↑	2↗
3↑	4↑	3↑	1↗	1↗

6. A gyenge határvonalak analízise: amennyiben egy gyenge határvonal erős határvonalhoz kötődik, megtartjuk, amennyiben nem, akkor töröljük.



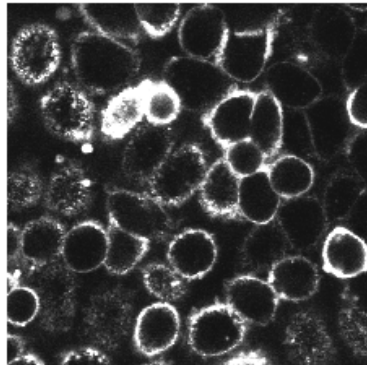
Szegmentálás: vízfeltöltési algoritmus (watershed segmentation)

- A képet egy 3D felszínként kell elképzelni, ahol a pixelek értéke a felszín magasságával egyenlő.

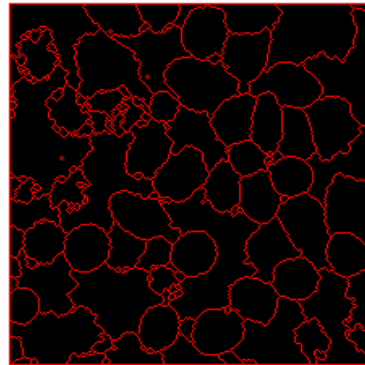


- Ha képzeletben vizet engedünk a fenti domborzati térképre, a víz a lokális minimumokat fogja megtalálni. Egy adott vízgyűjtő területről mindig ugyanazt a lokális minimumot. Két vízgyűjtő terület határa megfelel a képen azonosított objektumok határainak.
- Az algoritmus automatikus változatában az állítható paraméterektől függően általában vagy túl sok, vagy túl kevés határvonalat talál.
- Ezért sokkal hatékonyabb, ha a felhasználó kijelöli a sejteket („seeded watershed”) és akkor az algoritmus pontosan annyi sejtet talál, mint amennyi „seed” a képre lett helyezve.

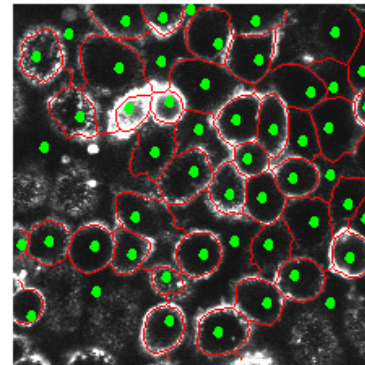
Szegmentálás: vízfeltöltéses algoritmus (watershed segmentation)



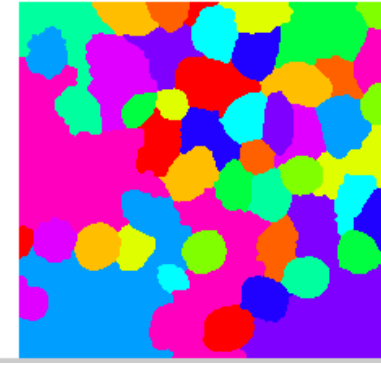
Eredeti kép



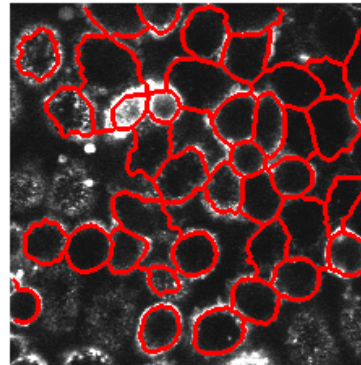
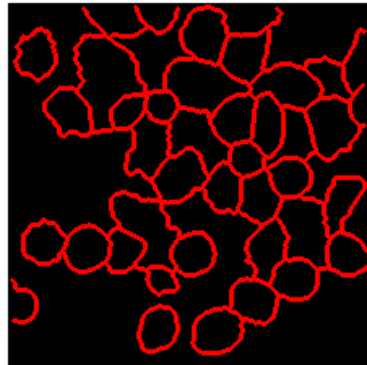
Automatikus watershed algoritmus



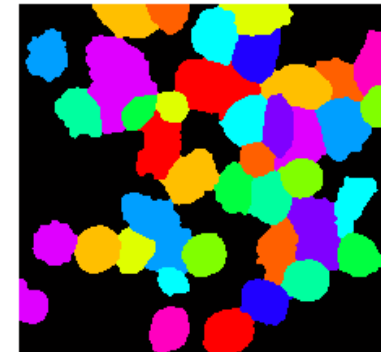
Manually-seeded watershed algoritmus a „seed”-ekkel



Elsődleges szegmentálás a manuálisan indított algoritmussal



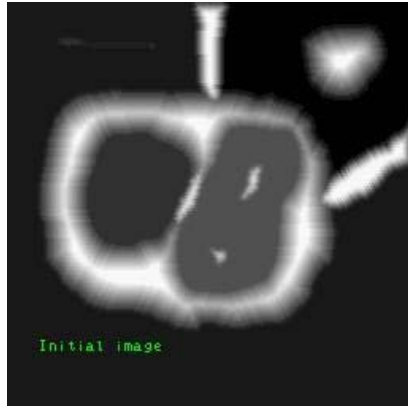
A manuális indított watershed algoritmus eredménye: balra a membránok, jobbra a membránok az eredeti képen



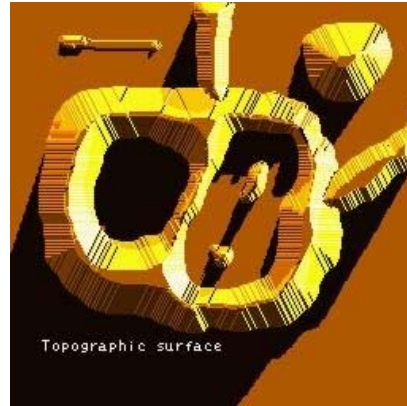
A nem sejtek tűnő objektumok törlése, mások összevonása

Szegmentálás: vízfeltöltéses algoritmus (watershed segmentation)

fluoreszcens kép
(szürke-skála)



