

Tömörítési hiba illusztrálása Gimppe

Lépések:

1. Importáltam az arcképet gimpbe
2. Ezután a Layer -> Transparency beállításokban hozzáadtam egy Alpha-csatornát
3. Majd Free Select tool segítségével körbevágtam magam
4. Hozzáadtam egy új layert, majd rátettem egy háttérképet
5. A layerek sorrendjét megfordítottam
6. Átméreteztem magam, hogy ráférjek a képre
7. Átméreteztem a canvas méretet 600x600-ra



Image	
Image ID	
Dimensions	600 x 600
Width	600 pixels
Height	600 pixels
Horizontal resolution	300 dpi
Vertical resolution	300 dpi
Bit depth	24

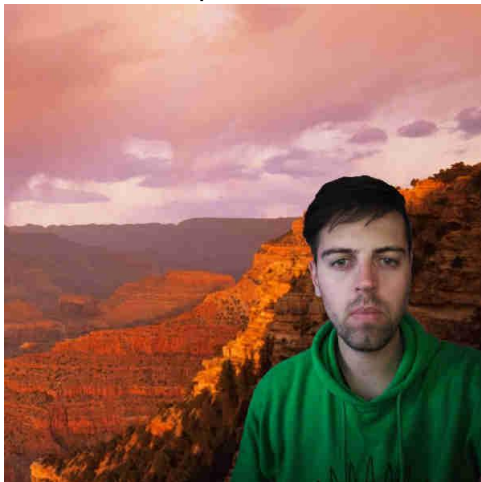
Alap

A tömörítés mértékét a megadott formula segítségével tudjuk kiszámolni

tömörítés mértéke = $24 \text{ [bit/pixel]} / (\text{adott képfájl mérete [bit]} / \text{píxelek száma [db]})$

$$24/(762168/(600*600)) = 11.336$$

Az elkészült kép 20-as és 90-es minőségben:

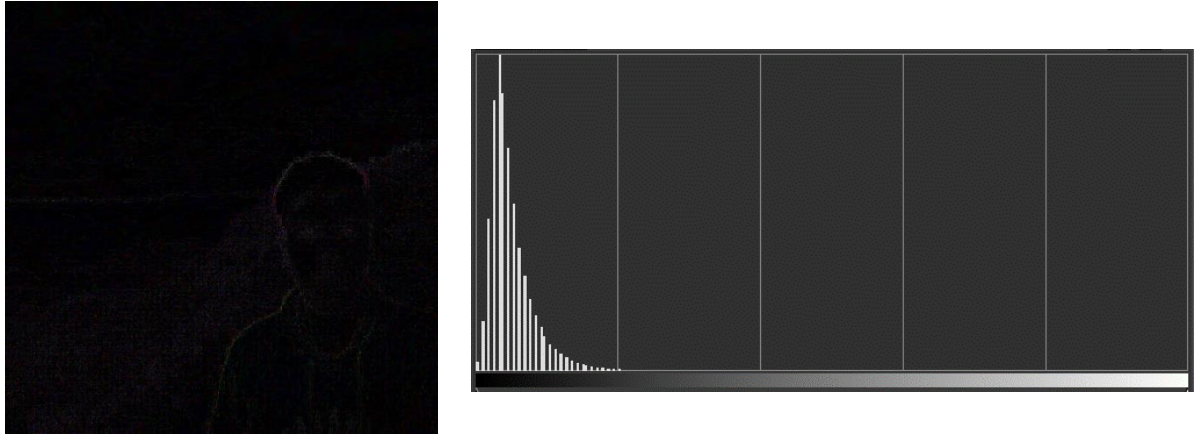


20



90

Az exportált képeket megnyitottam Gimp-ben 2 db layerként. Átállítottam a layer műveletet különbségre, majd utána a 2 layert összemostam, világosítottam és az alábbi kép lett az alábbi hisztogram értékekkel



A képek a Beadando1/Beadando1/images/original mappában találhatóak

Hisztogramtranszformációk

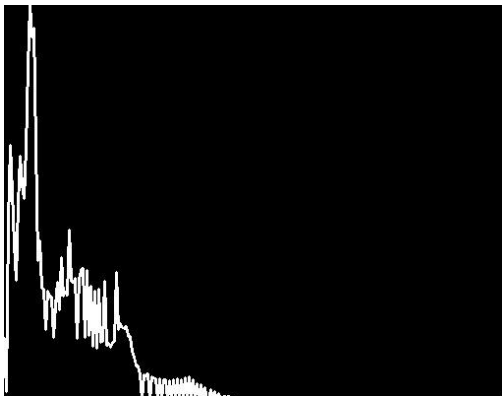
Lineáris széthúzás

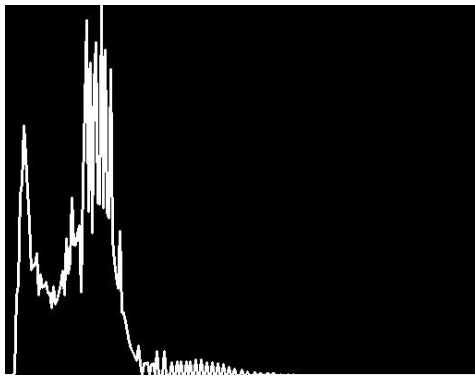
A képek világosodtak, és részletesebbek lettek. peppers_sotet

