

Digitális képfeldolgozás feladatgyűjtemény

Khoros Cantata és VisiQuest rendszerhez

Készítették:

Dr. Tanács Attila, Domokos Csaba, Gara Mihály,
Kardos Péter, Németh Gábor, Németh József

Szegedi Tudományegyetem
Képfeldolgozás és Számítógépes Grafika Tanszék

2009 - 2010.

Verzió 1.3

Tartalomjegyzék

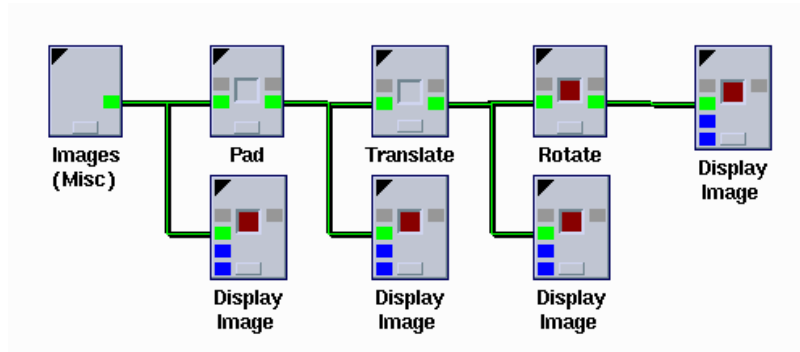
1. Affin transzformációk	4
2. Képmanipulációk, mintavételezés, kvantálás.....	11
3. Fourier-transzformáció, konvolúció.....	17
4. Pontoperációk.....	23
5. Programozási feladatok.....	33

1. Affin transzformációk

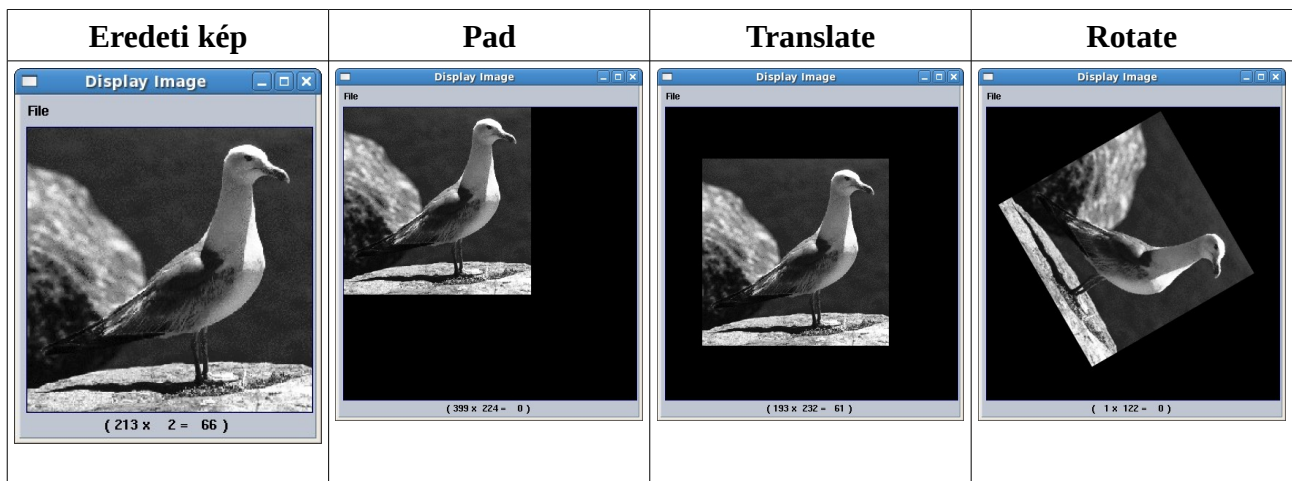
1.a) Példa.

Forgassuk el a sirályos képet 45°-kal! Ügyeljünk arra, hogy forgatás során a teljes kép látható legyen!

Megoldás:



A sirályos kép mérete 256x256. Növeljük meg a kép méretét a **Pad** dobozzal 512x512-re! Ekkor a kép mérete már nagyobb lesz, de a sirály a kép bal felső negyedében fog elhelyezkedni, ezért toljuk el a képet a **Translate** dobozzal X és Y irányban is 128-cal. Így a sirály a kép középre kerül. Ezt követően forgassuk el a **Rotate** dobozzal 45°-kal!



1.1. Kép transzponálása.

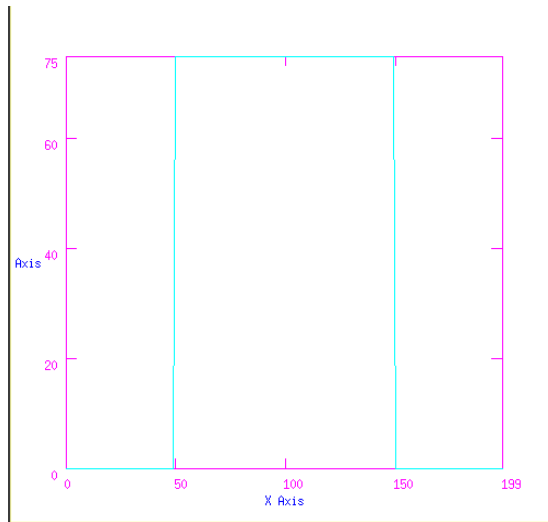
Nehézségi fok: 1

Transzponáld a sirályos képet és jelenítsd meg az eredeti és a transzponált képet is!

1.2. Állítsd elő a képen látható jelet!

Nehézségi fok: 2

Az 1-dimenziós kép mérete legyen 200. A jel 50-nél kezdődjön és 150-nél szálljon le. A jel tetőpontja legyen 75.



1.3. Kép tükrözése.

Nehézségi fok: 1

Készíts műveletsort, amely a beépített szürkeárnyaltos képeket függőlegesen, vízszintesen, és mindkét tengely mentén tükrözi.

1.4. Kép eltolása.

Nehézségi fok: 1

Egészítsd ki a Spanish Seagull kép méretét kétszeresére és told el a sirályt a kép közepére!

1.5. Kép forgatása.

Nehézségi fok: 2

Egészítsd ki a Spanish Seagull kép méretét kétszeresére, told a kép közepére és forgasd el 60°-kal!

1.6. Affin transzformációk.

Nehézségi fok: 2

Forgasd el a Marine képet 90°-kal úgy, hogy az eredménykép mérete megegyezzen az eredeti képpel! Ne legyenek levágott képrészletek! (A kép méretét lekérdezheted a **Print Data** és **File Viewer** dobozokkal.)

1.7. Képrészlet kivágása.

Nehézségi fok: 2

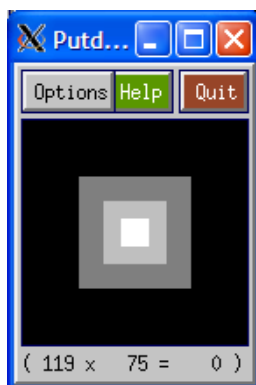
Nyisd meg a sirályos és a labdás képet! Vágd ki a sirály fejét, és helyezd el a labdás kép közepén!

1.8. Képrészletek összeillesztése.

Nehézségi fok: 2

Állítsd elő az ábrán látható képet! A kép mérete 128x128, a négyzetek méretei és intenzitásai:

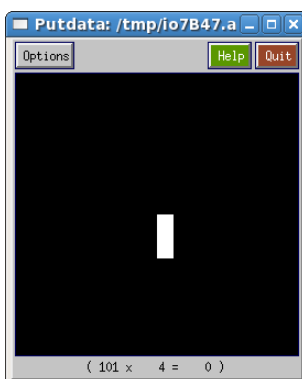
- méret: 64x64 intenzitása: 100
- méret: 36x36 intenzitása: 150
- méret: 16x16 intenzitása: 200



1.9. Téglalap rajzolása.

Nehézségi fok: 2

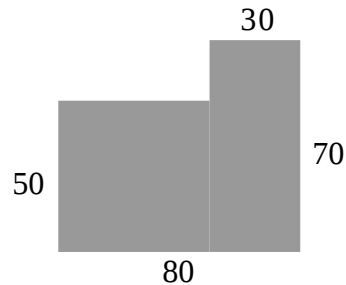
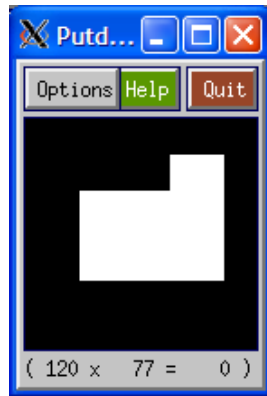
Készíts műveletsort, amely az ábrán látható téglalapot! A kép mérete legyen 256x256-os. A fehér téglalap mérete legyen 15x40. A téglalapot helyezd a kép közepére! Forgasd el 45°-kal! Jelenítsd meg mindkét képet!



1.10. Alakzat rajzolása.

Nehézségi fok: 2

Készíts műveletsort, amely az alábbi alakzatot rajzolja ki! A képméret 128x128 legyen! A pozíció nem fontos, de ne a kép szélén helyezkedjen el!



1.11. Kép transzformálása.

Nehézségi fok: 3

Növeld meg a sirályos kép méretét, helyezd a sirályt a kép közepére, forgasd el 45°-kal és nyújtsd meg az Y tengely mentén kétszeresére!

1.12. Kép nagyítása, kicsinyítése.

Nehézségi fok: 1

Kicsinyítsd a Spanish Seagull képet negyedére, felére, és nagyítsd kétszeresére, négyszeresére! Jelenítsd meg az összes eredményt.

1.13. Mozaik kép készítése.

Nehézségi fok: 2

Készíts műveletsort, amely a *mosaic.xv* kép felhasználásával az alábbi mozaik képet eredményezi!



mosaic.xv



1.14. Mozaik kép készítése.

Nehézségi fok: 2

Készíts műveletsort, amely a *mosaic.xv* kép felhasználásával az alábbi mozaik képet eredményezi!



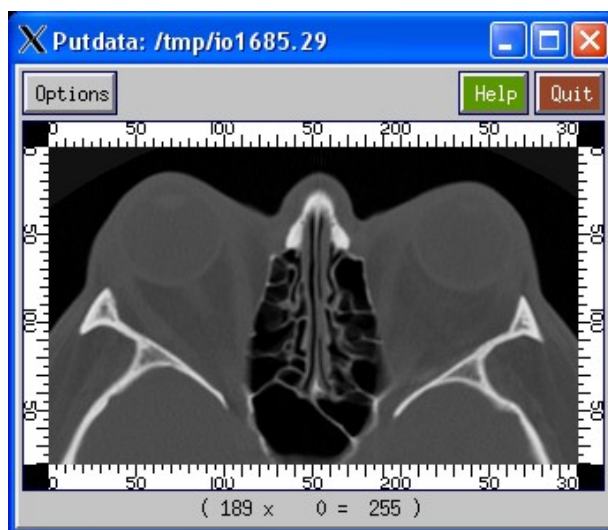
mosaic.xv



1.15. Kép köré vonalzó illesztése.

Nehézségi fok: 3

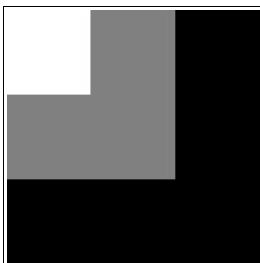
Helyezd el a vonalzókat a kép szélén az ábrán látható módon! A mérce a kép felől legyen!
A mérce és CT-kép részlet megtalálható a *Glyphs* → *DIP Tools* → *DIP Data* → *DIP Images* képtárban.



1.16. Képrészletek összeillesztése.

Nehézségi fok: 2

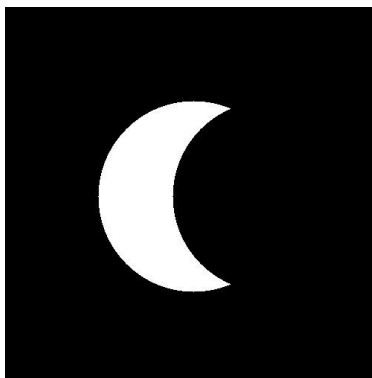
Készítsd el az alább látható 192*192 méretű ábrát, melyen a 64 oldalhosszúságú fehér négyzet és a szürke alakzat együttesen egy 128 oldalhosszú négyzetet eredményez!



1.17. Alakzat készítése.

Nehézségi fok: 2

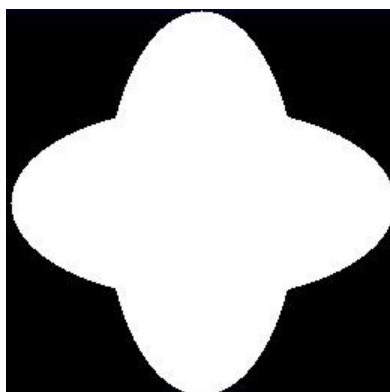
Készíts az ábrához hasonló kifli-szerű alakzatot, amelynek két határolóvonala 1-1 azonos sugarú körív!



