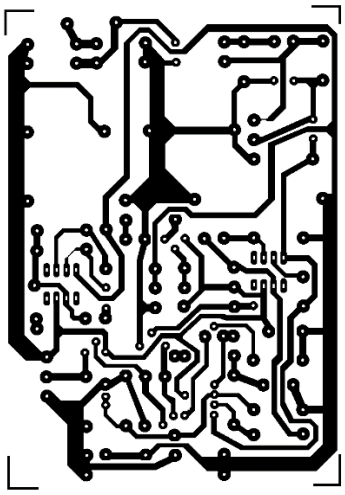
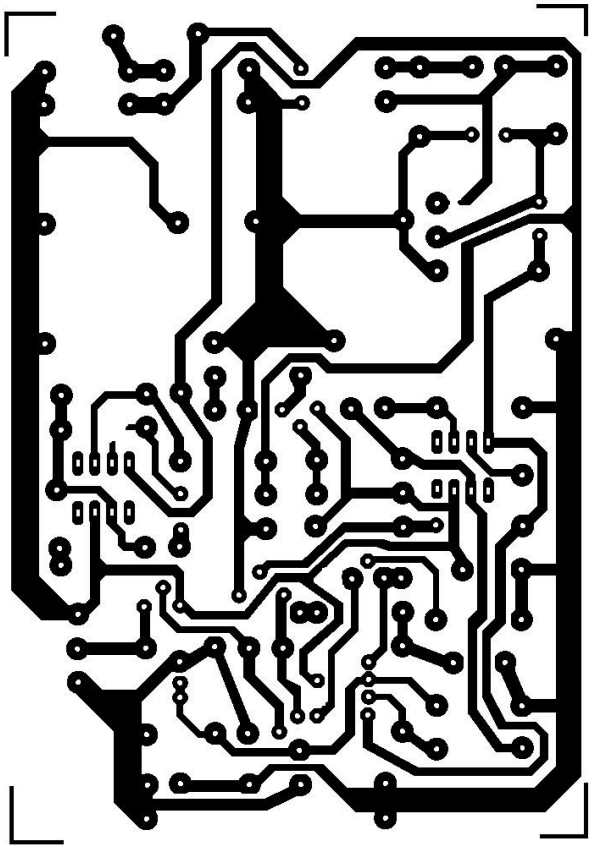
Képi morfológia

Az algoritmusok a main.cpp fájlban találhatóak, 110 – 222 sor, a függvények meghívása a 73-74 sorokban történnek. A feladatban a pcb_hibas_8bpp és pcb_hibas_8bpp7 képeket használtam. Utóbbi azért, mert a neptun kódomban az első számjegy 8.

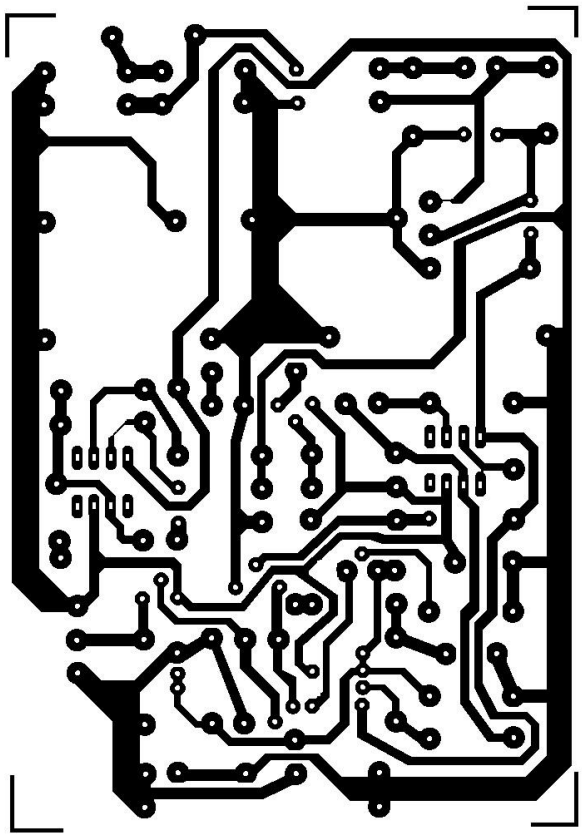
pcb_hibas_8bpp eredeti:



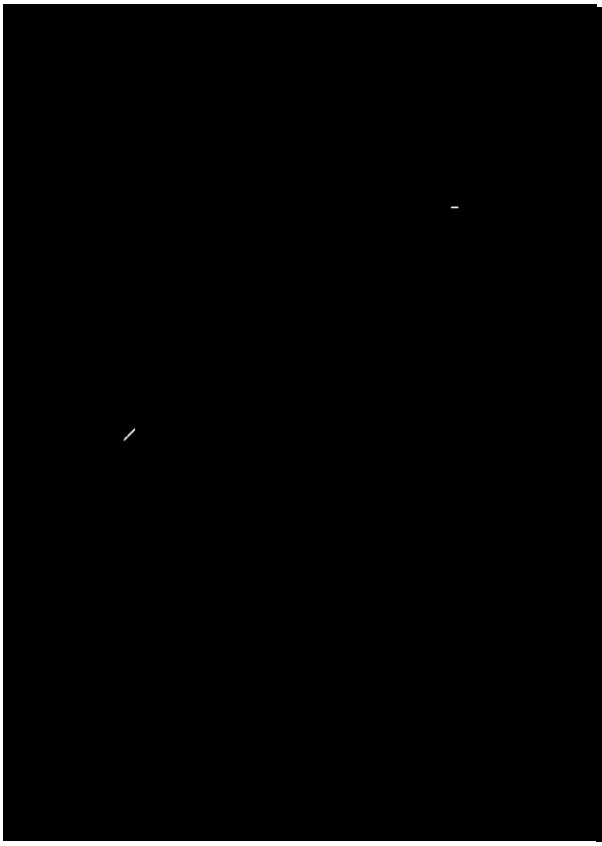
pcb_hibas_8bpp nyitás után:



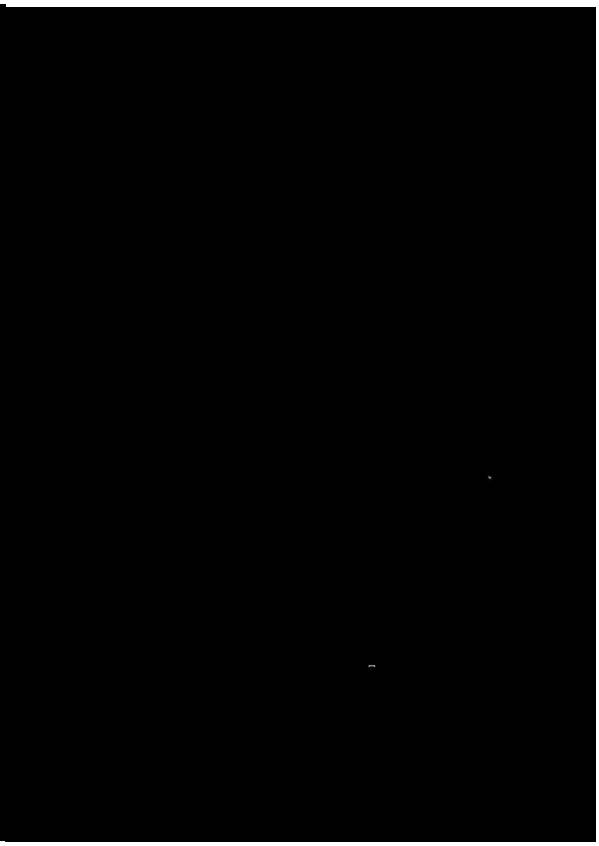
pcb_hibas_8bpp zárás után:



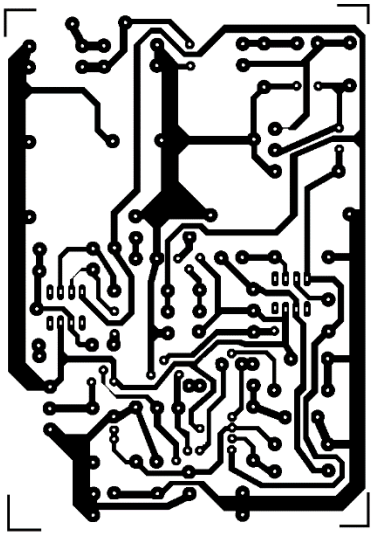
pcb_hibas_8bpp túl vékony:



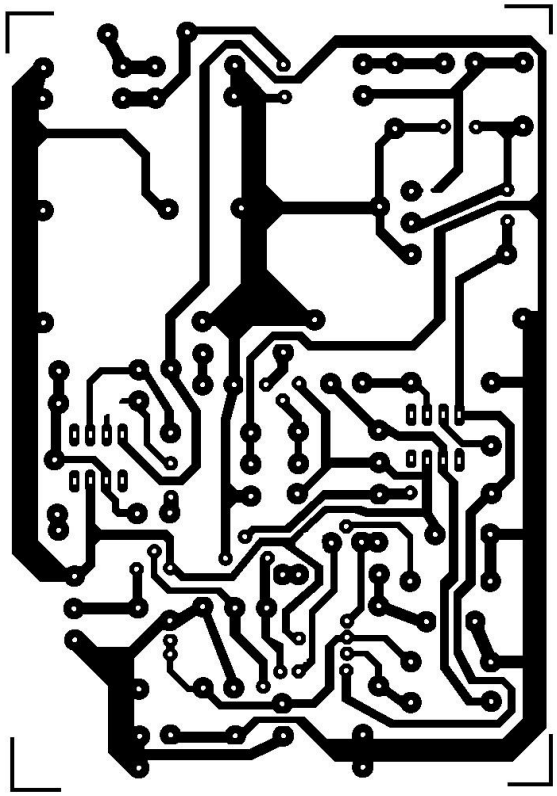
pcb_hibas_8bpp túl közeli:



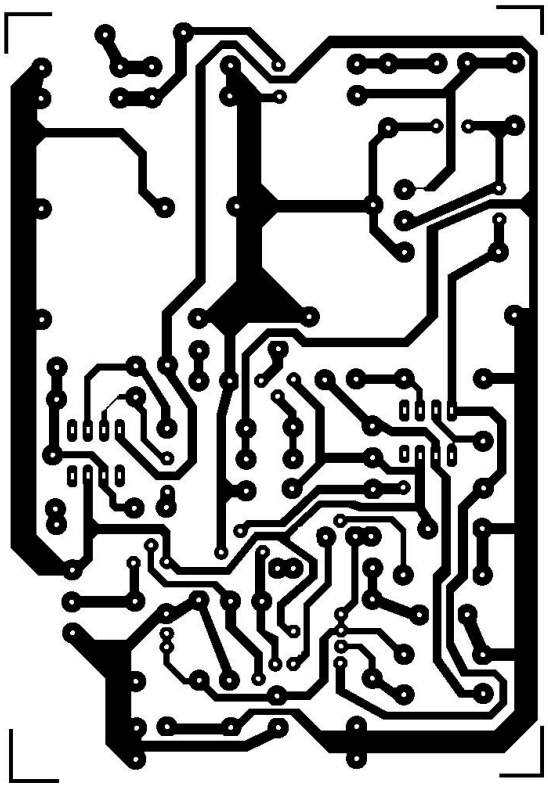
pcb_hibas_8bpp7 eredeti



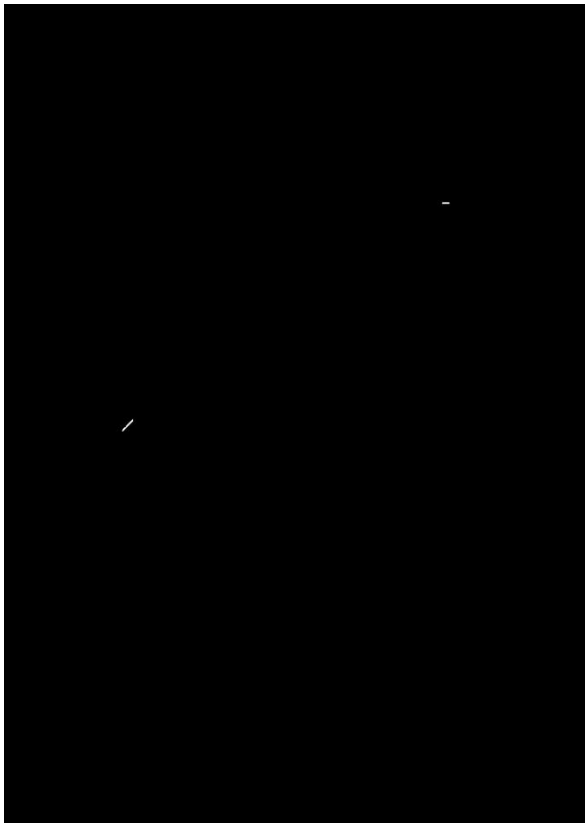
pcb_hibas_8bpp7 nyitás után:



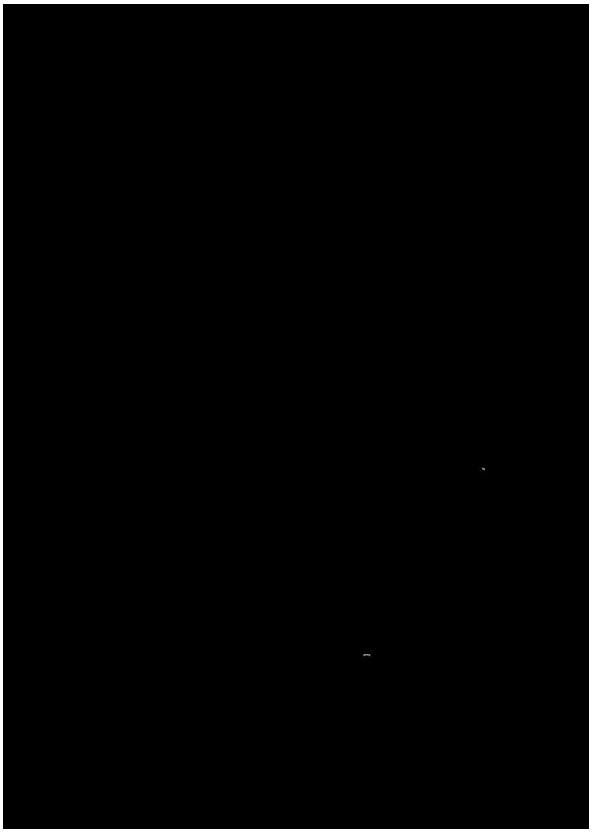
pcb_hibas_8bpp7 zárás után:



pcb_hibas_8bpp7 túl vékony:



pcb_hibas_8bpp7 túl közeli

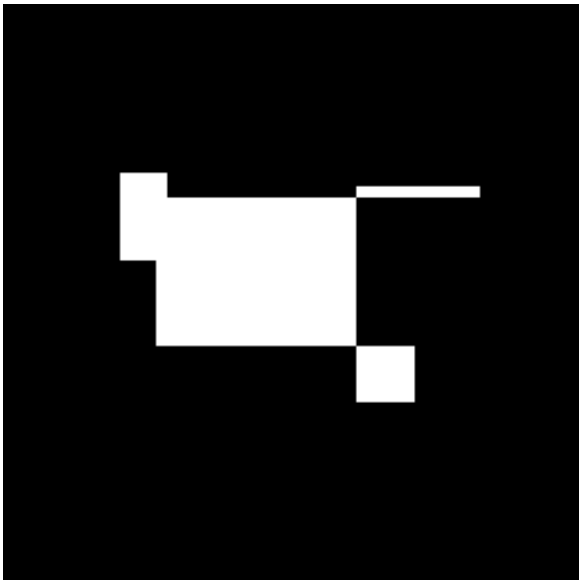


Bogárkövető algoritmus

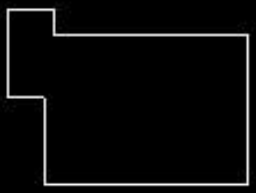
Az algoritmusok a main.cpp fájlban találhatóak, 224 – 400 sor, a függvények meghívása a 76-77 sorokban történnek. A feladatban a bug és bug7 képeket használtam. Utóbbi azért, mert a neptun kódomban az első számjegy 8.