Eredeti

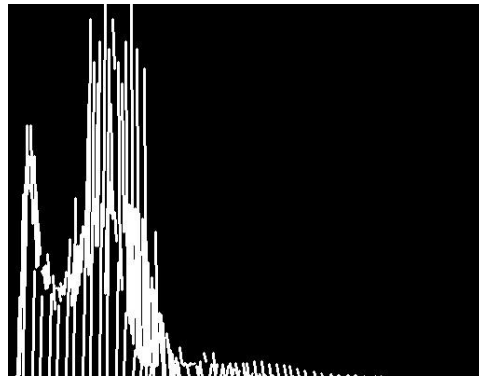


Lineárisan széthúzott





boat_sotet alap



boat_sotet lineárisan széthúzott

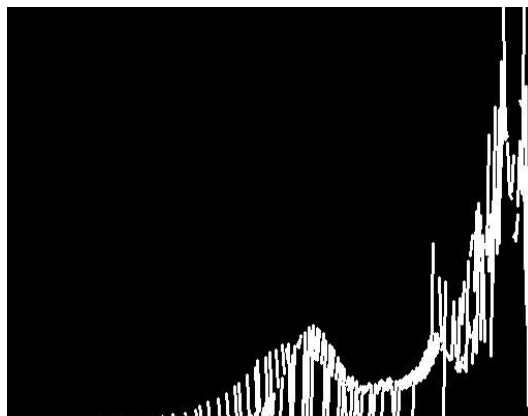
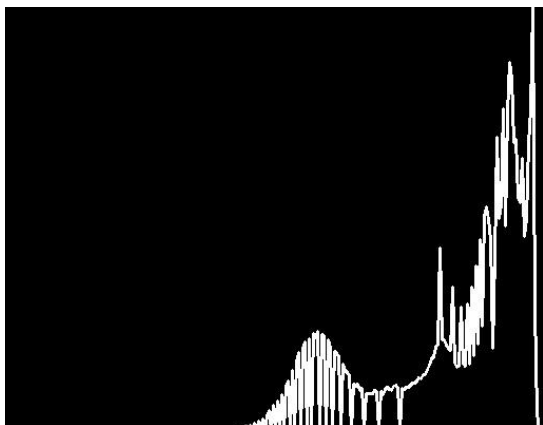
Gyökös széthúzás

A peppers_sötet, peppers_vilagos és lena_vilagos-ra teszteltem le a gyökös széthúzást.
lena_vilagos

Alap



Módosított



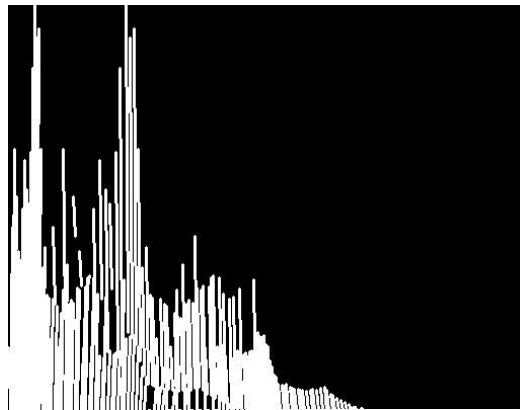
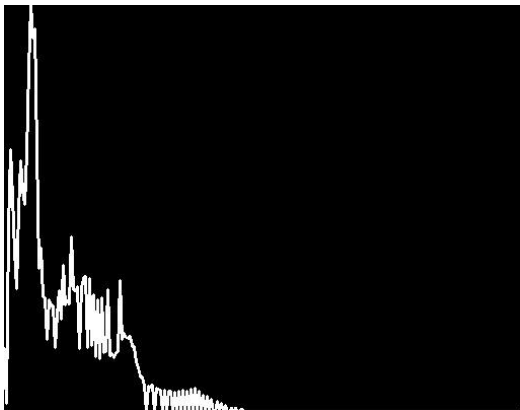
A lena_vilagos kép világosabb és kevésbé részletes lett

peppers_sotet

Alap



Módosított



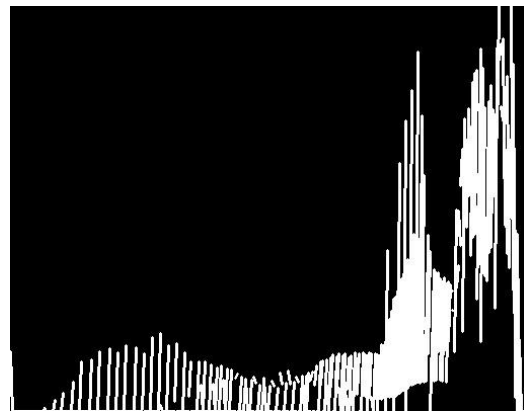
A peppers_sotet kép világosabb és részletesebb lett