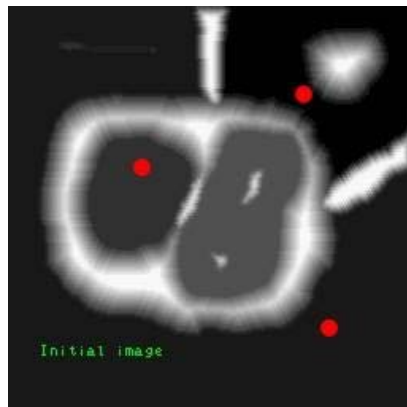
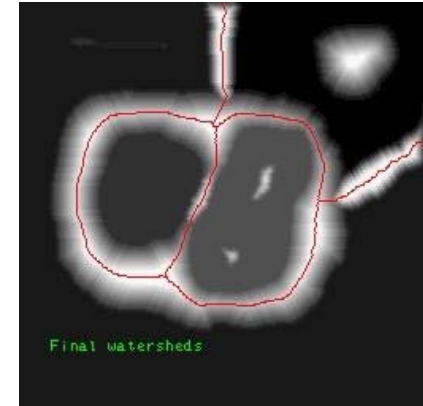
topográfiai kép



a képet „elárasztjuk”,
hogy az objektumok
határait megtaláljuk



az azonosított
határvonalak



az objektumok
azonosítása céljából a
felhasználó „magokat”
(jeleket) rak a képre



az elárasztást a
felhasználó által
kijelölt helyekről
indítjuk el



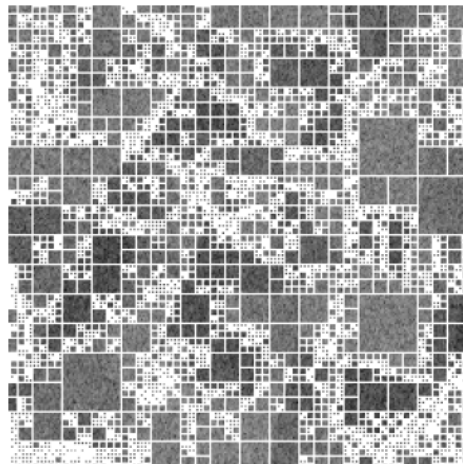
az azonosított
határvonalak

Szegmentálás: split and merge

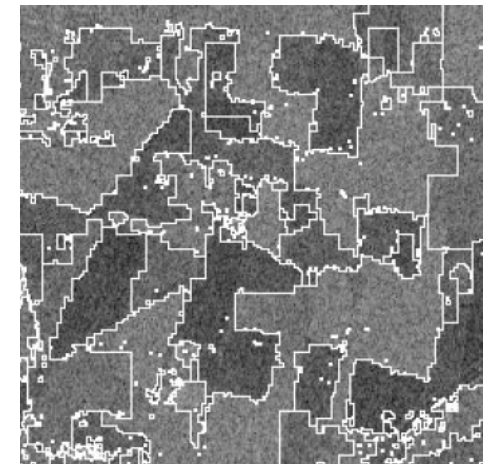
1. Válasszunk egy tetszőleges értéket, amihez az algoritmus során a területek varianciáját fogjuk hasonlítani.
2. Hasítási fázis (splitting):
 1. Számítsuk ki a kép varianciáját. Ha ez nagyobb, mint az első pontban választott érték, akkor hasítsuk a képet négy kvadránsra.
 2. Ha akármelyik kvadráns varianciája meghaladja a választott értéket, azt is hasítsuk négy kvadránsra.
 3. Ezt a folyamatot addig kell folytatni, amíg minden halmaz (kvadráns) varianciája a választott érték alá csökken.
3. Összeolvasztás (merging):
 1. Minden egymással határos területnél meg kell vizsgálni, hogy a két határos terület összeolvasztása olyan halmazt eredményez-e, aminek a varianciája az egyes pontban választott érték alatt van. Ha igen, akkor össze kell őket olvasztani.
 2. A folyamatot az így keletkezett új területekkel is meg kell ismételni, amíg további összeolvasztás már nem végezhető a fenti szabály szerint.



Eredeti



A hasítási fázis végeredménye



Az összeolvasztási fázis végeredménye

Szegmentálás: k-means clustering

1. Minden pixelre adott n különböző mérés (pl. pixel intenzitás $\lambda_{em}=520$ nm, $\lambda_{em}=670$ nm, stb). n lehet egy is. Tehát a következő adathalmazból indulunk ki:

	1. mérés	2. mérés	...	n. mérés
1. pixel	$f(1,1)$	$f(1,2)$	...	$f(1,n)$
2. pixel	$f(2,1)$	$f(2,2)$	...	$f(2,n)$
3. pixel	$f(3,1)$	$f(3,2)$	...	$f(3,n)$
...	...	...	...	...
k. pixel	$f(k,1)$	$f(k,2)$	...	$f(k,n)$

2. Elhatározzuk, hogy hány klasztert szeretnénk azonosítani (N).
3. Minden klaszterhez meghatározzuk a kezdeti középpértékeket:

	1. mérés	2. mérés	...	n. mérés
1. klaszter	$\mu(1,1)$	$\mu(1,2)$	...	$\mu(1,n)$
2. klaszter	$\mu(2,1)$	$\mu(2,2)$	...	$\mu(2,n)$
3. klaszter	$\mu(3,1)$	$\mu(3,2)$	...	$\mu(3,n)$
...	...	...	...	...
N. klaszter	$\mu(N,1)$	$\mu(N,2)$	...	$\mu(N,n)$

4. Minden pixelnél meghatározzuk, hogy melyik klaszter középpontjához van legközelebb, tehát minden pixelhez az (1...N) számok valamelyikét rendeljük.
5. Az így létrehozott klasztereknek megint kiszámoljuk a középpontját.
6. A ciklust addig folytatjuk, míg egyetlen pixel sem vált klasztert.

Pontok távolságának meghatározása: Mahalanobis távolság

Egy vektor $x=(x_1,x_2,x_3,...,x_n)$ távolságát egy másik vektortól $\mu=(\mu_1,\mu_2,\mu_3,...,\mu_n)$ a következő képlettel kell kiszámolni:

$$D(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}) \mathbf{V}^{-1} (\mathbf{x} - \boldsymbol{\mu})^T}$$

ahol $\mathbf{V}$ a variancia-kovariancia mátrix, -1 felső indexben a mátrix inverziót és a T felső indexben a mátrix transzpozíciót jelzi.

