УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

по дисциплине **«Алгоритмические основы современной компьютерной графики»**

на тему: «Реализация программ для применения графических фильтров»

Выполнил: студент гр. ИТ-11,
Петров Н.Н.
Принял: доцент каф. «ИТ»,
к.ф.-м.н., доц.,
Иванов В.В.

Гомель 2021

Цель работы: получить теоретические и практические знания по созданию и применению графических фильтров.

Задание

Разработать программу, использующую следующие графические фильтры:
– инверсия цветов;
– размытие;
– увеличение резкости;
– тиснение;
– акварельный эффект.

Теоретические сведения

Инверсия цветов

Это наиболее простой способ редактирования изображения. Для реализации эффекта «инверсия цветов» достаточно изменить составляющие цвета на противоположенные.

Матрица – ядро свертки

Указанные выше графические эффекты «размытие», «увеличение резкости», «тиснение» и «акварельный эффект» создаются с помощью применения матрицы чисел размерности (3x3), которая называется ядром свертки. Для преобразования пикселя изображения он умножается на значение в центре ядра, а значения пикселей, находящихся вокруг данного, умножаются на соответствующие им коэффициенты ядра, после чего все значения суммируются, и мы получаем новое значение для изменяемого пикселя. Данный процесс должен быть последовательно выполнен с каждым пикселем редактируемого изображения. От коэффициентов ядра зависит то, как изменится редактируемое изображение. Для достижения некоторых эффектов необходимо последовательно применить к изображению не одну, а несколько матриц.

Алгоритм размытия

Для того, чтобы размыть изображение, необходимо считать в память значения RGB-составляющих цвета каждого пикселя. Затем, ядро размывания будет применено ко всем составляющим компонентам цвета всех пикселей редактируемого изображения:

$$R = \begin{vmatrix} 0{,}05 & 0{,}05 & 0{,}05 \\ 0{,}05 & 0{,}60 & 0{,}05 \\ 0{,}05 & 0{,}05 & 0{,}05 \end{vmatrix}$$

Рисунок 1 - Матрица для фильтра «Размытие»

Для того чтобы определить цвет пикселя, находящегося под центром ядра, необходимо провести умножение весовых коэффициентов ядра с соответствующими значениями цвета редактируемого изображения. После этого результаты суммируются. Полученное изображение «размыто» по

сравнению с оригинальным, так как цвет каждого обработанного пикселя «распространился» среди соседних пикселей. Чтобы увеличить ядро размытия, вы можете: – использовать ядро большего размера (так цвет будет распределяться среди большего количества соседних пикселей); – изменять коэффициенты таким образом, чтобы уменьшить влияние центрального коэффициента; – многократно выполнить фильтрацию изображения;

Алгоритм увеличения резкости

Создавая эффект увеличения резкости, выполняем все тот же алгоритм, но используем другое ядро, так как теперь нашей целью является увеличение резкости изображения. Ядро G для увеличения резкости:

$$G = \begin{vmatrix} -0{,}1 & -0{,}1 & -0{,}1 \\ -0{,}1 & 1{,}8 & -0{,}1 \\ -0{,}1 & -0{,}1 & -0{,}1 \end{vmatrix}$$

Рисунок 2 - Матрица для фильтра «Увеличение резкости»

Как и в предыдущем случае, отдельно обрабатываем RGB-составляющие, после чего формируем значения цвета обрабатываемого пикселя. Для увеличения контраста между центральным пикселем и соседями используются отрицательные весовые коэффициенты. Таким образом, результирующее изображение стало более четким, чем оригинал. Дополнительные детали возникли из ничего – это просто увеличенный контраст между цветами пикселей.

Алгоритм тиснения

Тиснение выполняется аналогично, но в данном случае используем не одну матрицу, а несколько.

$$T_1 = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad T_2 = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix} \quad T_3 = \begin{vmatrix} -0 & -1 & 0 \\ -1 & 4 & -1 \\ -0 & -1 & 0 \end{vmatrix}$$

Рисунок 3 - Матрица для фильтра «Тиснение»: шаг первый

В то время как ядра размытия и резкости имели сумму коэффициентов равную единице, в данном случае сумма весов в ядре тиснения равна 0. Если сумма коэффициентов не будет равна 0, мы получим отклонение к какому-то конкретному цвету. Полученное значение цвета будет дополнительно обработано (усреднено) и приведено к диапазону 0-255 (подробнее это можно увидеть при реализации данного фильтра). Меняя значения позиций 1 и -1, мы можем получить измененное направление подсветки.

$$T_4 = \begin{vmatrix} 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & -1 & -2 \end{vmatrix} \quad T_5 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{vmatrix} \quad T_6 = \begin{vmatrix} 2 & -1 & 0 \\ -1 & 0 & -1 \\ -0 & -1 & 2 \end{vmatrix}$$

Рисунок 4 - Матрица для фильтра «Тиснение»: шаг второй

Алгоритм акварелизации

Название акварельного фильтра говорит само за себя: результирующее изображение будет выглядеть так, как будто его нарисовали акварелью. На первом этапе применения данного фильтра сгладим цвета редактируемого изображения.

$$M = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

Рисунок 5 - Матрица для фильтра «Акварельный эффект»: шаг первый

На следующем этапе увеличим резкость переходов для завершения создания эффекта акварели.

$$R = \begin{vmatrix} -0,5 & -0,5 & -0,5 \\ -0,5 & 5 & -0,5 \\ -0,5 & -0,5 & -0,5 \end{vmatrix}$$

Рисунок 6 - Матрица для фильтра «Акварельный эффект»: шаг второй

Ход работы

Листинг программы

```
//IT-11 KaitanovaA.N.GSTU
//Lab №6
im=imread('el.jpg')
//reading of image
imrazm=im;
imrez=im;
imtesn=im;
iminvers=im;
imakv=im;
//seting of matrix filters
T1=[-1,0,1;-2,0,2;-1,0,1]
F1=[0.05,0.05,0.05;0.05,0.6,0.05;0.05,0.05,0.05]
F2=[-0.1,-0.1,-0.1;-0.1,1.8,-0.1;-0.1,-0.1,-0.1]
M=(1/16)*[1,2,1;2,4,2;1,2,1]
R=[-0.5,-0.5,-0.5;-0.5,5,-0.5;-0.5,-0.5,-0.5]
function [F1, imrazm]=razm()
   imrazm=imfilter(imrazm,F1)
   imshow(imrazm)
endfunction
```

```
h=uicontrol("style","pushbutton","string","картинка","Callback",...
"imshow(imrazm)","Callback_Type",2,"position",[50, 20, 90, 30])
h=uicontrol("style","pushbutton","string","размытие","Callback",...
"imrazm=imfilter(imrazm,F1)","Callback_Type",2,"position",[50, 50, 90, 30])
//funkciya rezkosti
function [F2, imrez]=rezk()
   imrez=imfilter(imrez,F2)
   imshow(imrez)
endfunction
h=uicontrol("style","pushbutton","string","картинка","Callback",...
"imshow(imrez)","Callback_Type",2,"position",[140, 20, 90, 30])
h=uicontrol("style","pushbutton","string","резкость","Callback",...
"imrez=imfilter(imrez,F2)","Callback_Type",2,"position",[140, 50, 90, 30])
//funktiya tesneniya
function [T1, imtesn]=tesn()
   imtesn=imfilter(imtesn,T1)
   imshow(imtesn)
endfunction
h=uicontrol("style","pushbutton","string","картинка","Callback",...
"imshow(imtesn)","Callback_Type",2,"position",[230, 20, 90, 30])
h=uicontrol("style","pushbutton","string","теснение","Callback",...
"imtesn=imfilter(imtesn,T1)","Callback_Type",2,"position",[230, 50, 90, 30])
 function [iminvers]=invers()
   iminvers=iminvers*(-1)
   imshow(iminvers)
endfunction
h=uicontrol("style","pushbutton","string","картинка","Callback",...
"imshow(iminvers)","Callback_Type",2,"position",[320, 20, 90, 30])
h=uicontrol("style","pushbutton","string","инверсия","Callback",...
"iminvers=iminvers*(-1)","Callback_Type",2,"position",[320, 50, 90, 30])
function [M, R, imakv]=akv()
   imakv=imfilter(imakv,M)
   imakv=imfilter(imakv,R)
   imshow(imakv)
endfunction
h=uicontrol("style","pushbutton","string","картинка","Callback",...
"imshow(imakv)","Callback_Type",2,"position",[410, 20, 90, 30])
h=uicontrol("style","pushbutton","string","акв1","Callback",...
"imakv=imfilter(imakv,M)","Callback_Type",2,"position",[410, 50, 90, 30])
h=uicontrol("style","pushbutton","string","акв2","Callback",...
"imakv=imfilter(imakv,R)","Callback_Type",2,"position",[410, 80, 90, 30])
```

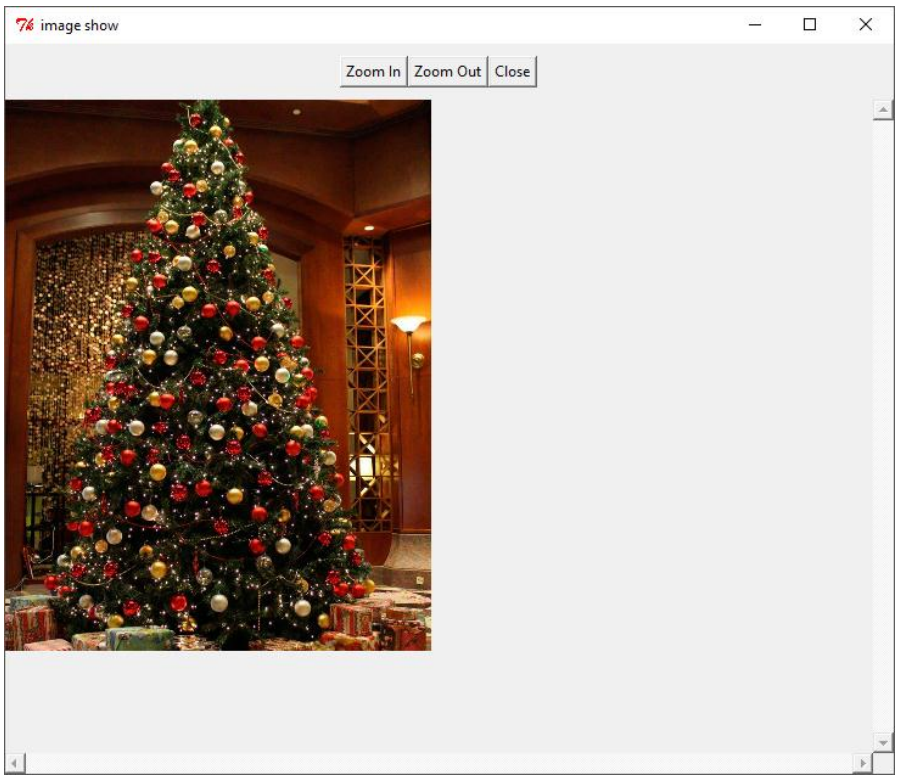

Рисунок 7 – Исходное изображение

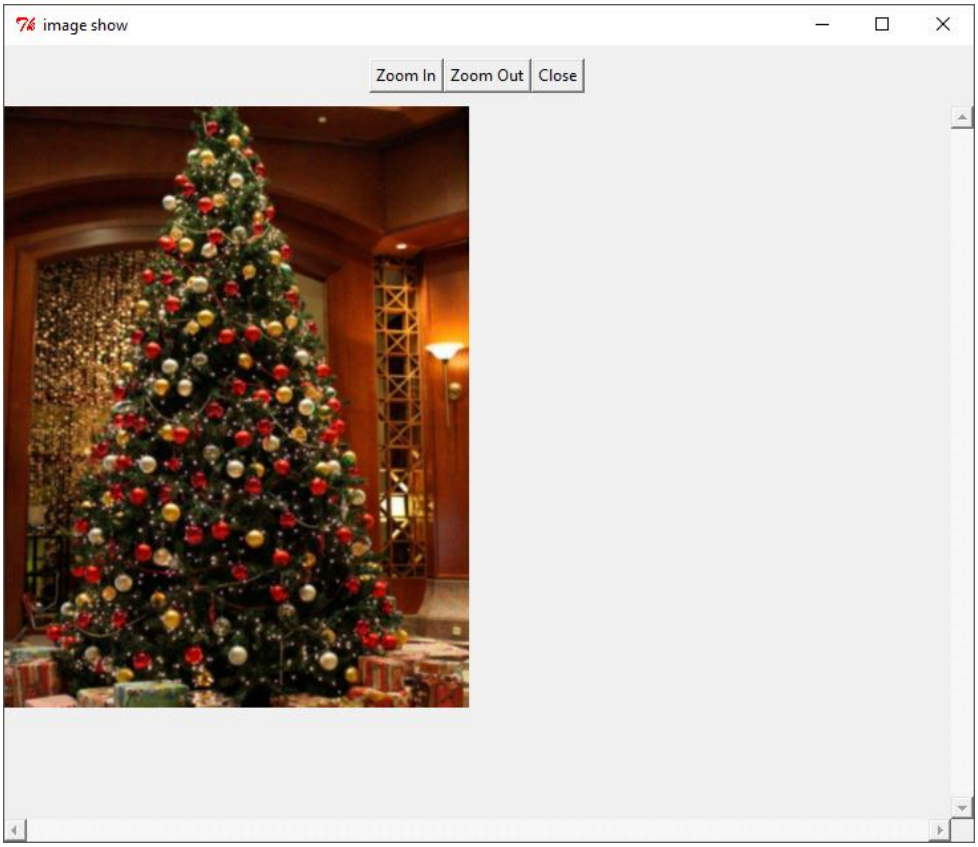

Рисунок 8 - Эффект размытия

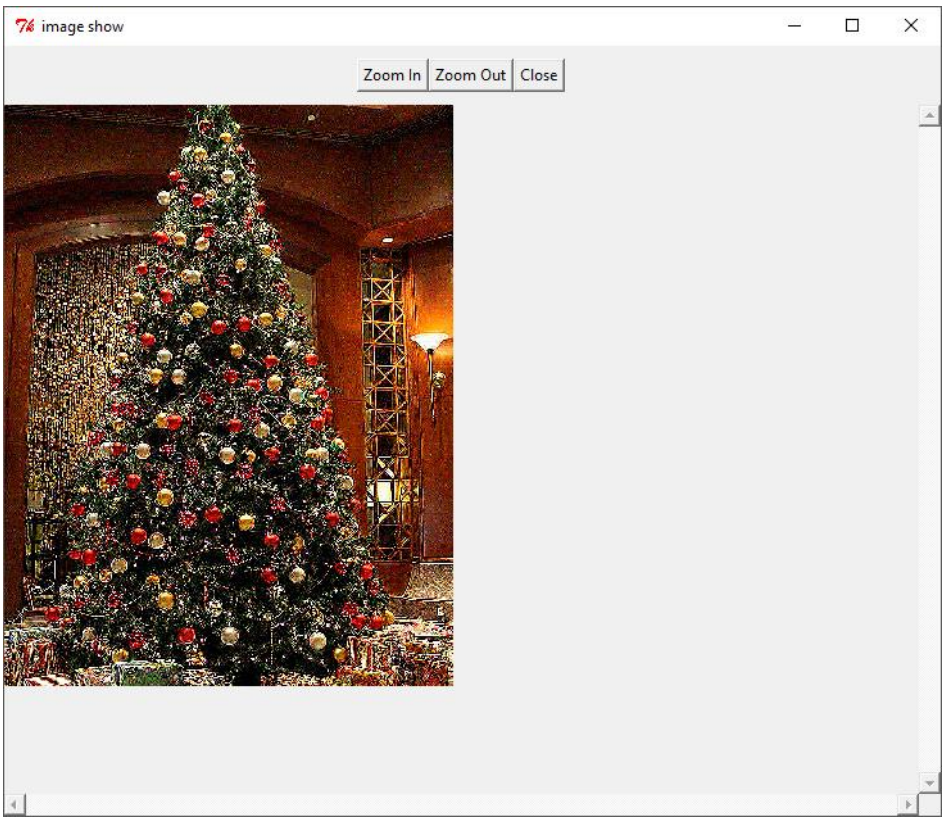

Рисунок 9 - Эффект увеличения резкости

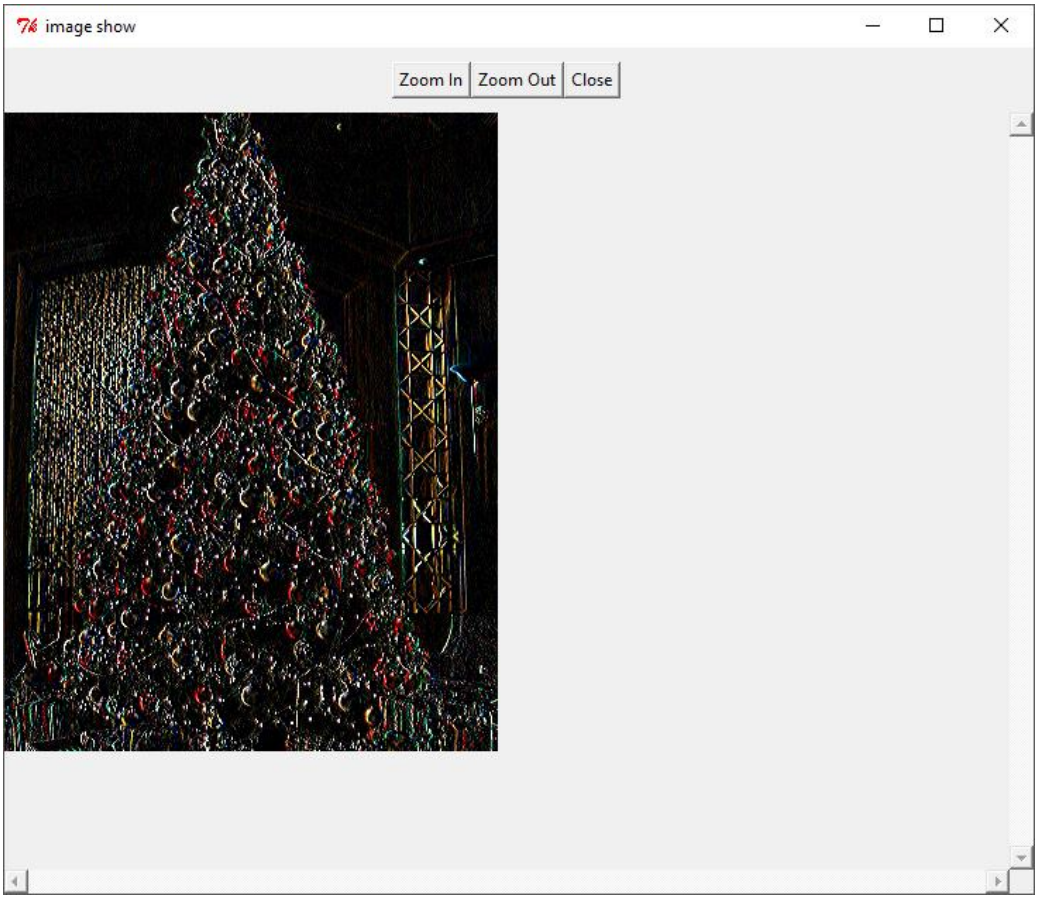

Рисунок 10 - Эффект тиснения

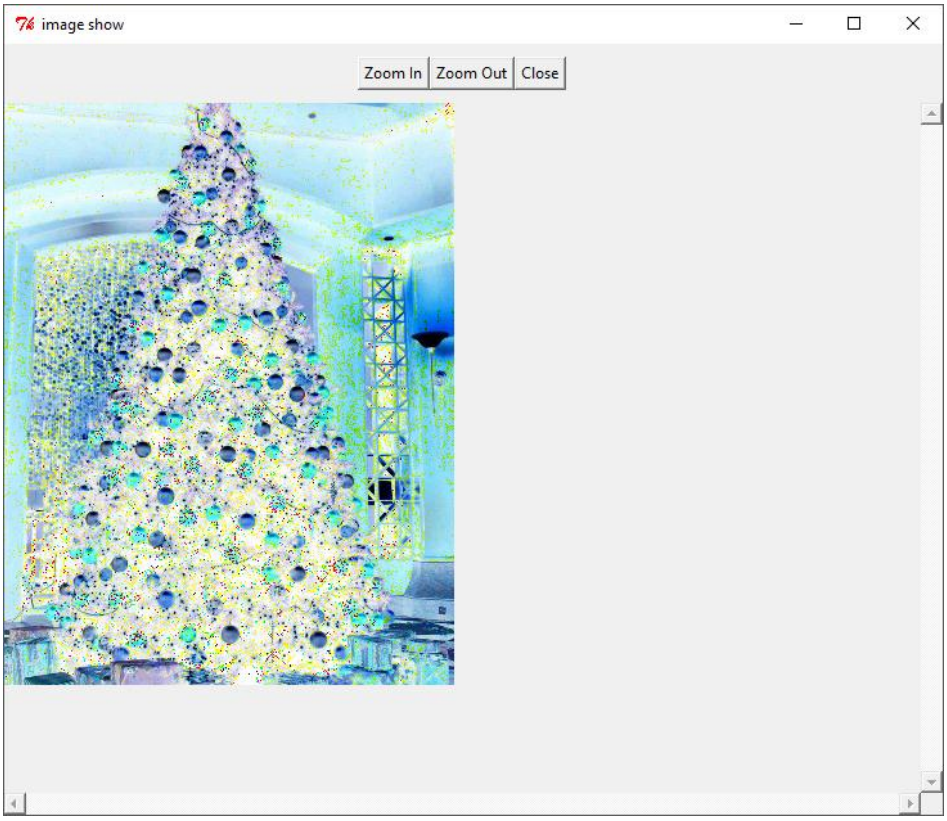

Рисунок 11 - Эффект инверсии

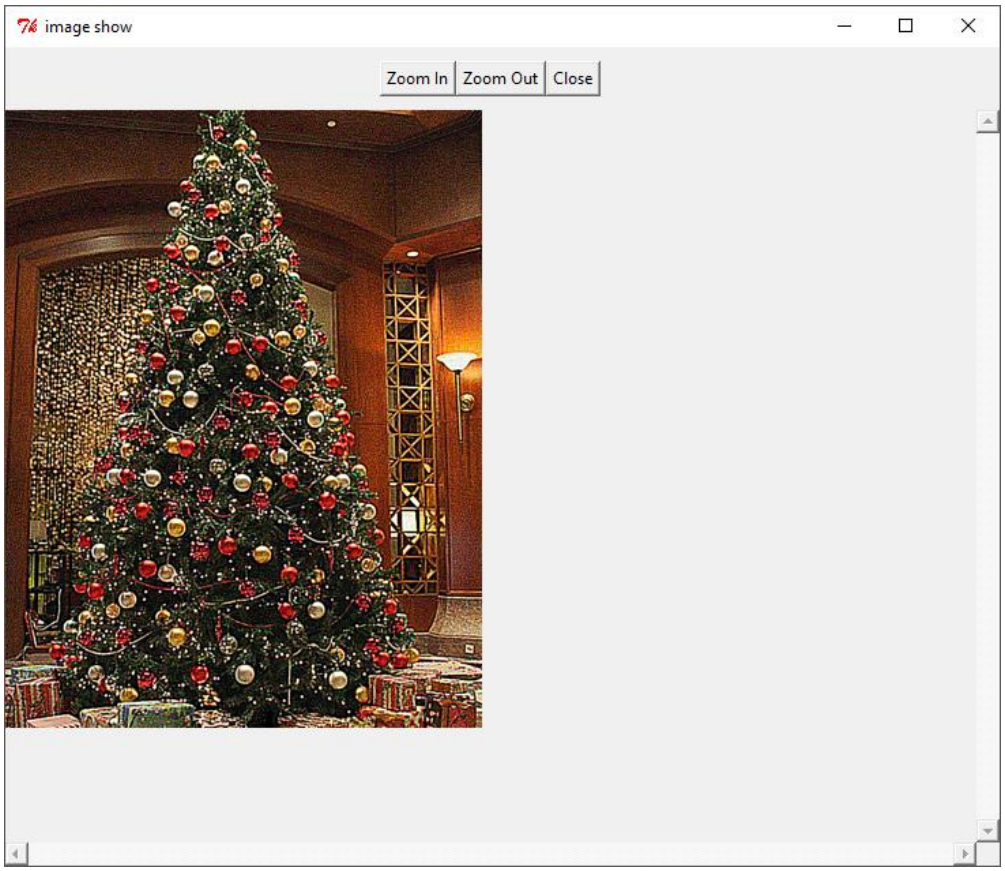

Рисунок 12 - Эффект акварелизации

Вывод: в ходе работы были получены навыки по разработке программы, которая использует графические фильтры : инверсия, размытие, акварелизация, тиснение, увеличение резкости в среде Scilab; была реализована программа для применения графических фильтров изображения в среде разработки Scilab.

Список использованных источников:

1. Шикин Е.В., Боресков А.В. Компьютерная графика. – М.: Диалог-МИФИ, 1996. 288 с.

2. Роджерс Д., Алгоритмические основы машинной графики. – М.: Мир, 1989, 512 с.