

Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный технический университет
имени П.О.Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине
«АЛГОРИТМИЧЕСКИЕ ОСНОВЫ СОВРЕМЕННОЙ КОМПЬЮТЕРНОЙ
ГРАФИКИ»
для студентов специальности
1-40 05 01 «Информационные системы и технологии (по направлениям)»
направление специальности 1–40 05 01–01 «Информационные системы и
технологии (в проектировании и производстве)»

Цитринов А.В.

СОДЕРЖАНИЕ

Введение	3
Лабораторная работа №1	4
Лабораторная работа №2	9
Лабораторная работа №3	12
Лабораторная работа №4	15
Лабораторная работа №5	17
Лабораторная работа №6	20
Список использованных источников	24
Приложение А	26
Приложение Б	27

ВВЕДЕНИЕ

Дисциплина «Алгоритмические основы современной компьютерной графики» дает будущему специалисту знания и умения, позволяющие в дальнейшем эффективно использовать их в практической работе. Полученные знания и навыки будут также востребованы при изучении специальных дисциплин и станут инструментом в реализации программных продуктов в конкретной предметной области при выполнении курсовых и дипломных работ. Дисциплина знакомит студентов с основами компьютерной графики и способами и алгоритмами отображения её на устройствах вывода.

Цель дисциплины – получение теоретических и практических знаний с использованием современных инструментов в области компьютерной графики, овладение практическими приемами работы с геометрическими объектами.

Задачи дисциплины – овладеть основами представления изображения в компьютерной графике. Изучить основы работы с двумерной и трёхмерной графикой, методы вывода изображения на экран. Приобрести навыки работы с алгоритмами отображения проекций трёхмерных объектов на плоскость, методами работы с цветом в задачах визуализации.

В результате изучения учебной дисциплины студент должен **знать:** области применения компьютерной графики; основные концепции и принципы отображения графической информации; технические средства поддержки компьютерной графики; программные средства поддержки компьютерной графики; цветовые модели, переход от одной модели к другой; алгоритмы геометрических преобразований, таких как параллельный перенос, масштабирование, вращение; **уметь:** работать с программными средствами представления компьютерной графики; использовать технологии обработки графической информации; применять языки программирования для отображения геометрических объектов; **владеть:** основами работы с двумерной и трёхмерной графикой; способами представления геометрической информации; алгоритмами отображения проекций трёхмерных объектов на плоскость; методами работы с цветом в задачах визуализации; принципами разработки программ для представления геометрической информации различными способами.

Настоящий лабораторный практикум рассчитан на 17 аудиторных часов и состоит из 6 лабораторных работ, по тематике охватывающих основную часть лекционного материала. В пособии для каждой лабораторной работы приведены задание, краткий теоретический материал, примеры результатов выполнения работы и список контрольных вопросов. В конце пособия приведены списки основной и дополнительной литературы, а так же, приложение А, с примером оформления титульного листа отчета по лабораторной работе, и приложение Б, в котором представлена структура отчета по лабораторной работе и требования по оформлению.

ЛАБОРАТОРНАЯ РАБОТА №1

«Реализация программ построения и преобразования двумерных объектов»

Цель работы: изучить геометрические преобразования двумерных объектов (перенос, масштабирование, поворот).

Задание.

Для выбранного варианта необходимо:

1. Создать матрицу ключевых точек для каждой фигуры, образующей замкнутый контур. По координатам ключевых точек построить изображение заданной фигуры, используя 2D график.
2. Провести перенос фигуры на расстояние вдоль горизонтали и вертикали. Построить изображение преобразованной фигуры.
3. Провести неоднородное и однородное масштабирование заданной фигуры. Построить изображение преобразованной фигуры.
4. Провести поворот заданной фигуры. Построить изображение преобразованной фигуры.

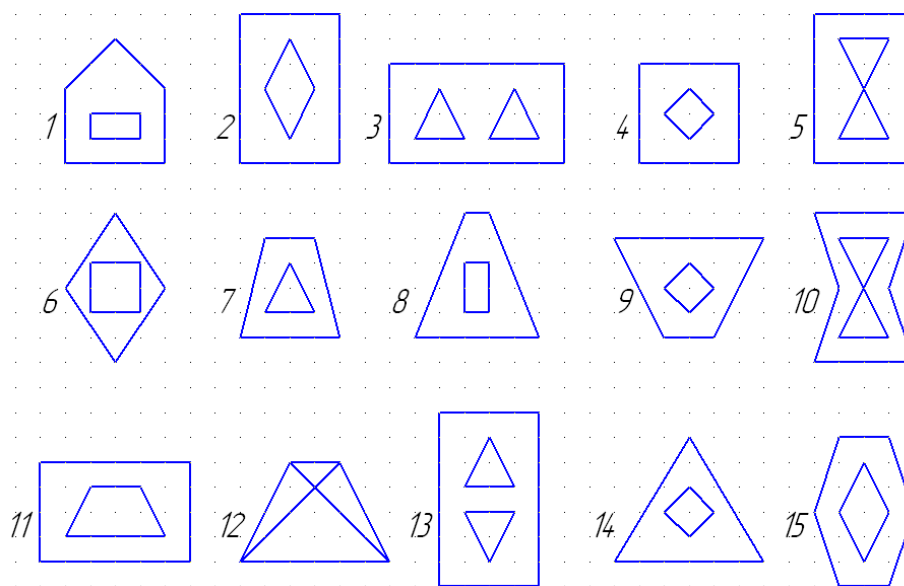


Рисунок 1.1 – Варианты заданий

Теоретические сведения.

Геометрические объекты на плоскости можно подвергать ряду различных преобразований. Наиболее употребительными в задачах компьютерной графики являются: перемещение (параллельный перенос); изменение размеров (масштабирование); повороты вокруг некоторой точки на плоскости (вращение). В далее будем отождествлять точки пространства с радиус-вектором, определяемым этой точкой.

Перенос

Параллельный перенос объекта сводится к перемещению всех его точек на одно и то же расстояние D в одном и том же направлении, заданном определённым вектором \vec{v} . Если вектор имеет длину D , то операция переноса может быть реализована путём сложения всех точек объекта с вектором \vec{v} . При такой операции сохраняются расстояния между точками и углы между отрезками, а также, отрезки прямых перейдут в отрезки прямых.

Математическая модель переноса: для каждой точки $P(x, y)$, которая перемещается в новую точку $P'(x', y')$, сдвигаясь на D_x единиц по оси X и на D_y по оси Y :

$$x' = x + D_x, y' = y + D_y.$$

Определим векторы-строки: $P = (x \ y)$, $P' = (x' \ y')$, $T = (D_x \ D_y)$, тогда уравнение переноса можно записать в виде: $(x' \ y') = (x \ y) + (D_x \ D_y)$, или в матричной форме: $P' = P + T$.

Масштабирование

Масштабирование объекта можно реализовать путём умножения координат всех его точек на некоторое число. Пусть имеются точки с координатами (x_1, y_1) и (x_2, y_2) , над которыми выполняется такое преобразование. Результатом будут новые точки с координатами $(S_x x_1, S_y y_1)$ и $(S_x x_2, S_y y_2)$. Если $S_x = S_y = S_0$, то обе точки переместятся вдоль прямых, проходящих через саму точку и начало координат. При этом расстояние между новыми точками будет в S_0 раз отличаться от прежнего, но углы между отрезками сохраняются. Если коэффициент масштабирования $S_0 > 1$, соответствующий отрезок растягивается, а если меньше, то сжимается. Кроме того, при таком преобразовании объект смещается. В случае, когда $S_x \neq S_y$, расстояния между точками изменятся неравномерно, поскольку растяжения в горизонтальном и вертикальном направлениях будут различными. Углы между отрезками также не сохраняются.

Математическая модель масштабирования при масштабировании в S_x раз по оси X и в S_y раз по оси Y , имеет вид:

$$x' = x S_x, y' = y S_y.$$

Определив матрицу масштабирования S в виде:

$$S = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix},$$

можно записать в матричной форме:

$$(x' \ y') = (x \ y) \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

или $P' = P S$.

Поворот

Вращения в плоскости перемещают точки по дуге окружности, центр которой находится в начале координат. Рассмотрим сначала движение одной точки при повороте на угол α (положительным является направление против часовой стрелки), т.е. поворот радиус-вектора на угол (рисунок 1.2). Пусть точка располагалась на расстоянии r от начала координат, а её радиус-вектор составлял угол β с осью абсцисс. Тогда координаты точки определяются формулами:

$$x = r \cos(\beta), y = r \sin(\beta)$$

После поворота вектор будет составлять угол $(\beta + \alpha)$, а новые координаты точки будут определяться соотношениями:

$$\begin{aligned} x' &= r \cos(\beta + \alpha) = r \cos(\beta) \cos(\alpha) - r \sin(\beta) \sin(\alpha) = x \cos(\alpha) - y \sin(\alpha), \\ y' &= r \sin(\beta + \alpha) = r \sin(\beta) \cos(\alpha) + r \cos(\beta) \sin(\alpha) = x \sin(\alpha) + y \cos(\alpha). \end{aligned}$$

При таком преобразовании сохраняются расстояния между точками и углы между отрезками.

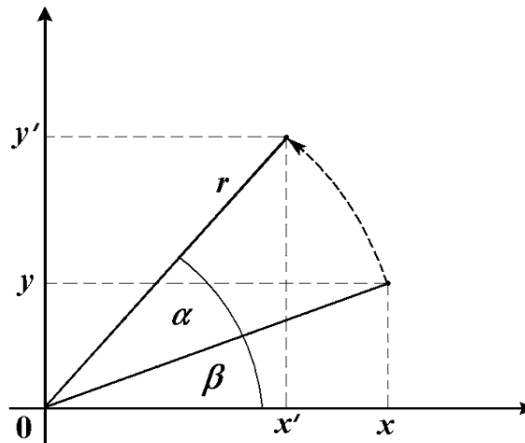


Рисунок 1.2 — Поворот на плоскости.

Преобразование поворота можно записать в матричной форме:

$$(x' y') = (x y) \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix},$$

или, определив матрицу поворота как:

$$R = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix},$$

в краткой форме соответствует записи:

$$P' = P R.$$

Примеры результатов выполнения лабораторной работы.

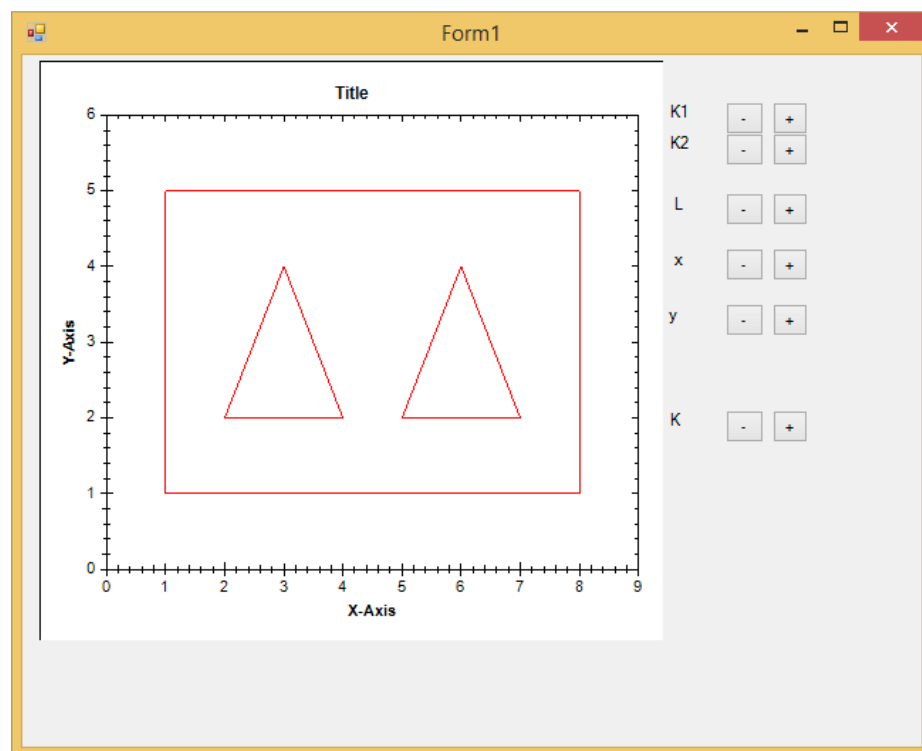


Рисунок 1.3 – Главное окно с исходным объектом

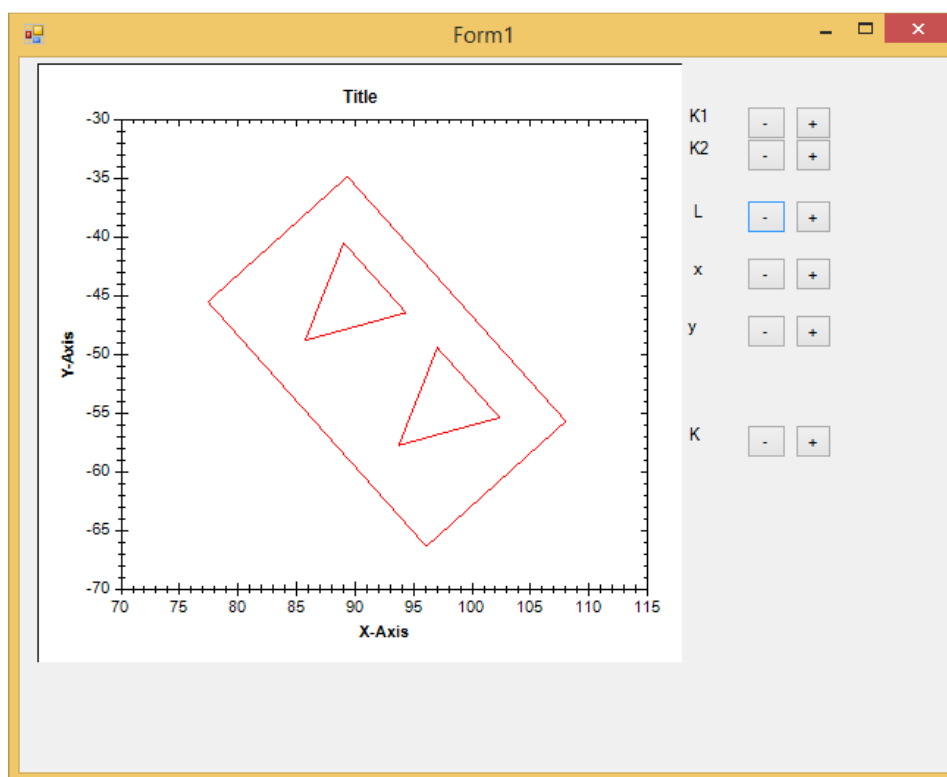


Рисунок 1.4 – Результат поворота, масштабирования и смещения по осям

Контрольные вопросы.

1. Какие способы задания координаты точки на плоскости вам известны?
2. Чем определяются декартовы координаты точки на плоскости?
3. Что такое геометрическое преобразование?
4. Как реализуется перенос фигуры на плоскости?
5. Какую операцию необходимо произвести с координатами точек фигуры для осуществления масштабирования фигуры?
6. Какие способы реализации поворота фигуры на плоскости вам известны?

ЛАБОРАТОРНАЯ РАБОТА №2

«Реализация программ построения и преобразования трехмерных объектов»

Цель работы: изучить геометрические преобразования трехмерных объектов.

Задание.

Для выбранного варианта необходимо:

1. Создать матрицу ключевых точек для каждой фигуры, образующей замкнутый контур. По координатам ключевых точек построить изображение заданной фигуры, используя 3D график.
2. Провести перенос фигуры на расстояние вдоль горизонтали и вертикали. Построить изображение преобразованной фигуры.
3. Провести неоднородное и однородное масштабирование заданной фигуры. Построить изображение преобразованной фигуры.
4. Провести поворот заданной фигуры. Построить изображение преобразованной фигуры.

Теоретические сведения.

Однородные координаты — координаты, обладающие тем свойством, что определяемый ими объект не меняется при умножении всех координат на одно и то же ненулевое число. Формирование суммарного преобразования сдвига, масштабирования и поворота в однородных координатах может быть представлено в виде произведения соответствующих матриц. Однородные координаты точки трехмерного пространства задаются в виде: (x, y, z, w) . Здесь w — произвольный множитель, не равный 0. Число w называется *масштабным множителем*. Для перехода от декартовой системы координат в трёхмерном пространстве каждой точке (x, y, z) будет сопоставляться точка четырёхмерного пространства $(x, y, z, 1)$.

Для выполнения преобразований можно воспользоваться следующими матрицами преобразований, умножив их на вектора-столбцы с координатами.

Матрица сдвига на вектор (D_x, D_y, D_z) :

$$P = \begin{pmatrix} 1 & 0 & 0 & D_x \\ 0 & 1 & 0 & D_y \\ 0 & 0 & 1 & D_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица масштабирования:

$$S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица поворота относительно соответствующей оси на угол α :

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_y = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_z = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Проекции точек на координатные плоскости:

$$P_{xy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad P_{yz} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad P_{zx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Примеры результатов выполнения лабораторной работы.

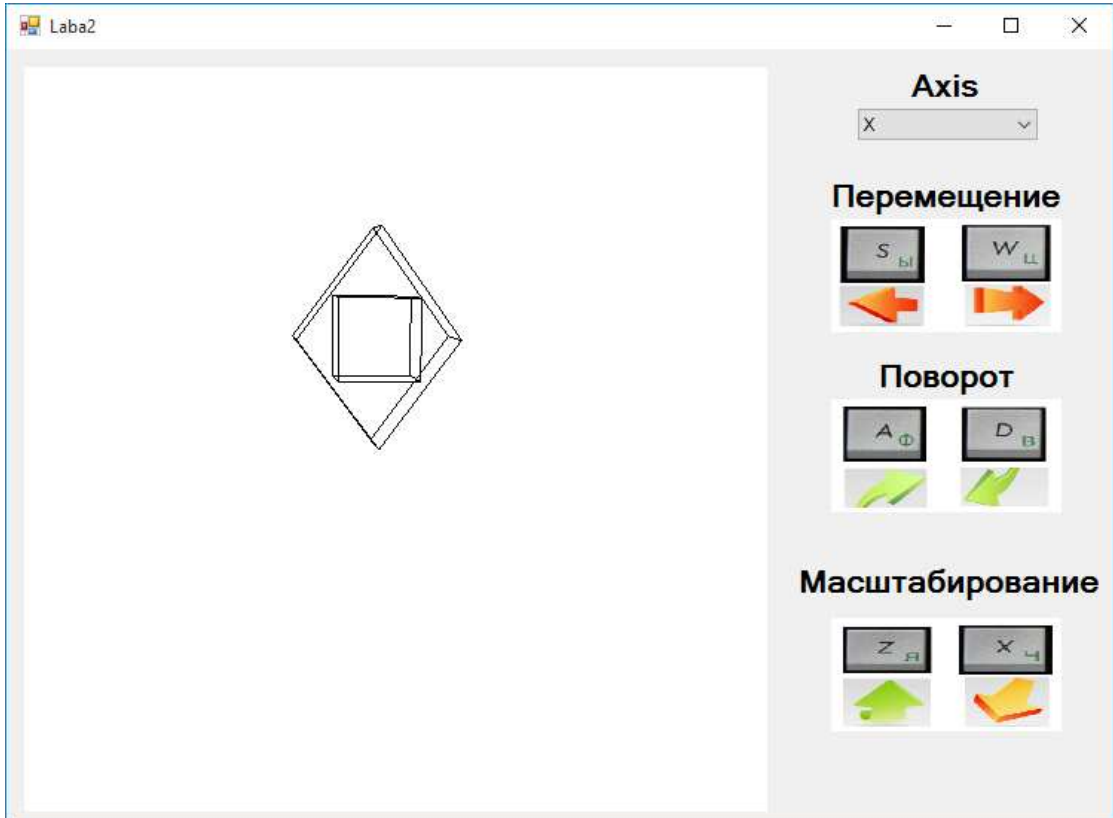


Рисунок 2. – Пример программы для преобразования 3D объектов

Контрольные вопросы.

1. В чем особенность однородных координат точки в пространстве?
2. Дайте определение понятия «аффинное преобразование».
3. Какой вид геометрических преобразований реализует операцию проецирования?
4. Какая операция над векторами используется для выполнения преобразования масштабирования?
5. Какие геометрические преобразования необходимо выполнить для того, чтобы сменить систему координат?
6. Что такое параллельная проекция.
7. Дайте определение центральной проекции.

ЛАБОРАТОРНАЯ РАБОТА №3

«Реализация программ отсечения»

Цель работы: получить теоретические и практические знания выполнения сечений геометрических двумерных фигур.

Задание.

Разработать программу, выполняющую рассечение заданной геометрической фигуры на части прямой линией, задаваемой пользователем в виде отрезка.

Теоретические сведения.

Для решения задачи необходимо: найти коэффициенты уравнения прямой, которая пересекает фигуру; найти точки пересечения этой прямой с прямыми, заданными сторонами фигуры, решая соответствующие системы уравнений; отобразить оставшуюся после обрезки часть фигуры.

Для построения линейной функции используется следующая формула:

$$y = kx + b ,$$

где k – угловой коэффициент $k = \frac{y_2 - y_1}{x_2 - x_1}$,

b – свободный коэффициент $b = \frac{x_2 y_1 - y_2 x_1}{x_2 - x_1}$

Для нахождения точки пересечения двух прямых с координатами (x, y) необходимо решить систему линейных уравнений вида:

$$\begin{cases} A_1 x + B_1 y + C_1 = 0 \\ A_2 x + B_2 y + C_2 = 0 \end{cases}.$$

Однако, решение этой системы можно найти в общем виде и в программе использовать готовое решение, пользуясь формулами:

$$x = -\frac{C_1 B_2 - C_2 B_1}{A_1 B_2 - A_2 B_1} ,$$

$$y = -\frac{A_1 C_2 - A_2 C_1}{A_1 B_2 - A_2 B_1} .$$

Примеры результатов выполнения лабораторной работы.

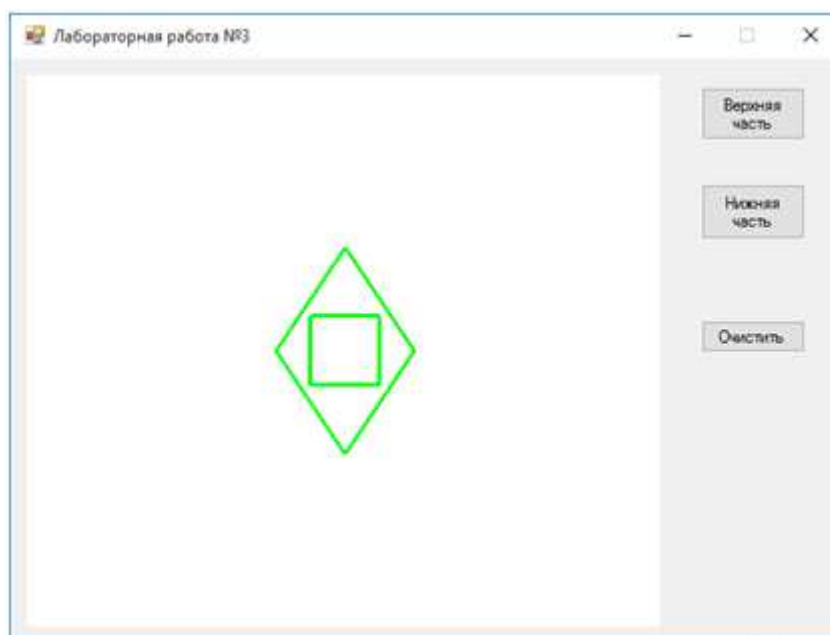


Рисунок 3.1 – Первоначальная форма фигуры

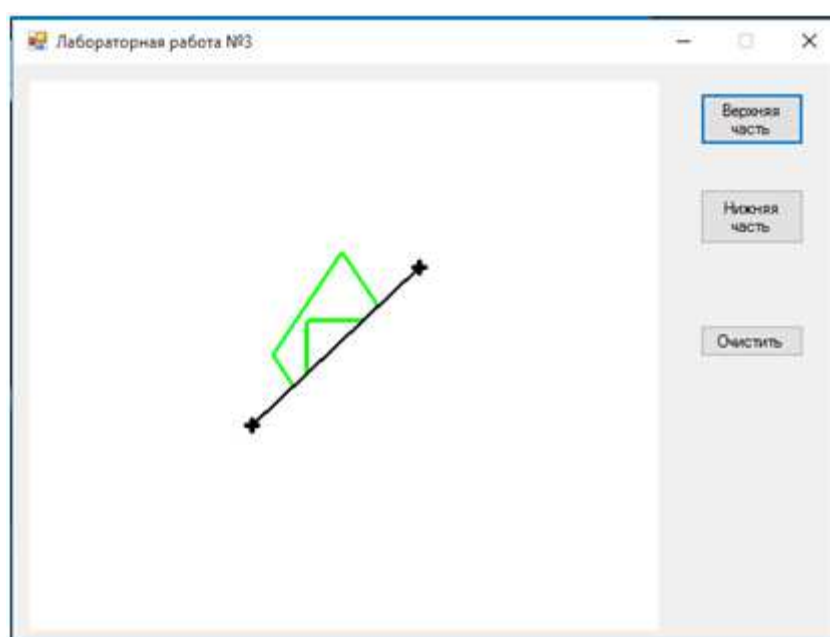


Рисунок 3.2 – Верхняя часть рассеченной фигуры

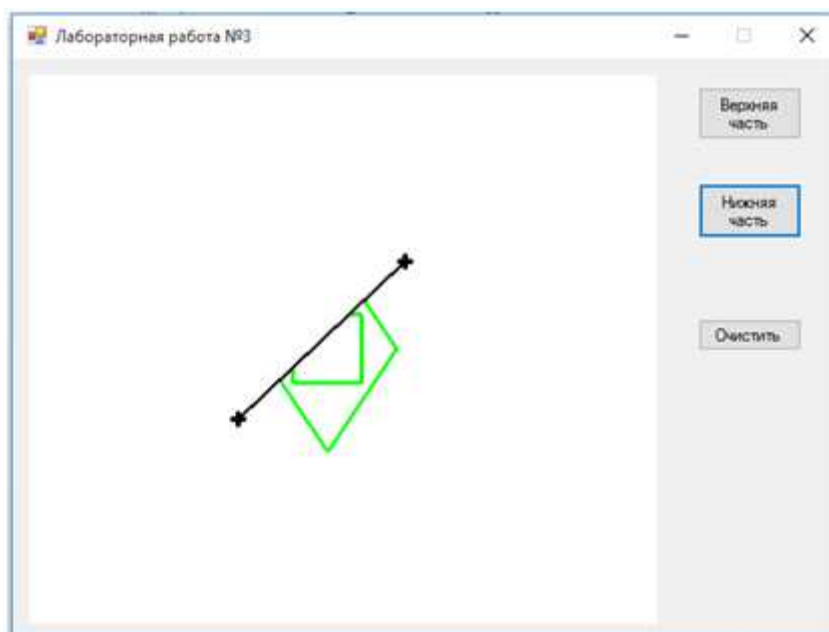


Рисунок 3.3 – Нижняя часть рассеченной фигуры

Контрольные вопросы.

1. Какие способы задания уравнения прямой на плоскости Вам известны?
2. Каким образом можно найти точку пересечения двух прямых на плоскости?
3. Для чего применяется алгоритм Сазерленда-Кодона и в чем его суть?
4. В чем заключается суть алгоритма отсечения выпуклым многоугольником?
5. Что такое клиппирование?

ЛАБОРАТОРНАЯ РАБОТА №4

«Реализация программ удаления невидимых линий и поверхностей»

Цель работы: получить теоретические и практические знания выполнения удаления невидимых поверхностей и линий.

Задание

Разработать программу выполняющую удаление невидимых поверхностей и линий, основанную на применении алгоритма Z-буфера.

Теоретические сведения.

Алгоритм Z-буфера:

1. Заполнить буфер кадра фоновым значением цвета.
2. Заполнить Z-буфер минимальным значением z (глубины).
3. Преобразовать изображаемые объекты в растровую форму в произвольном порядке.
4. Для каждого объекта:
 - 4.1 Для каждого пикселя (x, y) образа вычислить его глубину $z(x, y)$.
 - 4.2 Сравнить глубину $z(x, y)$ со значением глубины, хранящимся в Z-буфере в этой же позиции.
 - 4.3 Если $z(x, y) > Z\text{-буфер}(x, y)$, то занести атрибуты пикселя в буфер кадра и заменить Z-буфер (x, y) на $z(x, y)$. В противном случае никаких действий не производить.

Примеры результатов выполнения лабораторной работы.

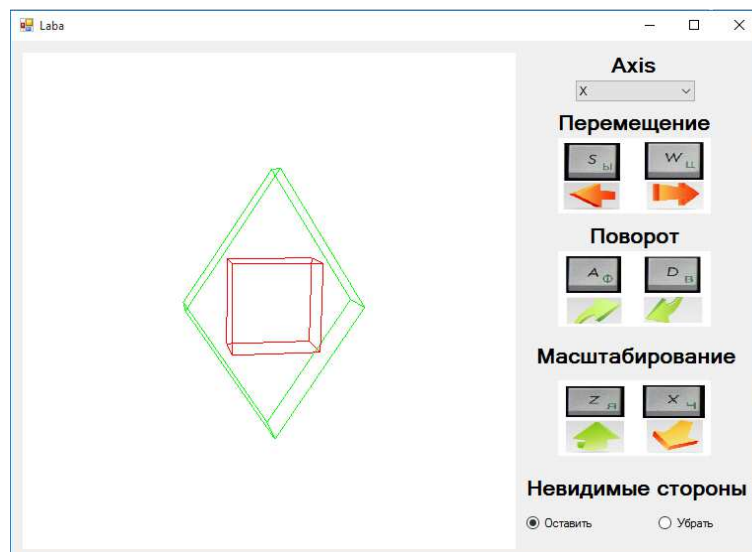


Рисунок 4.1 – Исходная фигура со всеми гранями

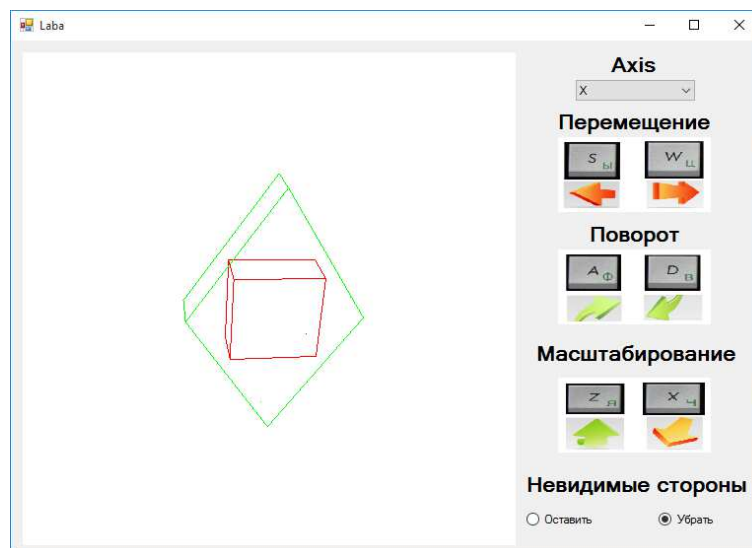


Рисунок 4.2 – Фигура без внутренних граней

Контрольные вопросы.

1. Сформулируйте основную идею алгоритма Робертса.
2. Какими достоинствами и недостатками обладает метод Z-буфера?
3. С какой целью выполняют удаление невидимых линий и поверхностей?

ЛАБОРАТОРНАЯ РАБОТА №5

«Реализация программ освещения и закраски»

Цель работы: получить теоретические и практические знания по созданию моделей освещенности.

Задание

Разработать программу, создающую простую модель освещения.

Теоретические сведения.

Простая модель освещения учитывает только отражение света. Свойства отраженного света зависят главным образом от направления лучей и характеристик отражающей поверхности.

Отражение может быть двух видов: диффузное и зеркальное. Первое из них возникает в ситуации, когда свет как бы проникает под поверхность объекта, поглощается, а потом равномерно излучается во всех направлениях. Поверхность в этом случае рассматривается как идеальный рассеиватель. При этом возникает эффект матового света, а видимая освещённость того или иного участка поверхности не зависит от положения наблюдателя. Зеркальное отражение, наоборот, происходит от внешней поверхности, интенсивность его неоднородна, поэтому видимый максимум освещенности зависит от положения глаза наблюдателя.

Свет точечного источника отражается от поверхности рассеивателя по закону Ламберта: интенсивность отражения пропорциональна косинусу угла между внешней нормалью к поверхности и направлением к источнику света.

Если I_S — интенсивность источника света, ϕ — угол между вектором внешней нормали к поверхности и направлением к источнику света, то интенсивность отраженного света определяется формулой:

$$I = \begin{cases} I_S \cos(\phi) & \text{при } 0 \leq \phi \leq \frac{\pi}{2} \\ 0 & \text{— в противном случае} \end{cases}.$$

При таком расчёте интенсивности получится очень контрастная картина, т.к. участки поверхности, на которые лучи от источника не попадают напрямую, останутся абсолютно черными. Для повышения реалистичности необходимо учитывать рассеивание света в окружающем пространстве. Поэтому вводится фоновая освещенность, зависящая от интенсивности рассеянного света I_F , и интенсивность отраженного света определяется выражением:

$$I = \begin{cases} I_F k_F + k_S I_S \cos(\phi) & \text{при } 0 \leq \phi \leq \frac{\pi}{2}, \\ I_F k_F & \text{— в противном случае} \end{cases},$$

где k_F — коэффициент диффузного отражения рассеянного света, $0 \leq k_F \leq 1$;
 k_S — коэффициент диффузного отражения падающего света, $0 \leq k_S \leq 1$.

В описанной модели пока никак не учитывалась удалённость источника света от поверхности, поэтому по освещённости двух объектов нельзя судить об их взаимном расположении в пространстве. Если нужно получить перспективное изображение, то необходимо включить затухание интенсивности с расстоянием. Обычно интенсивность света обратно пропорциональна квадрату расстояния от источника. В качестве расстояния до источника в случае перспективного преобразования можно взять расстояние до центра проекции, и если он достаточно удалён, то изображение будет достаточно адекватным. Но если этот центр расположен близко к объекту, то квадрат расстояния меняется очень быстро, и в этом случае лучше использовать линейное затухание. Тогда интенсивность отражённого света от непосредственно освещённых участков поверхности будет задаваться формулой:

$$I = I_F k_F + \frac{k_S I_S \cos(\phi)}{d + C},$$

где d — расстояние от центра проекции;

C — произвольная постоянная.

Более подробное описание модели освещения представлено в Главе 9 конспекта лекций по курсу «Алгоритмические основы современной компьютерной графики».

Примеры результатов выполнения лабораторной работы.

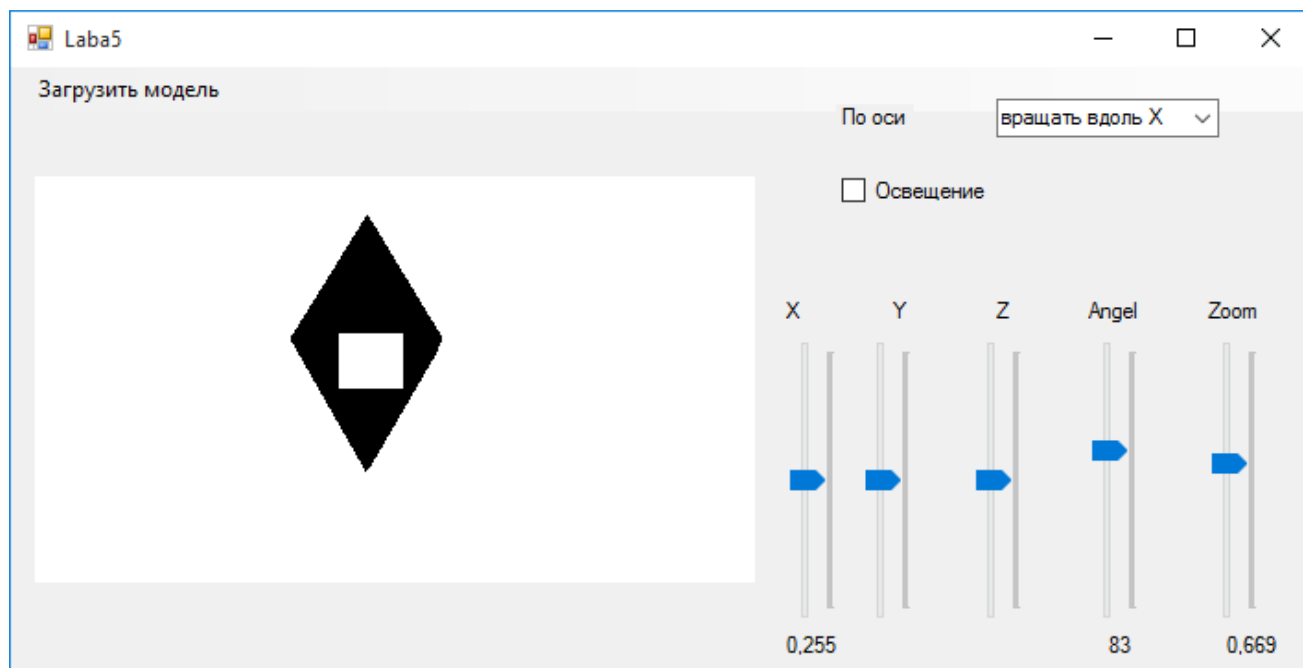


Рисунок 5.1 – Вид фигуры без освещения

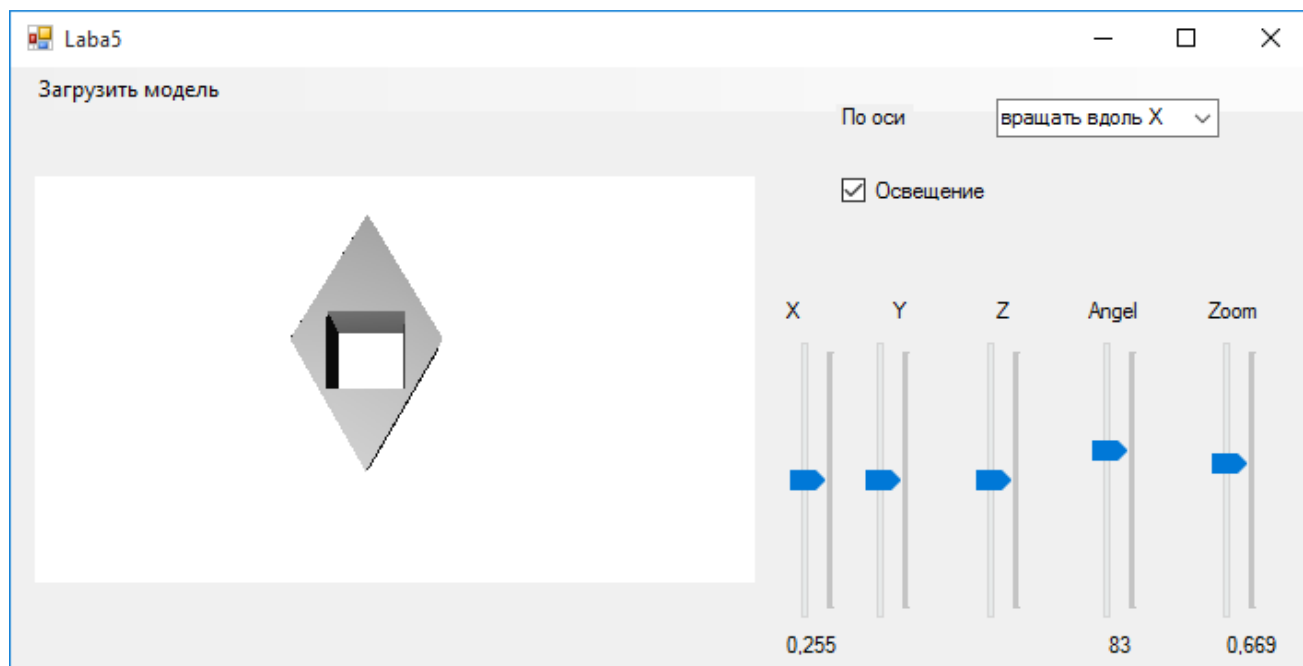


Рисунок 5.2 – Вид фигуры с освещением

Контрольные вопросы.

1. Что включает в себя простая модель освещения?
2. Чем отличается диффузное отражение от зеркального?
3. От чего зависит интенсивность освещения точки поверхности при диффузном отражении?
4. От чего зависит интенсивность освещения точки поверхности при зеркальном отражении?

ЛАБОРАТОРНАЯ РАБОТА №6

«Реализация программ для применения графических фильтров»

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 размытия 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 6.1 – 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 6.2 – 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 6.3 – 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 6.4 – 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 6.5 – 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 6.6 – Матрица для фильтра «Акварельный эффект»: шаг второй

Примеры результатов выполнения лабораторной работы.

Примеры применения рассмотренных графических фильтров приведены на рисунке 6.7.

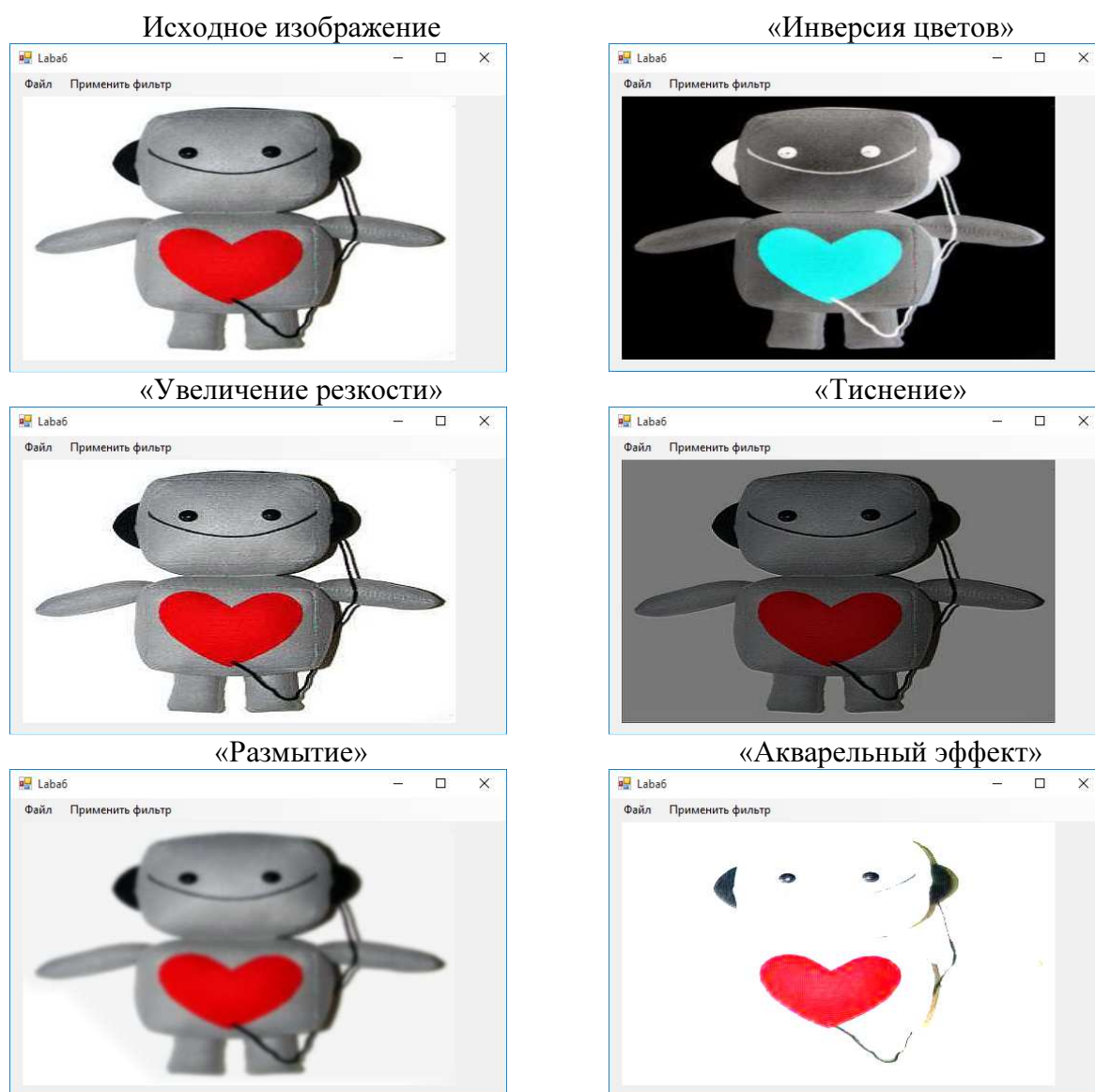


Рисунок 6.7 – Примеры применения графических фильтров

Контрольные вопросы.

1. Для чего применяются графические фильтры?
2. Какие графические фильтры Вам известны и как они реализовываются?
3. Что такое цветовой график МКО?
4. Дайте определение координат цветности.
5. Для чего используется опорный белый цвет на цветовом графике МКО?
6. Чем являются цветовые модели RGB и CMY по отношению друг к другу?
7. Перечислите основные недостатки цветовых моделей RGB и CMY.

Список использованных источников

Основная литература

1. Красильников Н.Н., Цифровая обработка 2D- и 3D-изображений: учебное пособие. – Спб.: БХВ-Петербург, 2011, 608с.
2. Роджерс Д., Алгоритмические основы машинной графики. – М.: Мир, 1989, 512 с.
3. Шикин Е.В., Боресков А.В. Компьютерная графика. – М.: Диалог-МИФИ, 1996. 288 с.
4. Абламейко С.В., Лагуновский Д.М. Обработка изображений: технология, методы, применение. – Мн.: Амалфея, 2000, 304 с.

Дополнительная литература

1. Bresenham, J.E. A Linear Algorithm for Incremental Digital Display of Circular Arcs / J.E. Bresenham // Communication of the ACM. – 1977. – V. 20(2). – P. 100–106.
2. Danielson P.E. Incremental curve generation / P.E. Danielson // IEEE Transaction on Computers. – 1970. – V. 19. – P. 23–28.
3. Suenaga, Yasuhito. A High Speed Algorithm for the Circular Arcs / Yasuhito Suenaga, Takahiko Kamae, Tomonori Kabayashi // IEEE Trans. on Computers. – 1979. – V. 28(10). – P. 728–736.
4. Pitteway, M.L.V. Integer circles etc. – Three move of Bresenham's algorithm / M.L.V. Pitteway, R.J. Botting // Computer Graphics and Image Processing. – 1974. – V. 3. – P. 260–261.
5. Horn, B.K.P. Circle generators for display devices / B.K.P. Horn // Computer Graphics, and Image Processing. – 1976. – V. 5. – P. 580–588.
6. Badler, N.I. Disk generators for a raster display device / N.I. Badler // Computer, Graphics and Image Processing. – 1977. – V. 6. – P. 589–593.
7. Chung W.L. On circle generation algorithms / W.L. Chung // Computer Graphics and Image Processing. – 1977. – V. 6. – P. 196–198.
8. Doros, M. Algorithms for Generation of Discrete Circles, Rings and Disks / M. Doros // Computer Graphics and Image Processing. – 1979. – V. 10(4). – P. 366–371.
9. Pitteway M. Bresenham's Algorithm with Gray Scale / M. Pitteway, D. Watkinson // Communication of the ACM. – 1980. – V. 23(11). – P. 625–626.
10. Moller G. Fast digital vector and circle generator with binary rate multipliers / G. Moller // Computer Graphics. – 1978. – V.12(4). – P. 81–91.
11. McCrea P.G. On DDA circle generation for computer graphics / P.G. McCrea, P.W. Baker // IEEE Trans. on Computers. – 1975. – V. 24. – P. 1109–1110.
12. Maxwell, P.C. The Generation of Polygons Representing Circles, Ellipses and Hyperbolas / P.C. Maxwell, P.W. Baker // Computer Graphics and Image Processing. – 1979. – V. 10. – P. 84–93.
13. Foley, James D. Computer graphics – principles and practice / James D. Foley [et al] // ADDISON-WESLEY PUBLISHING COMPANY. 1990.

14. Божко, А.Н. Компьютерная графика : учеб. пособие для вузов / А.Н. Божко, Д.М. Жук, В.Б. Маничев. – М. : Изд-во МГТУ, 2007.
15. Фоли Дж. Основы интерактивной машинной графики / Дж. Фоли, А. ван Дам. В 2 т. – М. : Мир, 1985.
16. Шикин, Е.В. Компьютерная графика. Динамика, реалистические изображения / Е.В. Шикин, А.В. Боресков. – М. : ДИАЛОГ-МИФИ, 1995.
17. Шикин Е.В. Кривые и поверхности на экране компьютера: Рук. по сплайнам для пользователей / Е.В. Шикин, А.И. Плис. – М. : Диалог-МИФИ, 1996.
18. Шикин, Е.В. Начала компьютерной графики / Е.В. Шикин, А.В. Боресков, А.А. Зайцев. – М. : Диалог-МИФИ, 1993.
19. Чириков, С.В. Алгоритмы компьютерной графики (методы растривования кривых) : учеб. пособие / С.В. Чириков. – СПб. : СПб ГИТМО(ТУ), 2001.
20. Джамбруно, М. Трехмерная графика и анимация : пер. с англ. / М. Джамбруно. – 2-е изд. – М. : Вильямс, 2002.
21. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики / Е.А.Никулин. – СПб. : БХВ-Петербург, 2003.

ПРИЛОЖЕНИЕ А

Пример оформления титульного листа отчета по лабораторной работе

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**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,
Иванов И.И.

Принял: доцент каф. «ИТ»,
к.ф.-м.н., доц.,
Цитринов А.В.

Гомель 2015

ПРИЛОЖЕНИЕ Б

Структура отчета по лабораторной работе

Текст отчета по лабораторной работе должен быть оформлен в соответствии с ГОСТ 2.105 и содержать следующие структурные части:

1. Титульный лист, оформленный в соответствии с приложением А.
2. Основная часть, включающая в себя:
 - 2.1 Цель работы;
 - 2.2 Теоретические сведения;
 - 2.3 Текст разработанной программы и рисунки-примеры ее работы.
3. Вывод, в котором представлен анализ реализованных алгоритмов с указанием их достоинств и недостатков, по сравнению с другими алгоритмами.
4. Список использованных источников, оформленный по ГОСТ 7.0.5-2008.