1.18. Alakzat készítése.

Nehézségi fok: 2

Készíts olyan egyszínű alakzatot, melyet egy ellipszisnek és középpontja körüli 90 fokos elforgatottjának uniójaként kapunk (ld. ábra)!



1.19. Képrészletek összeillesztése.

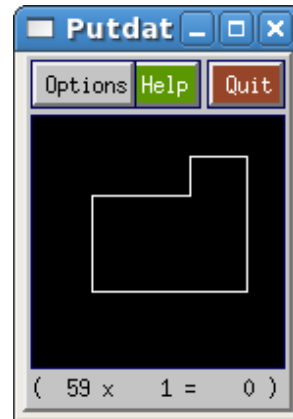
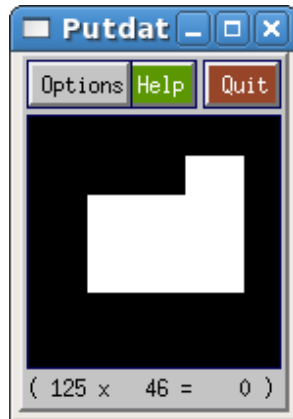
Nehézségi fok: 3

Olvasd be a *ruler.kdf (DIP Images)* képet, és rakd rá a holdfelszínt ábrázoló képre! Végezd el ugyanezt X irányú kétszeres nagyítás után is!

****1.20. Kontúr meghatározása.**

Nehézségi fok: 2

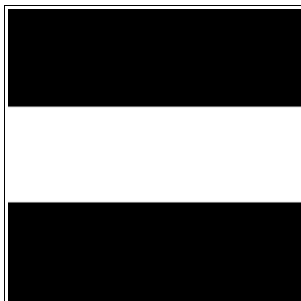
Állítsd elő az alakzatot és határozd meg a kontúrját!



****1.21. Alakzat előállítás**

Nehézségi fok: 2

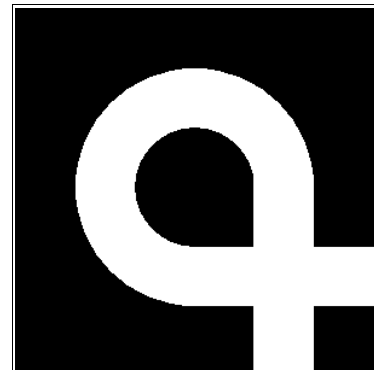
Állítsd elő a következő alakzatot a *koriv.xv* és *egyenes.xv* elemekből!



egyenes.xv



koriv.xv



alakzat

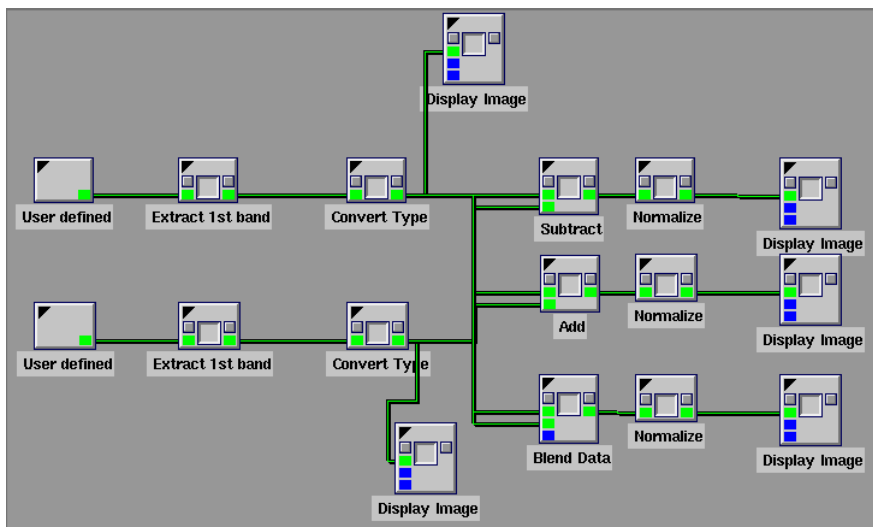
2. Képmanipulációk, mintavételezés, kvantálás

2.a) Példa.

Aritmetikai műveletek képeken.

Van két képünk, amely azonos pozícióból de különböző időpontban készült. Végezzünk rajtuk összeadást, kivonást, súlyozott összeadást!

Megoldás:



A két kép ugyanabból a pozícióból készült és ugyanakkora méretű. Nyissuk meg őket a **User Defined** dobozzal. Ha a képek színes RGB képek, akkor vagy átkonvertálhatjuk szürkeárnyaltosba, vagy pedig egyszerűen kiválaszthatjuk valamelyik színcsatornát. Most válasszuk ez utóbbi, egyszerűbb megoldást! A kép típusát alakítsuk a **Convert Type** dobozzal Integer típusúra! Ezt követően alkalmazzuk a **Subtract**, **Add** és **Blend** dobozokat úgy, hogy a dobozok egyik bemenetére az egyik képet, a másikra a másik képet kötjük. A dobozok kimenetét érdemes normalizálni.

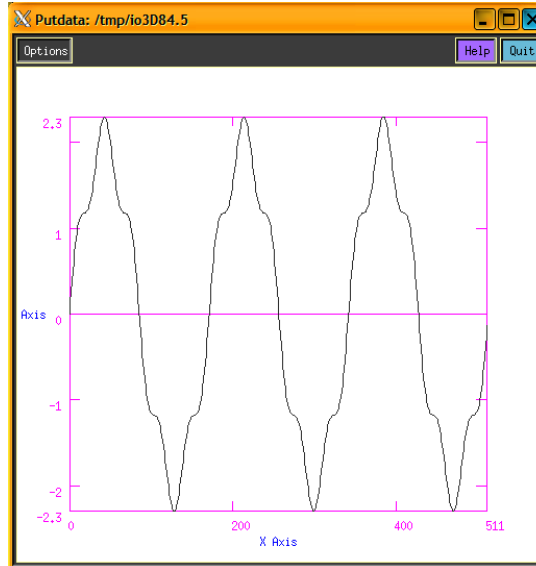
Ha inputként a *cam1.xv* és *cam2.xv* képeket használjuk, akkor ezt az eredményt kapjuk:



2.1. Szinusz hullámok összeadása.

Nehézségi fok: 2

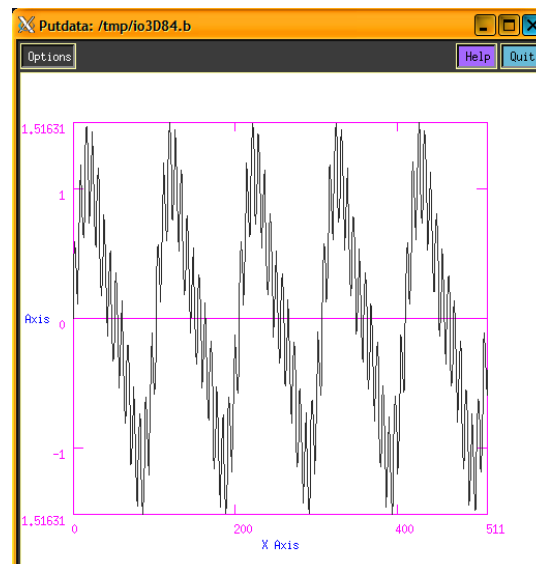
Állítsd elő a $2 \cdot \sin(3x) + 0.3 \cdot \sin(15x)$ jelet!



2.2. Szinusz hullámok összeadása.

Nehézségi fok: 2

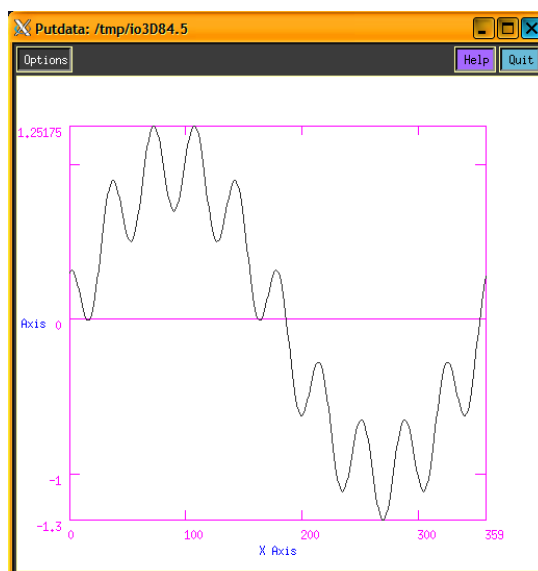
Állítsd elő a $\sin(5x) + 0.3 \cdot \sin(10x) + 0.4 \cdot \sin(70x)$ jelet!



2.3. Szinusz hullámok összeadása.

Nehézségi fok: 2

Állítsd elő a $\sin(x) + 0.3 \cdot \sin(10x + 90)$ jelet! A vízszintes tengely legyen 360 egység hosszú!



2.4. Négyzögjel közelítése szinusz hullámokkal.

Nehézségi fok: 3

Közelítsd a négyzögjelet 5 szinuszos felharmonikussal! A közelítések kiszámíthatók a következő képlettel:

$$\sum_{i=0}^5 \frac{\sin((2i+1)x)}{2i+1}$$

2.5. Színes kép szürkeárnyalatossá konvertálása.

Nehézségi fok: 2

Konvertáld át a Famous Mandril (RGB Images) képet szürkeárnyalatossá! Használd a színcsatornák következő lineáris kombinációját: $Grey = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B$!

2.6. Háttér eltüntetése.

Nehézségi fok: 1

Vond ki a háttérret a *cam1.xv* és *cam2.xv* képről!



cam1.xv



cam2.xv

2.7. Képrészlet kinyerése.

Nehézségi fok: 2

Jelenítsd meg a *cam1.xv* képen látható gyalogos alakját a *cam1.xv* és *cam2.xv* képek felhasználásával! (Apróbb képrészletek még látszódnak a képen.)



A gyalogos alakja a cam1.xv képen



A gyalogos alakja a cam2.xv képen

2.9. Képrészlet megjelenítése.

Nehézségi fok: 2

Jelenítsd meg a *cam2.xv* képen látható gyalogos alakját a *cam1.xv* és *cam2.xv* képek felhasználásával! (Apróbb képrészletek még látszódnak a képen.)

2.10. Képek súlyozott összemosása.

Nehézségi fok: 1

Készítsd el a *cam1.xv* és a *cam2.xv* súlyozott összegét! A *cam1.xv* és *cam2.xv* kép aránya 30% / 70% legyen!

2.11. Aliasing.

Nehézségi fok: 2

Mutass egy példát az átfedés (aliasing) hatásra!

2.12. Képek súlyozott összemosa.*Nehézségi fok: 1*

Készítsd el a *cam1.xv* és a *cam2.xv* súlyozott összegét! A *cam1.xv* és *cam2.xv* kép aránya 65% / 35% legyen!

2.13. Kvantálás.*Nehézségi fok: 1*

Kvantáld a Spanish Seagull képet úgy, hogy csak 4 intenzitásérték legyen rajta!

2.14. Kvantálás.*Nehézségi fok: 1*

Kvantáld a Floating Ball képet úgy, hogy csak 6 intenzitás legyen rajta!

2.15. Kvantálás.*Nehézségi fok: 1*

Egy tetszőleges szürkeárnyaltos kép intenzitásstartományát csökkentsd le felére, negyedére, nyolcadára!

2.16. Mintavételezés.*Nehézségi fok: 1*

Alkalmazz mintavételezést a Spanish Seagull képen. A képet először csökkentsd felére, majd növeld kétszeresére.

2.17. Zaj hozzáadása képhez.*Nehézségi fok: 1*

Adj a Spanish Seagull képhez Gauss-zajt 30-as középértékkel és 145-ös szórással!

2.18. Zaj hozzáadása képhez.*Nehézségi fok: 1*

Adj a Spanish Seagull képhez Gauss-zajt 120-as középértékkel és 40-es szórással!

2.19. Zaj hozzáadása képhez.*Nehézségi fok: 1*

Adj a Moon képhez uniform zajt! A zaj minimális értéke legyen 1, a maximális pedig 200!

2.20. Zaj hozzáadása képhez.*Nehézségi fok: 1*

Adj a Marine képhez uniform zajt! A zaj minimális értéke legyen 45, a maximális pedig 156!

2.21. Zaj hozzáadása képhez.*Nehézségi fok: 1*

Adj a sirályos képhez exponenciális zajt! A zaj szórása 230 legyen!