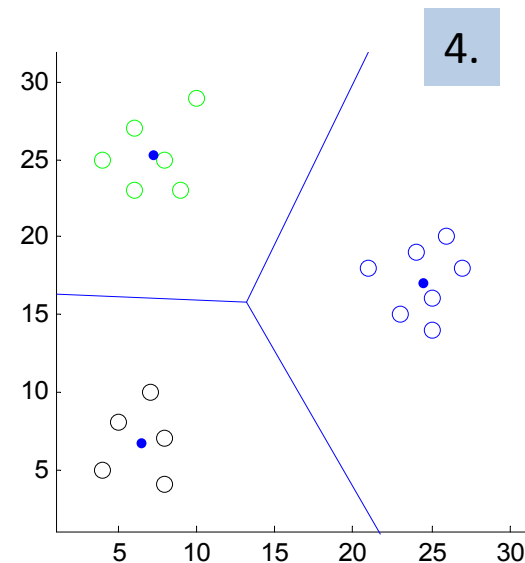
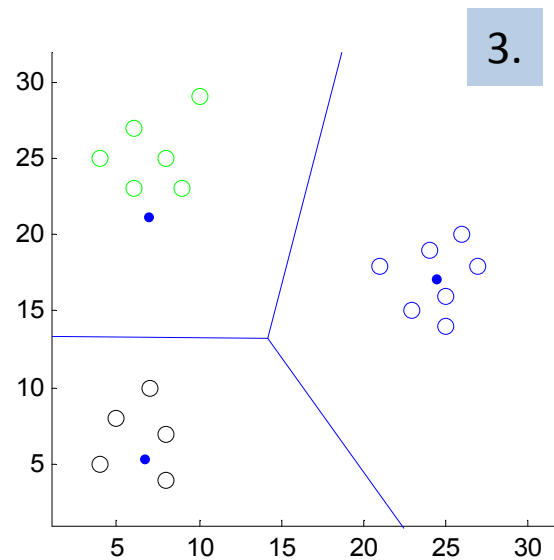
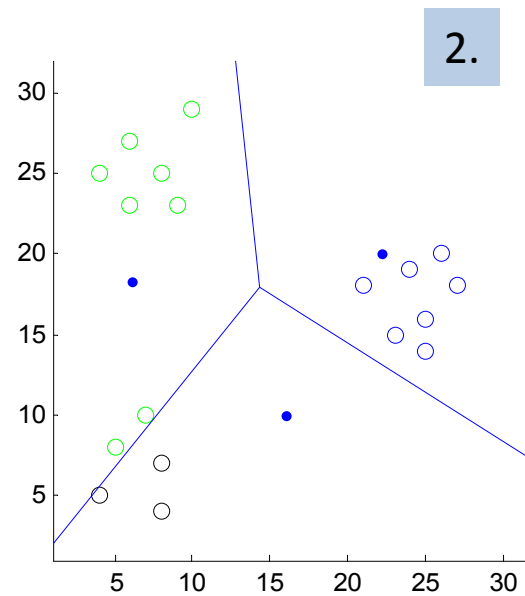
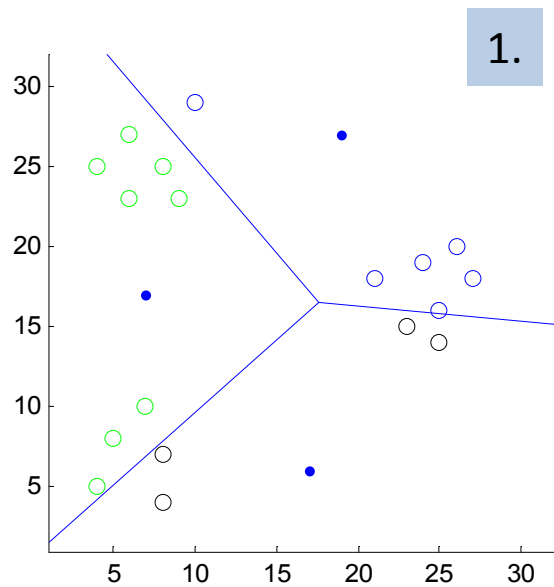
Variancia-kovariancia mátrix:

$$\begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 & \dots & \sigma_{1,n}^2 \\ \sigma_{2,1}^2 & \sigma_{2,2}^2 & \dots & \sigma_{2,n}^2 \\ \dots & \dots & \dots & \dots \\ \sigma_{n,1}^2 & \sigma_{n,2}^2 & \dots & \sigma_{n,n}^2 \end{bmatrix}, \sigma_{x,y}^2 = M \left[(x - \mu_x)(y - \mu_y) \right]$$

$\sigma_{1,1}$ az első mérés varianciája, $\sigma_{1,2}$ az első mérés és második mérés kovarianciája.

diagonális mátrix	normalizált euklideszi távolság	egység mátrix	euklideszi távolság
Ha $\begin{bmatrix} \sigma_{1,1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{2,2}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_{n,n}^2 \end{bmatrix}$	$\Rightarrow D(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{\sum_i \frac{(x_i - \mu_i)^2}{\sigma_{i,i}^2}}$	Ha $\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$	$\Rightarrow D(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{\sum_i (x_i - \mu_i)^2}$

Szegmentálás: k-means clustering



A k-means clustering algoritmus négy lépés alatt konvergál ezen két változós klaszterizációs probléma esetén.

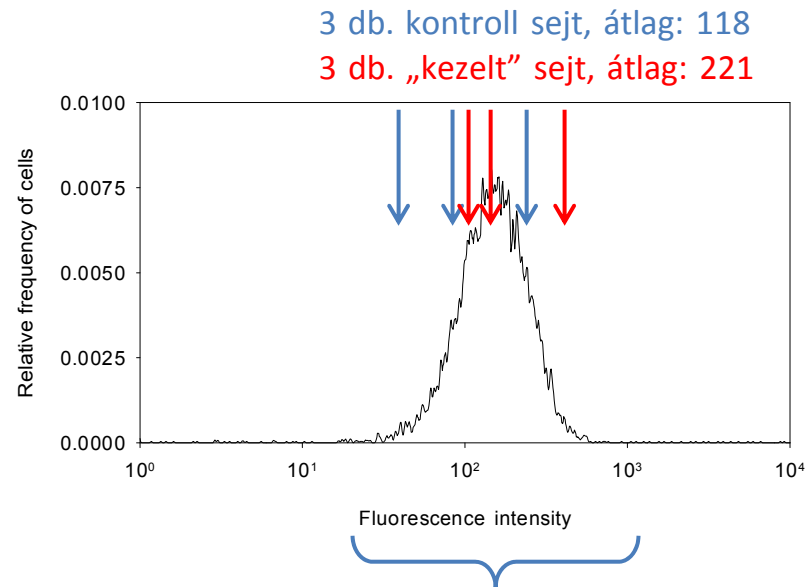
Háttérkivonás

Olyan algoritmusok, melyek a eltávolítják a háttér fluoreszcencia intenzitást és csak az objektumok specifikus jelöléséből származó intenzitást hagyják meg.