peppers_vilagos

Alap



Módosított



A peppers_vilagos világosabb és kevésbé részletes lett

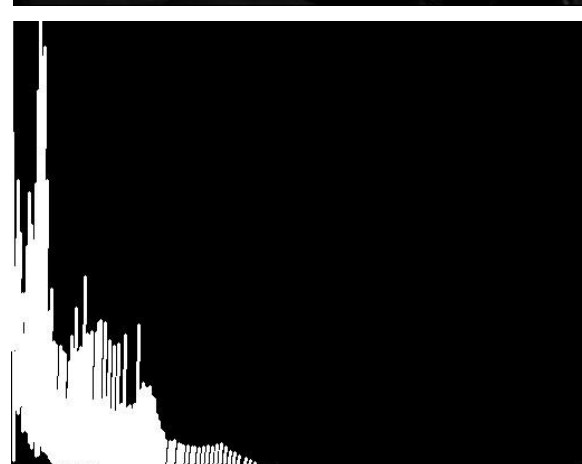
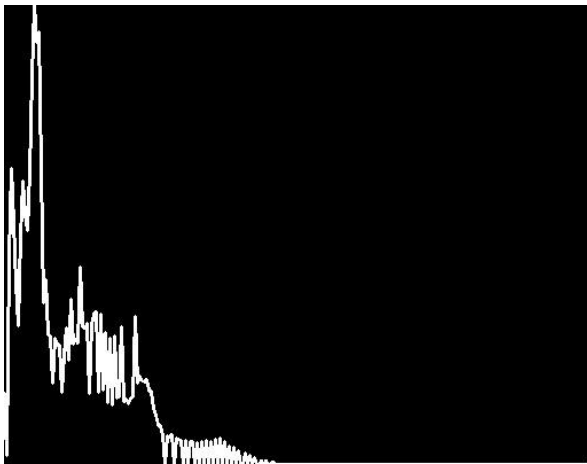
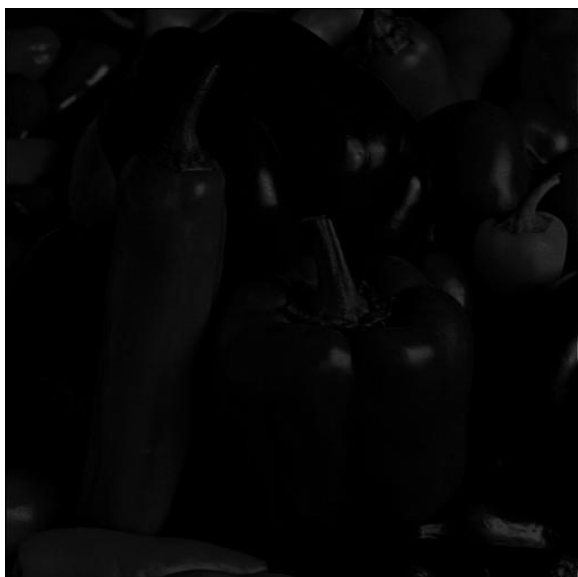
Négyzetes széthúzás

peppers_sotet

Alap



Módosított



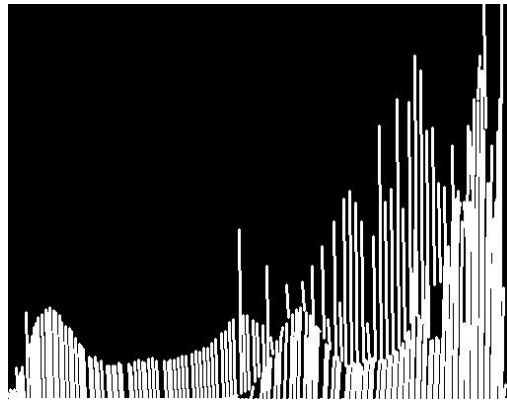
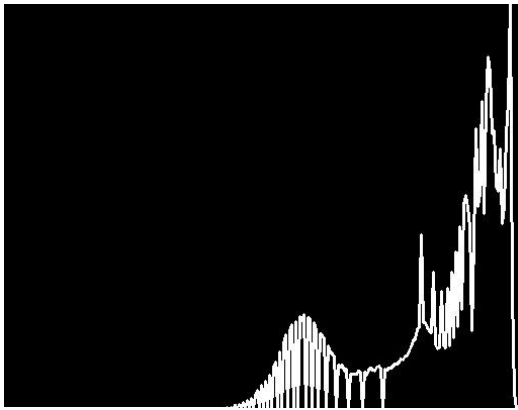
A peppers_sotet sötétebb és kevésbé részletes lett

lena_vilagos

Alap



Módosított



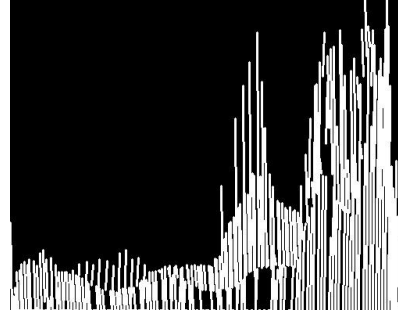
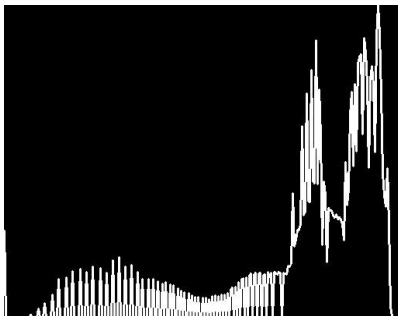
A lena_vilagos sötétebb és részletesebb lett

peppers_vilagos

Alap



Módosított



a peppers_vilagos sötétebb és részletesebb lett

A hisztogramokon megfigyelhető, hogy az algoritmus feltölti színekkel a képeket, ettől a világos képek részletesebbek lesznek, de a sötétek kevésbé részletesek.