2.22. Zaj hozzáadása képhez.*Nehézségi fok: 1*

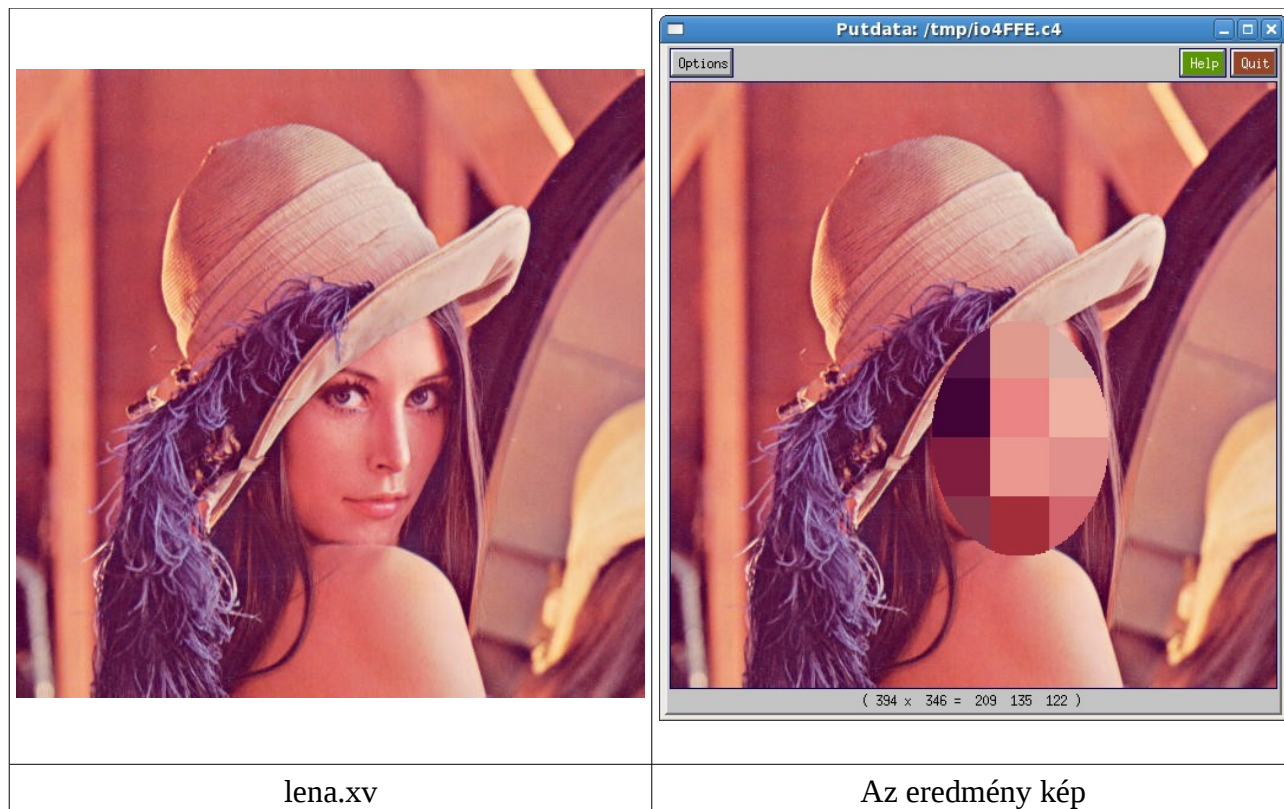
Adj a Marine képhez exponenciális zajt! A zaj szórása 140 legyen!

**** 2.23. Képrészlet eltakarása újramintavételezéssel.**

Nehézségi fok: 2

Újramintavételezéssel takard el Lena arcát egy ellipszis alakú régióban az ábrán látható módon. Az eltakart rész bármekkora lehet, de ennek méretét célszerű globális változókkal továbbadni a többi műlethez.

A példában egy olyan ellipszist használtam, melynek méretei 150×200 pixel. A mintavételezésnél a skálatényező 50 volt.

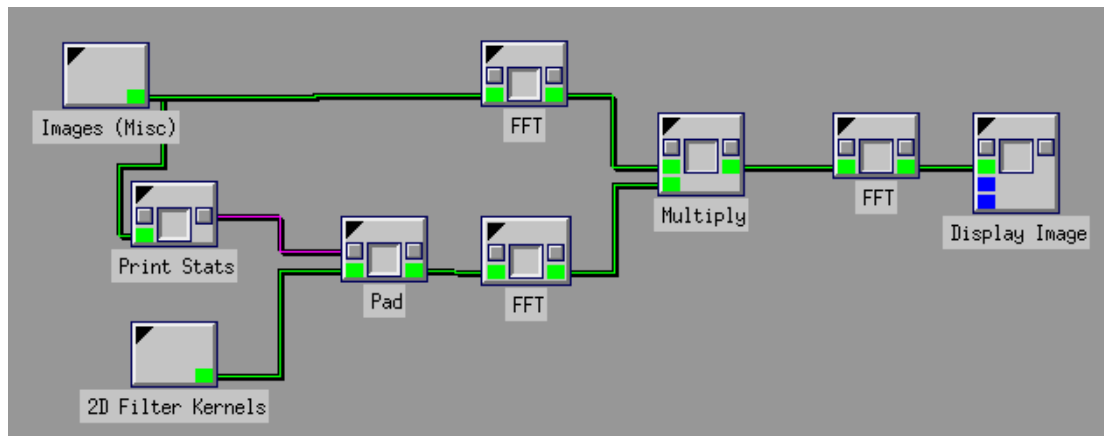


3. Fourier-transzformáció, konvolúció

3.a) Példa.

Végezzünk simítást frekvencia térben a sirályos képen 7x7-es átlagoló szűrővel!

Megoldás:



A sirályos kép méretét kinyerjük a **Print Stats** dobozzal és a **2D Filter Kernels** dobozban kiválasztjuk a 7x7-es átlagoló szűrőt. A kernel kép méretét meg kell növelni akkora, amekkora a sirályos kép. Itt használjuk fel a **Print Stats** doboz globális változóit. A sirályos képet és az átméretezett kernel képet is Fourier-transzformáljuk és a Fourier-képeket összeszorozzuk. Ezután inverz Fourier-transzformációt hajtunk végre. Az IFFT eredményét átvezethetjük egy **Complex To Real** dobozon, mielőtt megjelenítjük.



3.1. Fourier-transzformáció.

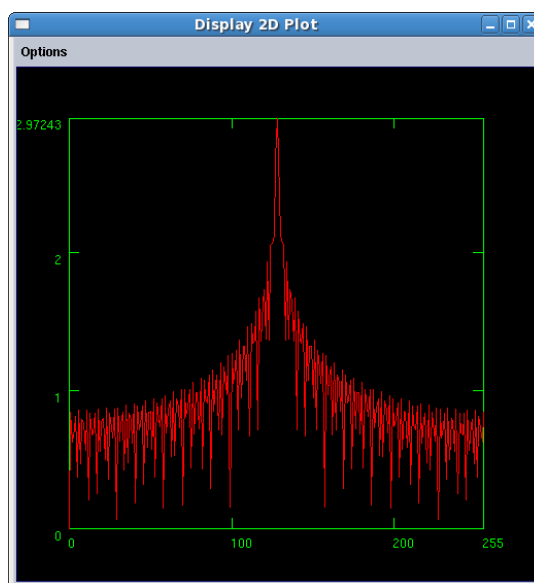
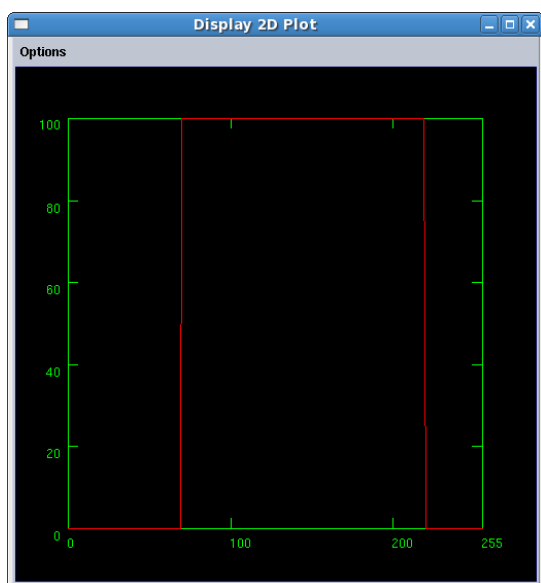
Nehézségi fok: 2

Ábrázold a $\sin(x) + 0.4 \cdot \sin(20x)$ Fourier-transzformált képét!

3.2. Fourier-transzformáció.

Nehézségi fok: 2

Állítsd elő az alábbi jelet és Fourier-transzformáltját! (A jel 75-nél fut fel és 225-nél ereszkedik le, az amplitúdója pedig 100.)



3.3. Fourier-transzformáció.

Nehézségi fok: 1

Hajts végre Fourier-transzformációt a Spanish Seagull képen! Jelenítsd meg a Fourier-képet!

3.4. 1D Fourier-transzformáció

Nehézségi fok: 2

Állíts elő egy szinuszhullámot és hajts rajta végre Fourier-transzformációt! Vizsgáld meg a Fourier-képének valós, képzetes részét valamint magnitúdóját!

3.5. Fourier-transzformáció.

Nehézségi fok: 3

Hajts végre Fourier-transzformációt a Spanish Seagull képen, majd végezz inverz Fourier-transzformációt! Nézd meg, hogy az eredeti és az inverz Fourier-transzformáció között, mekkora a maximális intenzitáskülönbség!

3.6. Simítás képtérben.

Nehézségi fok: 1

Hajts végre simítást a Spanish Seagull képen képtérben 3x3-as átlagoló szűrővel!

- 3.7. Simítás képtérben.** *Nehézségi fok: 1*
Hajts végre simítást a Marine képen képtérben 5x5-ös átlagoló szűrővel!
- 3.8. Simítás képtérben.** *Nehézségi fok: 1*
Hajts végre simítást a Moon képen képtérben 7x7-es átlagoló szűrővel!
- 3.8. Simítás képtérben.** *Nehézségi fok: 1*
Hajts végre simítást a Nerve Cell képen 11x11-es medián szűrővel!
- 3.10. Simítás képtérben.** *Nehézségi fok: 1*
Hajts végre simítást a Marine képen 7x7-es medián szűrővel! Az ismétlések száma legyen 3!
- 3.11. Simítás frekvencia-térben.** *Nehézségi fok: 2*
Hajts végre simítást a Spanish Seagull képen frekvencia-térben 3x3-as átlagoló szűrővel!
- 3.12. Simítás frekvencia-térben.** *Nehézségi fok: 2*
Hajts végre simítást a Marine képen frekvencia-térben 5x5-ös átlagoló szűrővel!
- 3.13. Simítás frekvencia-térben.** *Nehézségi fok: 2*
Hajts végre simítást a Moon képen frekvencia-térben 7x7-es átlagoló szűrővel!
- 3.14. Frekvencia-szűrés.** *Nehézségi fok: 2*
Hajts végre frekvencia-szűrést a Spanish Seagull képen ideális felüláteresztő szűrővel! A szűrő mérete legyen 0.20!
- 3.15. Frekvencia-szűrés.** *Nehézségi fok: 2*
Hajts végre frekvencia-szűrést a Spanish Seagull képen ideális felüláteresztő szűrővel! A szűrő mérete legyen 0.05!
- 3.16. Frekvencia-szűrés.** *Nehézségi fok: 2*
Hajts végre frekvencia-szűrést a Spanish Seagull képen ideális aluláteresztő szűrővel! A szűrő mérete legyen 0.20!

3.17. Frekvencia-szűrés.

Nehézségi fok: 2

Hajts végre frekvencia-szűrést a Spanish Seagull képen ideális aluláteresztő szűrővel! A szűrő mérete legyen 0.05!

3.18. Zajszűrés képtérben.

Nehézségi fok: 3

Adj a Spanish Seagull képhez uniform zajt (minimum: 30, maximum: 255)! Használj globális változókat! Simítsd el a képet képtérben 3x3-as átlagoló szűrővel! Jelenítsd meg a zajos és a simított képet!

3.19. Zajszűrés képtérben.

Nehézségi fok: 3

Adj a Spanish Seagull képhez uniform zajt (minimum: 15, maximum: 60)! Használj globális változókat! Simítsd el a képet képtérben 7x7-as átlagoló szűrővel! Jelenítsd meg a zajos és a simított képet!

3.20. Zajszűrés képtérben.

Nehézségi fok: 3

Adj a Spanish Seagull képhez uniform zajt (minimum: 0, maximum: 100)! Használj globális változókat! Simítsd el a képet képtérben 5x5-as medián szűrővel! Jelenítsd meg a zajos és a simított képet!