Az algoritmus segítségével körbe járjuk a képeket.

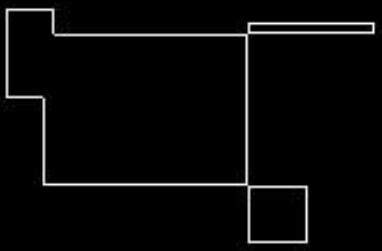
bug:



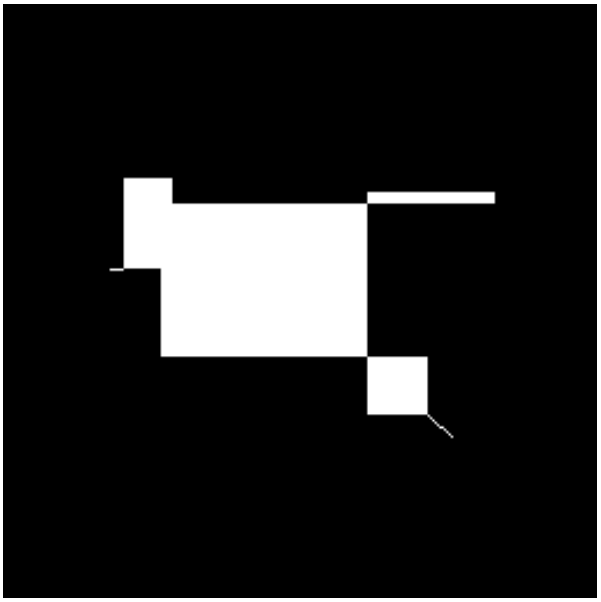
bug bogárkövető algoritmussal:



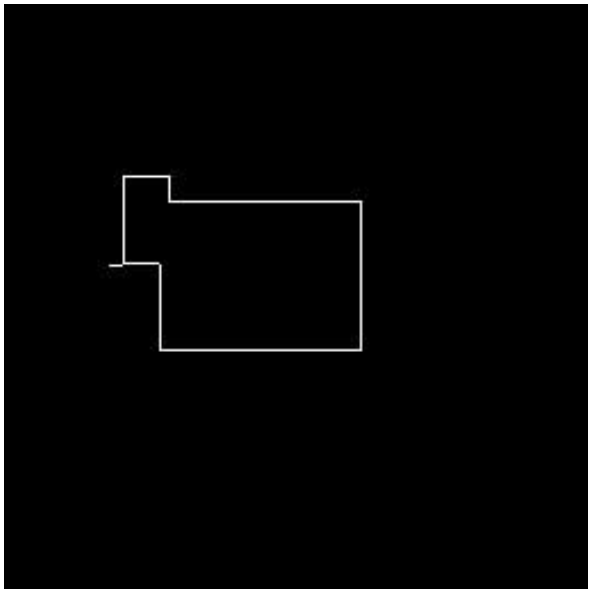
bug visszalépéses bogárkövető algoritmussal:



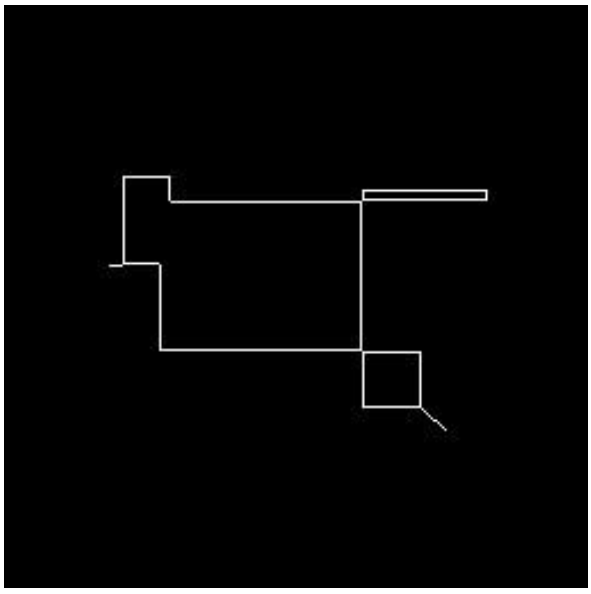
bug7:



bug7 bogárkövető algoritmussal:



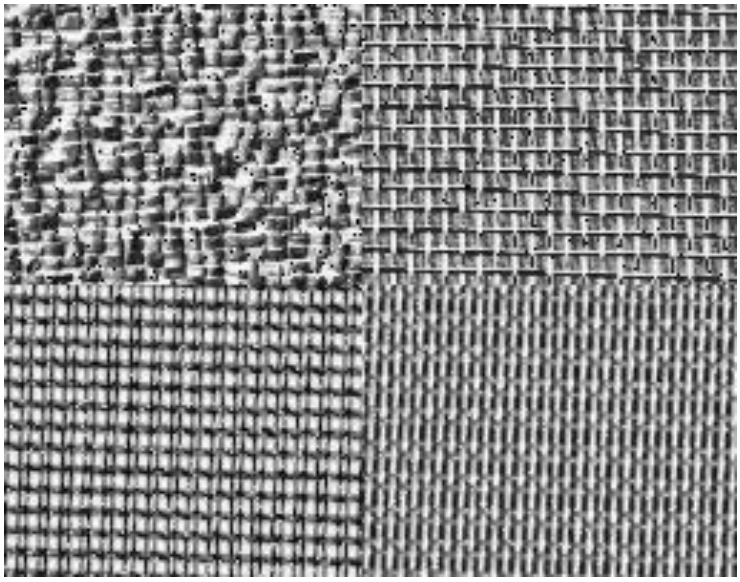
bug7 visszalépéses bogárkövető algoritmussal:



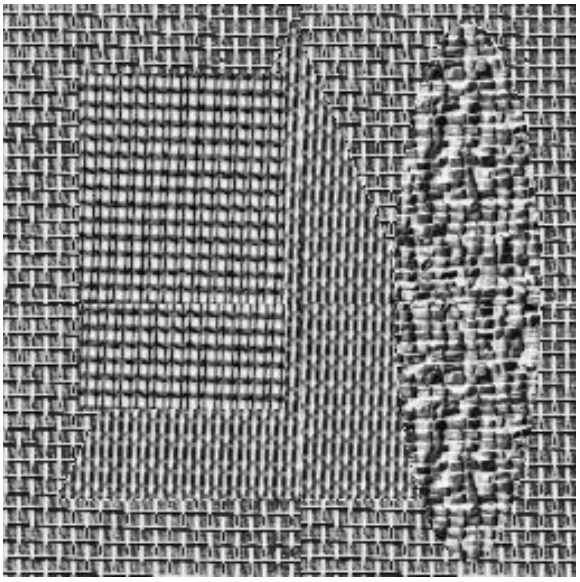
Law szűrő

Az algoritmusok a main.cpp fájlban találhatóak,403 – 565 sor, a függvények meghívása a 79-80 sorokban történnek. A feladatban a laws_input, laws_texture és laws_input5, laws_texture5 képeket használtam. Utóbbi azért, mert a neptun kódomban az első számjegy 8.