A hisztogram széthúzáshoz a main.cpp fájlban található 68-77 sorban a függvény meghívása a képekre, a függvények a 349, 159, 168, 177 sorokban találhatóak

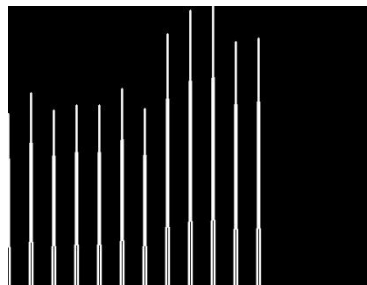
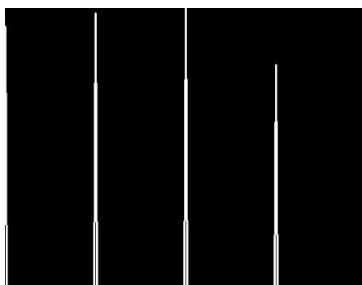
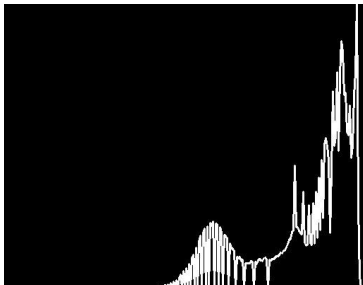
Hisztogramkiegyenlítés

A lineáris kiegyenlítés során a program jobban kiegyenlíti a színeket.

lena_vilagos: Alap

K4

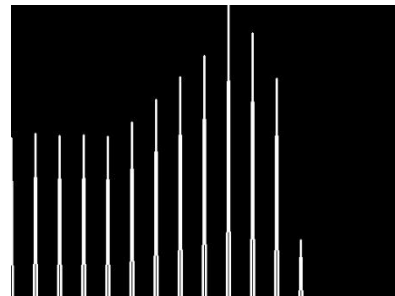
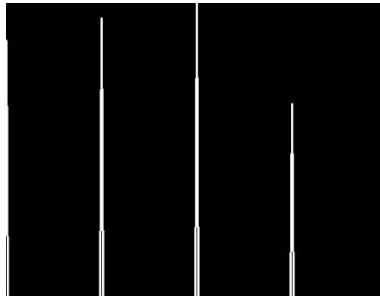
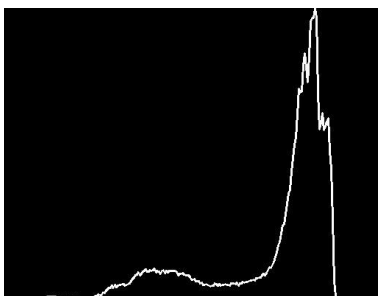
K16



airplane: Alap

K4

K16



Kiegyenlítette a képen a színeket, tehát nem volt a kép egyik fele túl sötét, a másik túl világos, ezzel az algoritmussal egységesebbé tettük.

A hisztogram kiegyenlítéshez a main.cpp fájlban található 80-81 sorban a függvény meghívása a képekre, a függvények a 414, 186 sorokban találhatóak

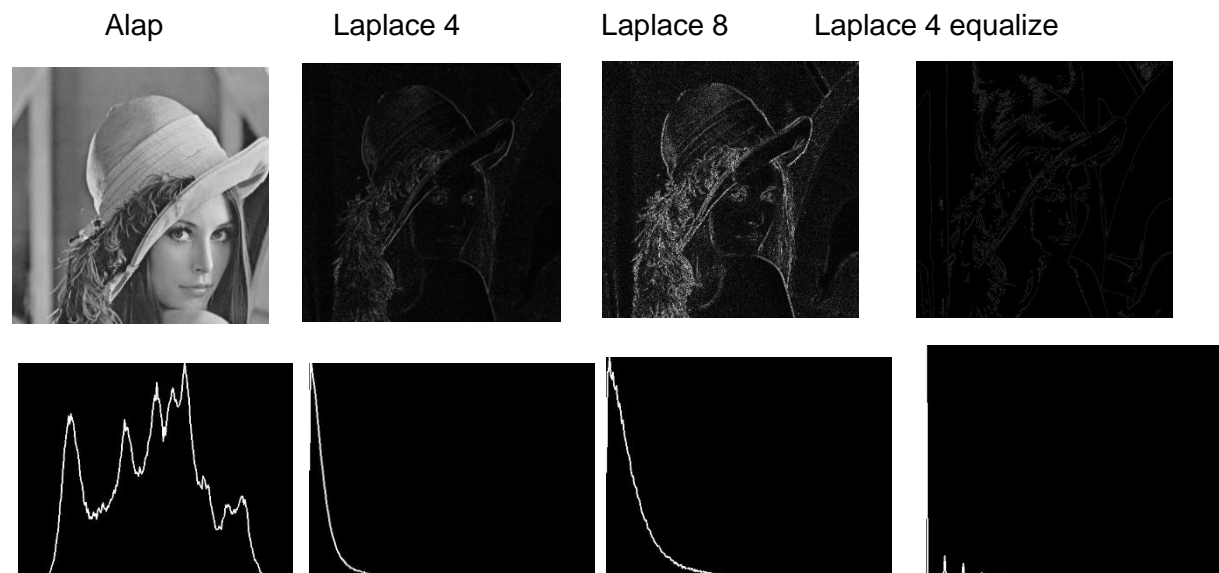
Konvolúciók

Laplace és a Sobel algoritmus eltünteti a színeket, és csak a sötét területek körvonalait rajzolja ki.