3.21. Zajszűrés frekvencia-térben.

Nehézségi fok: 3

Adj a Moon képhez uniform zajt (minimum: 30, maximum: 255)! Használj globális változókat! Simítsd el a képet frekvencia-térben 5x5-as átlagoló szűrővel! Jelenítsd meg a zajos és a simított képet!

3.22. Zajszűrés frekvencia-térben.

Nehézségi fok: 3

Adj a Nerve Cell képhez uniform zajt (minimum: 15, maximum: 60)! Használj globális változókat! Simítsd el a képet frekvencia-térben 7x7-as átlagoló szűrővel! Jelenítsd meg a zajos és a simított képet!

3.23. Zajszűrés frekvencia-térben.

Nehézségi fok: 3

Adj a Nerve Cell képhez uniform zajt (minimum: 40, maximum: 200)! Használj globális változókat! Simítsd el a képet frekvencia-térben 5x5-as átlagoló szűrővel! Jelenítsd meg a zajos és a simított képet!

**3.24. Zajszűrés.

Nehézségi fok: 3

Állítsd elő a sirályos tesztképnek 4 különböző zajos példányát úgy, hogy mind a 4 esetben 2-2 %-os só-és-bors zajjal legyen terhelve a kép! Távolítsd el a zajnak a lehető legnagyobb részét simító szűrők alkalmazása nélkül! Ezután végezz medián szűrést az egyik zajos példányra, és hasonlítsd össze mindkét módszer eredményét az eredeti képtől való átlagos eltérésük alapján! (Az összehasonlításnál javasolt a **Statistics** és a **File Viewer** dobozok használata.)

****3.25. Zaj eltávolítása.**

Nehézségi fok: 3

Állítsd helyre a lehető legjobban a *fruits.kdf* színes képet a rajta levő csíkozódás eltávolításával! (Segítség: a zaj teljesen a kép vörös csatornájában szerepel.)



***3.26. Konvolúció.**

Nehézségi fok: 2

Ellenőrizd egy tetszőleges tesztképen, hogy a

$$G_x = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

vízszintes Sobel-szűrővel történő konvolúció eredménye megegyezik-e azon kétlépéses műveletsor eredményével, amelynek első lépésében a képet a

$$H_1 = \begin{array}{|c|} \hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array}$$

maszkkal konvolváljuk, második lépésében pedig az első lépés eredményére hajtunk végre konvolúciót a

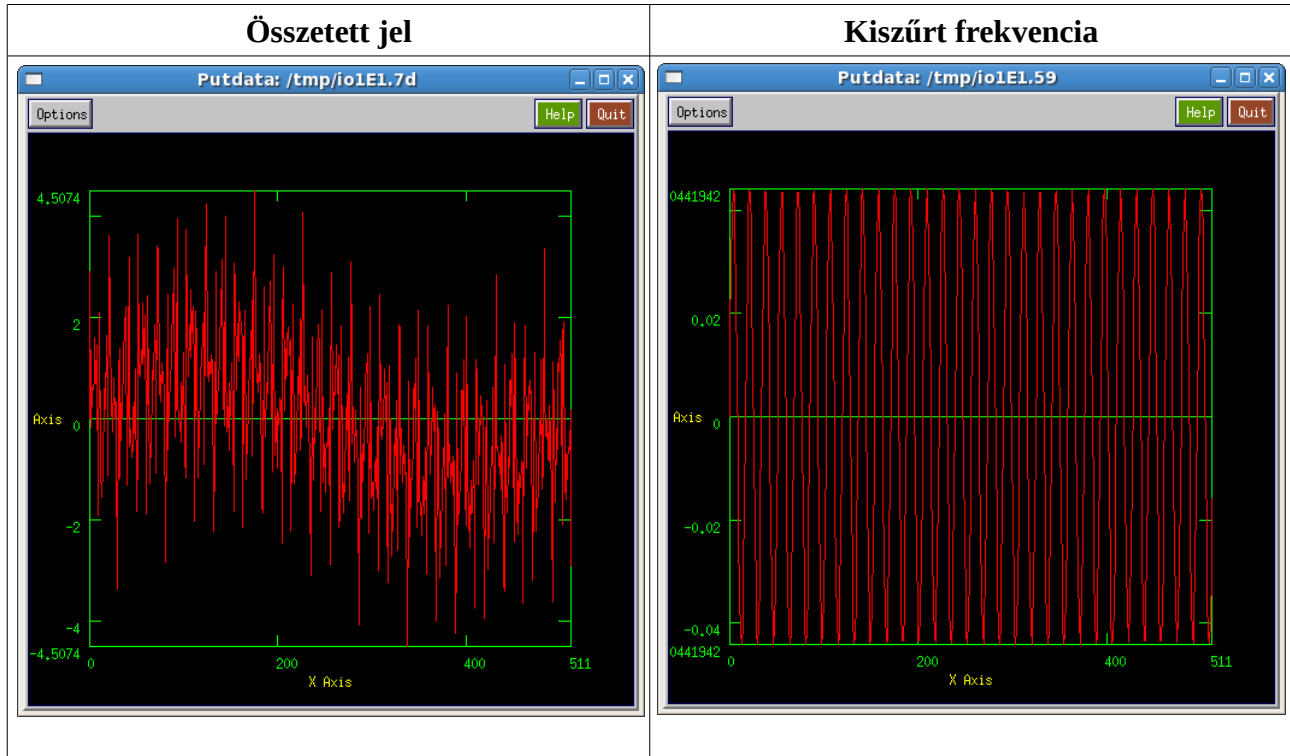
$$H_2 = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array}$$

maszkkal! (Az összehasonlításnál javasolt az *Absolute Diff*, *Statistics* és *File Viewer* dobozok használata.) A H_1 és H_2 szűrők megtalálhatóak a *h1.ascii* és *h2.ascii* fájlokban.

**3.27. Frekvencia szűrés.

Nehézségi fok: 2

Adott 5 azonos amplitúdójú, de különböző frekvenciájú (1D) szinuszoid hullám összegéből képzett jel. A frekvencia (hullámok száma) rendre 1, 30, 100, 150, 200. Ebből az összetett jeltől szűrd ki a 30-as frekvenciájú szinuszoidot! (Javasol képméret: 512)



3.28. Fourier-transzformáció tulajdonságai.

Nehézségi fok: 3

Készíts egy 128x128-as kép közepére egy 4x32-es konstans téglalapot! Hajts végre ezen a képen Fourier-transzformációt! Vizsgáld meg, hogy mi történik akkor, ha a téglalapot elforgatod 45°-kal és eltolod 20 pixellel X és Y irányban. Hasonlítsd össze a Fourier-képeket és a fázisképeket!

3.29. Konvolúciós tétel.

Nehézségi fok: 3

Szemléld a konvolúciós tételt (képtérben a konvolúció megegyezik a frekvencia-térben történő szorzással)!

3.30. Korrelációs tétel.

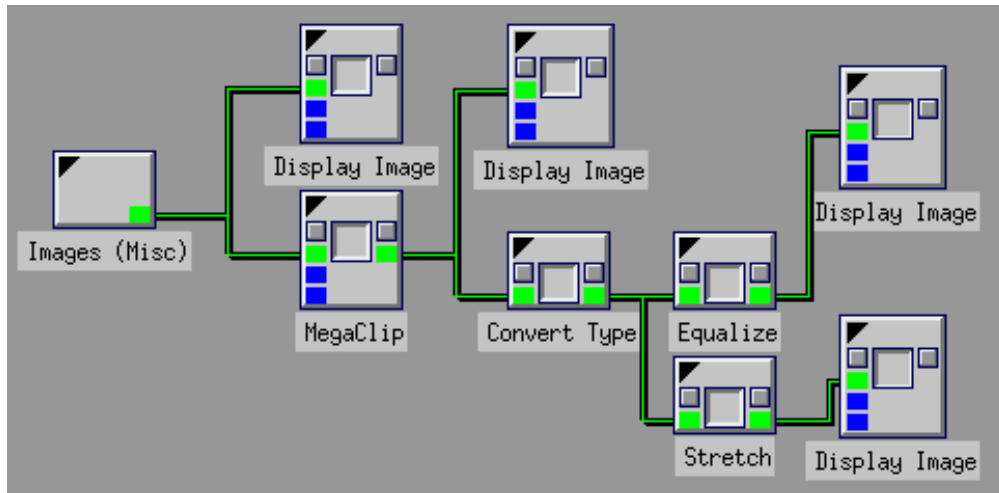
Nehézségi fok: 3

Szemléld a korrelációs tételt!

4. Pontoperációk

4.a) Példa.

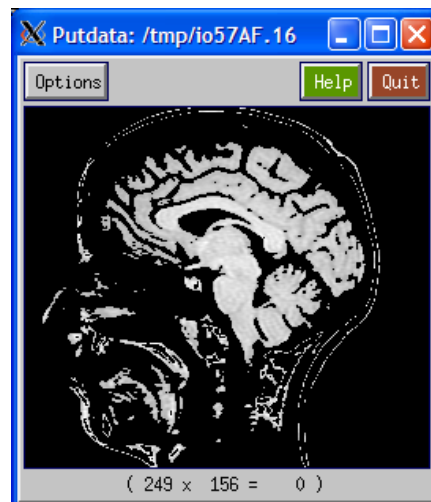
Jelöld ki at MRI of Head képen az agyi szöveteket és tedd kontrasztosabbá a képet!
Megoldás:



Nyissuk meg a képet és a **MegaClip** dobozzal vágjuk ki az 1098-1130 közötti intenzitásértékeket! Alakítsuk át a képet adattípusát *Unsigned Byte*-tá és alkalmazzunk hisztogram kiegyenlítést (**Equalize**) és hisztogram széthúzást (**Stretch**).



Hisztogram kiegyenlítés



Hisztogram széthúzás

4.1. Küszöbölés.

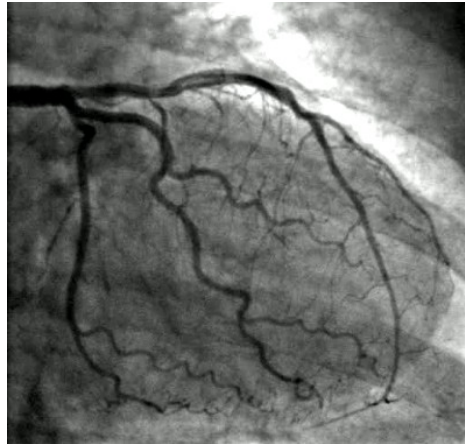
Nehézségi fok: 1

Végezz küszöbölést a *Nerve Cell* képen! Állítsd be a küszöbértéket úgy, hogy a sejtfalak legyenek láthatóak!

4.2. Küszöbölés.

Nehézségi fok: 1

Küszöböld az ereket az *angiography1.xv* képen!



angiography1.xv

4.3. Vágás.

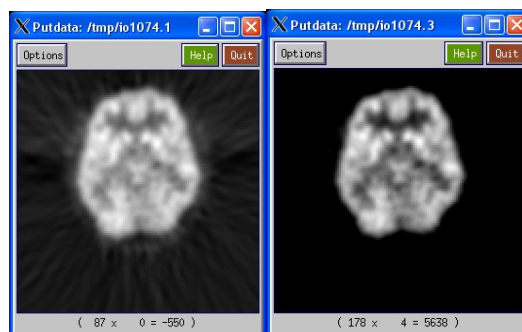
Nehézségi fok: 2

Határozd meg az *MRI of Spine* képen az a vágási küszöbértéket, amely mellett csak a csontok látszódnak. A levágott intenzitások helyén 0 legyen az intenzitás. Tedd kontrasztosabbá az eredményt!

4.4. Vágás.

Nehézségi fok: 2

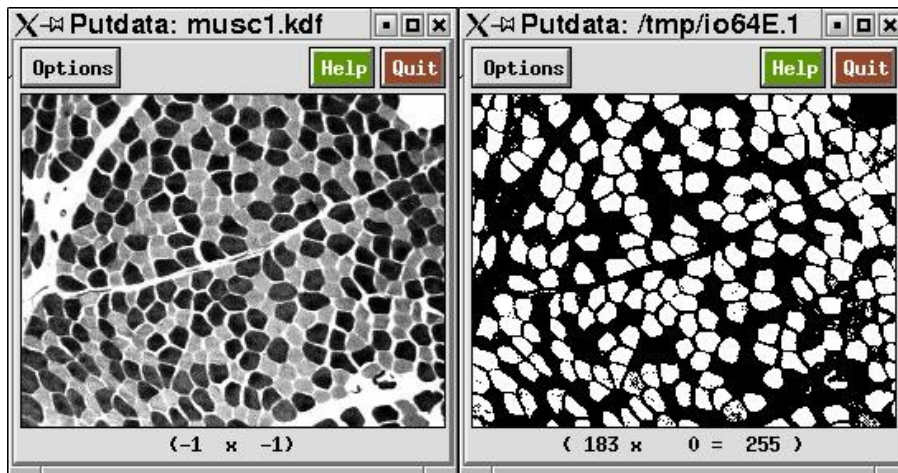
Végezz 20%-os háttérlevonást a *pet02_raw010.ascii* képen, azaz a maximális intenzitásérték 20%-ánál kisebb intenzitásokat vágjuk le a képen! Az alsó vágási küszöb meghatározásához használd a **Print Stats** dobozt!



