Assignment 2 : Create A B-spline Surface

NAME: CHENGHUA YANG
STUDENT NUMBER: 2019533089
EMAIL: YANGCHH@SHANGHAITECH.EDU.CN

1 INTRODUCTION

1.1 Contents

1.Explanation about the B-spline 2.Get a position of knot values u, v on a surface by construct curves. 3.Get the tangent of a postion with knot value on 2 curve and construct the normal on the surface 4. UV and Texture

2 IMPLEMENTATION DETAILS

2.1 My study about the B-spline

B-spline is quite different from the Beizer Curve.
Beizer method simply use every point as a position effector and iterately interpolates all of them.
However, B-spline only want to use part of the node to get a point a the curve. Well, you can simply restrict the effector nodes to a part of them, but as a result, you'll get several seperated curve according to different u ranges.

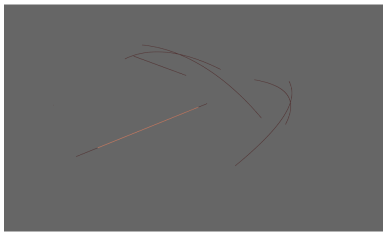


Fig. 1. Straightly use Beizer method to get the B-spline curve

Here it leads to a question of how to get a continuous curve and another question of how to get the point of u but not between some 2 control points.
The knot vector is designed to sovle this. It constrcut the N coefficients of controll point.
I have tried 3 ways of calculate the curve point. The DeBoor method, the Beizer method mentioned above and the N basis function method.
Well, I have to say they got similar result, and made curves mess up together. I don't know why this happens.

© 2021
Computer Graphics I Course Assignment

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Fig. 2. The basis function N of B-spline

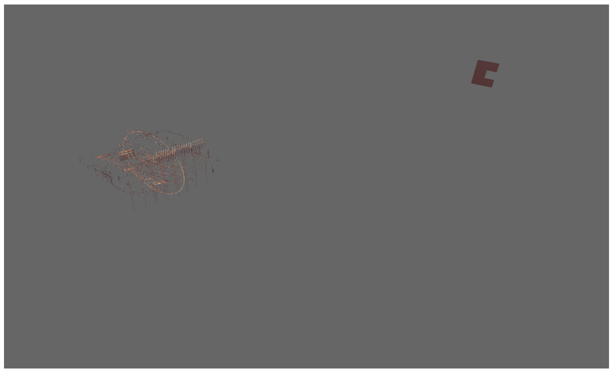


Fig. 3. Using the point render mode and DeBoor to get this

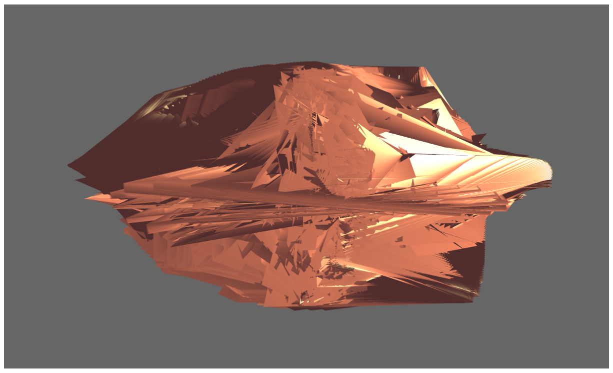


Fig. 4. Using the face render mode and N method to get this

2.2 get position on surface

There are 2 kinds of definations of a B-spline surface. One straightly uses the N functions' multiplication, while another construct new subcurves to find point. I realise both of them.

Name: Chenghua Yang
student number: 2019533089
email: yangchh@shanghaitech.edu.cn

2.3 Get the normal

The normal of a point on the surface must be gain from the 2 direction tangents on the curves that are through that point.
The derivative method try to calculate the derivative of a B-spline curve. According to the calculation, the deriviation of a B-spline is just another B-spline with modified control points and knots vector. Here I'd like to talk about the clamped curve case. The clamped curve has most of its knot be 0 or 1, which will make the coefficient of the new control points calculation be a constant.