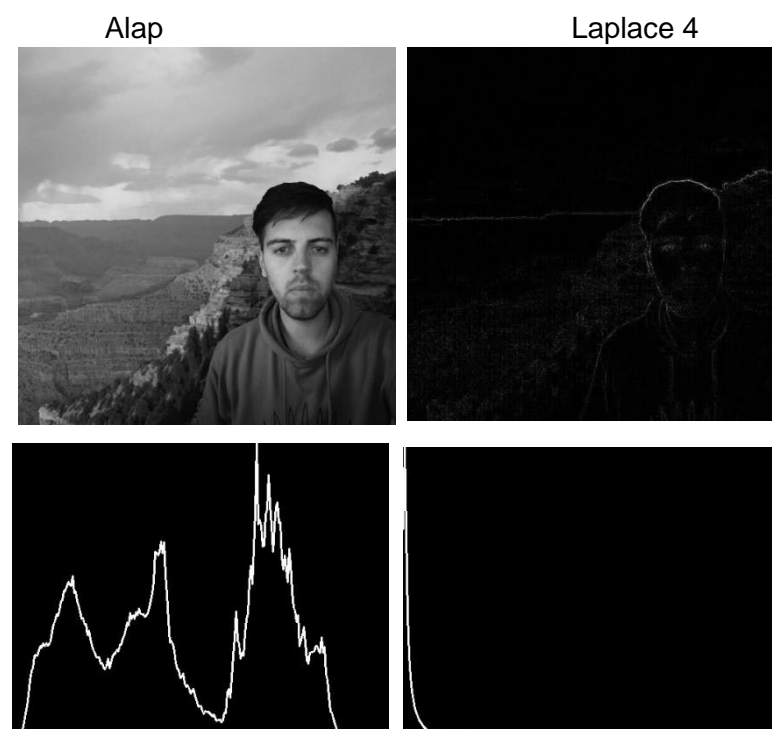
4 operátorral, 8 operátorral, 4 operátorral és hisztogramkiegyenlítéssel.

A saját képem (montage) és lena képeket teszteltem.

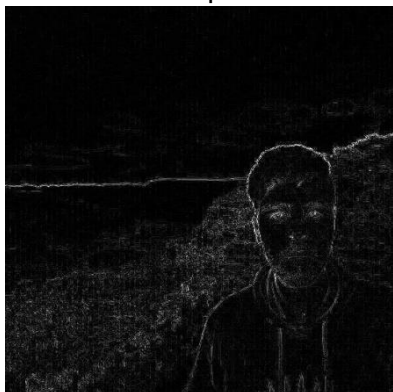
lena



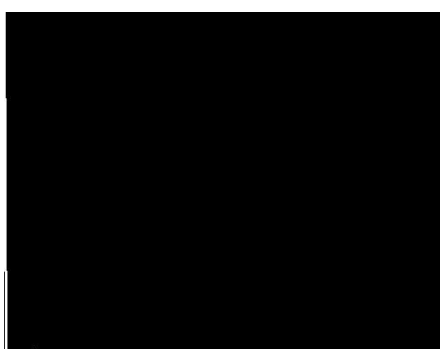
montage



Laplace 8



Laplace 4 equalize

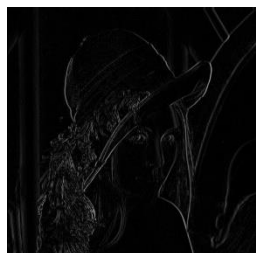


lena:

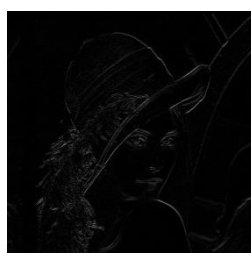
Alap



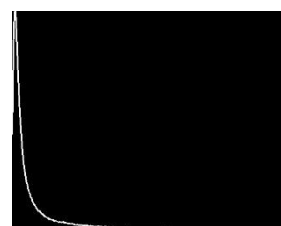
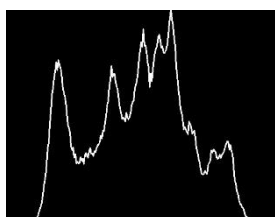
függőleges



vízszintes



összetett

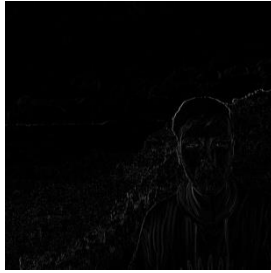


montage

alap



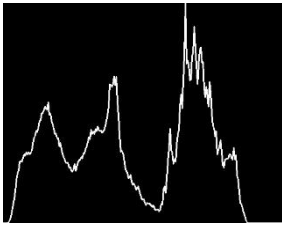
függőleges



vízszintes



összetett



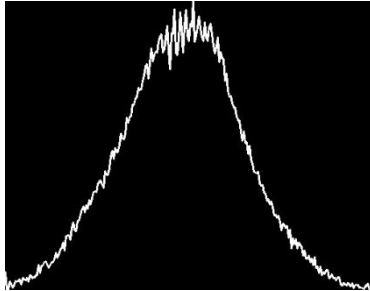
A konvolúciókhoz a main.cpp fájlban található 85-86 sorban a függvény meghívása a képekre, a függvények a 465, 284, 309, 327, 186 sorokban találhatóak

Wallis

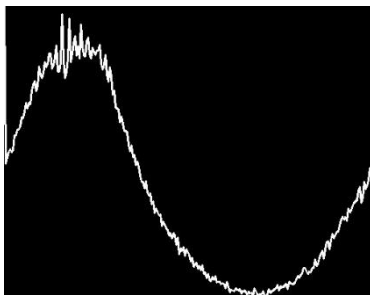
A kontraszt és világosság értékének állításával a következő képeket kaptuk:

bridge:

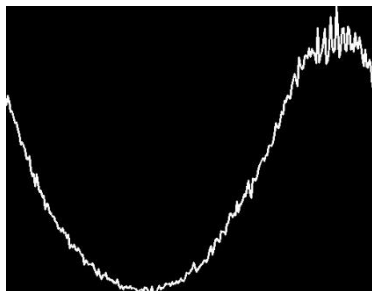
Md:128,Sd:100



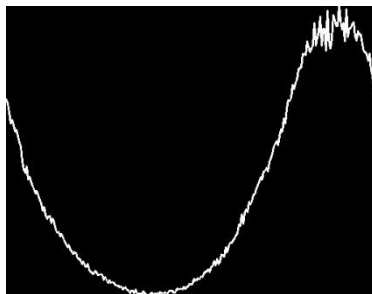
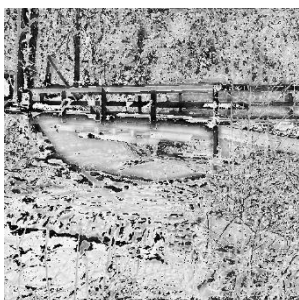
Md: 32, Sd: 100