4.5. Küszöbölés, címkézés.

Nehézségi fok: 2

Számold meg a *musc1.kdf* képen a sötét sejteket!



**4.6. Küszöbölés, kontúr meghatározása.

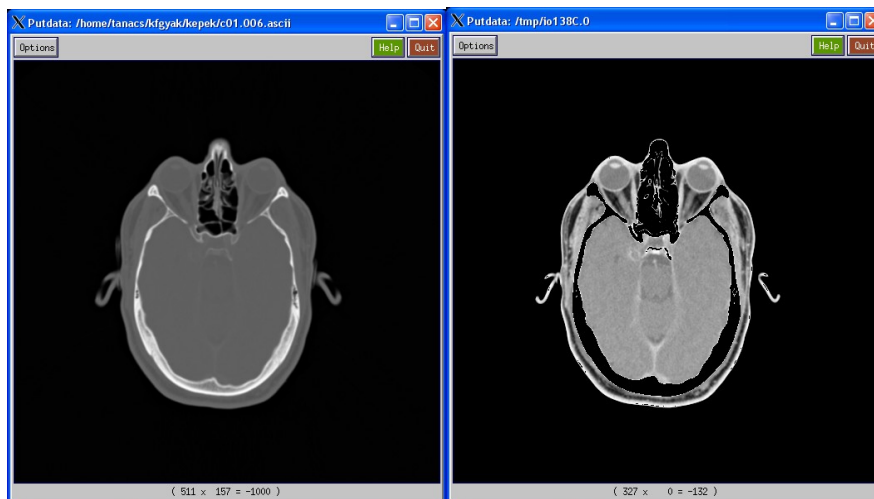
Nehézségi fok: 3

Küszöböld a *musc1.kdf* képet úgy, hogy a sötét sejtek látszódjanak! Határozd meg ezek kontúrját és vetítsd az eredeti képre! Tipp: Használd az *Add*, *Blend*, *Maximum*, *Gradient* dobozokat!

4.7. Vágás.

Nehézségi fok: 2

Jelenítsd meg a *c01.006* képen látható CT szeleten a lágy szöveteket! A kép RAW formátumban van és *Short* típusú.



*4.8. Bitsíkok kiemelése.

Nehézségi fok: 2

Emeld ki egy tetszőleges kép bitsíkjait!

***4.9. Álszínezés.**

Nehézségi fok: 1

Végezz álszínezést egy beolvasott, tetszőleges szürkeárnyaltos képen!

4.10. Küszöbölés.

Nehézségi fok: 2

A *digits.xv* képből állíts elő 1-1 bináris képet, amelyen csak az 1-es és a 2-es számjegy pontjai fehérek, a többi pont fekete!



digits.xv

4.11. Vágás.

Nehézségi fok: 2

A *digits.xv* képből állíts elő 1-1 olyan képet, melyen az 1-es / 2-es számjegy "le van törölve", azaz a hozzá tartozó pixelek a háttér intenzitását veszik fel!

4.12. Intenzitás-manipuláció.

Nehézségi fok: 1

Olvasd be a *log-enhance-problem.kdf* képet és javíts a kontraszton! A kép a **DIP Images** dobozból betölthető.

4.13. Kontraszt növelés.

Nehézségi fok: 1

Tedd kontrasztosabbá a *myelin.kdf* képet! A kép a **DIP Images** dobozból betölthető.

4.14. Kontraszt növelés.

Nehézségi fok: 1

Tedd kontrasztosabbá a *cell-tissue-saturate.kdf* képet! A kép a **DIP Images** dobozból betölthető.

4.15. Gamma-korrekción.

Nehézségi fok: 1

Hajts végre Gamma-korrekción a *spine.kdf* képen!

4.16. Kép invertálása.

Nehézségi fok: 1

Olvasd be egy szürkeárnyaltos képet, jelenítsd meg az invertált változatát!

4.17. Hisztogram megjelenítése.

Nehézségi fok: 2

Jelenítsd meg a labdás kép 250-edik sorának hisztogramját!

4.18. Hisztogram megjelenítése.

Nehézségi fok: 2

Jelenítsd meg a sirályos kép 100-adik oszlopának hisztogramját!

4.19. Hisztogram kiegyenlítés.

Nehézségi fok: 1

Végezz hisztogram kiegyenlítést a *Picture2.xv* képen!



Picture2.xv

4.20. Hisztogram széthúzás.

Nehézségi fok: 1

Végezz hisztogram széthúzást a *Picture2.xv* képen!

4.21. Hisztogram kiegyenlítés.

Nehézségi fok: 1

Alakítsd át Short típusúra az *MRI of Head* képet, majd hajts végre egy hisztogram kiegyenlítést!

4.22. Hisztogram kiegyenlítés.

Nehézségi fok: 1

Alakítsd át Short típusúra a *lenina.kdf* képet (**DIP Images**) és hajts végre egy hisztogram kiegyenlítést!

4.23. Hisztogram kiegyenlítés.

Nehézségi fok: 1

Alakítsd át Short típusúra a *cells-tissue-saturate.kdf* képet (**DIP Images**) és hajts végre egy hisztogram kiegyenlítést! Jelenítsd meg a hisztogramokat is!

4.24. Hisztogram széthúzás.

Nehézségi fok: 2

Csökkentsd a sirályos képen a maximális intenzitást 16-odára, majd végezz hisztogram széthúzást a képen! Jelenítsd meg a hisztogramokat is!

4.25. Zajszűrés.

Nehézségi fok: 2

Végezz zajszűrést a *barcode.kdf* képen!

***4.26. Kontraszt javítása.**

Nehézségi fok: 2

Tedd láthatóvá a rendszámot a VW Boot képen!



****4.27. Gyertyák számolása.**

Nehézségi fok: 3

Hány gyertya látható a képen? Készíts műveletsort, a gyertyák megszámlálására!



candles2.jpg



candles3.xv



candles6.xv

*****4.28. Lámpák számolása.**

Nehézségi fok: 3

Számold meg az útszéli lámpákat a *lampak_kicsi.xv* képen! (3 lámpa látszik)

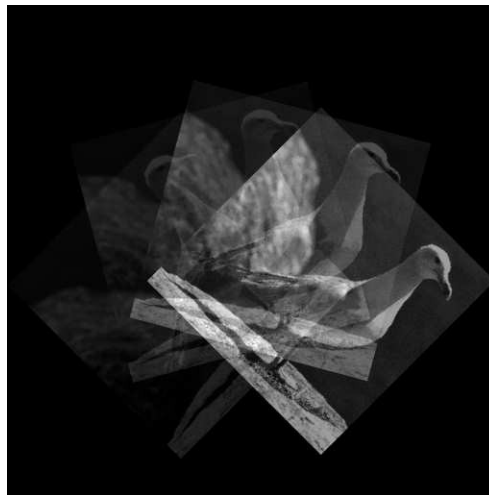


lampak_kicsi.xv

***4.29. Kép előállítás.**

Nehézségi fok: 2

Állítsd elő az alábbi képet (vagy hasonlót)!



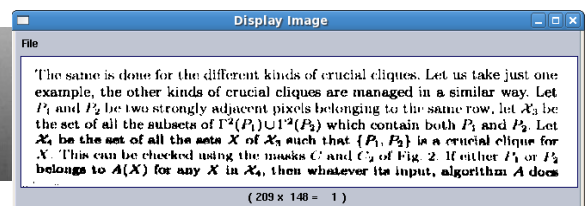
*****4.30. Küszöbölés.**

Nehézségi fok: 3

Állítsd elő a szkennelt lapból az alábbi eredményt (vagy hasonlót)! A küszöböléshez használhatsz statisztikai értékeket!

The same is done for the different kinds of crucial cliques. Let us take just one example, the other kinds of crucial cliques are managed in a similar way. Let P_1 and P_2 be two strongly adjacent pixels belonging to the same row, let \mathcal{X}_3 be the set of all the subsets of $\Gamma^2(P_1) \cup \Gamma^2(P_2)$ which contain both P_1 and P_2 . Let \mathcal{X}_4 be the set of all the sets X of \mathcal{X}_3 such that $\{P_1, P_2\}$ is a crucial clique for X . This can be checked using the masks C and C_2 of Fig. 2. If either P_1 or P_2 belongs to $A(X)$ for any X in \mathcal{X}_4 , then whatever its input, algorithm A does

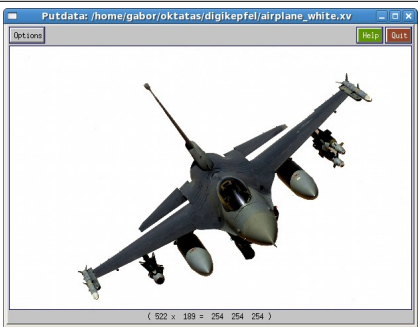


szkennelt_lap_sotet.jpg



*****4.31. Két kép kombinálása.**

Nehézségi fok: 3




Kombináld a képet a háttérrel az minta szerint! Az objektumot és a háttérrel tartalmazó képek azonos méretűek. A repülőgép takarja a felhőt!

Objektum	Háttér	Kombinált kép
		
airplane.xv	clouds.xv	Clouds and airplane

*****4.32. Két kép kombinálása.**

Nehézségi fok: 3

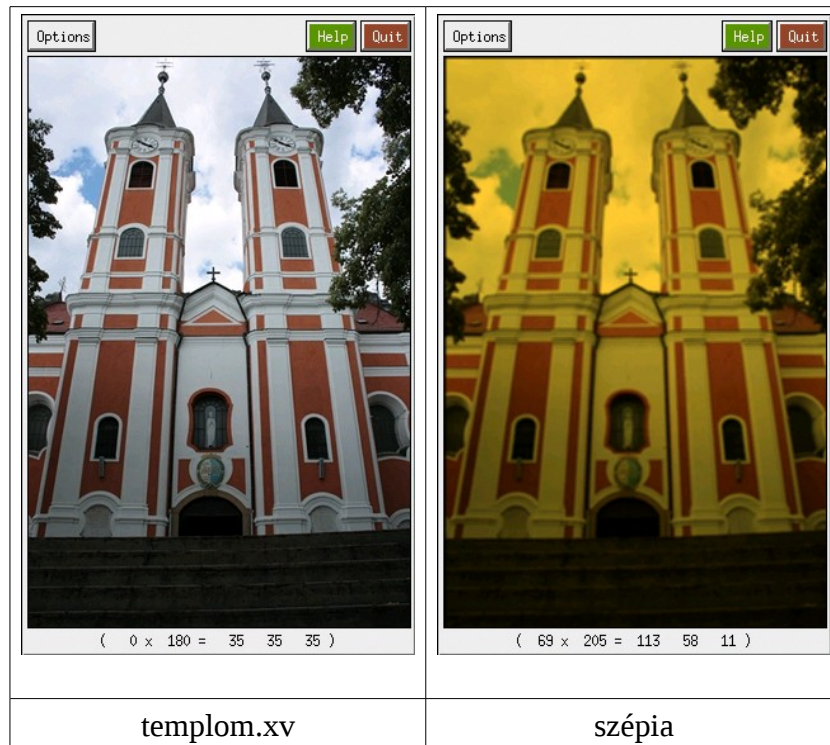
Kombináld a képet a háttérrel az minta szerint! Az objektumot és a háttérrel tartalmazó képek azonos méretűek. A pálmafa takarja a felhőt!

Objektum	Háttér	Kombinált kép
		
palm_tree.xv	clouds5.xv	Palm tree and clouds

****4.33. Szépia kép előállítás (kép „öregítése”).**

Nehézségi fok: 2

Adott egy színes kép (pl. a templom.xv) és végezz rajta olyan képműveleteket, amely szépia-hatást eredményez a képen.



***** 4.34. Képrészlet hangsúlyozása a háttér elmosásával.**

Nehézségi fok: 3

Hangsúlyozd ki egy színes kép adott részletét egy kör alakú régióban a háttér (a kép régió kívüli részének) elmosásával!

Példa:



***** 4.35. Képrészlet megjelölése a képen.**

Nehézségi fok: 3

Adott egy színes kép, jelöld be rajta a fontos képrészletet egy piros gyűrű segítségével!