- konstans háttér levonás: a kép minden pixeléből levonjuk ugyanazt a háttér
- minden felüláteresztő szűrő: a háttér alacsony frekvenciájú komponenseket tartalmaz, amelyek felüláteresztő szűrővel eltávolíthatók
- max-min filter
- tophat filter

Mérések

- Az objektumok („foreground”) azonosítása után (szegmentálás) a kvantitatív információ kinyerése a kép feldolgozásának utolsó lépése.
- Miért kell kvantitatív kiértékelés?
 - hogy a képek alapján levont konklúzió ne legyen szubjektív
 - mikroszkópiával kevés sejtet tudunk megmérni, ami a mikroszkópiát statisztikailag kevésbé robosztussá teszi
 - ezt a képek kvantitatív kiértékelésével lehet csökkenteni
 - minél több sejtet érdemes lemérni és kiértékelni, hogy csökkenjen a statisztikák hibája

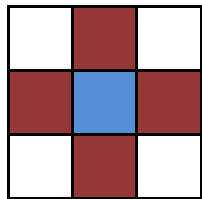


Gyakori, hogy a biológiai minták 1-2 nagyságrendnyi variabilitást mutatnak. Ha néhány sejt mikroszkópos analízise alapján vonunk le következtetést, könnyen hibát véthetünk.

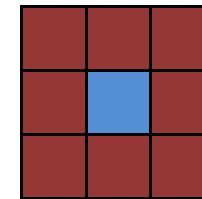
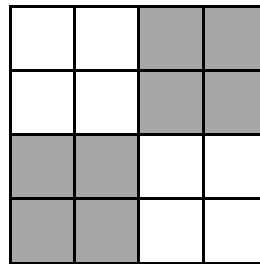
Mérések: 4- vagy 8-szoros kapcsolódás, Freeman lánckód

A mérések szempontjából fontos fogalom a pixelek összekapcsoltsága („connectivity”), tehát hogy milyen viszonyban levő pixeleket tekintünk egymással határosnak:

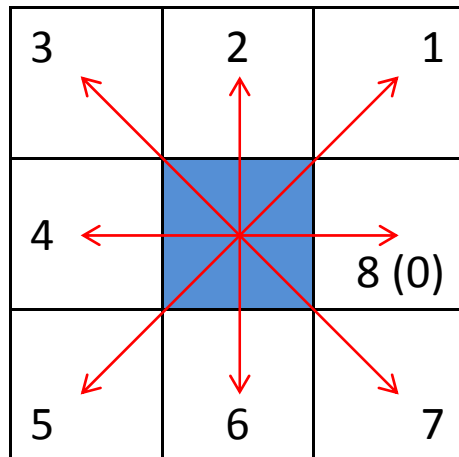
- 4-szeres kapcsolat (4-connected):
egy pixel négy másikkal határos (csak az oldalak menti kapcsolat számít)



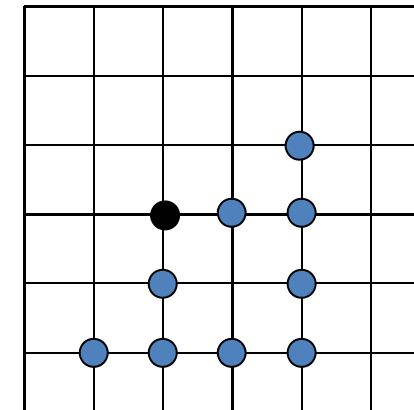
- 8-szoros kapcsolat (8-connected)
egy pixel nyolc másikkal határos (az oldalaknál és csúcsoknál levő kapcsolat is számít)



4-szeres kapcsolódás szerint ez két objektumnak számít, 8-szoros kapcsolódás szerint viszont egynek.



Egy adott pixeltől az óramutató járásának megfelelően indulva minden következő pixelnek egy kódot adunk annak megfelelően, hogy milyen irányban kellett azt elérni az előző pixelből.



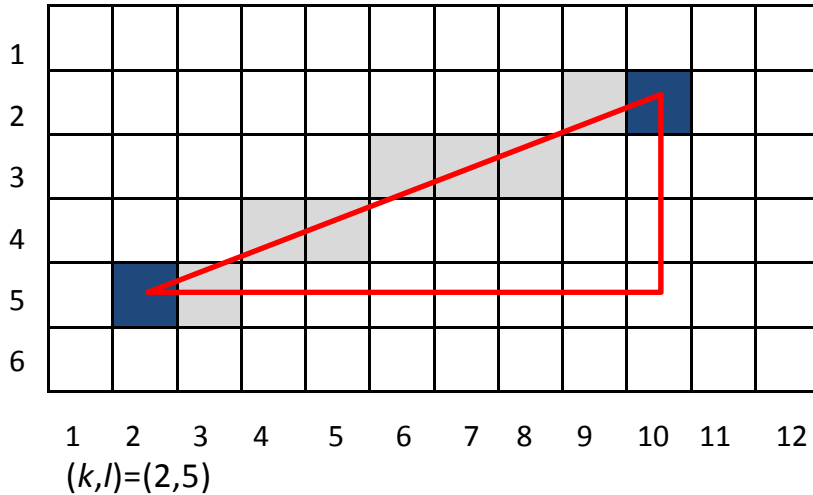
A fenti területet az alábbi kód írja le a fekete ponttól indulva:

8 → 1 → 6 → 6 → 6 → 4 → 4 → 4 → 1 → 2

Mérések: távolságmérés

Mennyi a két kék pont távolsága?

