ანიზოტროპული დიფუზიის მოდელების შედარება

ჯგუფი 3

ვილიამ სააკიანი

ნოდარ გიკაშვილი

ლადო ბარნოვი

ბაქარ ანდღულაძე

ხელმძღვანელები:

პროფ. რამაზ ბოჭორიშვილი

ასისტ.პროფ. თინათინ დავითაშვილი

სარჩევი

შესავალი

ჩვენს ნაშრომში განვიხილავთ ციფრული სურათების ფილტრაციის ერთ-ერთ მეთოდს. პირველ რიგში უნდა ვიცოდეთ თუ როგორ ხდება ციფრული სურათების წარმოდგენა, კერძოდ შავ-თეთრის, ფერადის, ორ და სამგანზომილებიანის. ამ ყველაფერს ჩვენს ნაშრომში დაწვრილებით განვიხილავთ.

სურათები ყოველთვის იდეალური არ გამოდის. მათ შექმნაზე შეიძლება გარე ფაქტორებმა იმოქმედონ და შედეგად გამოსახულება დამახინჯებული გამოვიდეს. გარე ფაქტორებში ელექტრონული ხმაური იგულისხმება. ხმაური არის სხვადასხვა სიხშირის არასასურველი ელექტრომაგნიტური სიგნალები, რომლისგანაც თავის დაღწევა და მოშორება ძალიან მნიშვნელოვანი საკითხია. ხმაური სურათებში გარკვეულ ცვლილებებს იწვევს და ქმნის არასასურველ შეუსაბამ წერტილებს.

ჩვენს მიერ განხილულ მეთოდს ანიზოტროპიული დიფუზია ეწოდება რომელიც ხმარებაში 90-იანი წლების დასაწყისში შემოვიდა. მისი შემქმნელები არიან პიტრო პერონა და ჯიტენდრა მალიკი და ამ მეთოდს ხშირად მათი სახელითაც მოიხსენიებენ.

ანიზოტროპიული დიფუზიის უპირატესობა იმაში მდგომარეობს რომ სხვა მეთოდებისგან განსხვავებით შეუძლია სიმკვეთრე შეუნარჩუნოს გამოსახულებაში მოცემული ობიექტის კონტურებს და წიბოებს, თუმცა მეთოდი იდეალური არ არის და მაინც იწვევს სურათის გაბუნდოვნებას და პროცესის მრავალად ჩატარების შემდეგ შეიძლება გამოსახულება მთლიანად გაურჩევადი გახდეს. ამ ნაშრომში განვიხილავთ ანიზოტროპიული დიფუზიის მოქმედების პრინციპს, დიფერენციალურ განტოლებას, მის დისკრეტულ სახეს და წარმოვადგენთ ჩვენს მიერ შექმნილი პროგრამით მიღებულ შედეგებს.

პირველ თავში ჩვენ გავეცნობით იმ სურათის სახეებს რომლის დამუშავებაცაა შესაძლებელი. ესენია შავ-თეთრი სურათები და ფერადი (RGB) სურათები. ასევე აღვწერთ მეთოდს, თუ როგორ შეიძლება გადავიყვანოთ ფერადი სურათი შავ-თეთრ სურათში და დავამუშავოთ იგი პროგრამულად.

მეორე თავში განვიხილავთ იზოტროპულ დიფუზიას, დავსვამთ ამოცანას და რიცხვითი მეთოდებით შევისწავლით მას. ასევე გიჩვენებთ როგორი გამოყენება აქვს დიფუზიას ციფრულ ფოტოსურათებში და ჩავატარებთ რიცხვით ექსპერიმენტებს.

მესამე თავში გადავალთ უშუალოდ ჩვენს თემაზე - ანიზოტროპული დიფუზია. მოვიყვანთ ამ მოდელს და შევისწავლით მას მათემატიკურად.

თავი 1. ციფრული სურათები

რა არის პიქსელი?

ციფრულ ფოტო-გამოსახულებებში პიქსელი არის უმცირესი მართვადი ელემენტი ციფრულ სურათში. პიქსელის კოორდინატები შეესაბამება მის ფიზიკურ კოორდინატებს.

გრაფიკულ მონიტორებზე გამოსახულება მიიღება ათასობით და მილიონობით პიქსელისაგან, რომლებიც რიგებად და სვეტებადაა განლაგებული და ქმნიან ერთ მთლიან გამოსახულებას.

რაც უფრო დიდია სურათში პიქსელების რაოდენობა, მით უფრო ზუსტი და სუფთაა გამოსახულება. პიქსელთა ინტენსივობა განსხვავებულია. ფერად სურათებში ფერი ძირითადად წარმოდგენილია 3 ან 4 კომპონენტის ინტენსივობით. ესენია: წითელი, მწვანე და ლურჯი ან ცისფერი, მეწამული, ყვითელი და შავი.

ბიტების რაოდენობა პიქსელში განსაზღვრავს ფერების მაქსიმალურ რაოდენობას, რომლებიც გვხვდება პიქსელში. მაგალითად, 8-ბიტიან პიქსელში შესაძლებელია 2^8 რაოდენობის ფერი იყოს მოთავსებული.

ფერად მონიტორებზე, თითოეული პიქსელი 3 წერტილითაა წარმოდგენილი - წითელი, ლურჯი და მწვანე. იდეალური გამოსახულებისთვის აუცილებელია სამივე ფერმა ერთ წერტილში მოიყაროს თავი, თუმცა ყველა მონიტორს აქვს ე.წ. კონვერგენციის ხარვეზი, რის გამოც ფერადი პიქსელები ბუნდოვანი ხდება ხოლმე.

სისტემის მიერ გამოსახულების ჩვენების ხარისხი დამოკიდებულია სისტემის გაფართოებაზე, ანუ იმაზე, თუ რამდენი პიქსელის ჩვენება შეუძლია და რამდენი ბიტითაა თითოეული პიქსელი წარმოდგენილი. VGA სისტემებს შეუძლიათ 640x480, ანუ დაახლოებით 300 000 პიქსელის ჩვენება. ხოლო SVGA სისტემებს 800x600 ანუ დაახლოებით 480 000 პიქსელის ჩვენება. ზუსტი ნატურალური გამოსახულების საჩვენებლად არსებობს სისტემები (ე. წ. **TRUE COLOR SYSTEMS**) , რომლებიც იყენებენ 24-ბიტიან პიქსელებს, რაც მათ 16 მილიონამდე ფერის წარმოდგენის საშუალებას აძლევს.

ორობითი გამოსახულება (ერთ-ბიტიანი პიქსელები)

ორობითი სურათი ეწოდება ციფრულ გამოსახულებას, რომელსაც მხოლოდ 2 შესაძლო მნიშვნელობა აქვს თითოეულ პიქსელში. ძირითადად ორი ფერი, რომლითაც ორობითი სურათია წარმოდგენილი არის შავი და თეთრი, თუმცა შესაძლებელია სხვა ფერების გამოყენებაც. სურათზე ობიექტების წარმოსადგენად გამოიყენება ერთი ფერი, რომელსაც წინა პლანის ფერი ეწოდება, ხოლო მეორეს, რომელიც გამოიყენება ფონის შესაქმნელად, ეწოდება ფონის ფერი. ორობითი სურათი ინახება **bitmap** ფორმატით. 640×480 გაფართოების ასეთი სურათი იკავებს 37.5 კბ მეხსიერებას.

Grayscale (8-ბიტიანი პიქსელები)

ციფრულ ფოტოგრაფიაში ნაცრისფერი ტონალობის სურათები ეწოდება გამოსახულებას, სადაც პიქსელი შეიცავს მხოლოდ ნაცრისფერი ფერის ინტენსივობის ინფორმაციას შეიცავს. ასეთი სურათებს ვუწოდებთ შავ-თეთრს. ასეთ გამოსახულებებს ვიღებთ ნაცრისფერი ფერის ინტენსივობის ცვალებადობით შავი ფერიდან ყველაზე ღია ფერამდე. greyscale სურათები ორობითი გამოსახულებებისგან საკმაოდ განსხვავდება. ორობით სურათებში მხოლოდ ორი ფერი გვაქვს მაშინ, როცა შავ-თეთრ სურათებში ნაცრისფერი ფერის უამრავი ტონალობის გამოყენება შეიძლება.

ფერადი სურათისათვის, პიქსელი განკარგავს ძირითადად 3 კომპონენტს RGB (წითელი, მწვანე, ცისფერი). ნაცრისფერ პიქსელს აქვს 3 იდენტური მნიშვნელობა RGB. ფერადი სურათის ნაცრისფერი ფერის დონეში გარდაქმნა არის მარტივი მეთოდი, სადაც უნდა მოხდეს ამ სამი კომპონენტის, წმც-ს მნიშვნელობის გამოყენება თითოეული კომპონენტისათვის.

ნაცრისფერი = 0.299 * წითელი + 0.587 * მწვანე + 0.114 * ცისფერი

ნაცრისფერი სურათის მაგალითი:

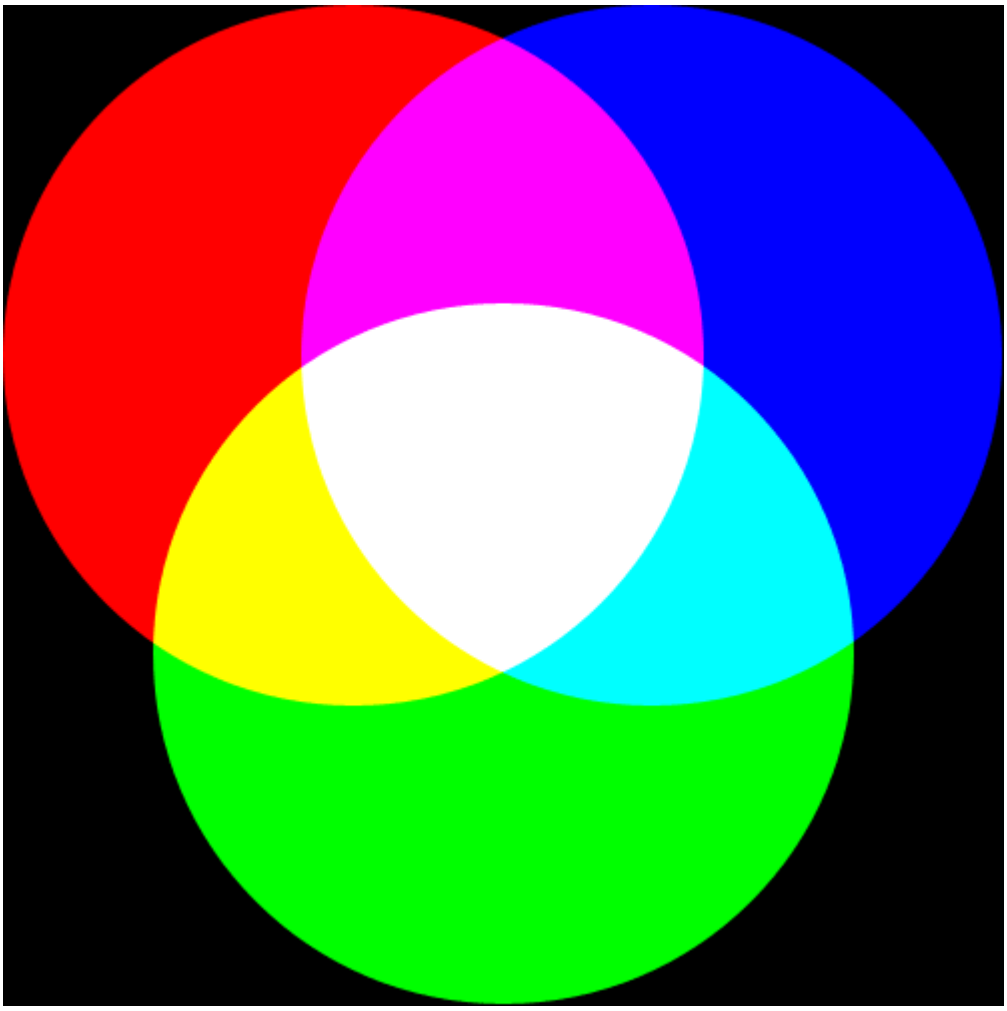
RGB მოდელი

RGB (Red, Green, Blue) ფერთა მოდელია, რომელშიც ყველა ფერი მიიღება ამ სამი ძირითადი ფერისგან: წითელი, მწვანე და ლურჯი. RGB ფერთა მოდელის ძირითადი დანიშნულებაა სენსიტიურობა და სურათის კარგად წარმოდგენა ელექტრონულ სისტემებში, როგორიცაა ტელევიზორი, კომპიუტერი, მობილური ტელეფონი და სხვა თანამედროვე ტექნიკა. ამ მოდელს მნიშვნელოვანი ადგილი უკავია ფოტოგრაფიაში. ელექტროობის ხანამდეც კი, RGB მოდელის უკან უკვე არსებობდა თეორია, რაც ძირითადად ადამიანის ფერთა აღქმას ეყრდნობოდა.

RGB ფერთა სისტემა, როგორც უკვე ვთქვით, 3 ძირითადი ფერისგან შედგება, რომელთაგან თითო ფერი იყენებს 8 ბიტს, მთლიანობაში მთელი რიცხვების დიაპაზონს 0-დან 255-მდე (unsigned int). შესაბამისად მთლიანად გვექნება 256*256*256=16777216 შესაძლებელი ფერი.

LCD მონიტორის ეკრანის ყოველი პიქსელიც ამგვარადაა წარმოდგენილი: წითელი, მწვანე და ლურჯი ფერების LED (light emitting diodes)- **“შუქდიოდური”** კომბინაცია.

შუქდიოდური ინტეგრირებული წრედი (LED die) ასხივებს მონოქრომატულ სინათლეს, რომელიც ამ 3 ფერთაგან ერთ-ერთია. ამ კომპონენტის მეშვეობით 3 ძირითადი ფერისგან მარტივი კომბინაციით მიიღება 3 შუალედური (მეწამული, ცისფერი, ყვითელი) და თეთრი ფერი, რომლებიც სიკაშკაშის ცვალებადობის მიხედვით წარმოქმნიან ყველა დანარჩენ ფერს.