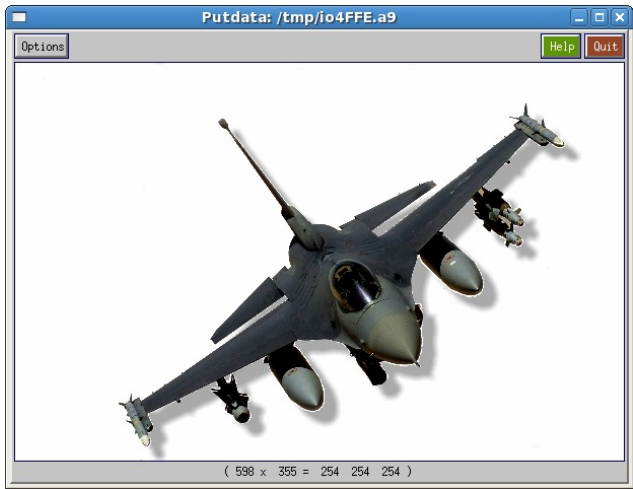
Példa:

	 <p>(248 x 111 = 46 42 30)</p>
loverseny.xv	eredmény

***** 4.36. Árnyék illesztése alakzathoz konstans háttérren.**

Nehézségi fok: 3

Tegyél árnyékot a repülőgépre az ábrán látható módon! Az árnyék legyen valamilyen halványszürke színű és alkalmazd rá egy simítást is, hogy ne legyen éles kontúrja!

	 <p>(598 x 355 = 254 254 254)</p>
airplane.xv	A repülőgép árnyékkal

5. Programozási feladatok

5.a) Példa.

Készítsünk programot egy egyszerű küszöbölésre!

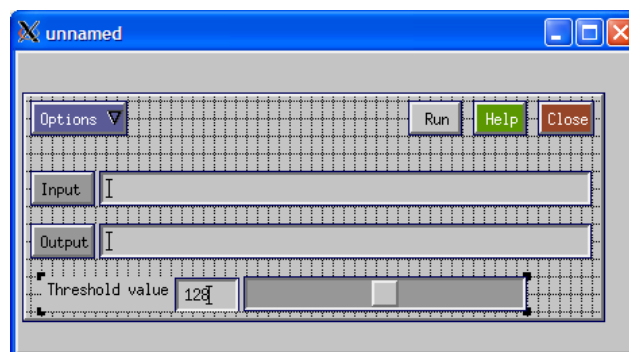
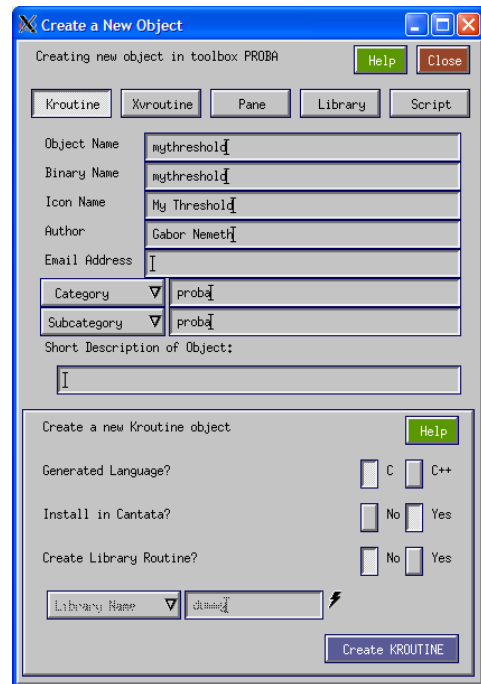
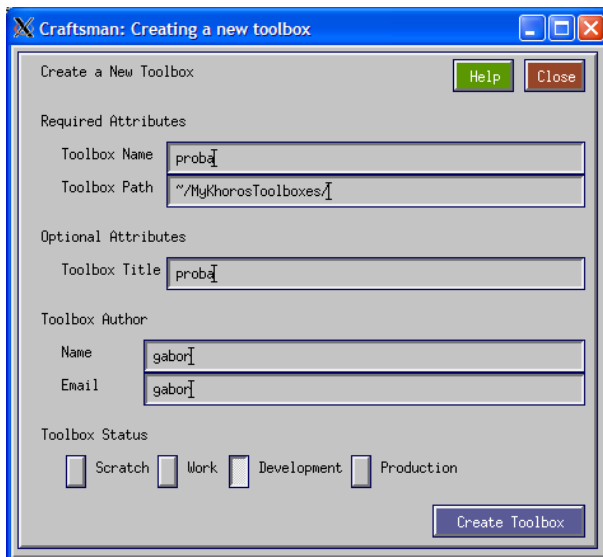
Megoldás:

1. Készítsünk egy könyvtárat *MyKhorosToolboxes* néven a home-ba! Hozzunk létre egy proba Toolbox-ot.

Toolbox Name: proba
Toolbox Path: ~/MyKhorosToolboxes
Toolbox Title: proba

2. Készítsünk egy dobozt!

Object Name: mythreshold
Binary Name: mythreshold
Icon Name: My Threshold
Author: ...
Category: proba
Subcategory: proba



3. Készítsük el a doboz paneljét. Alapértelmezett esetben egy input és egy output mező látható. Tegyük rá egy Integer változót! Szerkesszük ezt a változót át úgy, hogy az értéktartománya 0-255 között legyen, az alapértelmezett érték 128 legyen. A változó címkéje (felirata) „Threshold value” legyen, a változót nevezzük el *threshold*-nak!

Zárjuk be a szerkesztő ablakot.

4. A **Composer** ablakban kattintsunk oldalt a **Commands** gombra, majd a megjelenő panelen a **Generate Code** gombra. Ekkor le lesz generálva az alapvető kód, amit ki kell egészítenünk. A **Composer**-ben a **SOURCE** menü alatt megtaláljuk a forrásokat.

```
int run_mythreshold(void)
{
    /*-- Put Your Code Here --*/

    /* -main_variable_list */
    char lib = "mythreshold_obj"; /* ez a kerror() függvényhez kell */
    char rtn = "main";           /* ez a kerror() függvényhez kell */
    kobject in_object = NULL; /* input objektum, inicializálva */
    kobject out_object = NULL; /* output objektum, inicializálva */
    unsigned char *plane;        /* tömb az input adat számára */
    unsigned char *res_plane;    /* tömb az eredménykép számára */
    int w, h, d, t, e;           /* ezekbe a változókba kérdezzük le az attribútumokat */
    int cw, ch, ct, pos;         /* segédváltozók */
/* -main_variable_list_end */

/* -main_before_lib_call */
/* megnyitjuk az input fájlt és az adatot bemásoljuk az input struktúrába
   le kell ellenőrizni, hogy sikerült-e megnyitni a fájlt, és átmásolni a struktúrát,
   különben ki kell lépni a programból
*/
if ((in_object = kpds_open_input_object(clui_info->i_file))
    == KOBJECT_INVALID) {
    kerror(lib, rtn, "Can not open input object %s.\n", clui_info->i_file);
    kexit(KEXIT_FAILURE);
}

/* megnyitjuk az input fájlt és az adatot bemásoljuk az input struktúrába
   le kell ellenőrizni, hogy sikerült-e megnyitni a fájlt, és átmásolni a struktúrát,
   különben ki kell lépni a programból
*/
if ((out_object = kpds_open_output_object(clui_info->o_file))
    == KOBJECT_INVALID) {
    kerror(lib, rtn, "Can not open output object %s.\n", clui_info->o_file);
    kexit(KEXIT_FAILURE);
}

/* átmásoljuk az input képet az outputba, ezzel egyúttal be is állítjuk minden
   szegmensét és attribútumát
*/
if (!kpds_copy_object(in_object, out_object)) {
    kerror(lib, rtn, "Can not copy input object to output object.\n");
    kexit(KEXIT_FAILURE);
}

/* beállítjuk mindkét objektum adattípusát UNSIGNED BYTE-ra */
kpds_set_attribute(in_object, KPDS_VALUE_DATA_TYPE, KUBYTE);
kpds_set_attribute(out_object, KPDS_VALUE_DATA_TYPE, KUBYTE);

/* lekérdezzük a változókba a méret attribútumokat */
kpds_get_attribute(in_object, KPDS_VALUE_SIZE, &w, &h, &d, &t, &e);

/* létrehozunk egy tömböt a képi adat számára. A tömb egydimenzós lesz, mérete
   megegyezik a kép méretével.
*/
}
```

```

plane = (unsigned char *)kmalloc(w*h*sizeof(unsigned char));
res_plane = (char *)kmalloc(w*h*sizeof(unsigned char));
if (!plane || !res_plane) {
    kerror(lib, rtn, "Could not allocate memory for the image\n");
    kexit(KEXIT_FAILURE);
}
/* -main_before_lib_call_end */

/* -main_library_call */
/* Itt kezdődik a képfeldolgozó művelet érdemi része. Végighaladunk a képen, és
küszöböljök a paraméterként (clui_info->threshold változó) megadott értékkel.
Ami ez alatti intenzitás, az 0 lesz, az e fölöttiek pedig 255 értéket kapnak.
Most csak az általunk lefoglalt memória területen dolgozunk, nem manipuláljuk
a kobject struktúrák adatait.
*/
for (ct = 0; ct < t; ct++) {
    /* pozícionálás: a ct-edik kép kezdőpozíciójára állunk és kimásoljuk a képi adatot a
segéd tömbbe
*/
    kpds_set_attribute(in_object, KPDS_VALUE_POSITION, 0, 0, 0, ct, 0);
    kpds_get_data(in_object, KPDS_VALUE_PLANE, (kaddr)plane);
    kmemset(res_plane, 0, w*h*sizeof(char)); /* kinullázzuk az eredmény képet */

    for (ch = 0; ch < h; ch++) {
        for (cw = 0; cw < w; cw++) {
            pos = ch*w + cw; /* Az adott pozíciónak megfelelő tömb indexet
így számoljuk ki */
            if (plane[pos] <= clui_info->threshold_int)
                res_plane[pos] = (unsigned char)0;
            else
                res_plane[pos] = (unsigned char)255;
        }
    }

    /* beállítjuk a pozíciót úgy, hogy a struktúrán belüli mutató a képi adat
ct-edik képének kezdőpozíciójára mutasson és bemásoljuk erre a pozícióra az
eredmény képet
*/
    kpds_set_attribute(out_object, KPDS_VALUE_POSITION, 0, 0, 0, ct, 0);
    kpds_put_data(out_object, KPDS_VALUE_PLANE, (kaddr)res_plane);
}
/* -main_library_call_end */

/* -main_after_lib_call */
/* beállítjuk a history stringet (ezt mindig ugyanígy kell csinálni ) */
if (!kpds_set_attribute(out_object, KPDS_HISTORY, kpds_history_string())) {
    kerror(lib, rtn, "Unable to set history on the destination object");
    kexit(KEXIT_FAILURE);
}

/* felszabadítjuk a lefoglalt memóriaterületet */
if (plane)
    kfree(plane);
if (res_plane)
    kfree(res_plane);

/* a program végén lezárjuk az objektumokat, ezek a függvények elvégzik a struktúrán
belüli memória felszabdítást is.
*/
kpds_close_object(in_object);
kpds_close_object(out_object);
/* -main_after_lib_call_end */

return TRUE;
}

```

5. A kód szerkesztése után ismét kattintsunk a **Composer** ablakban a **Commds** gombra és fordítsuk le a programot a **Make** gombbal.

6. Indítsuk el a Cantata-t és próbáljuk ki a programot.

5.1. Invertálás.

Nehézségi fok: 1

Készíts programot, amely egy szürkeárnyaltos képet invertál. A képek intenzitástartománya 0-255 legyen!

5.2. Kép transzponálása.

Nehézségi fok: 1

Készíts programot, amely transzponál egy szürkeárnyaltos képet!

5.3. Simítás átlagoló szűrővel.

Nehézségi fok: 2

Készíts programot, amely átlagoló szűrővel simít egy szürkeárnyaltos képet! A doboz párbeszédablakában lehessen beállítani a simító maszk méretét 3-15-ig.

5.4. Körlap rajzolása.

Nehézségi fok: 1

Készíts programot, amely kirajzol egy kört! A doboznak nincsen bemenete. A kimenete egy olyan kép, amely egy kitöltött körlapot ábrázol. A háttér 0, a körlap pixelei 1 értékűek legyenek. A kép mérete legyen a körlap átmérője (pozitív egész), amit a doboz párbeszédablakában lehet beállítani.

5.5. Blending.

Nehézségi fok: 2

Készíts programot, amely két inputként kapott szürkeárnyaltos képből elkészíti azok súlyozott összegét (*blend*). A súlyozás a következőképpen értendő. Ha az első kép súlya p , akkor a második kép súlya $1-p$. A súlyozást a doboz párbeszédablakában lehessen beállítani egy csúszkával. A súlyokat 0-100 között lehessen állítani.