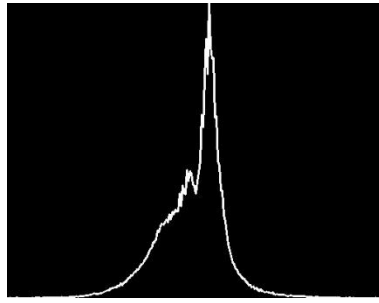
Md: 256, Sd: 100



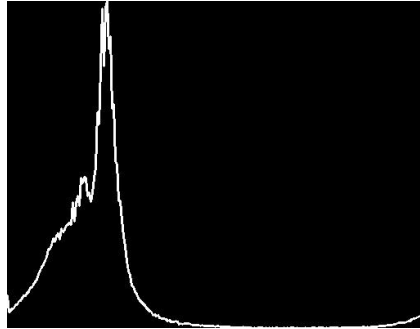
Md: 256, Sd: 50



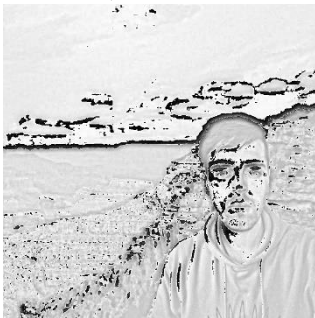
montage:
Md: 128, Sd: 100



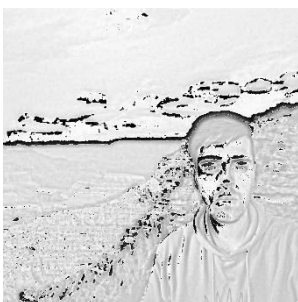
Md: 32, Sd: 100



Md: 256, Sd: 100



Md: 256, Sd: 50



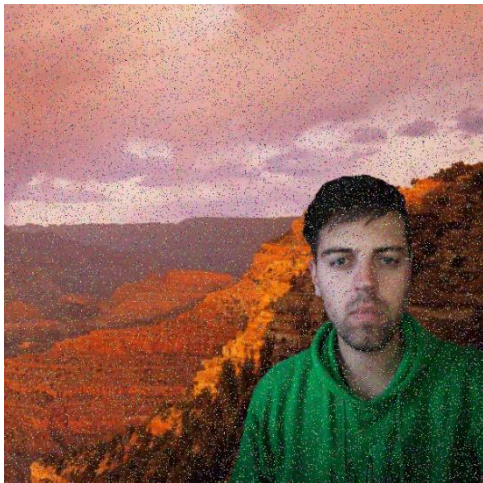
A wallis szűrőkhöz a main.cpp fájlban található 89-99 sorban a függvény meghívása a képekre, a függvények a 585, 208, 231 sorokban találhatóak

Nemlineáris szűrők

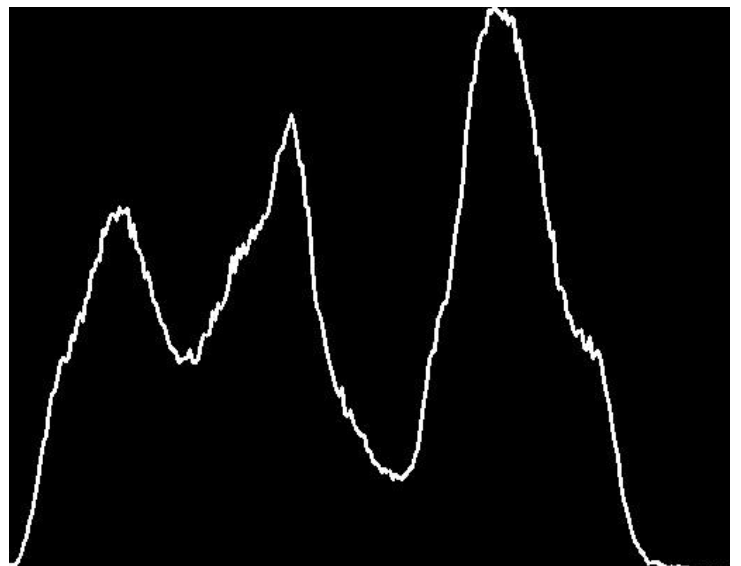
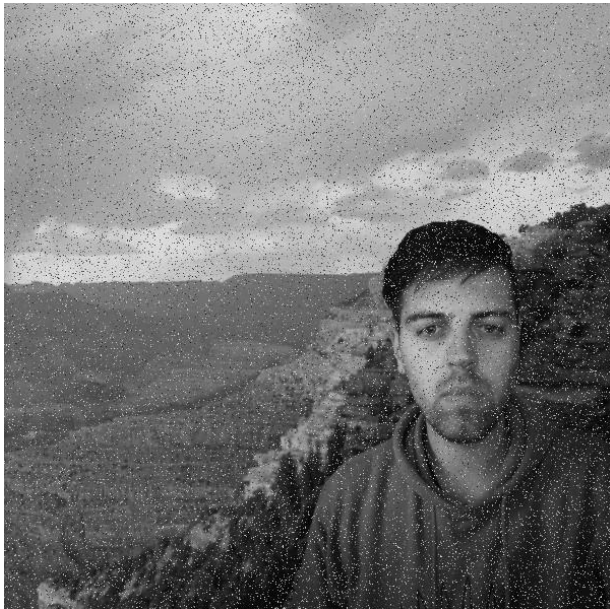
Ezek az algoritmusok zajok szűrésére alkalmasak. Összehasonlíthatjuk a pixelértékeinket a lokális átlagok értékeivel, amelyet egy értékhez hasonlítunk, vehetjük egy "ablakon" belül a pixelek mediánját, és külön meghatározzuk az oszlopok mediánját, majd ennek a számhalmaznak vesszük a mediánját.