როდესაც წითელი პიქსელის მნიშვნელობა 0-ია, მაშინ LED მთლიანად გამორთულია. ხოლო როცა ის 255-ია, მაშინ LED მთლიანად ჩართულია.

RGB კოდს აქვს 24-ბიტიანი ფორმატი, შესაბამისად 8-8 თითოეული ფერისთვის (255-ის ოროზითი ჩანაწერი მაქსიმალურად ავსებს 8 პოზიციას). RGB=(R*65536)+(G*256)+B. კონკრეტული ფერების RGB კოდი კი ასეთია:

თეთრი RGB კოდი = 256*65536+255*256+255 = #FFFFFF

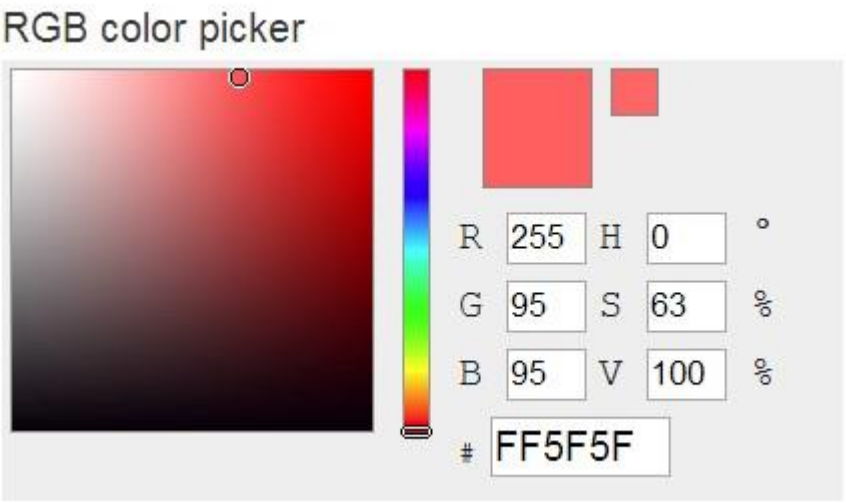
ლურჯი RGB კოდი = 0*65536+0*256+255 = #0000FF

წითელი RGB კოდი = 255*65536+0*256+0 = #FF0000

მწვანე RGB კოდი = 0*65536+255*256+0 = #00FF00

ნაცრისფერი RGB კოდი = 128*65536+128*256+128 = #808080

ყვითელი RGB კოდი = 255*65536+255*256+0 = #FFFF00w

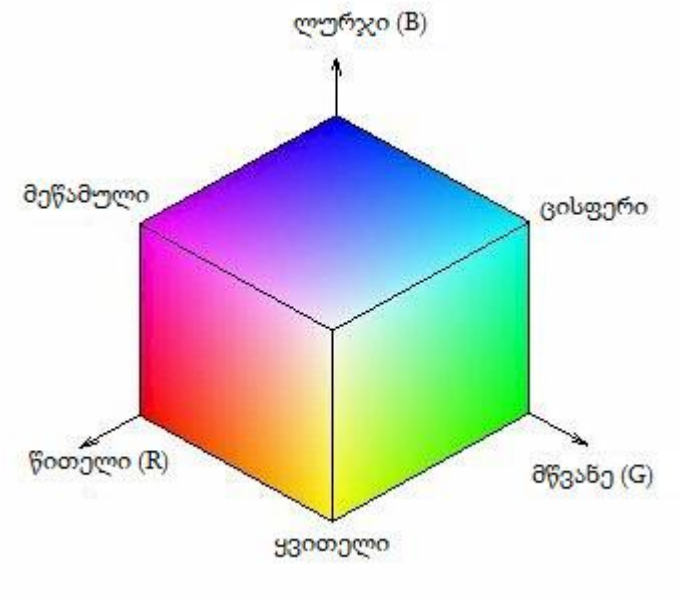


ძირითადი RGB ფერები, მათი შესაბამისი კოდებით:

Color	HTML/CSS Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
	Black	#000000	(0,0,0)
	White	#FFFFFF	(255,255,255)
	Red	#FF0000	(255,0,0)
	Lime	#00FF00	(0,255,0)
	Blue	#0000FF	(0,0,255)
	Yellow	#FFFF00	(255,255,0)
	Cyan / Aqua	#00FFFF	(0,255,255)
	Magenta / Fuchsia	#FF00FF	(255,0,255)
	Silver	#C0C0C0	(192,192,192)
	Gray	#808080	(128,128,128)
	Maroon	#800000	(128,0,0)
	Olive	#808000	(128,128,0)
	Green	#008000	(0,128,0)
	Purple	#800080	(128,0,128)
	Teal	#008080	(0,128,128)
	Navy	#000080	(0,0,128)

გეომეტრიული წარმოდგენა:

რადგანაც ყველა ფერი 3 ძირითადი ფერისგან მივიღეთ, RGB მოდელს ხშირად გამოსახავენ 3-განზომილებიან ევკლიდეს სივრცეში. ამ მოდელისთვის წარმოდგენილია კუბი, რომელიც თითოეულ ღერძზე იღებს მნიშვნელობებს [0;1] სეგმენტზე, მისი ერთი წვერო საკოორდინატო ღერძების გადაკვეთის წერტილს ემთხვევა, ხოლო მისი მოპირდაპირე წვეროს კოორდინატებია (1;1;1).



ამ მოდელზე ღერძებად გამოყენებულია 3 ძირითადი ფერის შესაბამისი ღერძები, სადაც სათავე (0; 0; 0) შეესაბამება შავ ფერს, თითოეულ ღერძზე მდებარე წერტილებისთვის კი შესაბამისი ფერები იცვლება შავიდან წითლამდე (ლურჯამდე, მწვანემდე). წერტილი კოორდინატებით (1;1;1) თეთრ ფერს წარმოადგენს, ხოლო კუბის მთავარ დიაგონალზე განლაგებულია მხოლოდ ნაცრისფერი ფერები, შავიდან თეთრამდე.

პროგრამული უზრუნველყოფა. ციფრული სურათის ტექსტურ სურათში გადაყვანა და პირიქით

ციფრული სურათის დამუშავებისთვის სურათი გადაიყვანება BMP ფორმატში. **Windows Bitmap (BMP)** არის ციფრული გამოსახულების სახე Microsoft-ის და IBM-ის მიერ განვითარებული.ეს არის გამოსახულების ერთ-ერთი ყველაზე მარტივი სახე განვითარებისთვის და პროგრამების გამოყენებისთვის. ის არის წაკითხვადი თითქმის ყველა სურათის შემდგენელი მოწყობილობისთვის. ის ეხება მატრიცულ გამოსახულებებს.

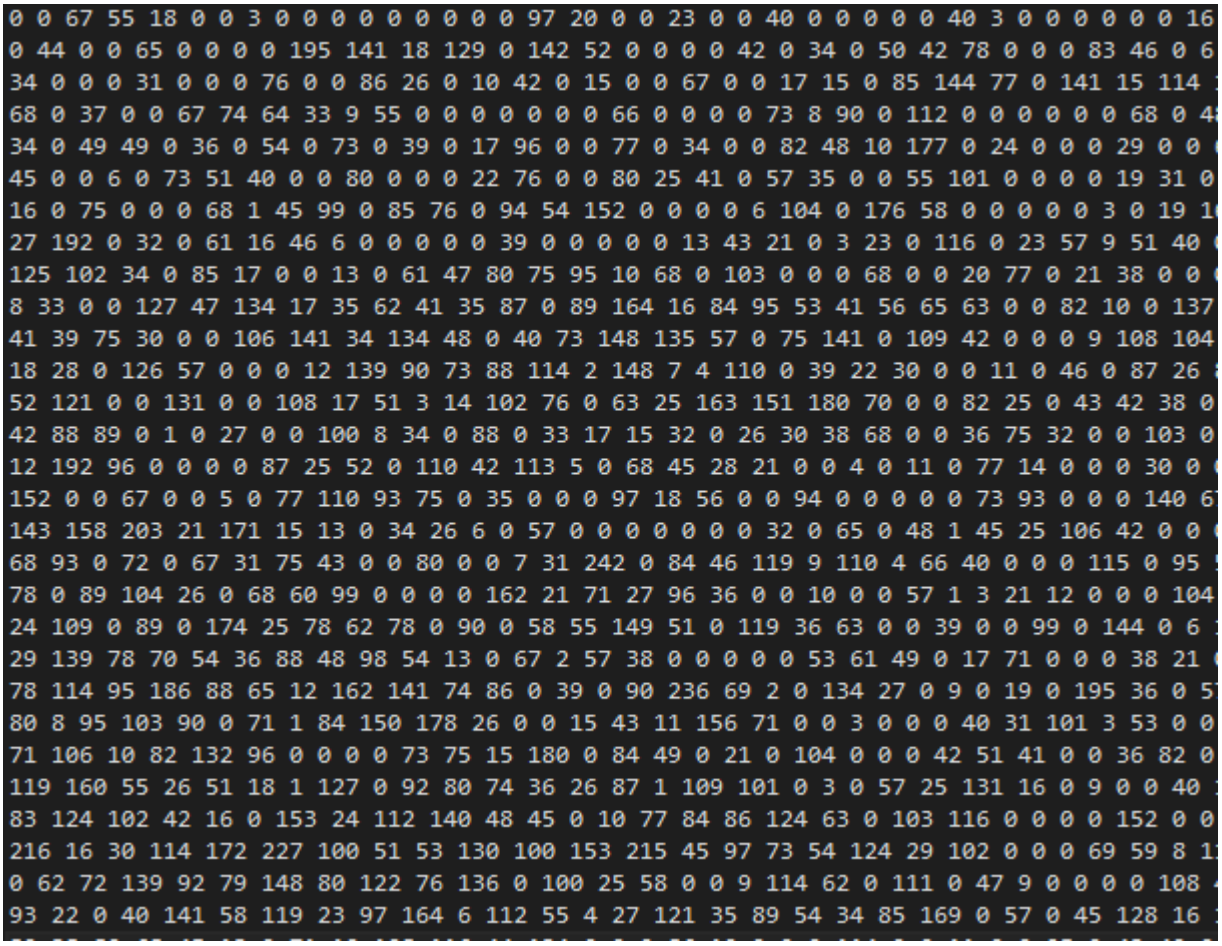
Bitmap კლასის დახმარებით, ჩვენ გვაქვს წვდომა სურათის ყველა პიქსელზე და ასევე ამ პიქსელში მოთავსებულ მნიშვნელობებზე. ალგორითმის გამოყენებით, ჩვენ ვცვლით ამ პიქსელებში მოთავსებულ მნიშვნელობებს. ამისთვის ვქმნით ორგანზომილებიან მატრიცას (ორგანზომილებიანი სურათისთვის), რომელშიც ჩაწერილია რიცხვები, რომელიც წარმოადგენს ნაცრისფერის შკალას. დასაწყისში ჩვენ განვიხილავთ ალგორითმს შავ-თეთრ სურათებზე, ამიტომ, თუ სურათი ფერადია, შემდეგი ფორმულის მეშვეობით სამი მნიშვნელობის ნაცვლად თითო პიქსელში, ჩვენ მივიღებთ ერთ მნიშვნელობას:

$$Y' = 0.299R + 0.587G + 0.114B$$

პროგრამულად ეს ჩაიწერება ასე:

```
for (int y = 0; y < bmp.Height; y++)
{
    for (int x = 0; x < bmp.Width; x++)
    {
        color[x, y] = bmp.GetPixel(x, y);
        I[x, y] = color[x, y].R * 0.299 + color[x, y].G * 0.587 + color[x, y].B * 0.114;
    }
}
```

ამის შემდეგ ჩვენ გვექნება შემდეგი სახის მატრიცა, რომელშიც ელემენტების დიაპაზონია [0, 255], სადაც 0 - შავი ფერია, 255 - თეთრი, დანარჩენი კი მათ შორის შესაბამისად ნაცრისფერი ფერის სიმკვეთრის მიხედვით იცვლება:



საბოლოო სურათის შესაბამისი მატრიცა (ჩვენი პროგრამის გამოყენების შედეგად მიღებული):

ამ მატრიცას თუ ისევ გადავიყვანთ სურათში, ჩვენ მივიღებთ შავ-თეთრ სურათს.

```
for (int i = 0; i < height; i++)
{
    for (int j = 0; j < width; j++)
    {
        bmp2.SetPixel(j, i, color1[j, i]);
    }
}
```

თავი 2: იზოტროპული დიფუზია

ამოცანის დასმა

ამ თავში ჩვენ განვიხილავთ „გაფუჭებული“ სურათის დამუშავების სქემას იზოტროპული დიფუზიის გამოყენებით. „გაფუჭებული“ სურათი თავის თავში შეიძლება მოიაზრებდეს ხმაურიან სურათს, რომელზეც არ ჩანს გამოსახულება კარგად, დაკარგულია საკვანძო ადგილები, პრობლემურია ამ სურათის გამოყენება სხვადასხვა მანიპულაციისთვის, შეიცავს ხელოვნურ სახეცვლილებებს (ნაკაწრი ფოტოზე და ა.შ). ჩვენი მიზანია მოვაშოროთ სურათს არასასურველი ხმაური და ამისათვის გამოვიყენებთ ყველაზე მარტივ და კარგად ნაცად იზოტროპული დიფუზიის მეთოდს რომელიც იყენებს გაუსის “გათანაბრების“ მეთოდს.