$(i,j)=(10,2)$



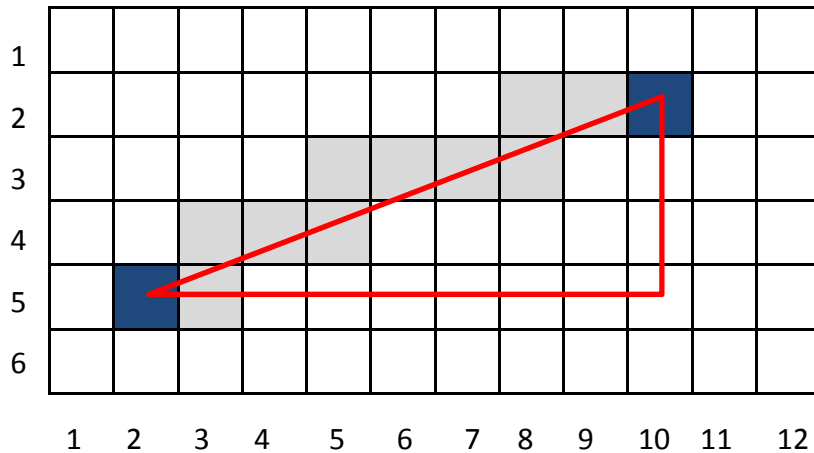
1. Euklideszi távolság:

$$\sqrt{(i-k)^2 + (j-l)^2} = \sqrt{(10-2)^2 + (2-5)^2} = 8.5$$

2. Sakktábla távolság (chessboard distance):

- a sakk király hány lépéssel tud az egyik pontból a másikba menni
- a 8-szoros összeköttetésű pixelek legrövidebb sorozata, ami összeköti a két pontot

$$\max(|i-k|, |j-l|) = 8$$



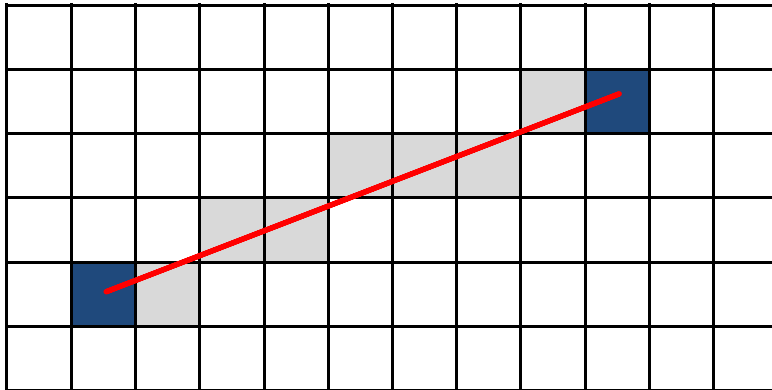
3. Városi blokk távolság (city-block distance):

- a 4-szeres összeköttetésű pixelek legrövidebb sorozata, amely összeköti a két pontot

$$|i-k| + |j-l| = 11$$

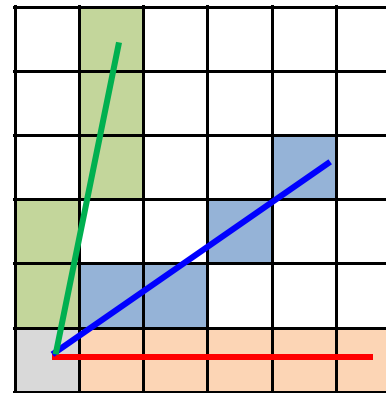
Mérések: egy görbe vagy vonal hosszának meghatározása

8-szoros kapcsolat

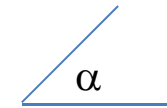


Ha egy vonal n pixellel írható le, akkor annak hossza $n/0.9$.

A fenti vonal hossza: $8/0.9=8.88$



d hosszúságú vonalak különböző szögökben



A pixelek száma:

• $5 \cdot \cos 0^\circ = 5$

• $5 \cdot \cos 30^\circ = 4$

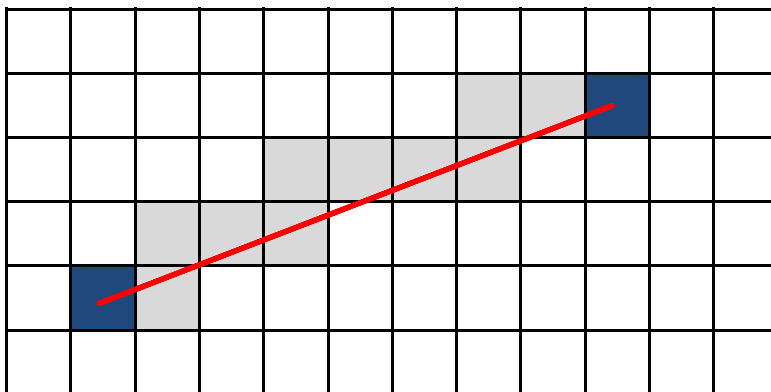
• $5 \cdot \sin 80^\circ = 5$

$n_{\text{pixel}} = d \cos \alpha$, ha $\alpha \leq 45^\circ$

$n_{\text{pixel}} = d \sin \alpha$, ha $\alpha > 45^\circ$

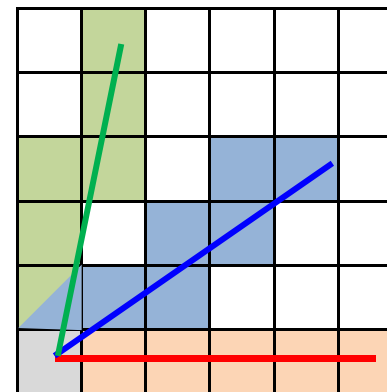
Ennek átlaga 0° és 45° között: $\frac{4}{\pi} \int_0^{\pi/4} \cos \alpha \, d\alpha = \frac{4}{\pi\sqrt{2}} = 0.9$

4-szeres kapcsolat



Ha egy vonal n pixellel írható le, akkor annak hossza $n/1.273$.

