COSC 471: Computer Graphics

Spring 2023, Assignment 5

Bezier Curve

(Total points: 100)

Due date: May 17th, 2023 (end of the day)

Submission Instructions

1. Compile your solutions and generate screenshots of your output, then write your understanding of output in a file. Compress source codes output and your explanation into one compressed file.
2. Rename the compressed file following this notation:
   FirstnameLastnamePA5.zip (or any format you prefer)
   Do not use a space in the filename.
3. Upload and submit the compressed file through Blackboard.
4. **NO** late submission is accepted.

---

1. Overview

Bézier Curve is a parameter-based curve used in Computer Graphics. In this assignment, you will need to implement **`de Casteljau`** algorithm and draw Bézier Curve using 4 control points (when you correctly implemented this algorithm, you should be able to use more control point on the Bézier Curve).

In this assignment, you need to modify the following functions in the **`main.cpp`** file:

- Modify function **`bezier`**: this function is used to implement rendering of Bézier Curve. It uses a list of control points and a object of **`OpenCV::Mat`** as inputs, and returns no value. It uses **`t`** in the range of 0 and 1 to do recursive computation, and in every recursive computing, the value of **`t`** value will increase slightly. For the value of **`t`**, it is computed using another function **`recursive_bezier`**, and the returned value is then used in rendering the object of **`OpenCV::Mat.`**
- Modify function **`recursive_bezier`**: this function uses a list of control points and floating point t as inputs to implement **`de Casteljau`** algorithm, and then return the corresponding coordinates on the Bézier Curve.

2. Algorithm

`De Casteljau` algorithm is explained as below:

1. Considering a list of control points p0, p1, … pn, which controls the Bézier Curve, firstly, we connect the control points to get line segments.
2. Subdivide each line segment with a ratio of t : (1-t), and find the split point.
3. The obtained split point is used as the new control point list, and the length of the new list will be reduced by one.
4. If the list contains only one point, return to that point and terminate. Otherwise, use the new control point list and Go to step 1 to continue.

Using multiple different t in [0,1] to execute the above algorithm, you can get the corresponding Bézier curve.

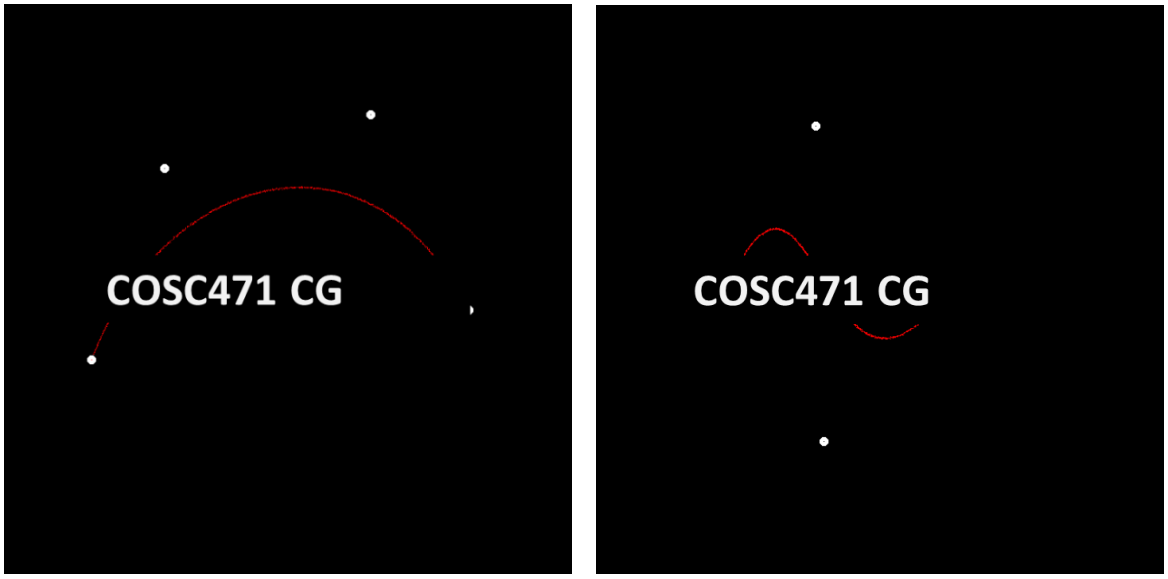
3. Compiling

3.1 Compiling and Run

In this assignment, you will write on a new code skeleton framework, which is much smaller than the previous code skeleton. Similar to the previous assignments, you can choose to complete this assignment on your own computer system or virtual machine.
Please download the skeleton code of the project and use the following command to build the project as before:

```
Mkdir build
cd ./build
cmake ..
make
```

Then, you can run the code like this: **`./BezierCurve.`** When its running, program will open a black window. In it, you can click on the screen to select points to control the Bézier curve. The program will wait you select 4 control points and then render the curve automatically according to the control points you selected. The implementation provided in the skeleton code framework uses polynomial equations to calculate Bézier curves and draw it in red. The following two pictures shown sample Bézier curves with control points:



After ensuring everything is good in the skeleton code, you can start to complete your own implementation. Comment out the line that calls the **`naïve_bezier`** function in the while loop in the **`main`** function, and cancels the comment on the **`bezier`** function. **You are required to implement your Bezier curve in Green color.**

If you want to ensure correct implementation, please call **`naive_bezier`** and **`bezier`** functions at the same time. If it is correct, both should write approximately the same pixels, so the curve will appear **yellow**. In this way, you can ensure that it is implemented correctly.

You can also try modifying code to have different numbers of control points and show Bezier curves accordingly.

4. Assignment submission and evaluation

4.1 Submission

Read the assignment description carefully to make sure you understand the programing assignment correctly. Modify the skeleton codes as required. Compile your solutions and generate screenshots of your output, then write your understanding of output in a file. **Please create an image folder and save all your images in this folder**. Compress source codes output, images and your explanation into one compressed file named **FirstnameLastnamePA4.zip (or any format you prefer).** Submit the compressed file through Blackboard before deadline.

When you complete the assignment, please check your project clearly and make sure your submission folder includes CMakeLists.txt file and all the source files no matter you changed them or not. In addition, please include a README.md file to explain if you complete the bonus question (if you complete bonus question, please include screenshot of it too), or explain this in the result analysis and screenshot file.

4.2 Evaluation

This programming assignment has two parts: basic parts and bonus part. Here is the grading details:

- Correctly implement de Casteljau algorithm: given control points, your code can generate correct Bezier curve (80%)
- Compile and run codes correctly and submit all files correctly (10%)
- Screens shot and result analysis (10%)
- **Bonus part**: try to add more control points (20%)