იზოტროპული დიფუზიის მთავარი "დევიზია" - დიფუზიის ჩატარება თანაბრად, ყველა მიმართულებით, ეს აჩენს რიგ პრობლემებს, რაც ძირითადად სურათის დაბურვაში გამოიხატება, რასაც მომდევნო თემაში განვიხილავთ. მიუხედავად ამ დიდი ნაკლისა, იზოტროპული დიფუზია მაინც რჩება დიფუზიის კლასიკურ მოდელად და ანიზოტროპული დიფუზიის განხილვა ამ თემის გვერდის ავლით შეუძლებელია

გაუსის გაგლუვება

სურათის დამუშავებაში გაგლუვება განიმარტება, როგორც დაახლოებით ისეთი ფუნქციის აგება, რომელიც სურათის მნიშვნელოვან ნაწილს მოუყრის თავს, ხოლო ხმაურს ან სხვა სტრუქტურულ ნაწილებს ცალკე დატოვებს. გაგლუვების დროს ცალკეული წერტილები (ჩვენი დაშვებით სწორედ ესენია ხმაური) შემცირებულია, ხოლო ის წერტილები, რომლებიც უფრო დაბალ მნიშვნელობას ატარებენ (მაგალითად ნაცრისფერ შკალაზე), ვიდრე მეზობელი წერტილები, გაგლუვების მეთოდით მნიშვნელობით იზრდებიან.

დასამუშავებლად შემოტანილი სურათი, როგორც უკვე განვიხილეთ რიცხობრივად შეიძლება წარმოვადგინოთ. სურათის პიქსელების მნიშვნელობების ნაცრისფერ შკალაზე წარმოდგენით, რაიმე შემოტანილი f სურათისთვის მისი გათანაბრება გამოითვლება კონვოლუციით, იგივე ხვეულით (ანუ 2 ფუნქციიდან ისეთი მე-3 ფუნქციის აგება, რომელიც გამოხატავს ერთ-ერთი საწყისი ფუნქციის მოდიფიცირებულ ვარიანტს და ნაწილობრივ ემთხვევა მას)

$$(K_\sigma * f)(p) := \int_{R^2} K_\sigma(p-q) f(q) dq$$

სადაც K_σ ორგანზომილებიანი, იზოტროპული (წრიულად სიმეტრიული) გაუსიანია, არგუმენტად შესაბამისი პიქსელის მნიშვნელობა, $\sigma>0$ სტანდარტული გადახრით. $x \in R^2$ კი პირობითად გარკვეული წერტილი უნდა იყოს სურათზე, რაც სინამდვილეში თავის თავში ორ კოორდინატს გულისხმობს (ორგანზომილებიანი სურათის შემთხვევა). გაუსიანი ასე გამოითვლება:

$$K_\sigma(p) := \frac{1}{2\pi\sigma^2} \cdot exp\left(-\frac{|p|^2}{2\sigma^2}\right)$$

და ეს პროცესი ტარდება სურათის ყოველი პიქსელისთვის. კერძოდ კი კონტურები ფორმულის მიერ აღქმულია როგორც კონცენტრირებული წრეწირების ცენტრები და მათზე ჩატარებულია გაუსის გაგლუვება ანუ ამ წრეწირების ცენტრებიდან ყველა მიმართულებით ხდება გაგლუვება. გაუსის ფუნქცია

ყველაზე დიდ მნიშვნელობას ცენტრებს ანიჭებს, მისგან დაშორებული წერტილები კი უფრო და უფრო უმნიშვნელოდ იცვლება.

აღწერილი ორი ფორმულის განხილვა სრულიად საკმარისია იზოტროპული დიფუზიის (წრფივი დიფუზიური ფილტრაციის) შესასწავლად.

იზოტროპული დიფუზია სითბოგამტარობის განტოლებით

იზოტროპულ დიფუზიას კლასიკურად სითბოს განტოლებით აღწერენ:

$$\partial_t u = \Delta u,$$

$$u(x, y, 0) = f(p), \qquad p = (x, y), \qquad f \in C(R^2)$$

სადაც $\frac{\partial u}{\partial t} = \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$

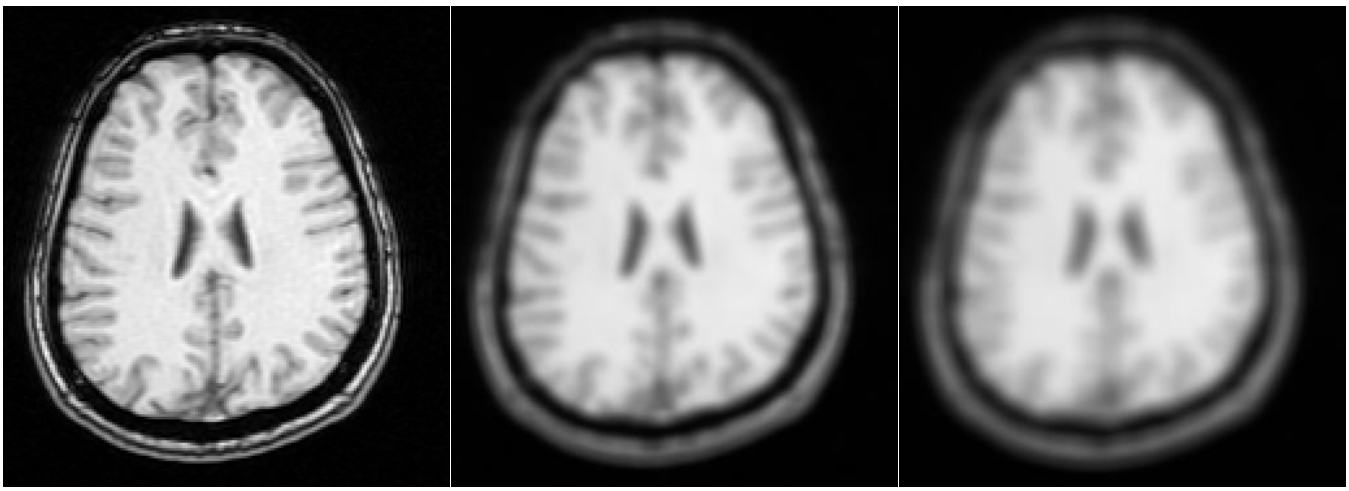
მისი ამონახსნი კი ერთ განზომილებაში გამოთვლების შედეგად მიღებულია ასეთი სახით:

$$u(x, y, t) = \begin{cases} f(p) & (t = 0) \\ (K_{\sqrt{2t}} * f)(p) & (t > 0) \end{cases}$$

$$\left(K_{\sqrt{2t}} * f\right)(p) = \int_{R^2} (K_{\sqrt{2t}}(p - q) f(q) dq = \int_{R^2} \frac{1}{2\pi \cdot 2t} exp\left(-\frac{|p - q|^2}{2 \cdot 2t}\right) \leq$$

$$\leq M \cdot \exp(a|p|^2),\ M, a > 0;$$

საზოგადოდ მივიღებთ, რომ $|u(x, y, t)| \leq M * exp(a|p|2), (M, a > 0)$. რაც იმას ნიშნავს, რომ სურათის აღმწერი u ფუნქცია შემოსაზღვრულია დროში. რადგანაც სტანდარტული გადახრა დროის მიხედვით ამ სახით გამოვსახეთ: $\sigma=\sqrt{2t}$, ამიტომ დიფუზიური პროცესის შეჩერება აუცილებელია $T=\frac{1}{2}\sigma^2$ დროში.



ორიგინალი σ=20 σ=40

დირიხლეს ამოცანისთვის ცხადი სხვაობიანი სქემის აპროქსიმაციის ცდომილება და მისი შეფასება

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,t), & a \le x \le b,\ c \le y \le d,\ 0 \le t \le T \\ u(0,x,y) = u_0(x,y), & a \le x \le b,\ c \le y \le d \\ u(t,x,y)|_\Gamma = u_\Gamma, & 0 \le t \le T \end{cases}$$

$$\begin{cases} \frac{y_{i,j}^{n+1} - y_{i,j}^n}{\tau} = \frac{y_{i+1,j}^n - 2y_{i,j}^n + y_{i-1,j}^n}{h_1^2} + \frac{y_{i,j+1}^n - 2y_{i,j}^n + y_{i,j-1}^n}{h_2^2} + f_{i,j}^n \quad (1) \\ y_{i,j}^0 = u_0(x_i, y_j), \quad i = 0,1,\dots,n_x;\ \ j = 0,1,\dots,n_y; \\ y_\Gamma^t = u_{\Gamma,}\ \ 0 \le t \le T \end{cases}$$

დირიხლეს ამოცანის ზუსტი ამონახსნი (x_i, y_j, t_n) წერტილში აღვნიშნოთ $u(x_i, y_j, t_n)$-ით. რადგან მიახლოებითი ამონახსნი ამავე წერტილში $y_{i,j}^n$-ით აღვნიშნეთ, ცდომილება ზუსტ და მიახლოებით ამონახსნს შორის იქნება:

$$z_{i,j}^n = y_{i,j}^n - u(x_i, y_j, t_n) \Rightarrow y_{i,j}^n = z_{i,j}^n + u(x_i, y_j, t_n)$$

მიღებული გამოსახულება ჩავსვათ (1)-ში:

$$\frac{y_{i,j}^{n+1} - y_{i,j}^n}{\tau} = \frac{y_{i+1,j}^n - 2y_{i,j}^n - y_{i-1,j}^n}{h_1^2} + \frac{y_{i,j+1}^n - 2y_{i,j}^n - y_{i,j-1}^n}{h_2^2} + f_{i,j}^n$$

$$\frac{z_{i,j}^{n+1} + u(x_i, y_j, t_{n+1}) - z_{i,j}^n - u(x_i, y_j, t_n)}{\tau} =$$

$$= \frac{z_{i+1,j}^n + u(x_{i+1}, y_j, t_n) - 2z_{i,j}^n - 2u(x_i, y_j, t_n) + z_{i-1,j}^n + u(x_{i-1}, y_j, t_n)}{h_1^2} +$$

$$\frac{z_{i,j+1}^n + u(x_i, y_{j+1}, t_n) - 2z_{i,j}^n - 2u(x_i, y_j, t_n) + z_{i,j-1}^n + u(x_i, y_{j-1}, t_n)}{h_2^2} + f_{i,j}^n$$

$$\frac{z_{i,j}^{n+1} - z_{i,j}^n}{\tau} = \frac{z_{i+1,j}^n - 2z_{i,j}^n + z_{i-1,j}^n}{h_1^2} + \frac{z_{i,j+1}^n - 2z_{i,j}^n + z_{i,j-1}^n}{h_2^2} +$$

$$+ \frac{u(x_{i+1}, y_j, t_n) - 2u(x_i, y_j, t_n) + u(x_{i-1}, y_j, t_n)}{h_1^2}$$

$$+ \frac{u(x_i, y_{j+1}, t_n) - 2u(x_i, y_j, t_n) + u(x_i, y_{j-1}, t_n)}{h_2^2} - \frac{u(x_i, y_j, t_{n+1}) - u(x_i, y_j, t_n)}{\tau} + f_{i,j}^n$$

აპროქსიმაციის ცდომილება (x, y, t) ამონახსნზე:

$$\Psi_{i,j}^{n}(u, h_1, h_2, \tau) = -\frac{u(x_i, y_j, t_{n+1}) - u(x_i, y_j, t_n)}{\tau} + \frac{u(x_{i+1}, y_j, t_n) - 2u(x_i, y_j, t_n) + u(x_{i-1}, y_j, t_n)}{h_1^2} +$$

$$+ \frac{u(x_i,y_{j+1},t_n)-2u(x_i,y_j,t_n)+u(x_i,y_{j-1},t_n)}{h_2^2};$$

$u(x_i, y_j, t_{n+1})$ გავშალოთ ტეილორის მწკრივად II რიგამდე, მაშინ:

$$\frac{u(x_i, y_j, t_{n+1}) - u(x_i, y_j, t_n)}{\tau} = \frac{u(x_i, y_j, t_n) + \frac{\tau}{1!}\frac{\partial u}{\partial t}(x_i, y_j, t_n) + O(\tau^2) - u(x_i, y_j, t_n)}{\tau} = \frac{\partial u}{\partial t}(x_i, y_j, t_n) + O(\tau)$$

$u(x_{i+1}, y_j, t_k), u(x_{i-1}, y_j, t_k), u(x_{i+1}, y_{j+1}, t_k)$, და $u(x_{i+1}, y_{j-1}, t_k)$, გავშალოთ IV რიგამდე:

$$\frac{u(x_{i+1}, y_j, t_n) - 2u(x_i, y_j, t_n) + u(x_{i-1}, y_j, t_n)}{h_1^2} =$$

$$= \frac{u(x_i, y_j, t_n) + \frac{h_1}{1!}\frac{\partial u}{\partial x}(x_i, y_j, t_n) + \frac{h_1^2}{2!}\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t_n) + \frac{h_1^3}{3!}\frac{\partial^3 u}{\partial x^3}(x_i, y_j, t_n) + O(h_1^4)}{h_1^2} - \frac{2u(x_i, y_j, t_n)}{h_1^2} +$$

$$+ \frac{u(x_i, y_j, t_n) - \frac{h_1}{1!}\frac{\partial u}{\partial x}(x_i, y_j, t_n) + \frac{h_1^2}{2!}\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t_n) - \frac{h_1^3}{3!}\frac{\partial^3 u}{\partial x^3}(x_i, y_j, t_n) + O(h_1^4)}{h_1^2} =$$

$$= \frac{\partial^2 u}{\partial x^2}(x_i, y_j, t_n) + O(h_1^2)$$

ანალოგიურად:

$$\frac{u(x_i, y_{j+1}, t_n) - 2u(x_i, y_j, t_n) + u(x_i, y_{j-1}, t_n)}{h_2^2} = \frac{\partial^2 u}{\partial y^2}(x_i, y_j, t_n) + O(h_2^2);$$

$$\Psi_{i,j}^{n}(u, h_1, h_2, \tau) = -\frac{\partial u}{\partial t}(x_i, y_j, t_n) + \frac{\partial^2 u}{\partial x^2}(x_i, y_j, t_n) + \frac{\partial^2 u}{\partial y^2}(x_i, y_j, t_n) + O(\tau) + O(h_1^2) + O(h_2^2) =$$

$$O(\tau + h_1^2 + h_2^2)$$

ცხადი სხვაობიანი სქემის მდგრადობის გამოკვლევის ჰარმონიკების მეთოდი

$$\frac{y_{m,k}^{n+1}-y_{m,k}^{n}}{\tau}=\frac{y_{m,k}^{n}-2y_{m,k}^{n}+y_{m,k}^{n}}{h_1^2}+\frac{y_{m,k}^{n}-2y_{m,k}^{n}+y_{m,k}^{n}}{h_2^2}$$