A fenti vonal hossza: $8/1.273=6.28$



$n_{\text{pixel}} = d (\cos \alpha + \sin \alpha)$

A pixelek száma:

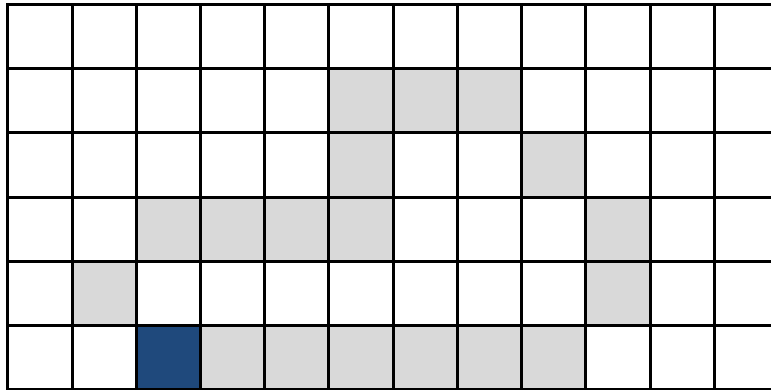
• $5 \cdot (\cos 0^\circ + \sin 0^\circ) = 5$

• $5 \cdot (\cos 30^\circ + \sin 30^\circ) = 7$

• $5 \cdot (\cos 80^\circ + \sin 80^\circ) = 6$

Ennek átlaga 0° és 45° között: $\frac{4}{\pi} \int_0^{\pi/4} (\cos \alpha + \sin \alpha) \, d\alpha = \frac{4}{\pi} = 1.273$

Mérések: egy görbe hosszának meghatározása

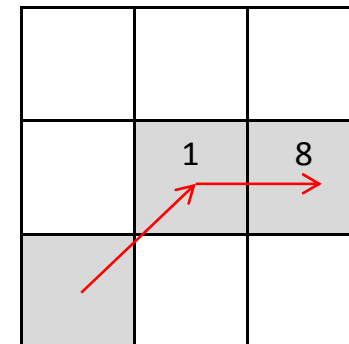


párhuzamos lépések – páros Freeman lánckód
átlós lépések – páratlan Freeman lánckód

	a hálózattal párhuzamos lépések (14)	átlós lépések (5)	sarkok száma (3)	a fenti kerület hossza
a pixelek száma	1	1	0	19
Freeman (1970)	1	$\sqrt{2}$	0	21.071
Proffit (1979)	0.984	1.340	0	20.476
Vossepoel (1982)	0.980	1.406	-0.091	20.477

Sarok:

- a páros-páratlan és páros-páratlan váltások száma a Freeman lánckódban.
- a sakkbeli huszárnak megfelelő lépések



$$D_{count} = n_e + n_o$$

$$D_{Freeman} = n_e + \sqrt{2}n_o$$

$$D_{Proffit} = 0.984n_e + 1.340n_o$$

$$D_{Vossepoel} = 0.98n_e + 1.406n_o - 0.091n_c$$

n_e – páros (even) Freeman kódok száma

n_o – páratlan (odd) Freeman kódok száma

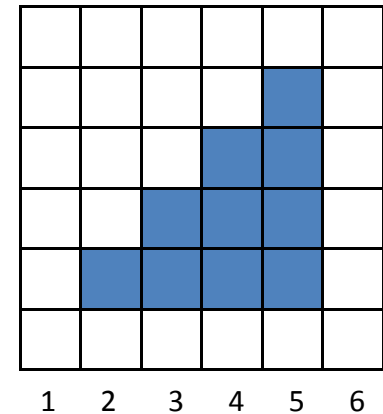
n_c – sarkok száma

Mérések: terület meghatározása

Mennyi a bal oldali háromszög területe?

- 1 becslés: $A_1 = a$ pixelek száma=10
- 2. becslés: $A_2 = \frac{1}{2} (3 \times 3) = 4.5$ (háromszög területének képlete alapján, ha az oldalhossz=végpont-kezdőpont=5-2=3)
- 3. becslés: $A_3 = \frac{1}{2} (4 \times 4) = 8$ (háromszög területének képlete alapján, ha az oldalhossz=pixelok száma=4)

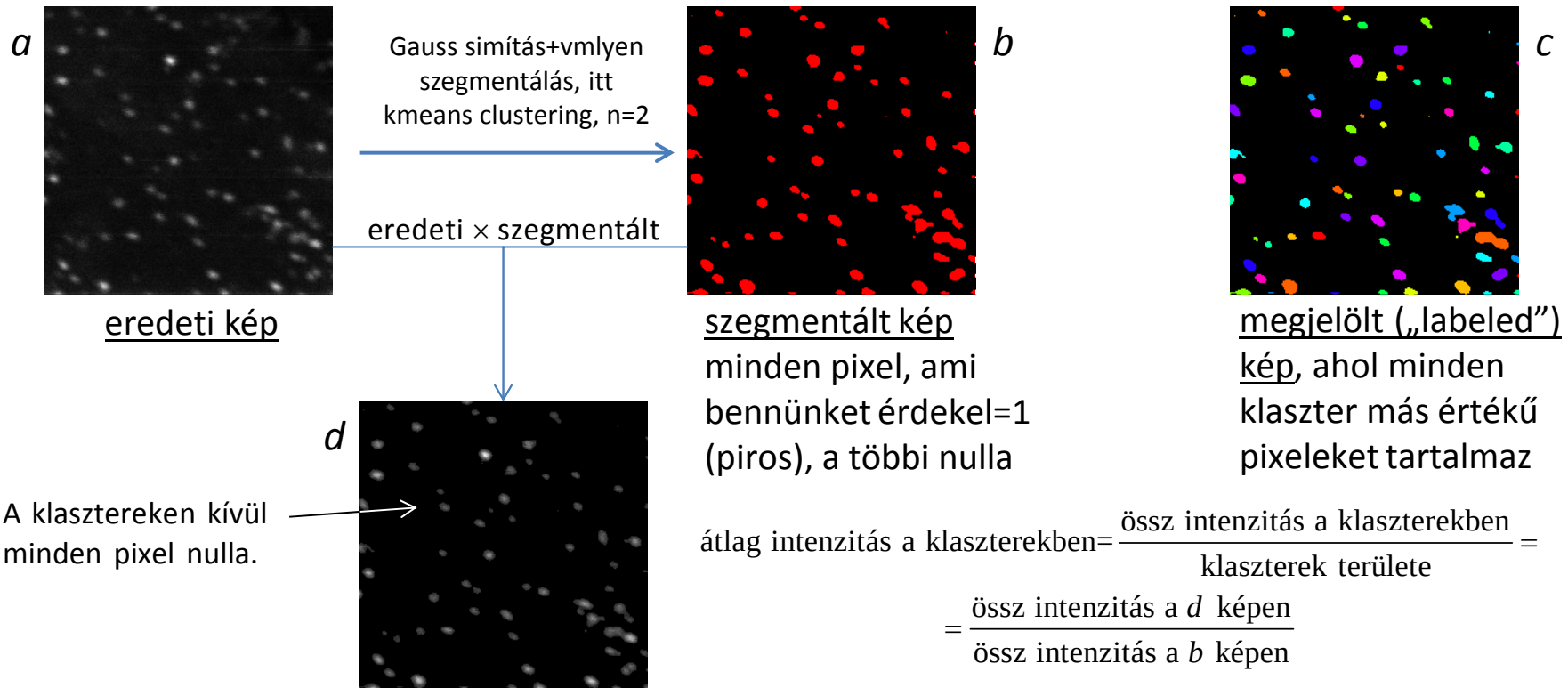
Mindhárom eljárás elfogadott.