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}u}\mathbf{C}(u) = \mathbf{C}'(u) = \sum_{i=0}^{n-1} N_{i+1,p-1}(u)\mathbf{Q}_i}$$

Fig. 5. The direvative curve of b-spline curve

$$\boxed{\mathbf{Q}_i = \frac{p}{u_{i+p+1} - u_{i+1}}(\mathbf{P}_{i+1} - \mathbf{P}_i)}$$

Fig. 6. Q are control points of the derivative curve

$$\boxed{\mathbf{Q}_0 = \frac{p}{u_{p+1}}(\mathbf{P}_1 - \mathbf{P}_0)}$$

Fig. 7. Clamped direvative curve has more convinient Q

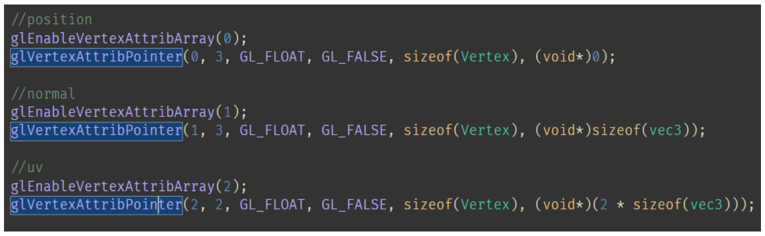
However, there is another much easier way to calculate the normal, just fetch the ajacent vertices of a vertex, contsruct 4 line and calculate their averages'cross. It is a approximation normal of that vertex

2.4 UV and Texture

OpenGl with the 3rd party library can load images and turn them into textures. A texture has a local coordinate called UV coordinate, it is used for vertices to sample color from the textures.
the (0,0) location is refered to the left bottom corner of the image, the (1,1) location is refered to the right up corner of the image. While a vertex can sample out of the range (0,0) (1,1). How this works can be setted as "repeat" or "clamped" to make a sample color rule.

2.5 Finally, I figure out the bugs

I used to think about the stripe argument of glVertexAttribPointer was the space between each element(As the default number is 0!!), yet it is actually the total length of the wrapped element pack. So the mess above was caused by the wrong reading stripe of the Vertex struct.



```
//position
glEnableVertexAttribArray(0);
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, sizeof(Vertex), (void*)0);

//normal
glEnableVertexAttribArray(1);
glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, sizeof(Vertex), (void*)sizeof(vec3));

//uv
glEnableVertexAttribArray(2);
glVertexAttribPointer(2, 2, GL_FLOAT, GL_FALSE, sizeof(Vertex), (void*)(2 * sizeof(vec3)));
```

Fig. 8. The glVertexAttribPointer function

3 RESULTS

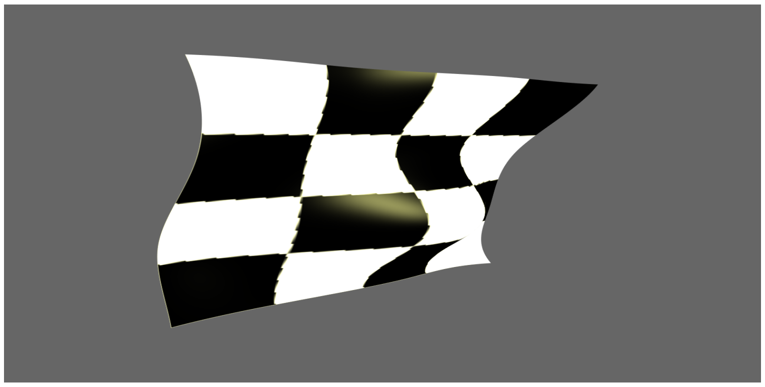


Fig. 9. The textured surface

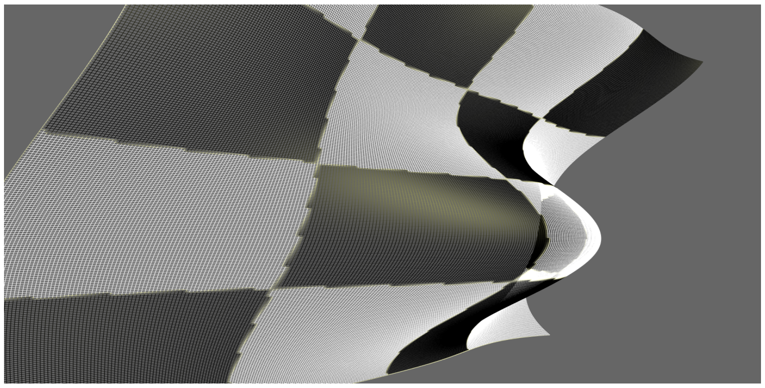


Fig. 10. The textured surface with line render mode