სქემის მდგრადობა გამოვიკვლიოთ ჰარმონიკების მეთოდის საშუალებით;

შემოვიღოთ აღნიშვნა: $y_{m,k}^{n}=\rho^n e^{i\beta mh}e^{i\gamma kh}, \qquad i^2=-1; \quad \rho,\beta,\gamma\in R$

$$\frac{\rho^{n+1}e^{i\beta mh}e^{i\gamma kh}-\rho^n e^{i\beta mh}e^{i\gamma kh}}{\tau}=$$

$$=\frac{\rho^n e^{i\beta(m+1)h}e^{i\gamma kh}-2\rho^n e^{i\beta mh}e^{i\gamma kh}+\rho^n e^{i\beta(m-1)h}e^{i\gamma kh}}{h_1^2}+$$

$$+\frac{\rho^n e^{i\beta mh}e^{i\gamma(k+1)h}-2\rho^n e^{i\beta mh}e^{i\gamma kh}+\rho^n e^{i\beta mh}e^{i\gamma(k-1)h}}{h_2^2}$$

განტოლება გავყოთ $\rho^n e^{i\beta mh}e^{i\gamma kh}$-ზე, მივიღებთ:

$$\frac{\rho-1}{\tau}=\frac{e^{i\beta h}-2+e^{-i\beta h}}{h_1^2}+\frac{e^{i\gamma h}-2+e^{-i\gamma h}}{h_2^2}$$

სადაც

$$\begin{cases} e^{i\beta h}=\cos(\beta h)+i\sin(\beta h) \\ e^{-i\beta h}=\cos(\beta h)-i\sin(\beta h) \end{cases} \Rightarrow e^{i\beta h}+e^{-i\beta h}=2\cos(\beta h)$$

ანალოგიურად β-ს მაგიერ γ-ს ჩასმით მივიღებთ, რომ

$$e^{i\gamma h}+e^{-i\gamma h}=2\cos(\gamma h)$$

მივიღებთ:

$$\frac{\rho-1}{\tau}=\frac{2(\cos(\beta h)-1)}{h_1^2}+\frac{2\,(\cos(\gamma h)-1)}{h_2^2}=\frac{2\left(-2sin^2\left(\frac{\beta h}{2}\right)\right)}{h_1^2}+\frac{2\left(-2sin^2\left(\frac{\gamma h}{2}\right)\right)}{h_2^2}$$

$$\rho=1-4\tau\left(\frac{sin^2\left(\frac{\beta h}{2}\right)}{h_1^2}+\frac{sin^2\left(\frac{\gamma h}{2}\right)}{h_2^2}\right)$$

$y_{m,k}^{0}=\rho^0 e^{i\beta mh}e^{i\gamma kh}=(\cos(\beta mh)+i\,sin(\beta mh))(\cos(\gamma kh)+i\sin(\gamma kh))$ შემოსაზღვრულია.

$|\rho|>1 \;\Rightarrow\; |y_{m,k}^{n}|=|\rho|^n\longrightarrow\infty \quad$ როცა $\quad n\longrightarrow\infty$

$|\rho|\le 1 \;\Rightarrow\; |y_{m,k}^{n}|=|\rho|^{\,2}\le 1$

$$\left|1-4\tau\left(\frac{sin^2\left(\frac{\beta h}{2}\right)}{h_1^2}+\frac{sin^2\left(\frac{\gamma h}{2}\right)}{h_2^2}\right)\right|\le 1 \quad\Rightarrow$$

$$\Rightarrow \qquad -1\le 1-4\tau\left(\frac{sin^2\left(\frac{\beta h}{2}\right)}{h_1^2}+\frac{sin^2\left(\frac{\gamma h}{2}\right)}{h_2^2}\right)\le 1 \;\Rightarrow$$

$$\Rightarrow 0 \leq 4\tau\left(\frac{sin^2\left(\frac{\beta h}{2}\right)}{h_1^2}+\frac{sin^2\left(\frac{\gamma h}{2}\right)}{h_2^2}\right)\leq 2 \Rightarrow \tau\left(\frac{1}{h_1^2}+\frac{1}{h_2^2}\right)\leq\frac{1}{2}$$

ე.ი. როცა $\tau\left(\frac{1}{h_1^2}+\frac{1}{h_2^2}\right)\leq\frac{1}{2}$

მაშინ $|\rho|\leq 1$ და სქემა პირობითად მდგრადია.

თავი 3. ანიზოტროპული დიფუზია

ამოცანის დასმა

იზოტროპული დიფუზიის ძირითადი ნაკლი ის იყო, რომ სურათიდან ზედმეტი ხმაურის მოცილებასთან ერთად, იკარგებოდა კონტურები (კიდეები) და სურათი იბურებოდა, რთული ხდებოდა მასზე არსებული გამოსახულების აღქმა. ასეთ შემთხვევებში სურათზე გამოსახული ადამიანის ამოცნობა რთულდება, ასევე რთულია მცირე დეტალების ამოცნობა და ამ სურათის გამოყენება შემდგომი მოდიფიკაციებისთვის თითქმის შეუძლებელია. ამ ნაკლის აღმოფხვრა შეძლეს პერონა-მალიკმა ანიზოტროპული დიფუზიის მათი მოდელით. პერონამ და მალიკმა შემოიღეს დიფუზიურობის ისეთი ფუნქცია g, რომ $g(0)=1$, $g(s)\geq 0$ და $\lim_{s\to\infty} g(s) = 0$, რაც იმას უზრუნველყოფს, რომ კონტურებში დიფუზია არ მიმდინარეობს (ანუ განსხვავება მეზობელ კვანძით წერტილებს შორის საკმარისად დიდია, ამიტომ უნდა მოხდეს კონტურების შენარჩუნება). მათი ალგორითმი იყენებს ფუნქციას g, რომელიც ახდენს კონტურების აღმოჩენას: ამ წერტილებში ყველაზე მცირეა მეზობელი კვანძითი წერტილების მნიშვნელობებს შორის განსხვავება, ანუ პატარაა გრადიენტი და დიფუზიური პროცესი ნაკლებად მიმდინარეობს. სამაგიეროდ მეტი დიფუზიური პროცესი ხმარდება კონტურებიდან უფრო და უფრო დაშორებულ წერტილებს. ეს უზრუნველყოფს კიდეების დაფიქსირებას, სურათი აღარ იბურება და უფრო ნათელი, წმინდა გამოსახულება მიიღება.

სხვადასხვა ცნობილი ფუნქციები ანიზოტროპული დიფუზიისთვის

პრაქტიკაში მრავალ დიფუზიურობის ფუნქციას იყენებენ, რომელთაგან უმეტესობა სულაც არ აკმაყოფილებს ზემოთ ნახსენებ შეზღუდვებს: $g(0)=1$, $g(|x|)\geq 0$ და $\lim_{|x|\to\infty} g(|x|) = 0$.

ამ ფორმულების ცვლილებით, იცვლება დიფუზიური პროცესი და შესაბამისად, მისი შედეგიც. აი რამდენიმე მათგანი:

პერონა მალიკის დიფუზიურობის ფუნქციები:

1) $g(|x|) = \frac{1}{1+\left(\frac{|x|}{\lambda}\right)^2}$

2) $g(|x|) = e^{-\left(\frac{|x|^2}{\lambda^2}\right)}$

წრფივი: $g(|x|) = 1$

კარბონიეს: $g(|x|) = \frac{1}{\sqrt{1+\frac{|x|^2}{\lambda^2}}}$

ვაიკერტის: $g(|x|)=\begin{cases}1 & |x| = 0\\ 1 - \exp(\frac{-3.31488}{(|x|/\lambda)^8}) & |x| > 0\end{cases}$

TV (Total variation): $g(|x|) = \frac{1}{|x|}$

ანიზოტროპული დიფუზია

ანიზოტროპული დიფუზია შემდეგი ფორმულით აღიწერება:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}\left(g\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(g\frac{\partial u}{\partial y}\right);$$

სადაც $g = g(|\nabla u|^2) = \frac{1}{1+\frac{|\nabla u|^2}{\lambda^2}}$ პერონა-მალიკის ფუნქციაა, $\lambda > 0$ მუდმივია, რომლის მაგივრადაც ექსპერიმენტულად 2-ს იღებენ ხმაურის მოცილების ამოცანაში.

$$|\nabla u|^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 =$$

$$= \left(\frac{u(x_{i+1}, y_j, t_n) - u(x_i, y_j, t_n)}{h_1}\right)^2 + \left(\frac{u(x_i, y_{j+1}, t_n) - u(x_i, y_j, t_n)}{h_2}\right)^2;$$

$\frac{\partial u}{\partial t}\Big|_{\partial\Omega} = 0$ სასაზღვრო პირობაა.

$u(0, x, y) = u_0(x, y)$ საწყისი დაუმუშავებელი სურათია.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(g\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(g\frac{\partial u}{\partial y}\right) \Leftrightarrow \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\tau} =$$

$$= \frac{1}{h_1}\left(\frac{g_{i+\frac{1}{2},j}^k\left(u_{i+1,j}^k - u_{i,j}^k\right)}{h_1} - \frac{g_{i-\frac{1}{2},j}^k\left(u_{i,j}^k - u_{i-1,j}^k\right)}{h_1}\right) + + \frac{1}{h_2}\left(\frac{g_{i,j+\frac{1}{2}}^k\left(u_{i,j+1}^k - u_{i,j}^k\right)}{h_2} - \frac{g_{i,j-\frac{1}{2}}^k\left(u_{i,j}^k - u_{i,j-1}^k\right)}{h_2}\right) =$$

$$= \frac{1}{h_1^2} g_{i+\frac{1}{2},j}^k\left(u_{i+1,j}^k - u_{i,j}^k\right) - \frac{1}{h_1^2} g_{i-\frac{1}{2},j}^k\left(u_{i,j}^k - u_{i-1,j}^k\right) + \frac{1}{h_2^2} g_{i,j+\frac{1}{2}}^k\left(u_{i,j+1}^k - u_{i,j}^k\right) - \frac{1}{h_2^2} g_{i,j-\frac{1}{2}}^k\left(u_{i,j}^k - u_{i,j-1}^k\right)$$

ანიზოტროპული დიფუზიის სქემის მდგრადობა

$$u_{i,j}^{k+1} = u_{i,j}^k + \frac{\tau}{h_1^2} g^k_{i+\frac{1}{2},j}\left(u_{i+1,j}^k - u_{i,j}^k\right) - \frac{\tau}{h_1^2} g^k_{i-\frac{1}{2},j}\left(u_{i,j}^k - u_{i-1,j}^k\right) + \frac{\tau}{h_1^2} g^k_{i,j+\frac{1}{2}}\left(u_{i,j+1}^k - u_{i,j}^k\right) -$$

$$- \frac{\tau}{h_1^2} g^k_{i,j-\frac{1}{2}}\left(u_{i,j}^k - u_{i,j-1}^k\right)$$

$$u_{i,j}^{k+1} = \left(1 - \frac{\tau}{h_1^2} g^k_{i+\frac{1}{2},j} - \frac{\tau}{h_1^2} g^k_{i-\frac{1}{2},j} - \frac{\tau}{h_1^2} g^k_{i,j+\frac{1}{2}} - \frac{\tau}{h_1^2} g^k_{i,j-\frac{1}{2}}\right) u_{i,j}^k +$$

$$+ \frac{\tau}{h_1^2}\left(g^k_{i+\frac{1}{2},j} u_{i+1,j}^k - g^k_{i-\frac{1}{2},j} u_{i-1,j}^k\right) + \frac{\tau}{h_2^2}\left(g^k_{i,j+\frac{1}{2}} u_{i,j+1}^k - g^k_{i,j-\frac{1}{2}} u_{i,j-1}^k\right)$$

$$\left|u_{i,j}^{k+1}\right| \le \left(1 - \frac{\tau}{h_1^2} g^k_{i+\frac{1}{2},j} - \frac{\tau}{h_1^2} g^k_{i-\frac{1}{2},j} - \frac{\tau}{h_1^2} g^k_{i,j+\frac{1}{2}} - \frac{\tau}{h_1^2} g^k_{i,j-\frac{1}{2}}\right)\left|u_{i,j}^k\right| +$$

$$+ \frac{\tau}{h_1^2}\left(g^k_{i+\frac{1}{2},j}\left|u_{i+1,j}^k\right| - g^k_{i-\frac{1}{2},j}\left|u_{i-1,j}^k\right|\right) + \frac{\tau}{h_2^2}\left(g^k_{i,j+\frac{1}{2}}\left|u_{i,j+1}^k\right| - g^k_{i,j-\frac{1}{2}}\left|u_{i,j-1}^k\right|\right)$$

მდგრადობისთვის გვჭირდება პირობა:

$$\frac{\tau}{h_1^2}\left(g^k_{i+\frac{1}{2},j} - g^k_{i-\frac{1}{2},j}\right) + \frac{\tau}{h_2^2}\left(g^k_{i,j+\frac{1}{2}} - g^k_{i,j-\frac{1}{2}}\right) < 1$$

გავითვალისწინოთ, რომ g შემოსაზღვრულია და $g < 1$, მაშინ

$\frac{\tau}{h_1^2} + \frac{\tau}{h_2^2} < \frac{1}{2}$; უზრუნველყოფს მდგრადობის პირობას.

$$\left|u_{i,j}^{k+1}\right| \le \left(1 - \frac{\tau}{h_1^2} g^k_{i+\frac{1}{2},j} - \frac{\tau}{h_1^2} g^k_{i-\frac{1}{2},j} - \frac{\tau}{h_1^2} g^k_{i,j+\frac{1}{2}} - \frac{\tau}{h_1^2} g^k_{i,j-\frac{1}{2}}\right)\tilde{u} +$$

$$+ \left(\frac{\tau}{h_1^2} g^k_{i+\frac{1}{2},j} + \frac{\tau}{h_1^2} g^k_{i-\frac{1}{2},j} + \frac{\tau}{h_2^2} g^k_{i,j+\frac{1}{2}} + \frac{\tau}{h_2^2} g^k_{i,j-\frac{1}{2}}\right)\tilde{u} = \tilde{u}$$

ე.ი სქემა მდგრადია პირობით: $\frac{\tau}{h_1^2} + \frac{\tau}{h_2^2} < \frac{1}{2}$

თავი 4. პროგრამული დამუშავების შედეგები

პროგრამაში განვიხილეთ პერონა-მალიკის ორივე, ვაიკერტის, წრფივი, შარბონიეს და სრული ვარიაციის დიფუზიის კოეფიციენტები. შედეგების გაუმჯობესების მიზნით მოვახდინეთ ჯერ ვაიკერტის ფორმულის გადაკეთება, რითაც $g(|x|) = \begin{cases} 1 & |x| = 0 \\ 1 - \exp(\frac{-3.31488}{(|x|/\lambda)^8}) & |x| > 0 \end{cases}$ ფორმულაში ექსპონენტას მნიშვნელში არსებული მე8 ხარისხი დავიყვანეთ 4-მდე, შემდეგ კი TV (სრული ვარიაციის) ფორმულა $g(|x|) = \frac{1}{|x|}$ შევცვალეთ და ავიღეთ $g(|x|) = \frac{1}{1+|x|}$, რამაც g მოაქცია სასურველ საზღვრებში. მიღებულ შედეგებზე ქვემოთ ვისაუბრებთ.

