

華東理工大學

计算机图形学

2024年1月

奉贤校区

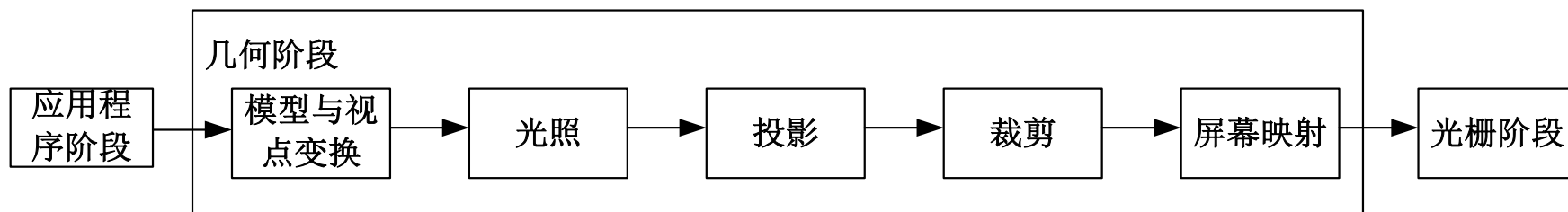
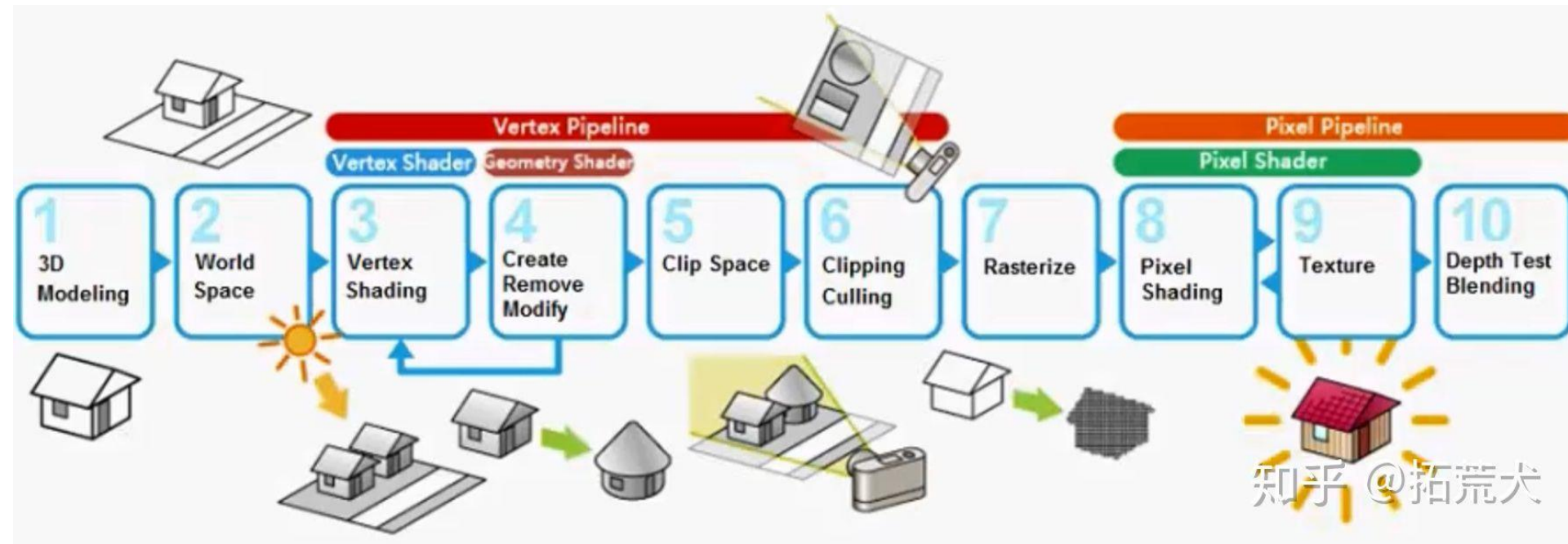


图2.22 绘制流水线的结构



知乎 @拓荒犬

CPU

GPU

Vertex buffer

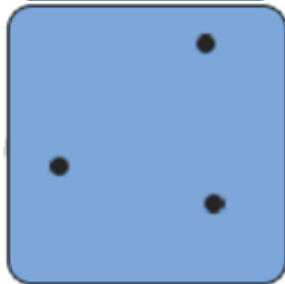
- Positions
- Attributes (color, normal...)



