CS606: Computer Graphics (Assignment 1)

2D Rendering of Lagrange interpolating polynomial and Bézier curve

Akshay Jindal
IMT2012002
IIIT-B
akshay.jindal@iiitb.org

***Abstract*—This report explains the algorithms used for 2D Rendering of Lagrange interpolating polynomial and Bézier curve.**

Keywords— translation, zooming, rotation, interpolation, Lagrange interpolating polynomial, Bézier curve;

I. Introduction

This assignment does a 2D rendering of Lagrange interpolating polynomial and Bézier curve and provides real-time interactive capabilities for panning (translation), zooming, rotation and addition/deletion of control points. The entire application is written using MVC architecture.

II. Summary

A. *Lagrange Interpolating Polynomial*[5]

The Lagrange interpolating polynomial is the polynomial $P(x)$ of degree $\leq(n-1)$ that passes through the n points, and is given by:

$$P(x) = \sum_{j=1}^{n} P_j(x),$$

where,

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

B. *Bézier curve*[3][4]

Given a set of $n+1$ control points $P_0, P_1, \ldots, P_n$, the corresponding Bézier curve (or Bernstein-Bézier curve) is given by:

$$\mathrm{B(t)} = \sum_{i=0}^{n} \binom{n}{i} \mathrm{t}^k (1-t)^{n-i} \mathrm{P_i}$$

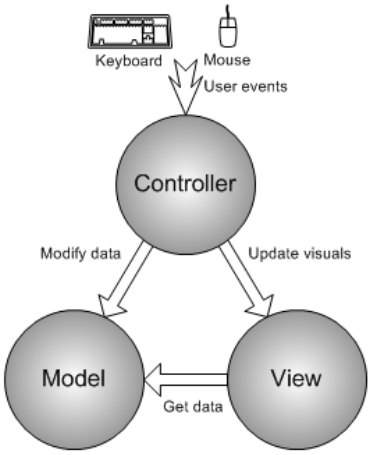
C. *MVC architecture*[2]

MVC paradigm is to divide an application into 3 separate components; Model, View and Controller components in order to minimize dependencies between them.

Model component is the brain part of the application, which contains all application data and implementations to tell how the application behaves.

View component is responsible to render the visual contents onto the screen.

Controller component is the bridge between users and the application by receiving and handling all user events, such as keyboard and mouse inputs.



III. User Interaction

The application provides three real-time interactive capabilities:

A. *Translation*

GLUT mouse coordinates are converted to OpenGL world coordinates and the image can be translated by clicking right mouse button and dragging it.

B. Zooming

The image can be scale up and down using '+' and '-' keys.

C. Rotation

The rendered image can be rotated clockwise and anticlockwise using 'l' and 'r' key, respectively around the centroid of control points in z-direction.

IV. INSIGHT

- MVC architecture is easy to implement and helps to maintain the code and also improves its reusability.
- Lagrange polynomial looks overly fitted and isn't appealing for large number of points.

REFERENCES

[1] Angel, E.; Shreiner, D. (2012). Ch. 2,3 Interactive Computer Graphics - A Top-Down Approach 6th ed. Pearson.

[2] OpenGL Windows GUI Application. (n.d.). Retrieved January 23, 2016, from http://www.songho.ca/opengl/gl_mvc.html

[3] Bézier Curve. (n.d.). Retrieved January 23, 2016, from http://mathworld.wolfram.com/BezierCurve.html

[4] Bézier Curve. (n.d.). Retrieved January 23, 2016, from https://en.wikipedia.org/wiki/Bézier_curve

[5] Lagrange Interpolating Polynomial. (n.d.). Retrieved January 23, 2016, from http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html