A nem-lineáris szűrőköz a main.cpp fájlban található 102-104 sorban a függvény meghívása a képekre, a függvények a 620, 250 sorokban találhatóak

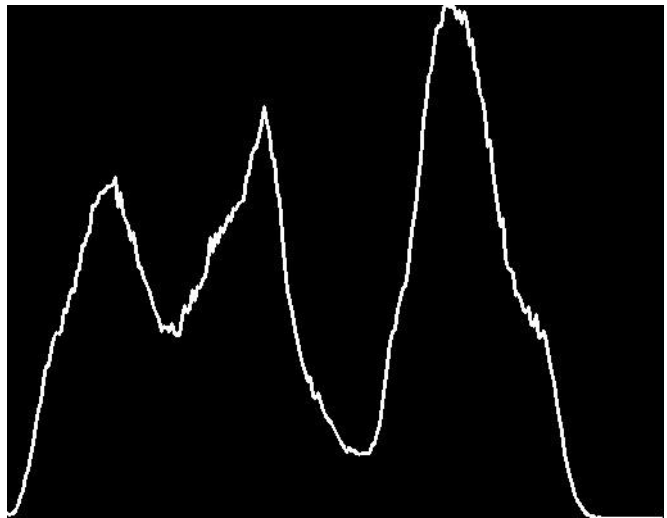
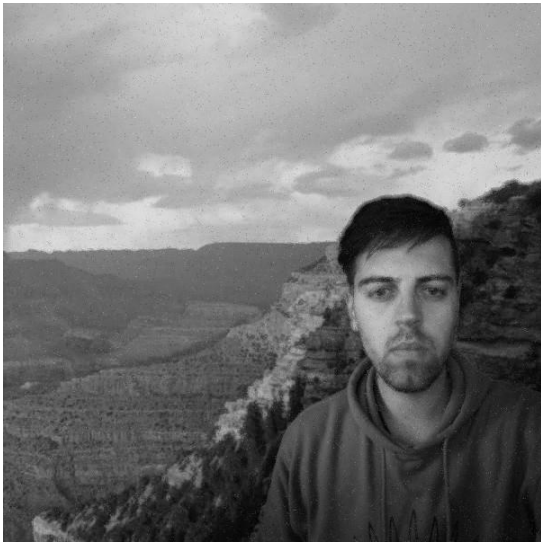
saját képem zajjal:



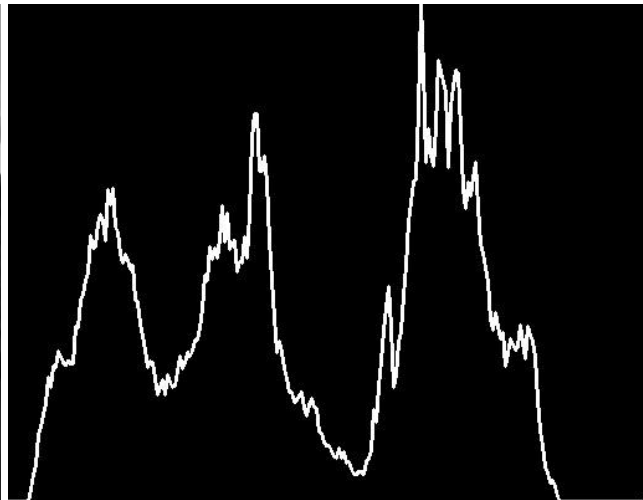
szürke zajos



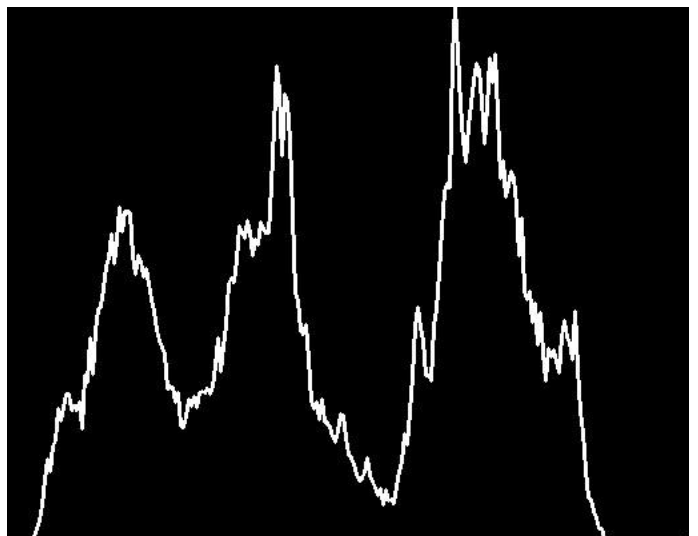
Outlier



Median

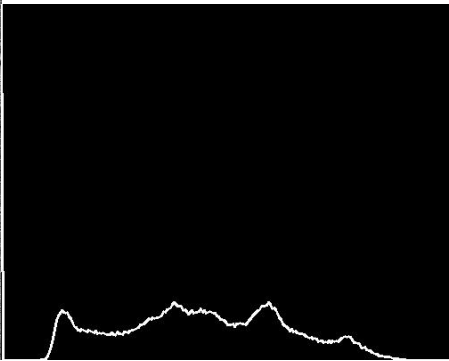


Fast Median

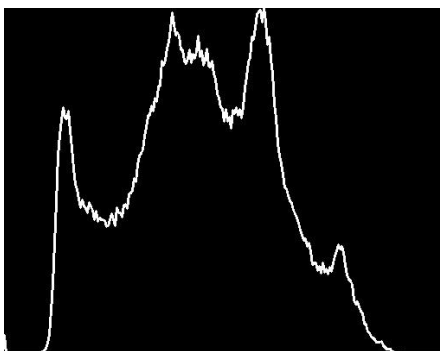


Barbara 0.01

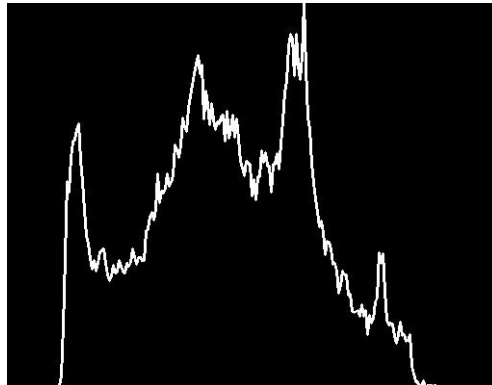
Zajos



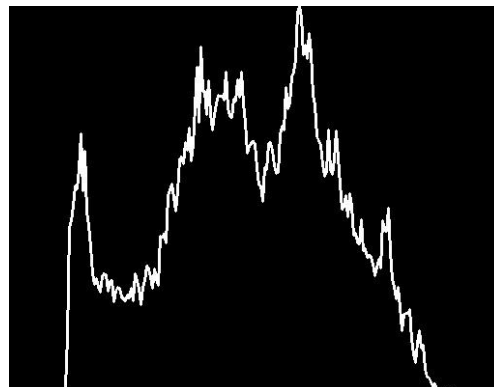
Outlier



Median

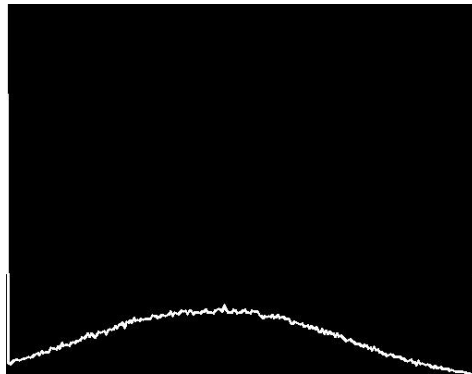
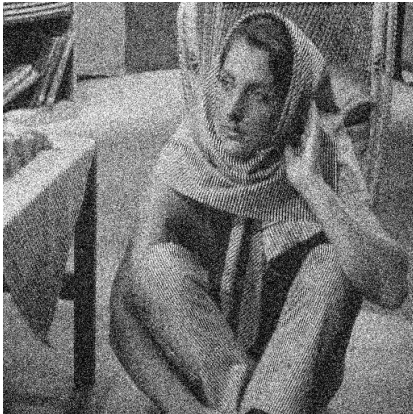


Fast Median

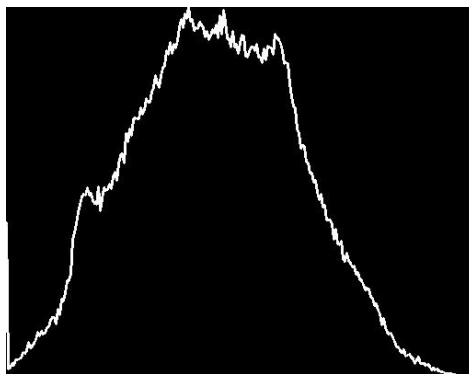
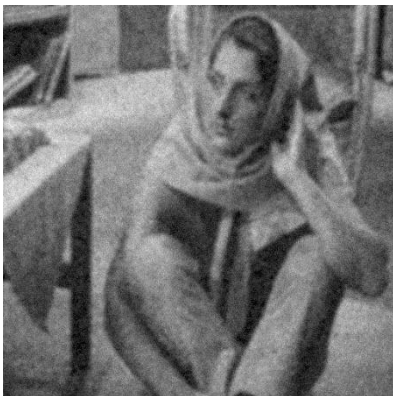


Barbara 0.025

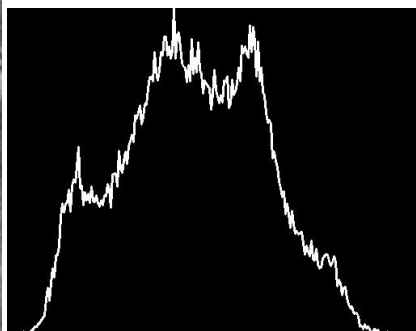
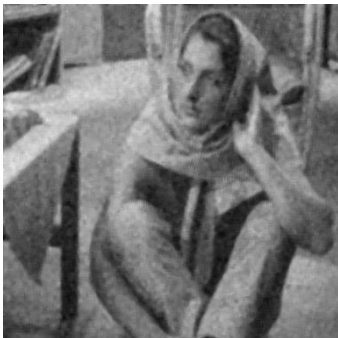
Zajos



Outlier



Median



Fast Median

