

Міністерство освіти і науки України
Національний авіаційний університет
Кафедра прикладної математики

Лабораторна робота №2

Виконав:
Студент ПМ-351
Тітенко Д.С.
Преревірів:
Юрчук І.А.

Київ 2016

Тема

Тема: Побудова просторових геометричних об'єктів.

Мета

Мета: Вивчити афінні перетворення в просторі, паралельні і перспективні проєкції та вміти застосувати їх до геометричних конструкцій.

Завдання

1. Вивчити всі типи афінних перетворень, паралельних і перспективних проєкцій в просторі та їх матричне представлення в однорідних координатах.
2. Реалізувати програмно алгоритми наступних рухів геометричного об'єкту GO (піраміда, в основі якої прямокутний трикутник, а ребро співпадає з висотою):
 - Елементарні афінні перетворення у просторі (повороти навколо координатних осей, зсув, розтяг/стиск та дзеркальне відображення відносно координатних площин);
 - Перетворення, що вказане у варіанті завдання, як комбінації елементарних перетворень, значення параметрів якого задаються користувачем. Якщо на вказані параметри існують обмеження, що наведені в індивідуальному заданні студента, то користувача про це необхідно повідомити;
 - Перспективні перетворення, що вказані у варіанті, з можливістю введення їх параметрів користувачем.
3. Розмір та координати вузлів об'єктів GO автор програми визначає на свій розсуд, виходячи параметрів монітору та естетичних міркувань (об'єкт має бути по центру, рух GO в полі зору і т.д.).

Теоретичні відомості

Афінні перетворення

Зазвичай лінійна алгебра використовує матриці для представлення лінійних перетворень, і векторну суму для представлення паралельних перенесень. За допомогою розширеної матриці можливо представити і те, і те як матричний добуток. Ця техніка вимагає розширити всі вектори додаванням «1» в кінці, всі матриці розширюються додаванням рядка нулів знизу, і колонки — вектора переноса — справа, а також одиниці в нижній правий кут.

Таке представлення показує набір оборотних афінних перетворень як напівпрямий добуток K^n і $GL(n, k)$. Афінні перетворення утворюють групу щодо операції композиції відображень. Ця група називається афінною групою.

Зазвичай матрично-векторний добуток завжди відображає початок координат на початок координат, і, таким чином, не може представляти перенесення, яке обов'язково переносить початок координат в іншу точку. Додаванням «1» до кожного вектора, вважаємо простір відображенням на підмножину простору з одним додатковим виміром. В цьому просторі, початковий простір займає підмножину в якій останній індекс 1. Таким чином початок координат початкового простору буде знаходитися в $(0, 0, \dots, 0, 1)$. Перенесення всередині початкового простору в термінах лінійного перетворення простору з більшою кількістю вимірів стає можливим. Це є приклад однорідних координат.

Перевагою використання однорідних координат є те, що можливо комбінувати будь-яку кількість перетворень в одне шляхом перемноження матриць. Ця можливість використовується графічними програмами.

Проекції

Проекції — зображення просторових об'єктів на площині або якій-небудь іншій поверхні.

Загальний опис

В основі побудови всіх зображень лежить спосіб проєціювання, відповідно до якого всі проєкції розділяються на перспективні (центральні) і паралельні. У кожній з них формування зображення здійснюється прямолінійними проєктувальними променями, які проходять через всі точки зображуваної фігури до перетину з площиною проєкції. У перспективних проєкціях промені виходять з однієї точки (центра проєкції), а в паралельних вони паралельні заданому напрямку. У окремому випадку проєктувальні промені можуть бути перпендикулярними площині проєкції, тоді проєкція називається ортогональною.

У математиці, астрономії, картографії, геології, кристалографії, маркшейдерії, гірничій справі та інших галузях розроблено і використовується велика кількість різновидів перспективних і паралельних проєкцій. До креслень, а отже і до проєкцій, висуваються такі вимоги: наочність, зручність вимірювань, динамічність (можливість доповнення креслень у процесі розвитку гірничих робіт), прочитність за допомогою ЕОМ, універсальність умовних позначень тощо. У гірництві, геодезії та маркшейдерії застосовують і перспективні, і паралельні проєкції. (аксонометричні, афінні, векторні, стереографічні, лінійні та інші).

Хід роботи

Код програми

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

```
import numpy as np
```

```
from .transformation import *
```

```
class PointF3D(QtCore.QPointF):
```

```
    def __init__(self, point, *args, **kwargs):
```

```
        super(PointF3D, self).__init__()
```

```
        point.append(1)
```

```
        self.M = np.array(point)
```

```
        self.makeProjection()
```

```
    def makeProjection(self):
```

```
        x = self.M[0]
```

```
        y = self.M[1]
```

```
        z = self.M[2]
```

```
        self.setX(x - 0.5 * y)
```

```
        self.setY(0.5 * y - z)
```

```
class LineF3D(QtCore.QLineF):
```

```
    def __init__(self, outsideCoords, *args, **kwargs):
```

```
        super(LineF3D, self).__init__()
```

```
        x = outsideCoords[0][0]
```

```
        y = outsideCoords[0][1]
```

```
        z = outsideCoords[0][2]
```

```
        _P1 = PointF3D([x, y, z])
```

```
        x = outsideCoords[1][0]
```

```
        y = outsideCoords[1][1]
```

```
        z = outsideCoords[1][2]
```

```
        _P2 = PointF3D([x, y, z])
```

```

        self.setP1(_P1)

        self.setP2(_P2)

        self._pen = QtGui.QPen()

    def setColor(self, qColor):
        self._pen.setColor(qColor)

    def color(self):
        return self._pen.color()

    def widthF(self):
        return self._pen.widthF()

    def setWidthF(self, p_float):
        self._pen.setWidthF(p_float)

    def pen(self):
        return self._pen

    def setPen(self, pen):
        self._pen = pen

    def translateToCenter(self, centerCoords):
        self.translate(centerCoords[0], centerCoords[1])

class AxisX(LineF3D):
    def __init__(self, length, *args, **kwargs):
        outsideCoords = [[0, 0, 0], [length, 0, 0]]
        super(AxisX, self).__init__(outsideCoords)
        self.setWidthF(2)
        self.setColor(QtGui.QColor.fromRgb(255, 0, 0))

```

```
class AxisY(LineF3D):  
    def __init__(self, length, *args, **kwargs):  
        outsideCoords = [[0, 0, 0], [0, length, 0]]  
        super(AxisY, self).__init__(outsideCoords)  
        self.setP1(QtCore.QPointF(0, 0))  
        self.setP1(QtCore.QPointF(0, 0))  
        self.setWidthF(2)  
        self.setColor(QtGui.QColor.fromRgb(0, 255, 0))
```

```
class AxisZ(LineF3D):  
    def __init__(self, length, *args, **kwargs):  
        outsideCoords = [[0, 0, 0], [0, 0, length]]  
        super(AxisZ, self).__init__(outsideCoords)  
        self.setWidthF(2)  
        self.setColor(QtGui.QColor.fromRgb(0, 0, 255))
```

Висновки

1. Під час виконання лабораторної роботи я розглянув та вивчив афінні перетворення над просторовими об'єктами та їх проекції.
2. В моєму програмному забезпеченні алгоритм афінні перетворення використовуються для дій над заданим точками об'єктом.
3. Після реалізації програмного забезпечення я провів його тестування та прийшов до висновків що воно є досить швидким та використовує дуже мало ресурсів комп'ютера.
4. Розглянувши та проаналізувавши результати тестування а також код програмного забезпечення, я встановив обчислювальну складність алгоритму яка дорівнює $O(n)$.