5.6. Blending.

Nehézségi fok: 2

Készíts programot, amely két inputként kapott azonos méretű szürkeárnyaltos képből elkészíti azok súlyozott összegét (*blend*). A súlyozást úgy kell elvégezni, hogy kép bal oldalán az első input kép 100%-osan látszik így a második input kép a kép bal oldalán 0%-osan látszik. Ahogy haladunk balról jobbra, az első input kép súlya folyamatosan csökken, míg a második kép súlya folyamatosan nő. A kép jobb oldalán az első input kép súlya 0 lesz, míg a második input kép súlya 100%.

5.7. Blending.

Nehézségi fok: 2

Készíts programot, amely két inputként kapott azonos méretű szürkeárnyaltos képből elkészíti azok súlyozott összegét (*blend*). A súlyozást úgy kell elvégezni, hogy kép tetején az első input kép 100%-osan látszik így a második input kép a kép tetején 0%-osan látszik. Ahogy haladunk fentről lefelé, az első input kép súlya folyamatosan csökken, míg a második kép súlya folyamatosan nő. A kép alján az első input kép súlya 0 lesz, míg a második input kép súlya 100%.

5.8. Képrészlet törlése.

Nehézségi fok: 1

Készíts programot, amely egy szürkeárnyaltos input képen törli (0-ra állítja) a főátló alatti pixeleket! A főátlót azon (x,y) pixelek alkotják, ahol $x = y$.

5.9. Képrészlet törlése.*Nehézségi fok: 1*

Készíts programot, amely egy szürkeárnyaltos input képen törli (0-ra állítja) a főátló feletti pixeleket! A főátlót azon (x,y) pixelek alkotják, ahol $x = y$.

5.10. Képrészlet törlése.*Nehézségi fok: 1*

Készíts programot, amely egy szürkeárnyaltos input képen törli (0-ra állítja) a mellékátló alatti pixeleket! A mellékátlót azon (x,y) pixelek alkotják, ahol $width - x - 1 = y$ és $width$ a kép szélessége.

5.11. Képrészlet törlése.*Nehézségi fok: 1*

Készíts programot, amely egy szürkeárnyaltos input képen törli (0-ra állítja) a mellékátló feletti pixeleket! A mellékátlót azon (x,y) pixelek alkotják, ahol $width - x - 1 = y$ és $width$ a kép szélessége.

5.12. Képrészlet kivágása.*Nehézségi fok: 1*

Készíts programot, amely egy input képből kivágja a középső sorát. Az output ez az egy sorból álló kép lesz.

- színes képekre
- szürkeárnyaltos képekre

5.13. Képrészlet kivágása.*Nehézségi fok: 1*

Készíts programot, amely egy input képből kivágja a középső oszlopát. Az output ez az egy oszlopból álló kép lesz.

- színes képekre
- szürkeárnyaltos képekre

5.14. Képrészlet kivágása.Nehézségi fok: 2*

Készíts programot, amely egy input képből kivág egy megadott téglalapot. A téglalap mérete és pozíciója legyen beállítható a doboz párbeszédablakában!

- színes képekre
- szürkeárnyaltos képekre

5.15. Kép tükrözése.Nehézségi fok: 1*

Készíts egy programot, amely vízszintesen és függőlegesen is tud tükrözni egy tetszőleges input képet! A doboz párbeszédablakában legyen beállítható, hogy függőleges, vízszintesen vagy mindkét tengelyre tükrözzön. Az output a tükrözött kép.

5.16. Hisztogram készítése.*Nehézségi fok: 1*

Készíts egy programot, amely kiszámolja egy tetszőleges szürkeárnyaltos kép hisztogramját! Az output a hisztogram legyen!

5.17. Hisztogram széthúzás.

Nehézségi fok: 2

Készíts olyan programot, amely egy szürkeárnyaltos képen végez hisztogram széthúzást!

5.18. Kontúr meghatározása.

Nehézségi fok: 1

Készíts egy programot, amely bináris (0-1) képen meghatározza az alakzatok kontúrjait! A kontúrt azon objektumpontok alkotják, amelyek 4-szomszédosak valamely háttérponttal.

5.19. Kontúr meghatározása.

Nehézségi fok: 2

Készíts olyan programot, amely bináris (0-1) képen házilégység-technikával (kontúrkövetés) határozza meg a kontúrt!

5.20. RGB kép szürkeárnyaltossá alakítása.

Nehézségi fok: 1

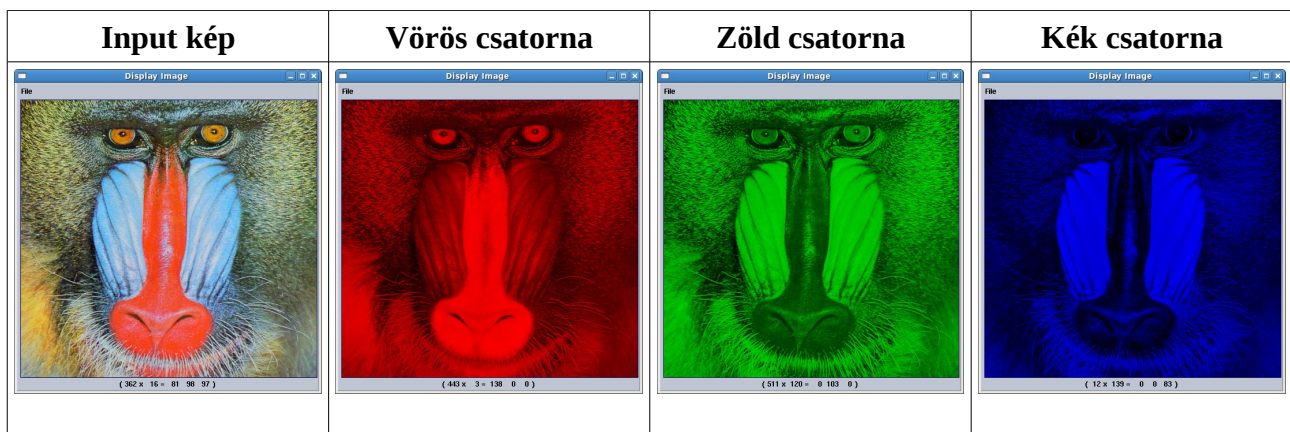
Készíts egy programot, amely egy színes RGB képet szürkeárnyaltossá alakít! A csatornák súlyait (valós értékek) a doboz párbeszédablakában lehessen beállítani!

5.21. Színcsatornák kinyerése.

Nehézségi fok: 2

Készíts egy programot, amely egy színes képből kinyeri a 3 csatornát, és egyenként előállít egy-egy olyan képet, amelyen csak a csatorna tartalma látszik. A három kimenetnek is ugyanolyan típusúnak kell lennie, csak annyi a különbség, hogy a három csatornából csak egy lesz használva.

Minta eredmények egy RGB képre:



5.22. 2D Gauss függvény

Nehézségi fok: 2

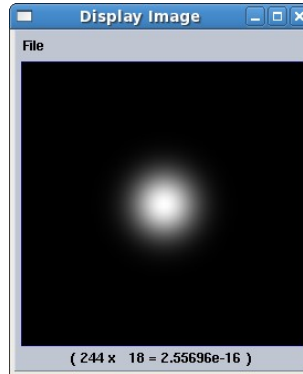
Készíts egy programot, amely a Gauss (G) függvényt adja ki az outputra. A doboznak nincsen inputja, az outputja egy kép, amely a G függvényt ábrázolja. A G függvény paramétereit és a kép méretét a doboz párbeszédablakában lehessen beállítani! A függvény a kép közepén helyezkedjen el! A G függvény képlete:

$$G_{\sigma}(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

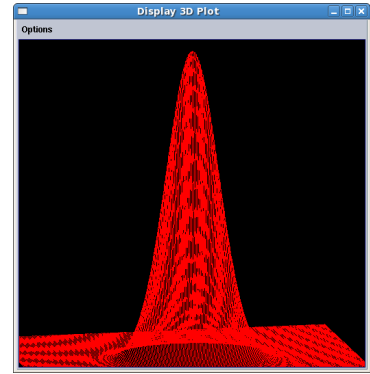
Minta a párbeszédablakhoz és egy eredményhez:



Gauss párbeszédablak



Gauss kép



Gauss plot

***5.23. Éldetektáló operátorok.**

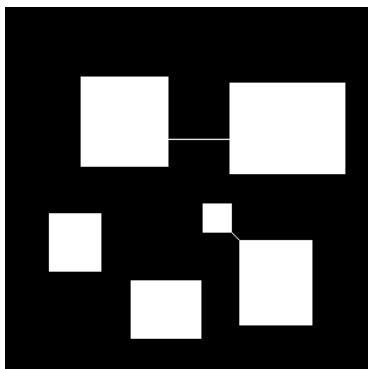
Nehézségi fok: 1

Sobel, Prewitt operátorok megvalósítása. A GUI-n lehet egy kapcsoló, amivel ki lehet választani, hogy melyik operátort hajtsa végre.

****5.24. Címkezés.**

Nehézségi fok: 2

Címkező algoritmus írása. Input egy bináris kép, fekete(0) háttéren fehér(1 vagy 255) objektumok vannak. A nem összefüggő objektumokat külön címkével lássa el. A címkezés 1-től kezdődjön. A GUI-n lehessen kiválasztani, hogy 4-szomszédság, vagy 8-szomszédság szerint történjen az összefüggő elemek meghatározása.



labeling_test.png

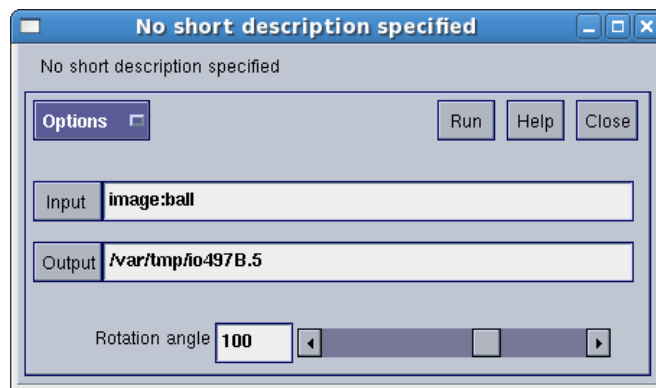


A címkező doboz párbeszédablaka

****5.25. Animáció készítése állóképből.**

Nehézségi fok: 2

Adott egy kép, készíts olyan dobozt, amely ezt a képet adott szögig forgatja animációban. Az animáció lejátszására megvan a megfelelő doboz a rendszerben, ezért a megjelenítéssel nem kell foglalkozni. A GUI-n legyen egy csúszka (egész számok), amelyet [-360, 360] között lehet állítani, alapállapota 0. A doboz nem csinál mást, mint előállít egy mozgóképet, amin az látszik, hogy elforgatja az inputot. Annyi frame-ből álljon az animáció, ahány fokkal lett elforgatva. Ügyelj arra, hogy a kép minden részlete látszódjon a forgatás közben! Az eredmény mérete lehet jóval nagyobb, mint az eredeti kép mérete.



**5.26. Niblack küszöbölés.

Nehézségi fok: 2

A Niblack küszöbölés során egy mozgó ablakot csúsztatunk végig a szürkeárnyaltos képen. Az ablak közepére eső pixelhez határozunk meg egy küszöb értéket. A küszöb értéket az ablakban eső pixelek intenzitása határozza meg a következő képlet alapján:

$$T(x, y) = \text{Avg}(E(x, y)) + k \cdot \text{Var}(E(x, y)),$$

ahol $T(x,y)$ az (x,y) ponthoz tartozó küszöbérték, $\text{Avg}(\dots)$ az ablakban szereplő intenzitások átlaga, $\text{Var}(\dots)$ az ablakban lévő intenzitások szórása, k egy valós konstans (lehet negatív is!!!). Az $E(x,y)$ az (x,y) ponthoz tartozó ablakban lévő intenzitások halmaza. Az algoritmus két paramétere (ablak méret és k) állítható legyen a GUI-n keresztül.

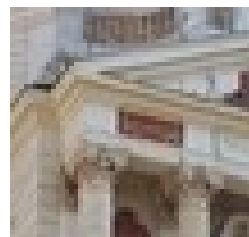
***5.27. Képrészlet keresése.