დამუშავებული სურათის მიხედვით დიფუზიის გავლენის კარგად აღსაქმელად ჩვენ ვიყენებთ ე.წ. სხვაობიან სურათს (ორიგინალ სურათსა და დამუშავებულ სურათს შორის განსხვავება, რომელიც ასახავს ყოველ პიქსელში grayscale მნიშვნელობათა სხვაობას), ასევე ამ განსხვავების რიცხობრივად აღსაწერად ვიყენებთ მატრიცულ ნორმას $\|A\|_F = \sqrt{\sum_{i,j=1}^{n} |a_{i,j}|^2}$.

მატრიცული ნორმის გამოყენებამ არამარტო მოდელების შედარების საშუალება მოგვცა, არამედ საუკეთესო გზა აღმოჩნდა კონკრეტული მოდელისთვის (დიფუზიურობის კოეფიციენტისთვის) საუკეთესო (ან მასთან მიახლოებული) შედეგი გვეპოვნა დაფიქსირებული λ კოეფიციენტის გათვალისწინებით.

ჯერ განვიხილავთ ნორმის შემოღებამდე მიღებულ შედეგებს, რითაც ერთმანეთს ვადარებდით სხვადასხვა კოეფიციენტებს და მათ გავლენას ხმაურიან სურათზე.

(As far as we know, photo is public domain)

(As far as we know, photo is public domain)

მარცხნივ წარმოდგენილია ორიგინალი სურათი, მარჯვნივ კი ხმაურიანი სურათი, რომელიც უნდა დავამუშავოთ.

კონკრეტული კოეფიციენტის შემთხვევაში პროგრამული დამუშავება ხდება λ კოეფიციენტის (პერონა-მალიკის, შარბონიეს და ვაიკერტის შემთხვევაში) ან იტერაციის (დროითი ბიჯის) ცვლილების მიხედვით. ჯერ განვიხილოთ მაგალითად λ=5, იტერაცია T=200 შემთხვევები სხვადასხვა კოეფიციენტისთვის შესაბამისი თანმიმდევრობით: *პერონა-მალიკი 1, პერონა-მალიკი 2, შარბონიე, ვაიკერტი, წრფივი და შეცვლილი სრული ვარიაციის.* შედეგის უკეთ აღსაქმელად ვნახოთ მათთვის მიღებული "სხვაობიანი" სურათი (პიქსელური მნიშვნელობების განსხვება ორიგინალ და საბოლოო სურათს შორის).

აშკარაა რომ პერონა-მალიკი 1 ბევრად უკეთეს შედეგს გვაძლევს ვიდრე დანარჩენი კოეფიციენტები. წრფივით (იგივე იზოტროპული დიფუზია) ძალიან ბუნდოვანი ფოტო მივიღეთ რადგან ის საერთოდ არაა ლამბდაზე დამოკიდებული და კონტურებს ვერ ინარჩუნებს g ფუნქციის მუდმივობის გამო. შარბონიეც, რაოდენ გასაკვირიც არ უნდა იყოს, დიდად არ „ჩამოუვარდება“ წრფივს, რაც იმის შედეგია რომ შარბონიეს კოეფიციენტი წარმოადგენს კვადრატულ ფესვს პერონა-მალიკ 2 (PM2)-ის კოეფიციენტიდან, შესაბამისად ყოველ ეტაპზე g კოეფიციენტი (რიცხვი 0-დან 1-მდე) უფრო დიდია ვიდრე PM2, რაც ნელნელა უახლოვდება წრფივი კოეფიციენტის მნიშვნელობას g=1.

სხვაობიანი სურათებისათვის გავითვალისწინოთ რომ სხვაობა განიხილება grayscale მნიშვნელობებს შორის და რაც უფრო ახლოა მიღებული შედეგი ორიგინალთან (ანუ პიქსელის მნიშვნელობა 0-ია), მით უფრო შავ ფერს ვიღებთ. შესაბამისად თუ კარგად დავუკვირდებით,

დავინახავთ რომ linear კოეფიციენტისთვის ყველაზე მეტი ღია ნაცრისფერი ნაწილია დარჩენილი, ანუ ცუდად დამუშავდა სურათი, პერონა-მალიკისთვის კი ყველაზე ნაკლები შავისგან განსხვავებული ფერია მიღებული.

ახლა კი გადავიდეთ თითოეული კოეფიციენტისგან მიღებულ საუკეთესო შედეგებზე რაც კი ნორმის შემოღებამდე გვქონდა.

PM1: λ=2, T=1000

PM2: λ=5, T=1000

Charbonnier: λ=2, T=100

Weickert: λ=8, T=300

Linear: T=5

როგორც ვხედავთ რიგი კოეფიციენტებისა ბევრად უკეთეს შედეგს გვაძლევს ვიდრე წინა შემთხვევაში, რაც გვარწმუნებს რომ ანიზოტროპული დიფუზიისთვის ნებისმიერი კოეფიციენტი შეიძლება გამოგვადგეს თუკი მათ კარგად შევურჩევთ პარამეტრებს.

მატრიცული ნორმის გამოყენება

როგორც უკვე ვთქვით, სურათის დამუშავებიდან მიღებული შედეგის რიცხობრივად აღსაქმელად შემოვიტანეთ მატრიცის ნორმა: $\|A\|_F = \sqrt{\sum_{i,j=1}^{n} |a_{i,j}|^2}$. ამ ნორმისა და პროგრამული უზრუნველყოფის დახმარებით გავარკვიეთ როგორია კონკრეტული კოეფიციენტისათვის საუკეთესო პარამეტრები (λ კოეფიციენტი და T დროის ბიჯი /იტერაცია/), რაც შეფასებულია ნორმის ყველაზე დაბალი მნიშვნელობით. ვნახოთ მიღებული შედეგები.

PM1: λ=37, T=6, norm=20396

PM2: λ=74, T=4, norm=20495

Charbonnier: λ=17, T=7, norm=20523

Weickert: λ=38, T=5, norm=20662

Linear: T=2, norm=21375

TV-changed: T=93, norm=20387

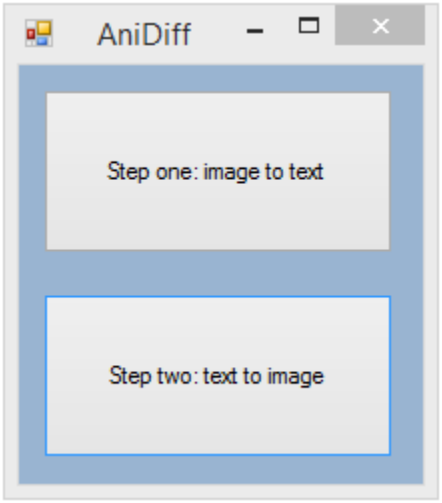
როგორც ვხედავთ მიღებულ სურათებს შორის ვიზუალური განსხვავება მცირეა, თუმცა მატრიცული ნორმა აშკარად მეტყველებს შედარებით უკეთეს და ნაკლებად უკეთეს კოეფიციენტებზე. ჩვენი შეცვლილი სრული ვარიაციის ფორმულა საუკეთესო ნორმას გვაძლევს, ხოლო წრფივი მეთოდი ყველაზე ნაკლებად კარგს, თუმცა ისიც უნდა აღინიშნოს რომ წრფივის შემთხვევაში ყველაზე ნაკლები იტერაცია დაგვჭირდა ამ მეთოდის საუკეთესო შედეგის მისაღებად.

პროგრამის დეტალური განხილვა, მუშაობის პრინციპი და ინსტრუქცია

როგორც იყო ადრე აღნიშნული, ჩვენ დაწერილი გვაქვს ორი პროგრამა: პირველი პროგრამა გამოიყენება იმისთვის, რომ გადავიყვანოთ სურათი ტექსტურ ფაილში რიცხვით მატრიცად; მეორე პროგრამა კი ამუშავებს ამ მატრიცას და შემდეგ სხვა ფაილში წერს დამუშავებულ მატრიცას და სხვაობის მატრიცას ორიგინალ და დამუშავებულ სურათებს შორის.

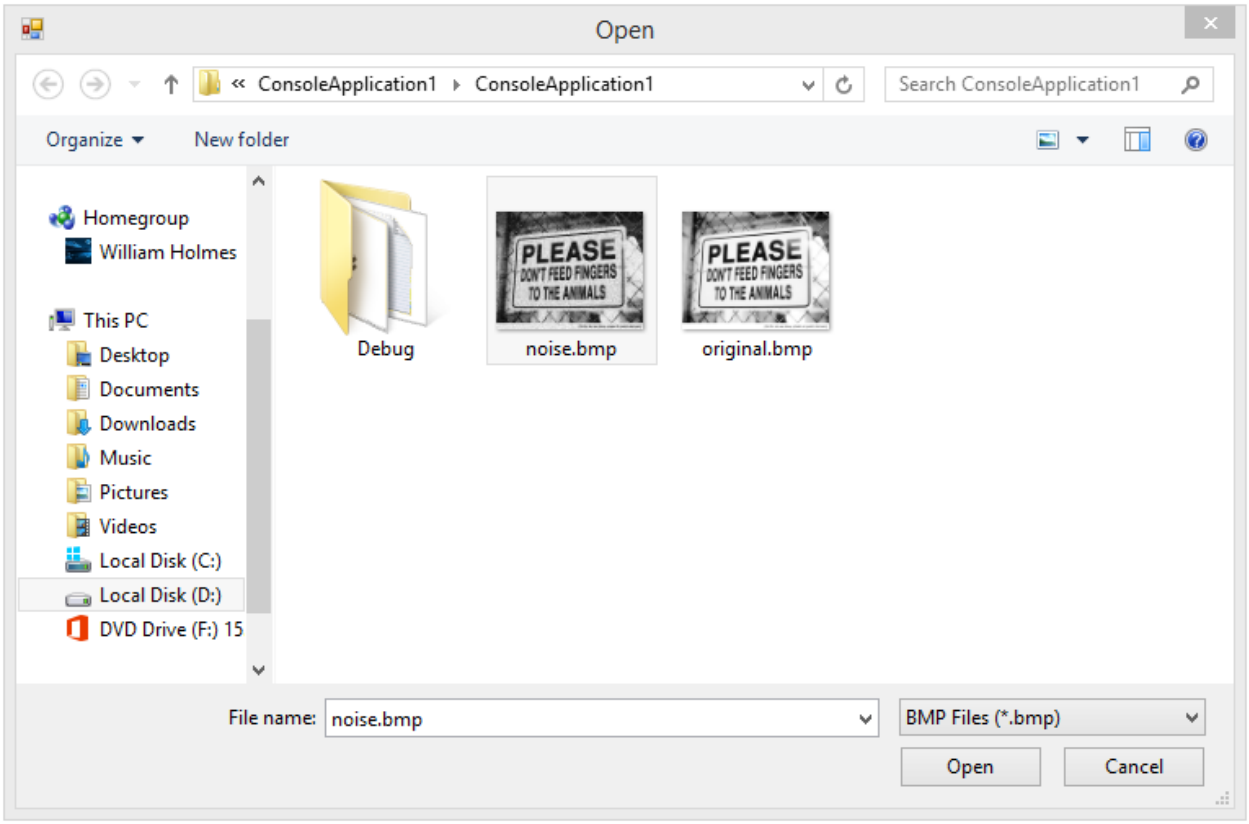
პირველი პროგრამა.

პირველი პროგრამა დაწერილია C# ენაზე. სურათთან მუშაობისთვის ვიყენებთ Bitmap კლასს.



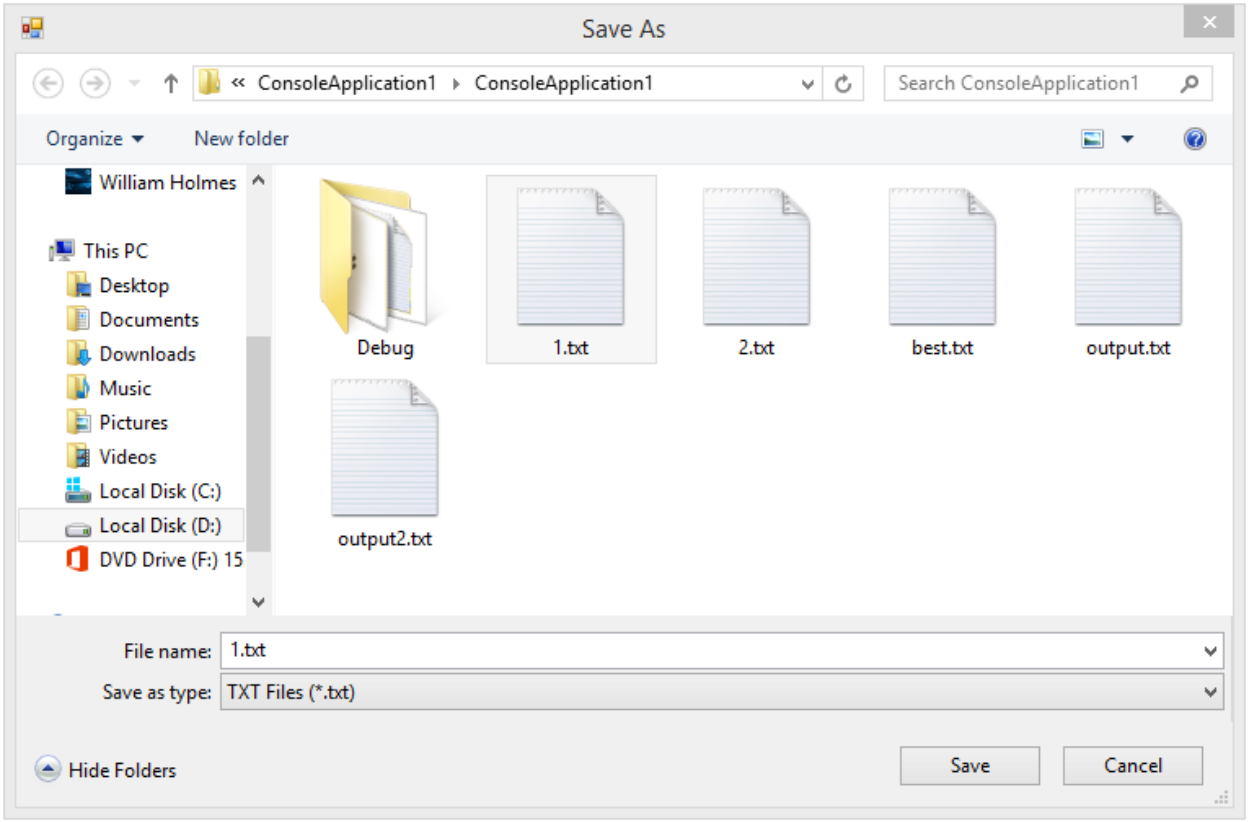
პროგრამის ინტერფეისი.

პირველი ღილაკის დაწკაპუნების შემდეგ გამოვა ფანჯარა, სადაც უნდა მივუთითოთ, თუ სად მდებარეობს ჩვენი ხმაურიანი სურათი.



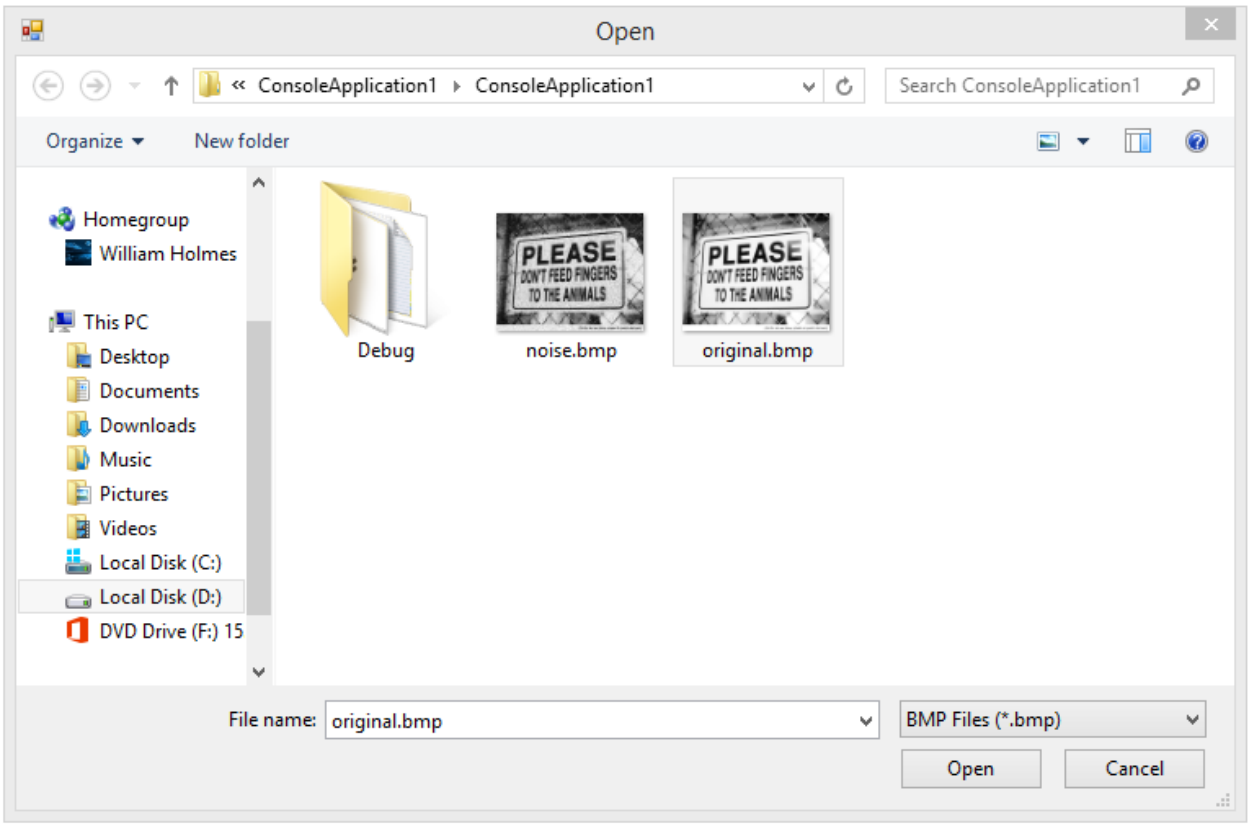
პირველი დიალოგური ფანჯარა.

ამის შემდეგ, უნდა ავირჩიოთ, თუ რომელ ფაილში ჩაწეროს სურათის მონაცემები. ჩვენ აუცილებლად უნდა ავირჩიოთ ფაილი სახელად „1.txt”, რადგან მეორე პროგრამა მონაცემებს ამ ფაილიდან იღებს.



მეორე დიალოგური ფანჯარა.

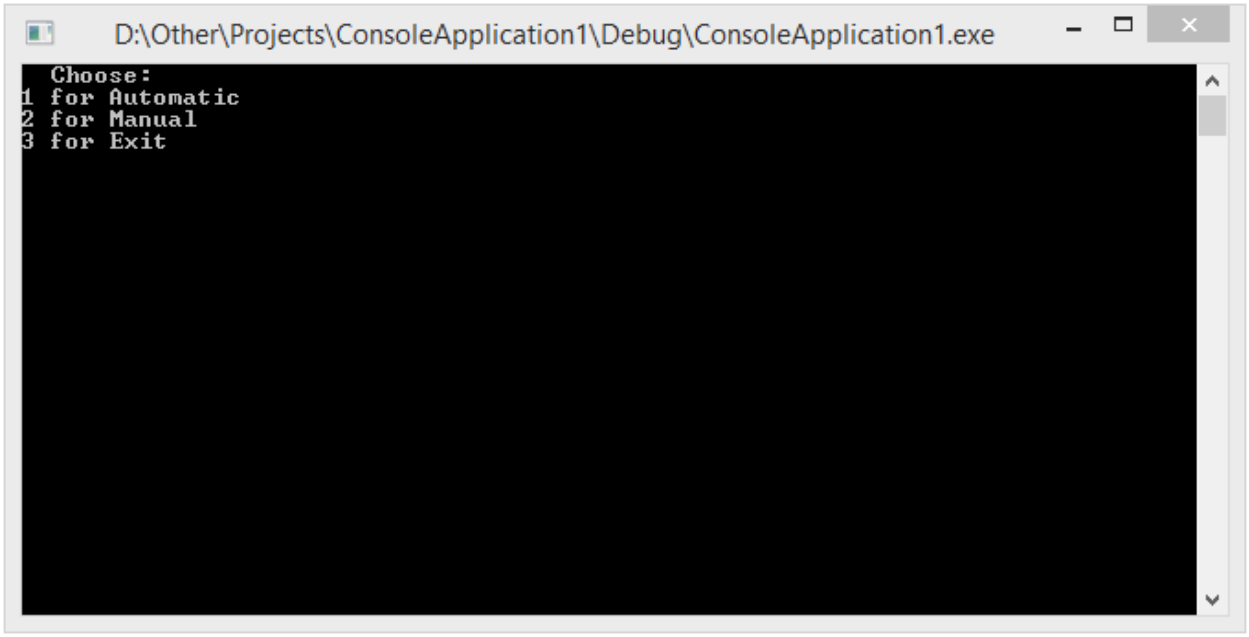
იმისთვის რომ ნორმა იპოვოს ორი მატრიცის სხვაობის (ორიგინალი და დამუშავებული) და სხვაობის სურათიც რომ ამობეჭდოს, რა თქმა უნდა, საჭიროა მივუთითოთ ორიგინალი სურათის მისამართი.



მესამე დიალოგური ფანჯარა.

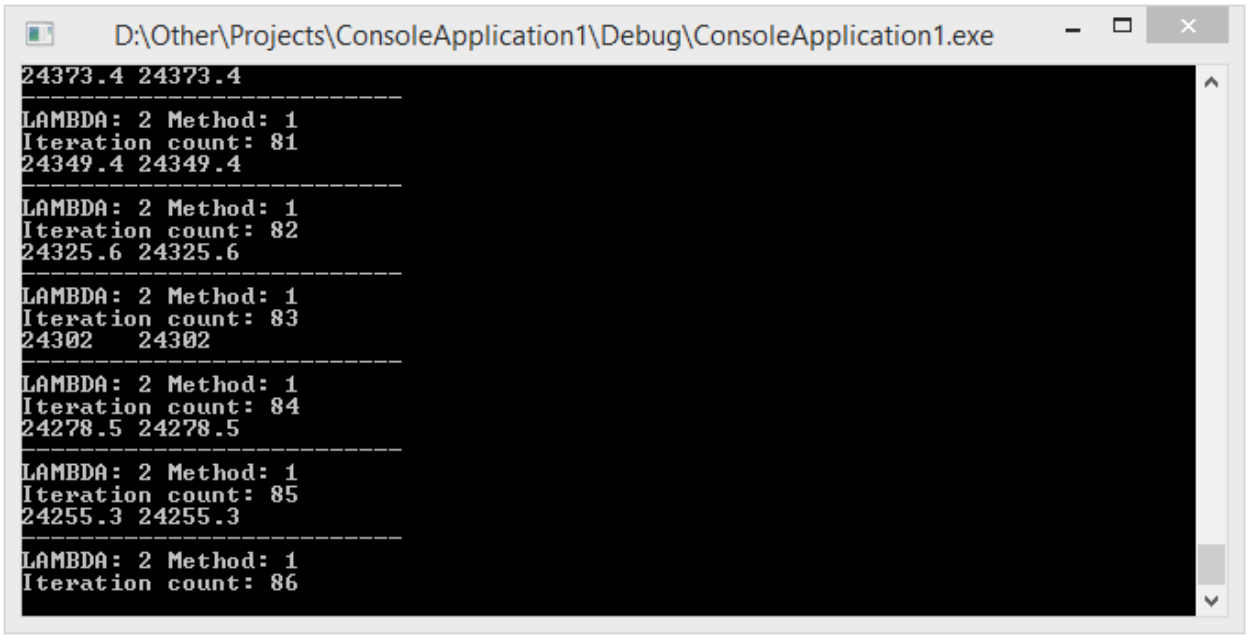
ასევე უნდა შევნიშნოთ, რომ პროგრამა ავტომატურად ჩაწერს ორიგინალი სურათის მონაცემებს იგივე ფოლდერში ფაილში სახელით „2.txt", რომელთანაც მუშობს მეორე პროგრამაც.

ამის შემდეგ უნდა ჩავრთოთ მეორე პროგრამა, რომელიც დაწერილია C++ ენაზე.



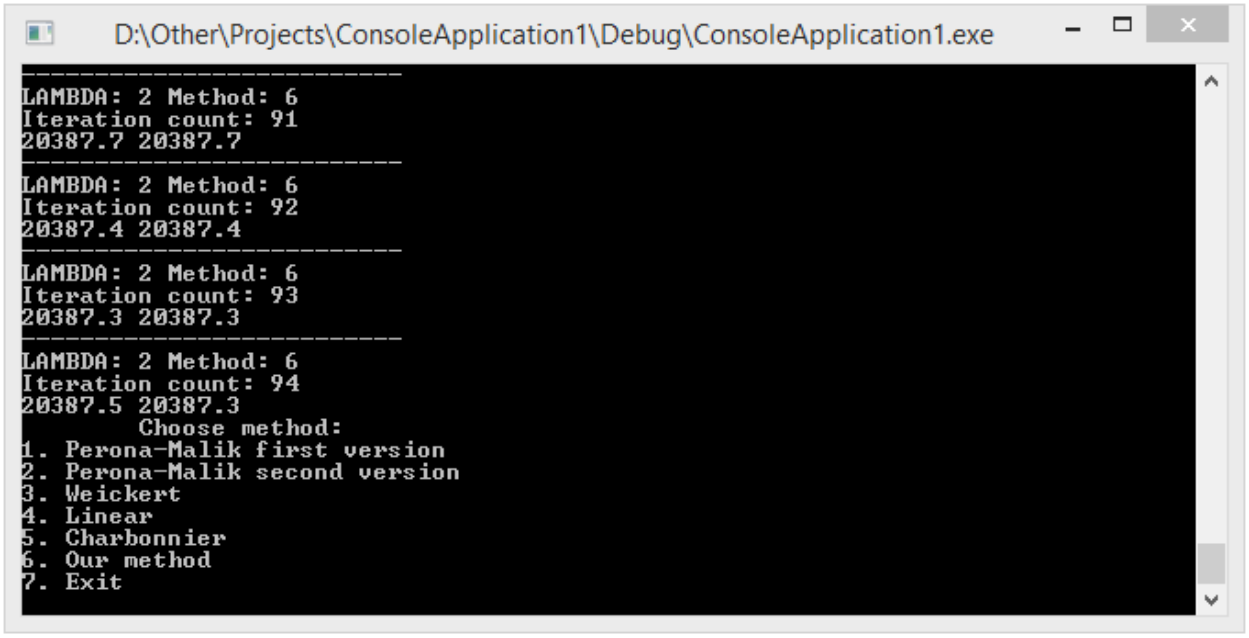
საწყისი ფანჯარა.

საწყის ფანჯარაში უნდა ავირჩიოთ ავტომატურად იპოვოს პროგრამა საუკეთესო შედეგი ყოველ მეთოდში, თუ ხელით შევიყვანოთ. ჯერ განვიხილოთ ავტომატურის პოვნის ფუნქცია. ანუ დავწეროთ „1" და დავაჭიროთ ენტერს.



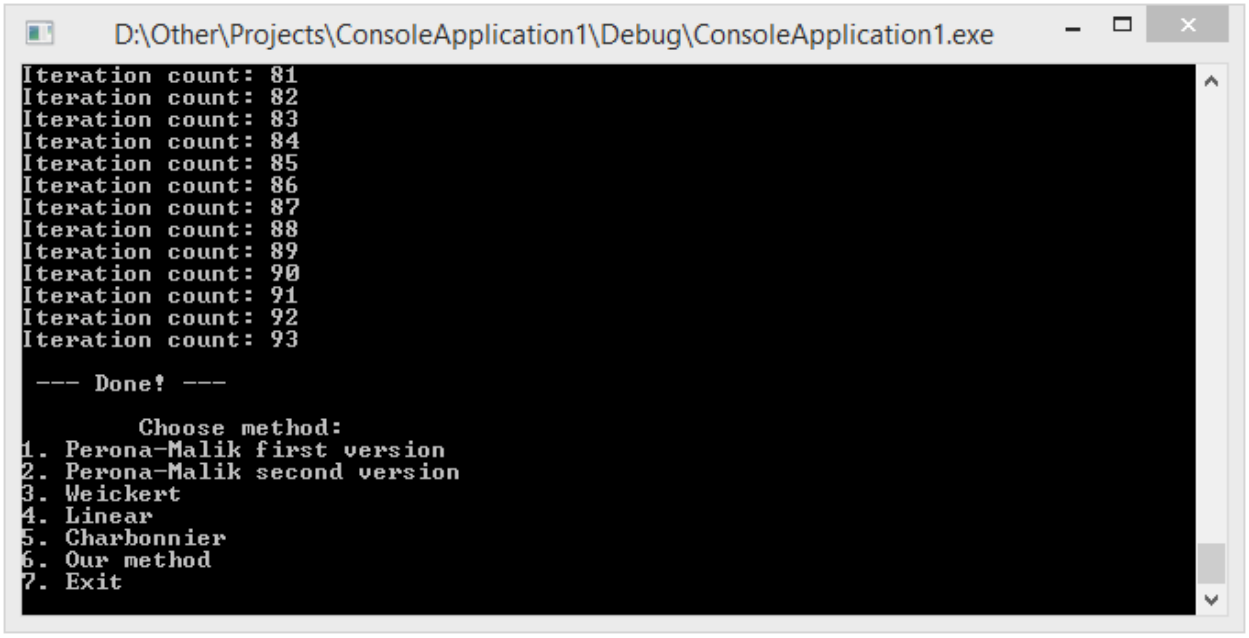
ავტომატური ფუნქციის მუშაობის პროცესი.

არჩევის შემდეგ პროგრამა ავტომატურად ამოწმებს ყოველ მეთოდს, თუ რომელი ლამბდის მნიშვნელობისთვის და რომელ იტერაციაზე იქნება საუკეთესო შედეგი. შემდეგ ამას იმახსოვრებს და როცა დაასრულებს, მოგვცემს მეთოდის არჩევის საშუალებას საუკეთესო პარამეტრებით.



პარამეტრების ავტომატური შერჩევის შემდეგ.

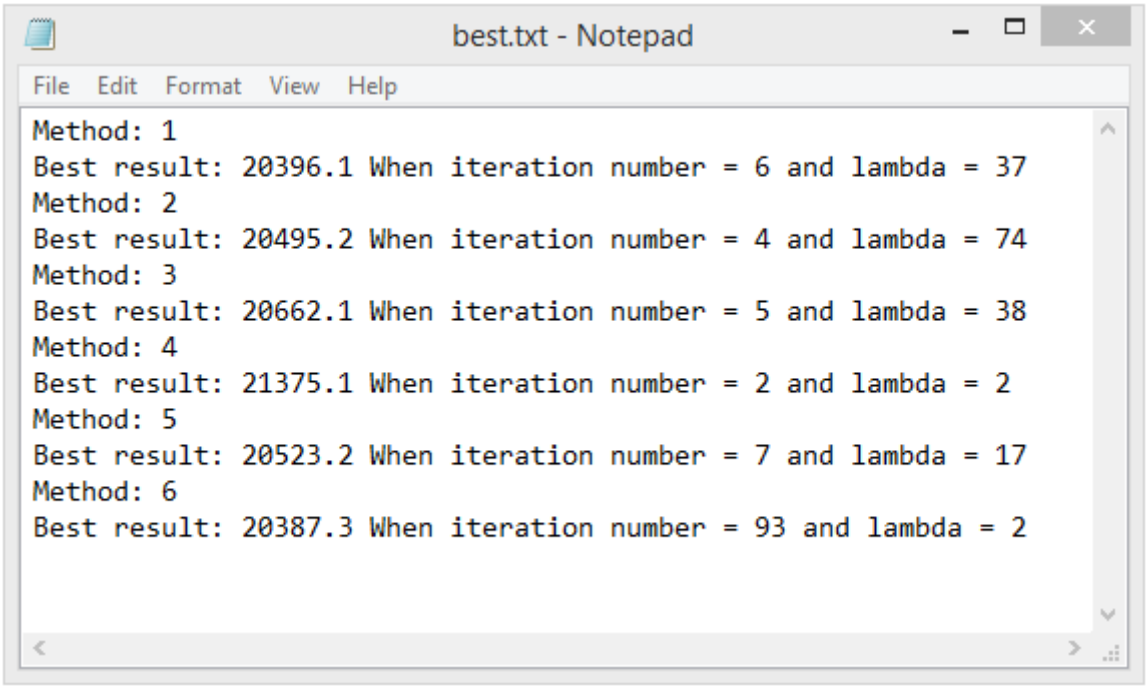
ავირჩიოთ, მაგალითად, მეექვსე მეთოდი. პროგრამამ დაასკვნა, რომ მეექვსე მეთოდი ყველაზე კარგად ამუშავებს სურათს 93-ე იტერაციაზე, ამიტომ ჩაატარა 93 იტერაცია და შემდეგ ისევ შეგვიძლია დავამუშავოთ საუკეთესო შედეგების მიხედვით, ოღონდ სხვა მეთოდით.



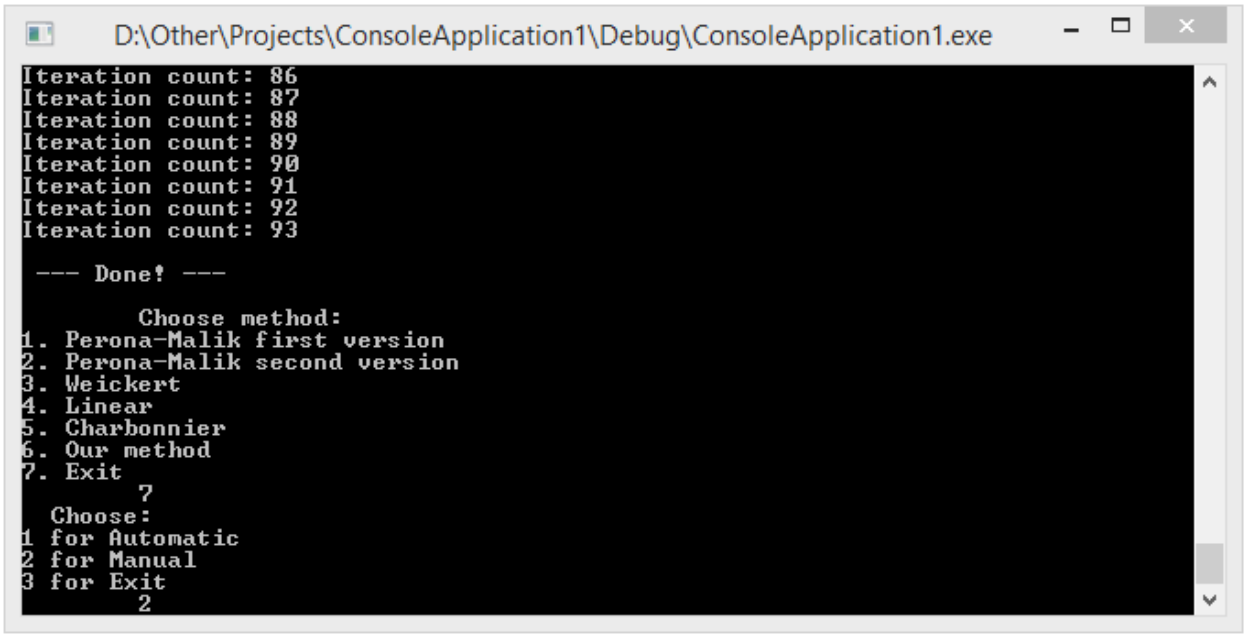
დამუშავების შემდეგ.

ასევე აღსანიშნავია, რომ საუკეთესო პარამეტრები ასევე ინახება ტექსტურ ფაილში სახელით:

„best.txt”.

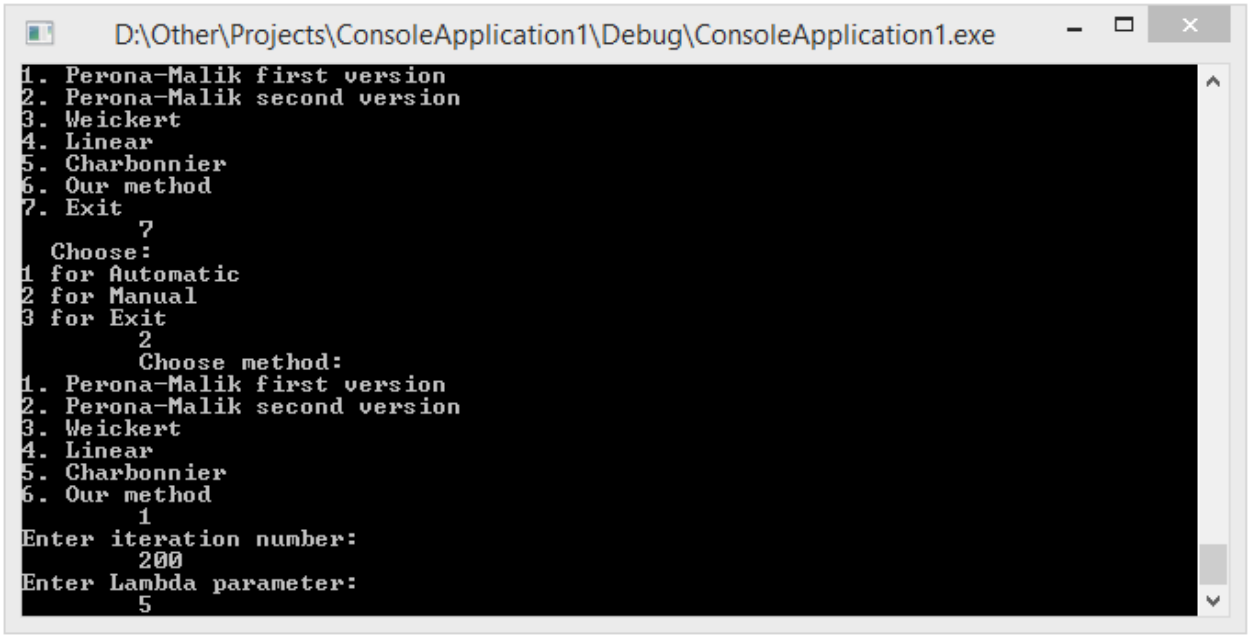


ახლა ავირჩიოთ მე-7, ანუ გასვლა და განვიხილოთ პარამეტრების ხელით შერჩევითი ფუნქცია.



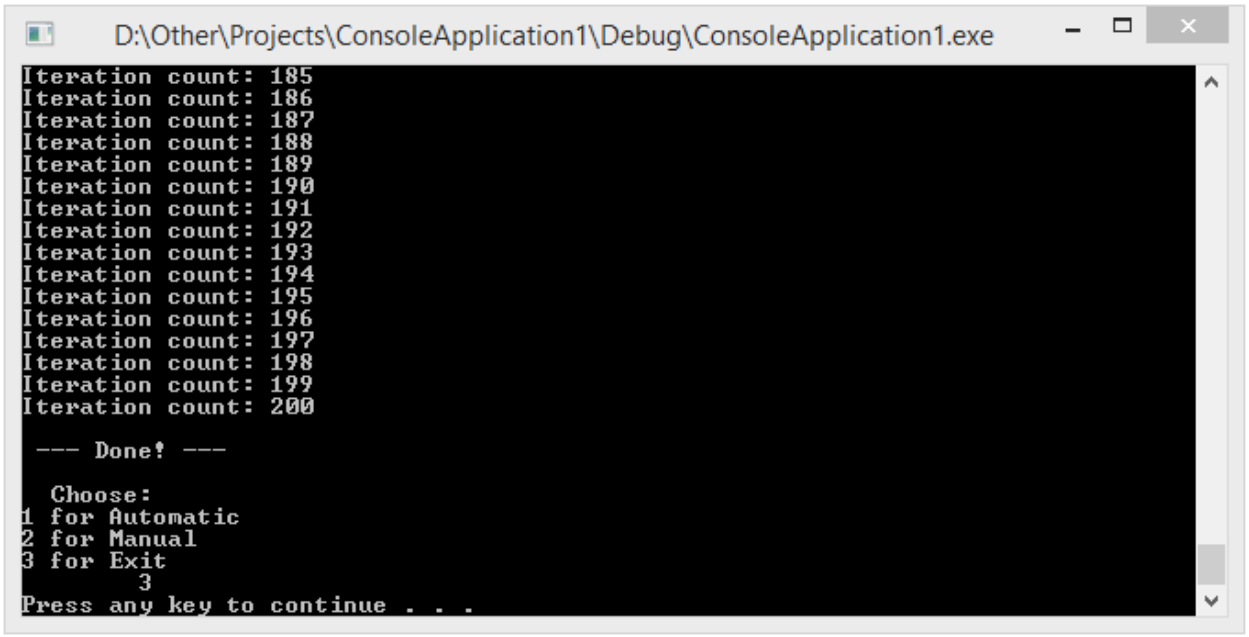
გასვლა პარამეტრების ავტომატური შერჩევის ფუნქციიდან.