Hossz, kerület és terület számolás konklúziója:

- a képeken a valódi objektum digitalizált változatát látjuk
- a mérés során a valódi objektum paramétereire (kerület, terület) vagyunk kíváncsiak
- mivel a valódi objektumnak a digitális kép csak közelítése, ezért a digitális kép alapján végzett számítások is csak becslést tudnak adni a valódi objektum paramétereire

Mérések: átlag intenzitás meghatározása

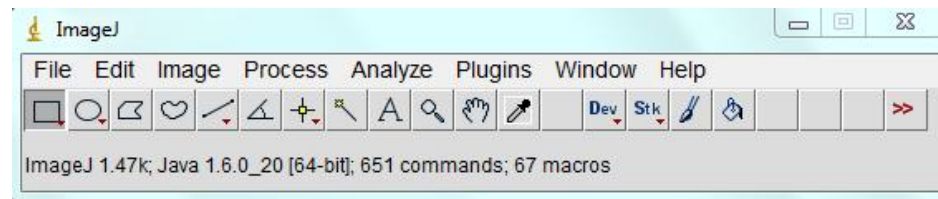


- Összes klaszter területe: a piros pixelek összege (hiszen minden piros pixel=1; a szegmentált kép az „*b*” változóban van):
 - Matlab: `area=sum(b(:))` DiplImage: `area=sum(b);`
- Az eredeti képen az összes intenzitás a klasztereknek megfelelő területen:
 - `d=a*b;`
 - Matlab: `totint=sum(d(:));` DiplImage: `totint=sum(d);`
- Átlag intenzitás a klaszterekben:
 - `totint/area`
- **A fenti számítást minden klaszterre egyenként is el lehet végezni a „*c*” kép alapján.**

DiplImage: `sum(a(b));`
 Matlab:
`sum(a (reshape(b,numel(b),1)==1))/`
`sum(b(:))`

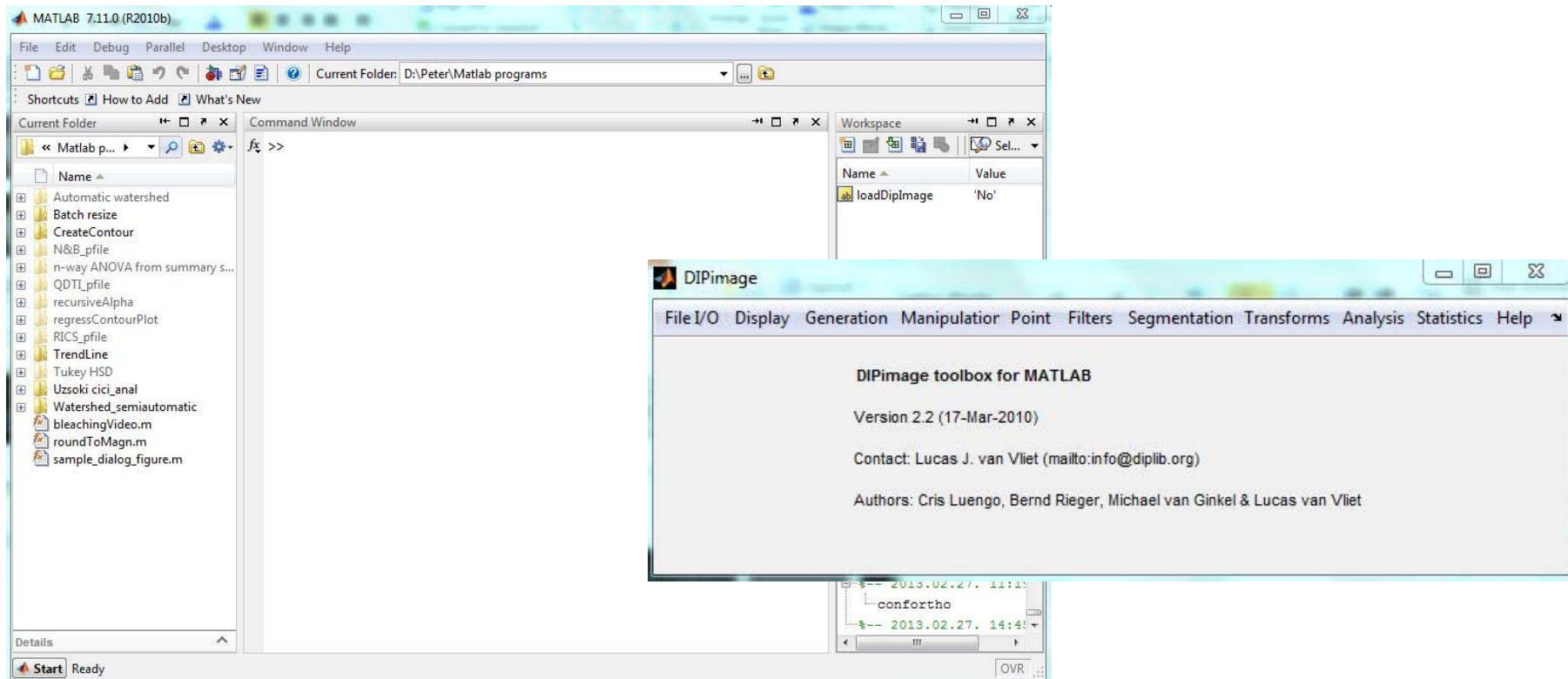
Tudományos képanalizáló programok

- Tudományos kép kiértékeléshez nem használhatók a legközismertebb programok (Powerpoint, Adobe Photoshop, Corel Photopaint)
- A tudományos kéпкиértékelő programok száma nagy, itt csak egy-két példa látható.