应用程序
数据、交互

Uniform variables
• P, V, M matrices

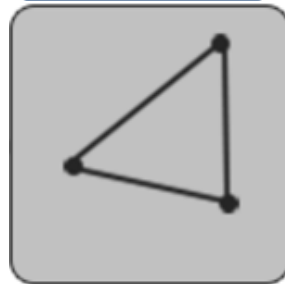
Vertex shader



投影计算
法向量变换、归一化
逐顶点光照计算

...
gl_Position
Varying
variables

Assembly



Viewport
Clipping
Culling

...
gl_Position
Varying
variables

Tessellation shader
Geometry shader



...
gl_Position
Varying variables

Frame buffer



Tests and Blending
• Color buffer
• Depth buffer
• Stencil buffer
• Multisample
• Anti-aliasing
• Alpha blending
• Fog

Pixels



渲染结果

Fragment shader

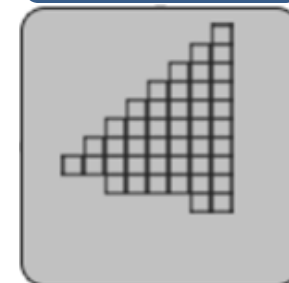


Colored
Fragments
Screen color

Fragments
Varying
variables

Uniform variables
• Texture

Rasterization



重心插值
(blending ratio)

题型

- 1. 选择 10分 （1空1分）
- 2. 填空 10分 （ 1空1分）

偏基础概念类，涉及到全部章节。

3.算法题（40分。每题20分）

- 涉及到图形学的多种算法（蕴含在程序代码里）

快速的绘制Bezier曲线的方法。请仔细阅读以下代码段：

//点类型point

typedef struct Point {

int x, y;

Point(int a = 0, int b = 0)

{

x = a, y = b;

}

} point;

//矩形类型rect

typedef struct Rectangle {

float w_xmin, w_ymin;

float w_xmax, w_ymax;

Rectangle(float xmin = 0.0, float ymin = 0.0, float xmax = 0.0, float yman = 0.0) {

w_xmin = xmin;w_ymin = ymin;

w_xmax = xmax;w_ymax = yman;

}

}rect;

//颜色类型color

typedef struct Color{

int r,g,b;

Color(int red= 0,int green = 0,int blue=0){

r=red;

g=green;

b=blue;

}

/// }color;

void de_Casteljau(vector<Point> &points,color &c){

int n=1000;

float step=1.0/n;

int iPointNum=points.size();

GLfloat FinalPoint[2][n+1], midlP[2][iPointNum];

int i, j;

double t=0.0;

for(int k = 0; k<=n; k++)

{

t=step*k;

for (i = 0; i < iPointNum; ++i)//迭代次数

{ midlP[0][i] = points[i].x ;

midlP[1][i] = points[i].y ;

}

for (i = 0; i < iPointNum-1; ++i)//迭代次数

{

for (j = 0; j < iPointNum - 1-i; ++j)//中间层次

{

//,通过上一次迭代的结果计算

midlP[0][j] = midlP[0][j] * (1 - t) + midlP[0][j+1] * t;

midlP[1][j] = midlP[1][j] * (1 - t) + midlP[1][j+1] * t;

}

}

FinalPoint[0][k] = midlP[0][0];

FinalPoint[1][k] = midlP[1][0];

}

glColor3f(c.r, c.g, c.b);

glPointSize(2);

glBegin(GL_LINE_STRIP);

for (int i = 0; i <=n; i++)

{

glVertex2f(FinalPoint[0][i],FinalPoint[1][i]);

}

glEnd();

}

仔细查看此代码，回答下述问题：

1. de_Casteljau算法的基本递推公式为什么？（2分）
2. 一条3次Bezier曲线顶点为(10,0),(20,60),(120,60)(180,0)请问 $t=1/2$ 处曲线的值为？（2分）
3. 参数points和c的作用是什么？（2分）
4. 局部变量n和step分别起着什么作用？（2分）
5. 数组FinalPoint起着什么作用？（2分）；
6. 数组midlP起着什么作用？（2分）；
7. de_Casteljau函数内是否绘制了Bezier的控制多边形？（2分）；
8. 目前的绘制中采用glBegin(GL_LINE_STRIP); 若改成GL_LINES，相应的代码如何修改？（2分）
9. 目前的生成方式是离散化生成曲线上的点，目前绘制出的曲线上的点数是多少个？（2分）
若加密一倍，如何修改代码？（2分）

4.程序题（40分， 每题20分）

- 涉及到图形学的编程基础、编程实践等。与前面有所不同。但类似

```
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(400,300);
    glutInitWindowPosition(100,100);
    glutCreateWindow(" 橡皮筋技术");
    glutDisplayFunc(Display);
    glutReshapeFunc(ChangeSize);
    glutMouseFunc(MousePlot);
    glutPassiveMotionFunc(PassiveMouseMove);
    Initial();
    glutMainLoop();
    return 0;
}
```

1. glutInit函数作用是什么？（2分）
2. 本程序涉及到的OpenGL库有哪些？（2分）
3. Display函数在程序中起什么作用？（2分）（2分）

- 还有更复杂的。

谢谢！

*Thank
You*