Nehézségi fok: 2

Hová illik a képrészlet? Keresd meg, hogy a kivágott képrészlet hová illik a nagy képen! Jelöld be 3 pixel vastagságú piros kerettel a képrészlet helyét a kimeneti képen!



szinhaz.jpg

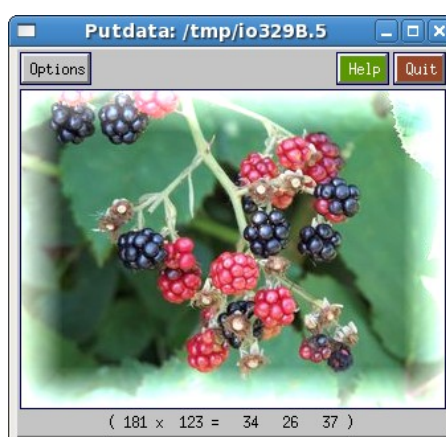
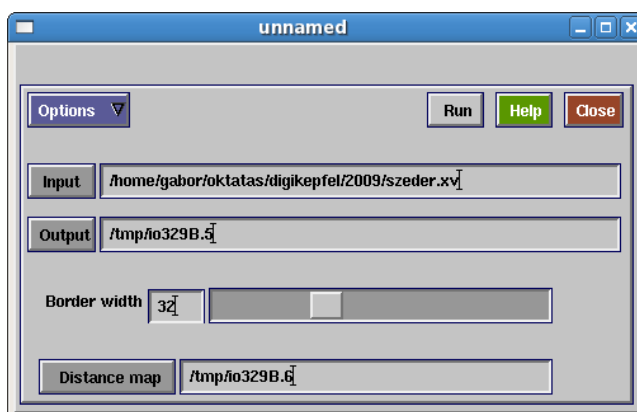


szinhaz_reszlet.jpg

***5.28. Fehéredő keret.

Nehézségi fok: 3

Készíts olyan dobozt, amely egy képet fehéredő kerettel lát el! A keret mérete állítható legyen 1 és 100 között! Tipp: a fehéredő kerethez célszerű távolságtérképet számolni (lásd: <http://nik.bmf.hu/vamossy/GepiLatas2007/10Skeleton/szkeleton.ppt> -- 15-26. dia vagy érdeklődj a gyakorlatvezetőktől). Több plusz pontot ér a feladat, ha színes (RGB) és szürkeárnyaltos képre egyaránt működik. Minta a doboz párbeszédablakához:



*****5.29. Két kép kombinálása.**

Nehézségi fok: 3

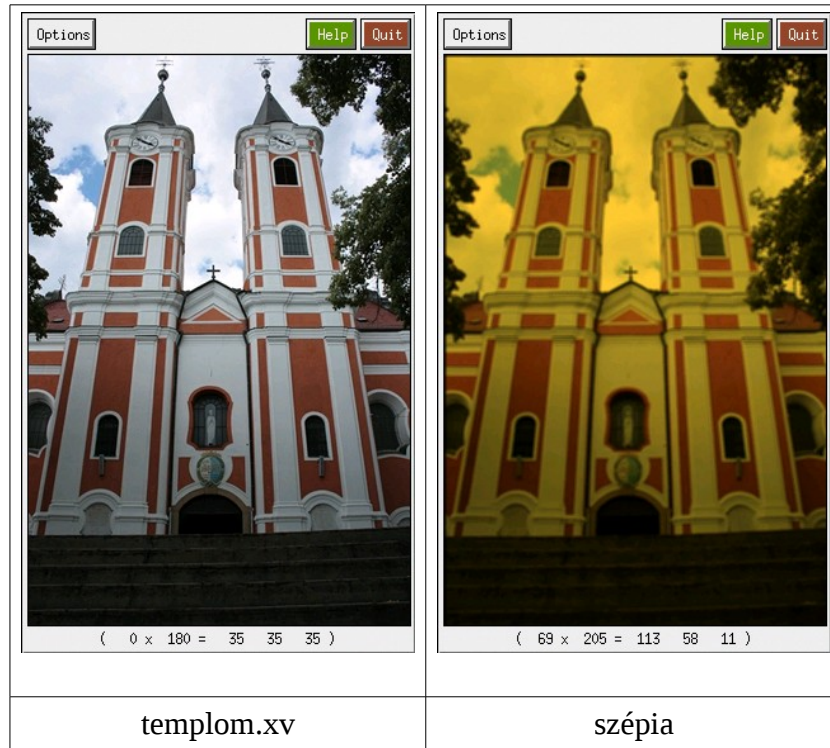
Készíts olyan dobozt, amely a megadott két input képet az alábbi módon kombinálja! A képek azonos méretűek!

Objektum	Háttér	Kombinált kép
airplane.xv	clouds.xv	Airplane and clouds

*****5.30. Szépia kép előállítás (kép „öregítése”).**

Nehézségi fok: 2

Adott egy színes kép (pl. a templom.xv), készíts olyan dobozt, amely szépia-hatást eredményez a képen.



***** 5.31. Képrészlet hangsúlyozása a háttér elmosásával.**

Nehézségi fok: 3

Készíts olyan dobozt, amellyel kihangsúlyozhatod egy színes kép adott részletét egy kör alakú régióban a háttér (a kép régió kívüli részének) elmosásával! Legyen beállítható a kör alakú régió átmérője, és középpontja! A konvolúciós maszk mérete lehet konstans.

Példa:

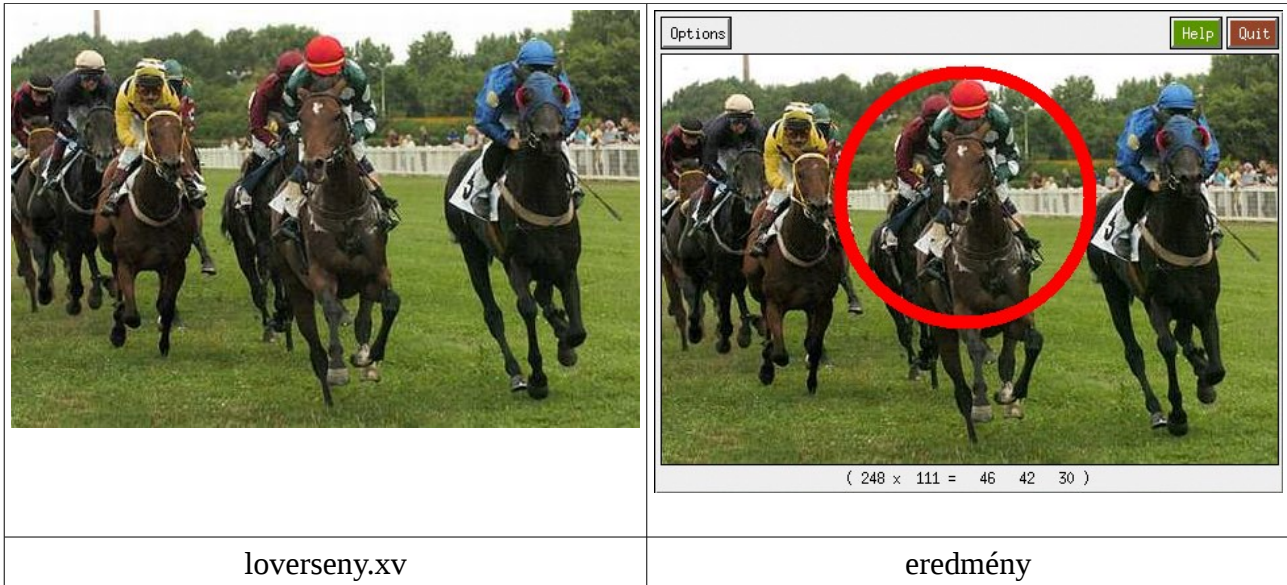


***** 5.32. Képrészlet megjelölése a képen.**

Nehézségi fok: 3

Adott egy színes kép. Készíts olyan dobozt, amellyel bejelölheted rajta a fontos képrészletet egy piros gyűrű segítségével! A doboz párbeszédablakában legyen beállítható a külső és belső körív átmérője és a gyűrű középpontja!

Példa:



***** 5.33. Harris sarokdetektáló operátor.**

Nehézségi fok: 3

Készíts dobozt a Harris-féle sarokdetektáló operátor megvalósítására!

***** 5.34. Moravec-féle sarokdetektor.**

Nehézségi fok: 3

Készíts dobozt a Moravec-féle sarokdetektáló operátor megvalósítására!

***** 5.35. Egyenes detektálása Hough-transzformáció segítségével.**

Nehézségi fok: 3

Készíts dobozt, amely Hough-transzformáció segítségével detektál egyeneseket egy bináris képen!

***** 5.36. Körvonal detektálása Hough-transzformációval.**

Nehézségi fok: 3

Készíts dobozt, amely Hough-transzformáció segítségével detektál körvonalakat egy bináris képen!

*** 5.37. Medián szűrő.**

Nehézségi fok: 1

Készíts dobozt, amely egy medián valósít meg! A doboz párbeszédablakában legyen beállítható a medián szűrő mérete.

*** 5.38. Hisztogram kiegyenlítés.**

Nehézségi fok: 1

Készíts dobozt a hisztogram kiegyenlítés megvalósítására!

*** 5.38. Álszínezés.**

Nehézségi fok: 2

Készíts dobozt, amely egy szürkeárnyaltos képen végez álszínezést. Az álszínezés azt jelenti, hogy a szürkeskálát egy színskálába transzformáljuk át. Ügyelj színskála folyamatos átmenetére!

***** 5.39. Régiónövelésen alapuló szegmentálás.**

Nehézségi fok: 2-3

Adott egy szürkeárnyaltos kép és egy vagy több kiindulási pont (seed points). A kiindulási pontokból párhuzamosan (egyidőben) végezz régiónövelést, és állítsd elő a szegmentált képet. A régiónövelés során a régió határával szomszédos pixeleket (amelyek még nem tartoznak egy régióhoz sem) vagy hozzávesszük a régióhoz vagy nem a pixel intenzitásától függően. A régiónövelés lehet

- globális küszöbszint alapú: ha régió határával szomszédos pixel még nem tartozik egy régióhoz sem, és küszöbértéken belül van, akkor vegyük hozzá a régióhoz, különben ne. (A globális alsó és felső küszöb legyen beállítható a doboz párbeszédablakában!)
- túrés alapú: ha régió határával szomszédos pixel még nem tartozik egy régióhoz sem, és „nincs nagy különbség” a vizsgált pixel és a vele szomszédos régióhoz tartozó pixel intenzitása között, akkor vegyük hozzá a régióhoz, különben ne. (A túréshatár legyen beállítható a doboz párbeszédablakában!)
- statisztika alapú: számoljuk ki a régió belüli pixel-intenzitások átlagát és szórását. Ha egy régió határával szomszédos pixel intenzitása kisebb és az átlag intenzitás különbsége kisebb, mint a szórás k -szorosa, akkor vegyük hozzá a pixelt a régióhoz. (Legyen k beállítható a doboz párbeszédablakában!)

**** 5.40. Morfológiai műveletek.**

Nehézségi fok: 1

Készíts dobozt, amely a morfológiai eróziót és dilatációt valósítja meg bináris képekre! A párbeszédablakban legyen kiválasztható, hogy eróziót vagy dilatációt akarunk végezni!

**** 5.41. Morfológiai műveletek.**

Nehézségi fok: 1

Készíts dobozt, amely a morfológiai nyitást és zárást valósítja meg bináris képekre! A párbeszédablakban legyen kiválasztható, hogy nyitást vagy zárást akarunk végezni!

***** 5.42. Morfológiai váz meghatározása.**

Nehézségi fok: 2

Készíts dobozt, amely előállítja egy bináris képen az objektumok morfológiai vázát! A morfológia váz iterációs lépések sorozataként áll elő. Legyen A egy bináris kép.

Legyen S a morfológiai vázat tartalmazó kép. Ez kezdetben üres.

REPEAT

// az i -edik iterációban végzett műveletek

- 1.) Erodáljuk az $A^{(i)}$ képet a T szerkesztőelemmel, jelölje $A^{(i)}_E$ az erodált képet.
- 2.) Dilatáljuk az $A^{(i)}_E$ képet a T szerkesztőelemmel, jelölje $A^{(i)}_D$ a dilatált képet
- 3.) Szinezük 1-re az S képen azokat a pixeleket, amelyek szerepelnek az $A^{(i)}$ képen, de nem szerepelnek az $A^{(i)}_D$ képen!
- 4.) $A^{(i+1)} = A^{(i)}_E$; $i = i + 1$;

UNTIL nincs változás az S képen

***** 5.43. Árnyék illesztése alakzathoz konstans háttérren.**

Nehézségi fok: 3

Tegyél árnyékot a repülőgép mögé az ábrán látható módon! Az árnyék legyen valamilyen halványszürke színű és alkalmazz rá egy simítást is, hogy ne legyen éles kontúrja! A doboz párbeszédablakában lehessen beállítani az árnyék intenzitását 0-255 között, az árnyék x és y távolságát a repülőhöz képest!