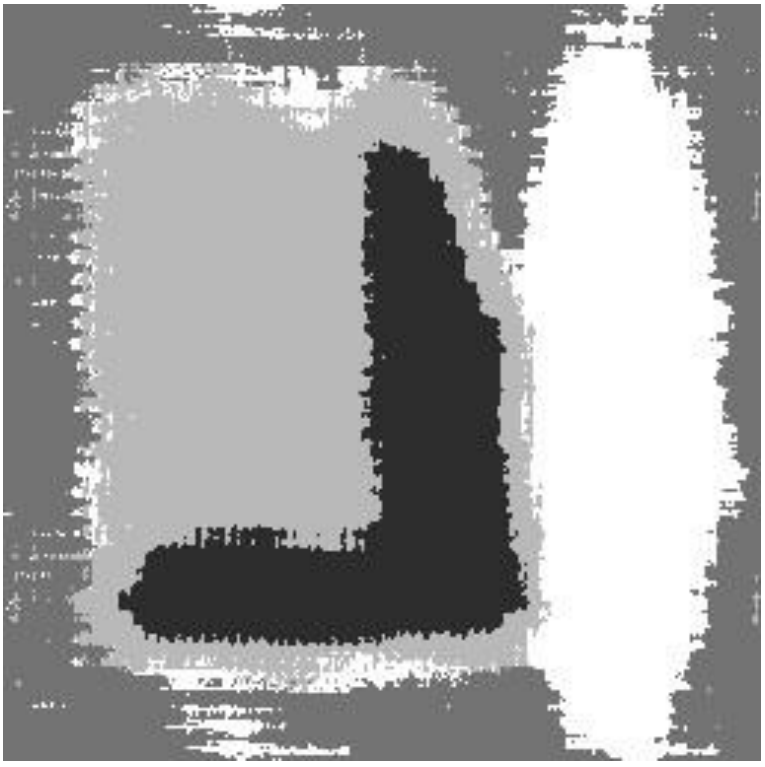
laws_input:



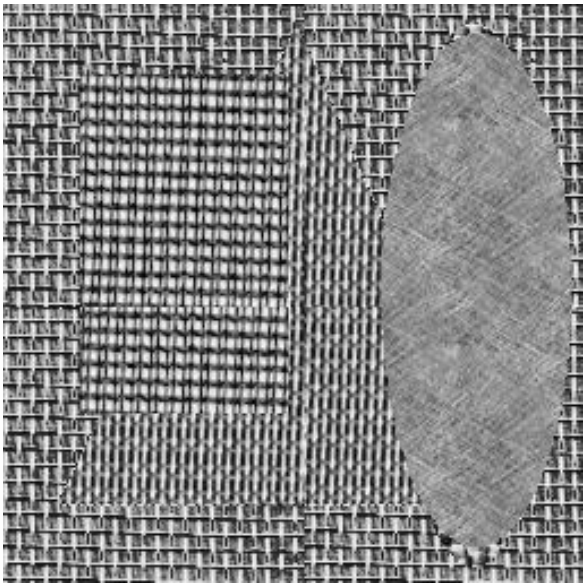
laws_texture:



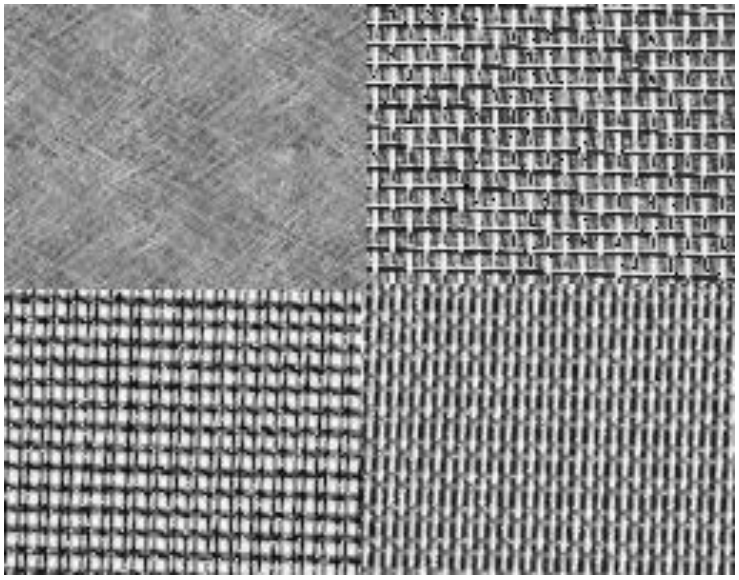
laws_input textúra szűrés után:



laws_input5:



laws_texture5:



laws_input5 textúra szűrés után: