МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

ПРЕДСТАВЛЕНИЕ И ПРОЕЦИРОВАНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ

Отчет по лабораторной работе № 11 дисциплины
«Компьютерная графика»

Выполнил студент группы ИВТ-21 ______________________________/Рзаев А.Э./
Проверил старший преподаватель _____________________________/Вожегов Д.В./

2016 г.

1 Постановка задачи

1. Описать брусок в приборной системе координат.
2. Вывести на экран три его ортогональные проекции (вид спереди, сверху, сбоку).
3. Продемонстрировать три прямоугольные аксонометрические проекции данного бруска (изометрию, диметрию, триметрию).
4. Построить две косоугольные аксонометрические проекции бруска (кавалье, кабине).
5. Показать одноточечную центральную проекцию бруска.

2 Краткие теоретические сведения

Изображение объектов на экране связано с такой геометрической операцией, как проецирование. В компьютерной графике используют в основном параллельное и центральное проецирование прямыми лучами на плоскость. Параллельное проецирование предполагает наличие:

- центра проецирования в бесконечности,
- вектора проецирования и проецирующих лучей, параллельных данному вектору,
- проецирующей (картинной) плоскости.

При центральном проецировании явно задаются:

- координаты точки - центра проецирования,
- картинная плоскость.

Центральное проецирование порождает визуальный эффект, аналогичный зрительному - перспективное укорачивание: с увеличением расстояния от центра проекции до объекта его размеры уменьшаются. Поэтому оно используется для создания реальных картин, но непригодно для представления точной формы и размеров объектов проецирования, необходимое, например, в чертежных задачах конструкторской графики.

Параллельное проецирование дает менее реалистичное изображение (нет перспективного укорачивания), но предоставляет пользователю истинные с точностью до скалярного множителя размеры.

Для описания преобразований проецирования используются однородные координаты и матрицы четвертого порядка, что упрощает изложение материала и облегчает решение задач геометрического моделирования.

Матрицы проецирования

1. Одноточечные (имеющие одну точку схода) центральные проекции характеризуются следующим: плоскость проекции совпадает с координатной Z=0, центр проекции имеет координаты (0,0,-d). Матрица проецирования имеет вид

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Для получения проекции точки в пространстве с координатами (x, y, z, 1) необходимо найти ее новые однородные, а затем - новые координаты (x’, y’) так

$$[xo \quad yo \quad zo \quad 1] = [x \quad y \quad z \quad 1] * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \quad y \quad 0 \quad z/d+1]$$

$$x' = x/(z/d+1)$$

$$y' = y/(z/d+1)$$

2. Ортографические (вид спереди, сверху, сбоку) параллельные проекции: картинная плоскость совпадает с одной из координатных, направление проецирования - с одной из главных осей.

Матрица проецирования вдоль оси X на плоскость YOZ $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

При построении вида сбоку x – координату точки проекции заменяют координатой z, y - координата остается без изменения.

Вдоль оси Y на плоскость XOZ: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

При построении вида сверху y – координату точки проекции заменяют координатой z. x - координата остается без изменения.

Вдоль оси Z на XOY: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

При построении вида спереди координаты z точек проекции отбрасываются.

3. Аксонометрические прямоугольные параллельные проекции – проецирующие прямые перпендикулярны картинной плоскости, которая не совпадает (не параллельна) ни с одной из координатных плоскостей.

Общий вид матрицы таких проекций: $\begin{bmatrix} \cos p & \sin f * \sin p & 0 & 0 \\ 0 & \cos f & 0 & 0 \\ \sin p & \sin f * \cos p & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

где p и f - углы, которые нормаль к картинной плоскости образует с ортами координатных осей (соответственно OY и OX).

Для построения стандартной изометрии следует взять p равным 45, f - 35.264 градусам. Для стандартной диметрии: p=22.208, f=20.705 градусов. При других значениях углов получается триметрию. Значения углов в матрицу подставляются в радианах.

4. Косоугольная параллельная аксонометрия - проекторы не перпендикулярны картинной плоскости, которая совпадает (параллельна) с одной из координатных плоскостей. Самые простые и наглядные из косоугольных - фронтальные проекции (картинная плоскость параллельна XOZ). Из них - косоугольная фронтальная диметрия (кабине) и косоугольная фронтальная изометрия (кавалье). Матрицы для этих двух проекций выглядят так:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ l * \cos \pi / 4 & l * \sin \pi / 4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где l = 1 для кавалье и 0.5 для кабине.

Для получения координат проекции любой точки изображения необходимо исходные координаты этой точки перемножить с соответствующей матрицей. Например, для получения проекции куба на экране, необходимо найти новые координаты восьми точек - вершин куба, затем соединить их отрезками в определенной последовательности. Процедура нахождения новых координат проекции кавалье, например, будет выглядеть так:

$$\begin{bmatrix} z' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} z & y & z & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ l \cos \pi / 4 & l \sin \pi / 4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x + z * l \cos \pi / 4 & y + z * l \sin \pi / 4 & 0 & 1 \end{bmatrix}$$

$$x' = x + z * l * 0.707$$

$$y' = y + z * l * 0.707$$

3 Вывод

В ходе данной лабораторной работы был изучен математический аппарат для визуализации объемных геометрических тел. Также были изучены различные способы проецирования объемных тел на некоторую плоскость. С помощью всего вышеперечисленного была написана программа, позволяющая строить различные проекции трехмерного тела на экран компьютера. Листинг и экранные формы приведены в приложениях А и Б.

Приложение А
(обязательное)
Листинг кода программы

```
from PyQt5 import QtCore, QtGui, QtWidgets, uic
import sys


class Window(QtWidgets.QWidget):
    def __init__(self, parent=None):
        super(Window, self).__init__(parent)
        uic.loadUi('window.ui', self)

        self.graphicsViewOrtho.setScene(QtWidgets.QGraphicsScene(self))
        self.graphicsViewAx.setScene(QtWidgets.QGraphicsScene(self))
        self.graphicsViewObliq.setScene(QtWidgets.QGraphicsScene(self))

        self.cube = ((-50, -50, 0), (-50, -50, 100), (50, -50, 100), (50, -50, 0),
                     (-50, 50, 0), (-50, 50, 100), (50, 50, 100), (50, 50, 0))

        # self.projOneDot.setVisible(False)
        self.projOneDot.clicked.connect(self.draw_onedot)

        self.projYOZButton.clicked.connect(self.draw_ortho_yoz)
        self.projXOZButton.clicked.connect(self.draw_ortho_xoz)
        self.projXOYButton.clicked.connect(self.draw_ortho_xoy)

        self.projIsoButton.clicked.connect(lambda: self.draw_axinomo(45, 35.264))
        self.projDiButton.clicked.connect(lambda: self.draw_axinomo(22.208, 20.705))
        self.projTriButton.clicked.connect(self.draw_trimetry)

        self.projCavButton.clicked.connect(lambda: self.draw_oblique(1))
        self.projCabButton.clicked.connect(lambda: self.draw_oblique(0.5))

    def draw_onedot(self):
        d = -200
        projection = [(x / (z / d + 1), y / (z / d + 1)) for x, y, z in self.cube]
        self.draw_cube(projection, self.graphicsViewOrtho)

    def draw_ortho_yoz(self):
        projection = [(z, y) for _, y, z in self.cube]
        self.draw_cube(projection, self.graphicsViewOrtho)

    def draw_ortho_xoz(self):
        projection = [(x, z) for x, _, z in self.cube]
        self.draw_cube(projection, self.graphicsViewOrtho)

    def draw_ortho_xoy(self):
        projection = [(x, y) for x, y, _ in self.cube]
        self.draw_cube(projection, self.graphicsViewOrtho)

    def draw_trimetry(self):
        p = float(self.lineEditP.text())
        f = float(self.lineEditF.text())
        self.draw_axinomo(p, f)

    def draw_axinomo(self, p, f):
        from math import sin, cos
        p = p * 3.1415 / 180
        f = f * 3.1415 / 180
        projection = [(x * cos(p) - y * cos(f),
                       x * sin(p) + y * sin(f) + z) for x, y, z in self.cube]
        self.draw_cube(projection, self.graphicsViewAx)
```

```
    def draw_oblique(self, l):
        from math import sin, cos, pi
        projection = [(x + z * l * cos(pi/4), y + z * l * sin(pi/4)) for x, y, z in
self.cube]
        self.draw_cube(projection, self.graphicsViewObliq)

    def draw_cube(self, proj, view):
        pen = QtGui.QPen()
        scene = view.scene()
        scene.clear()

        a1, b1, c1, d1 = proj[4:]

        pen.setColor(QtCore.Qt.blue)
        scene.addLine(*a1, *b1, pen)
        scene.addLine(*b1, *c1, pen)
        scene.addLine(*c1, *d1, pen)
        scene.addLine(*d1, *a1, pen)

        a, b, c, d = proj[:4]

        pen.setColor(QtCore.Qt.red)
        scene.addLine(*a, *b, pen)
        scene.addLine(*b, *c, pen)
        scene.addLine(*c, *d, pen)
        scene.addLine(*d, *a, pen)

        pen.setColor(QtCore.Qt.green)
        scene.addLine(*a, *a1, pen)
        scene.addLine(*b, *b1, pen)
        scene.addLine(*c, *c1, pen)
        scene.addLine(*d, *d1, pen)
```

Приложение Б
(обязательное)
Экранные формы программы

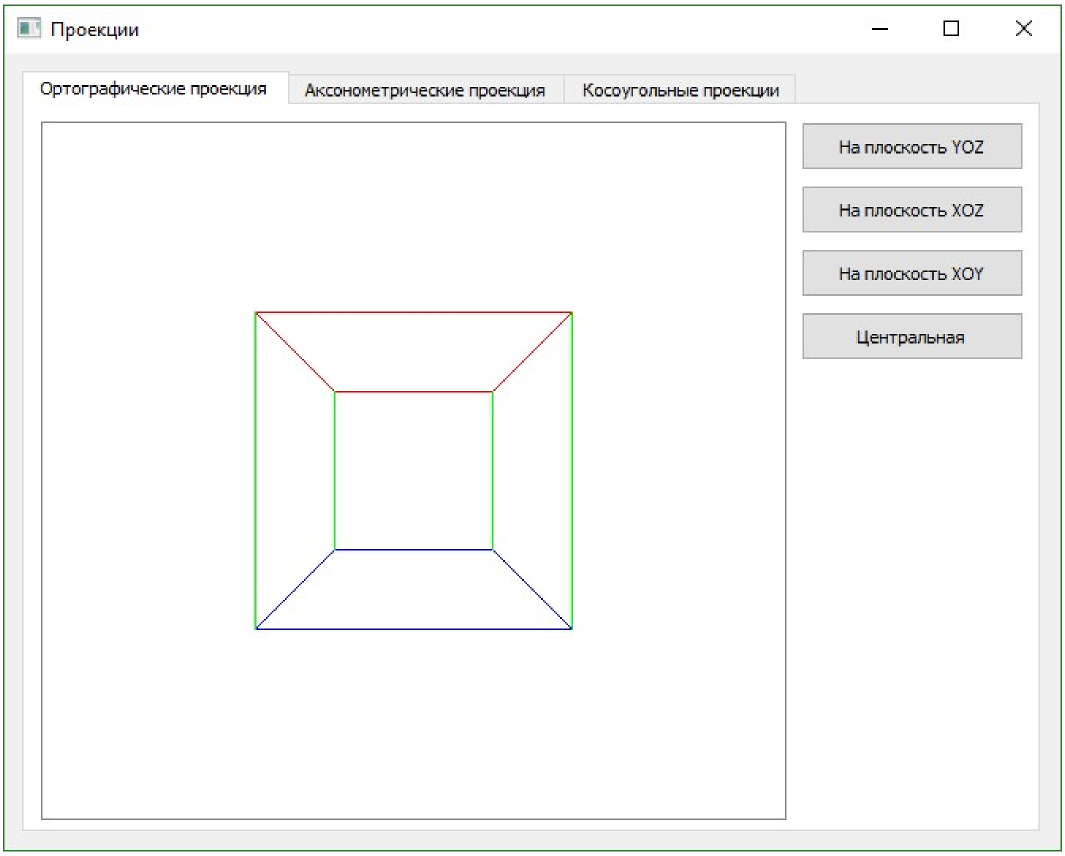

Рисунок Б.1 – Центральная проекция куба

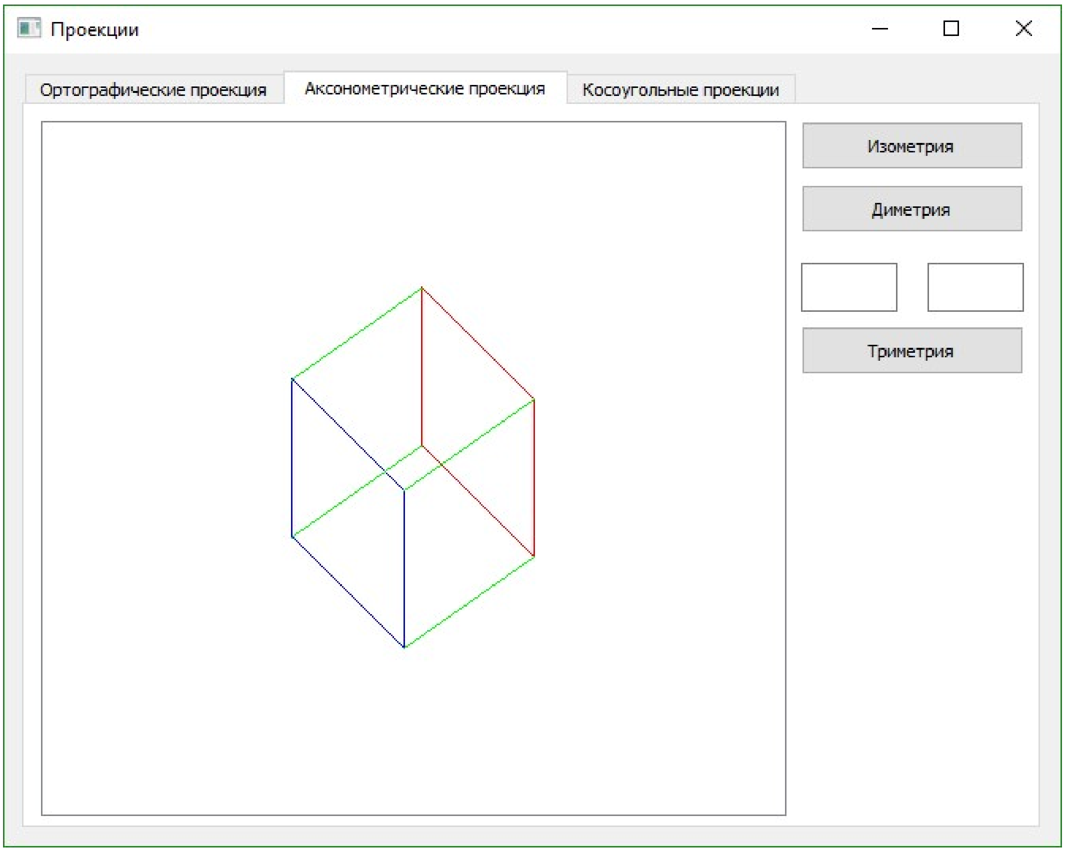

Рисунок Б.2 – Изометрическая проекция куба

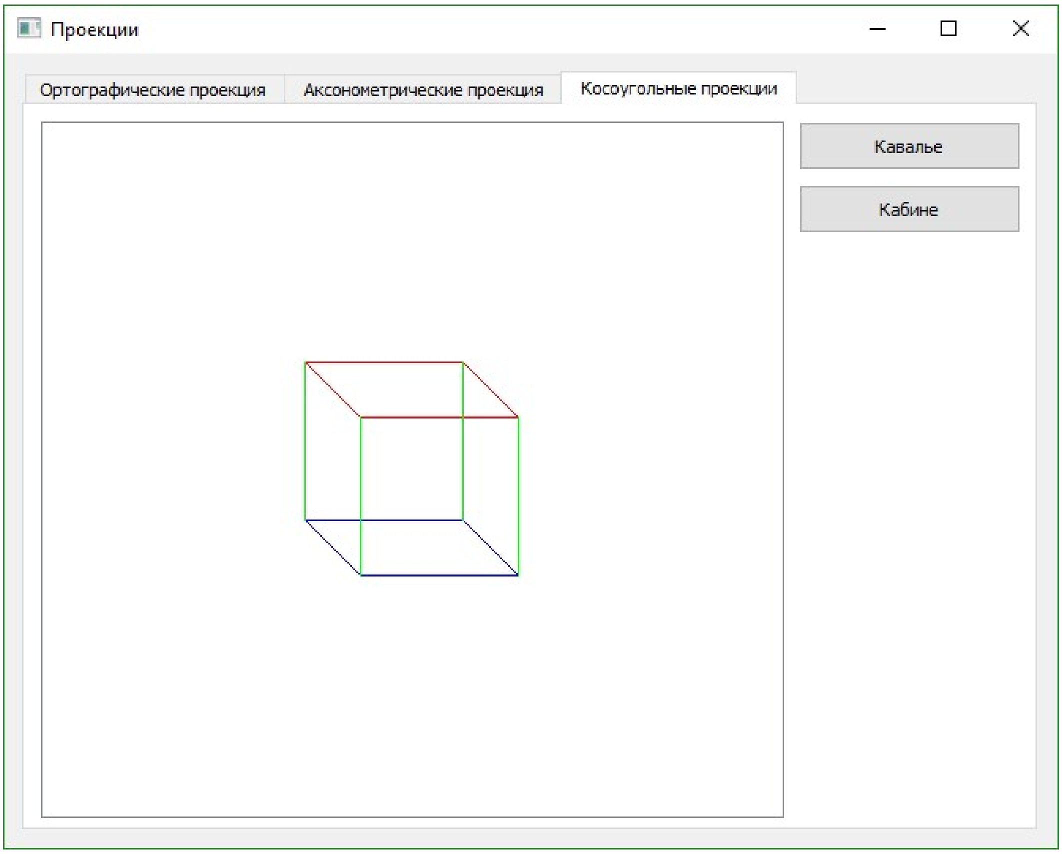

Рисунок Б.3 – Косоугольная проекция куба