ხელით საჭიროა მივუთითოთ მეთოდის ნომერი, იტერაციის რაოდენობა და ლამბდა პარამეტრის მნიშვნელობა. მაგალითისთვის ავიღოთ პირველი მეთოდი და დავამუშავოთ 200 იტერაციაზე და ლაბდა პარამეტრის მნიშვნელობით 5.



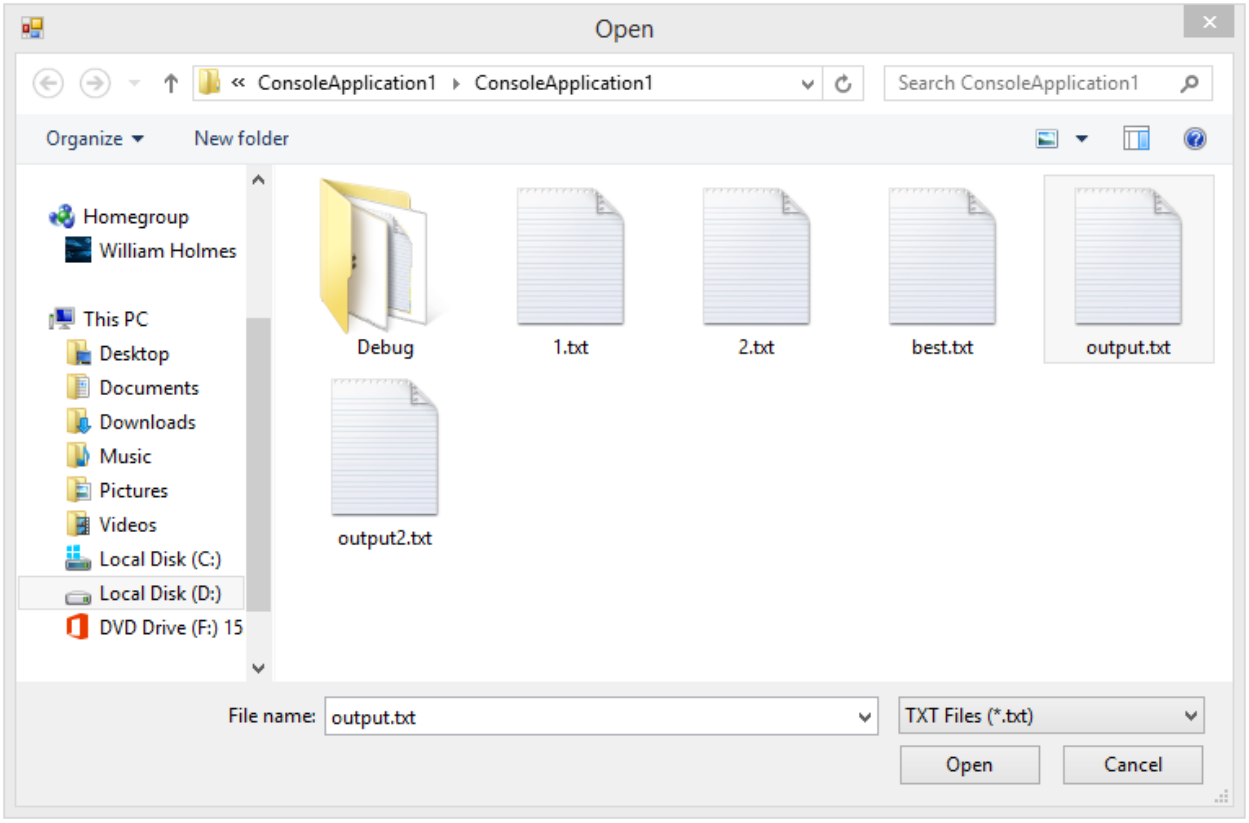
პარამეტრების ხელით შერჩევა.

პროგრამა დაამუშავებს მითითებულ პარამეტრებზე და შემდეგ ჩვენ შეგვიძლია ავირჩიოთ სხვა მეთოდი. მაგრამ ახლა გამოვიდეთ პროგრამიდან და განვიხილოთ ისევ პირველი პროგრამა, ვნახოთ რა შედეგი ექნება სურათს პირველი მეთოდის დამუშავების შემდეგ, 200 იტერაციაზე და ლამბდა 5 მნიშვნელობით.



მუშაობის დასრულება.

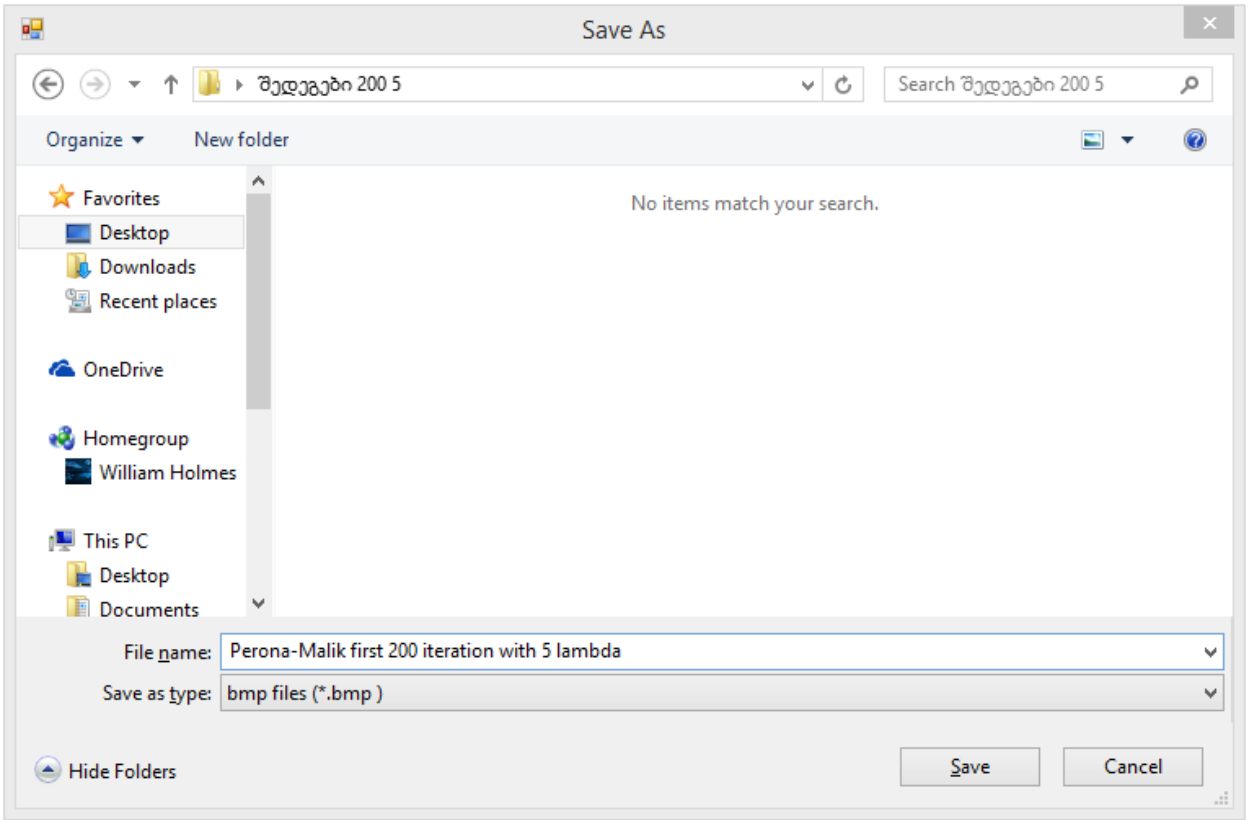
ახლა გადავიდეთ პირველ პროგრამაზე და დავაჭიროთ მეორე ღილაკს, რომელიც გადაიყვანს დამუშავებულ მატრიცას სურათში. მეორე პროგრამამ შეინახე მანოცემები ფაილში სახელით „output.txt”. ღილაკის დაწკაპუნების შემდეგ ავირჩიოთ ეგ ფაილი.



დამუშავებული მატრიცის არჩევა.

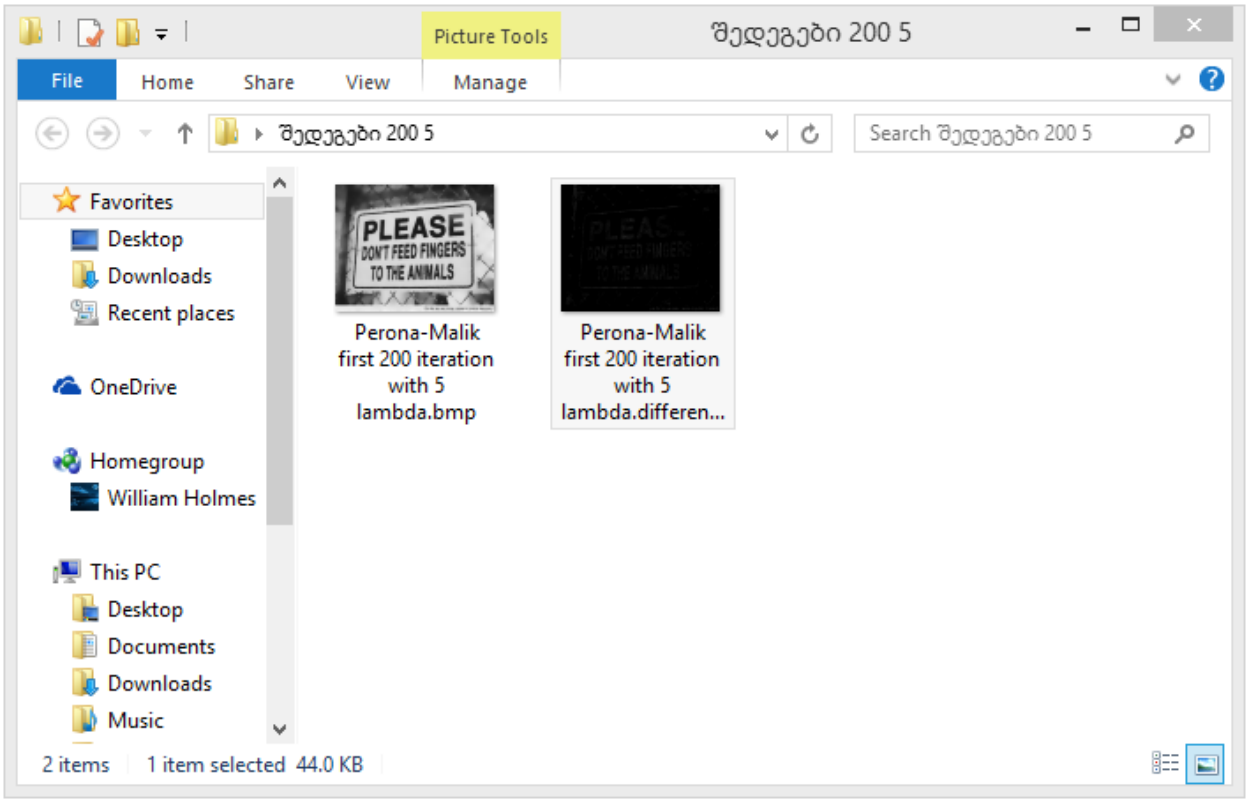
როგორც ვხედავთ ასევე არის ფაილი „output2.txt". ამ ფაილში ინახება სხვაობა. პროგრამა ავტომატურად იპოვის ამ ფაილს.

შემდეგ უნდა მივუთითოთ, თუ სად შეინახოს მიღებული სურათი.



სურათის შენახვა.

ამის შემდეგ შეგვიძლია პროგრამა გამოვრთოთ. შევამოწმოთ ფოლდერი, რომელშიც შეინახა სურათი.



დამუშავებული სურათები.

როგორც ვხედავთ, პროგრამამ ავტომატურად იგივე ფოლდერში შექმნა სხვაობის სურათიც, რომლის სახილეც იგივეა, ოღონდ დაემატა სიტყვა “diffirent”.

გამოყენებული ლიტერატურა:

http://www.lpi.tel.uva.es/muitic/pim/docus/anisotropic_diffusion.pdf
http://www.lems.brown.edu/~tek/research/morphology/image-smoothing.html
http://en.wikipedia.org/wiki/Gaussian_blur
http://www.cs.berkeley.edu/~malik/papers/MP-aniso.pdf
http://neumann.math.tufts.edu/research/denoising.pdf
http://web.cs.hacettepe.edu.tr/~erkut/bil717.s13/w03-nonlineardif.pdf
http://www.mia.uni-saarland.de/Publications/mrazek-scsp03.pdf
http://en.wikipedia.org/wiki/Anisotropic_diffusion
http://en.wikipedia.org/wiki/Grayscale
http://www.webopedia.com/TERM/P/pixel.html
http://www.rapidtables.com/web/color/RGB_Color.htm
http://en.wikipedia.org/wiki/RGB_color_model
http://koiavablog.blogspot.com/2013/06/hsl-hsv.html
http://hal.archives-ouvertes.fr/docs/00/33/27/98/PDF/Tschumperle-Deriche_AIEP_single.pdf
https://drive.google.com/file/d/0B3Glnf9SxRPMRjcwbUJwSHNSajA/edit?usp=sharing
http://en.wikipedia.org/wiki/Smoothing
http://en.wikipedia.org/wiki/Convolution
http://ka.wikipedia.org/wiki/BMP