- ImageJ (rsbweb.nih.gov/ij/)
 - ingyenes, biológia orientált, rendszeresen frissített, felhasználóbarát
 - ún. plugin-ek segítségével bővíthető (rsbweb.nih.gov/ij/plugins/index.html)
 - többféle szintű használat:
 - menüpontok segítségével
 - makrók írásával (a menüből elérhető parancsok sorozatának elmentése)
 - Java programok írása (plugin)

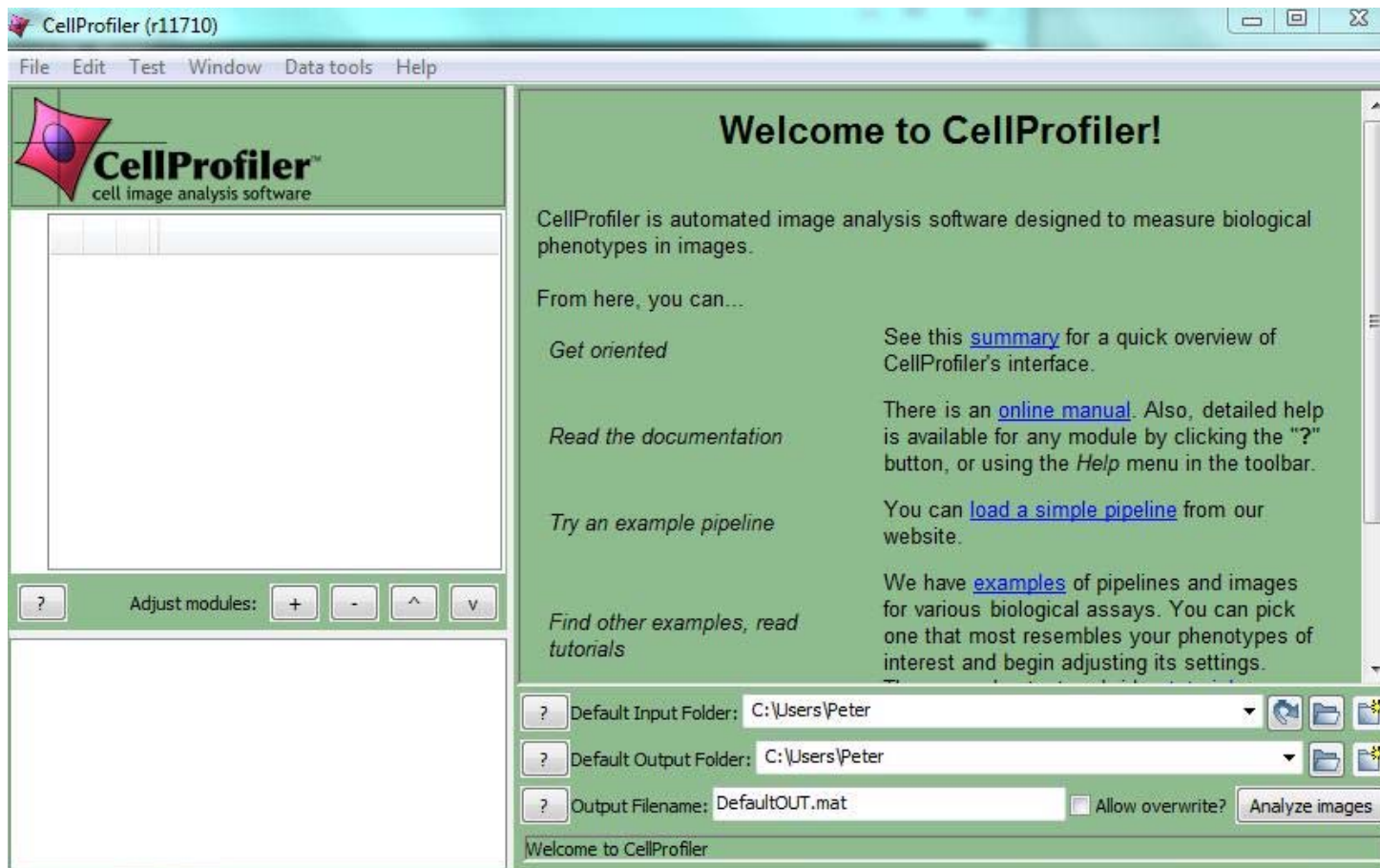
Tudományos képanalizáló programok



- Matlab (www.mathworks.com/products/matlab/)
 - nem ingyenes
 - begépelt parancsokkal vezérelhető → nem felhasználóbarát
 - saját, rendkívül hatékony programozási nyelven írt programokkal (ún. M-file) bővíthető (sok letölthető: www.mathworks.com/matlabcentral/)
 - Matlab-bal minden lehetséges...
- DipImage (<http://www.diplib.org/>)
 - Matlab alá installálható ingyenes kép analizáló eszköztár
 - menüvezérelt+a Matlab parancskészletét bővíti speciális képanalizáló függvényekkel

Tudományos képanalizáló programok

- Cell Profiler (www.cellprofiler.org/)
 - ingyenes
 - gyakran használt képkiértékelő lépésekből menü segítségével egy algoritmus építhető, ami tetszőleges számú kép kiértékelését elvégzi automatikusan („high-throughput” képkiértékelés)



Tudományos képanalizáló programok

- Metamorph (www.moleculardevices.com/products/software/meta-imaging-series/metamorph.html)
 - nem ingyenes
 - mikroszkóp vezérlés és kép kiértékelés
 - felhasználóbarát
 - jelentős kép kiértékelő és 3D megjelenítő képességek
- Image Pro (www.mediacy.com)
 - nem ingyenes
 - mikroszkóp vezérlés és kép kiértékelés
 - felhasználóbarát
 - jelentős kép kiértékelő és 3D megjelenítő képességek
- Általában minden konfokális mikroszkópnak van saját szoftvere, amivel egyszerű képkiértékelések elvégezhetők.
 - Ezen programok ingyenesen letölthető változata „lebutított” → csak a legegyszerűbb műveletek végezhető el, de azok effektíven és felhasználóbarát módon.