几何建模与处理基础
GAMES102
作业 5：细分曲线

Qingjun Chang

2020 年 11 月 21 日

1 目标

- 学习使用细分方法生成曲线的原理和方法

2 细分曲线

问题：对于给定 $\mathbb{R}^2$ 域内一组有序点 $\{\mathbf{Q}_i\}_{i=0}^n$，构成简单多边形，找到一条与之关联的光滑曲线。

2.1 Chaikin 细分曲线 (二次 B-spline 曲线细分)[1]

拓扑规则：

- 点分裂成边（割角），老点被抛弃（逼近型）
- 新点老点重新编号

几何规则：

- $\mathbf{Q}'_{2i} = \frac{1}{4}\mathbf{Q}_{i-1} + \frac{3}{4}\mathbf{Q}_i$
- $\mathbf{Q}'_{2i+1} = \frac{3}{4}\mathbf{Q}_i + \frac{1}{4}\mathbf{Q}_{i+1}$

下图展示了经过 5 次 Chaikin 细分得到的细分曲线：

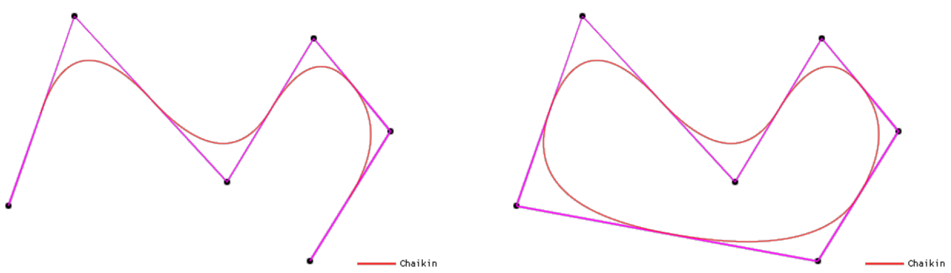

图 1: Chaikin 细分曲线（左：不封闭；右：封闭）

2.2 三次 B-spline 曲线细分 [1]

拓扑规则：

- 边分裂成两条新边

几何规则：

- $\mathbf{Q}'_{2i} = \frac{1}{8}\mathbf{Q}_{i-1} + \frac{3}{4}\mathbf{Q}_i + \frac{1}{8}\mathbf{Q}_{i+1}$
- $\mathbf{Q}'_{2i+1} = \frac{1}{2}\mathbf{Q}_i + \frac{1}{2}\mathbf{Q}_{i+1}$

下图展示了重复 5 次的“三次 B 样条”曲线细分得到的细分曲线：

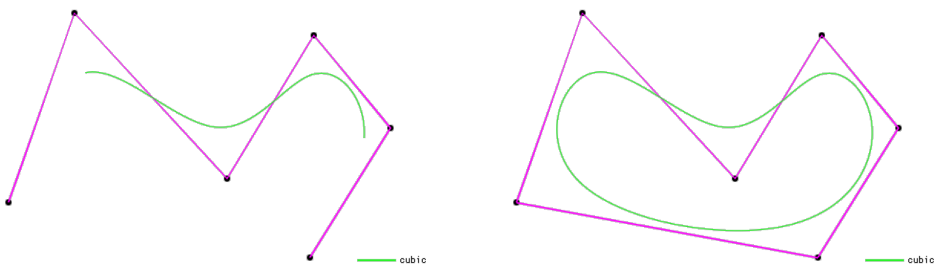

图 2: 三次 B 样条细分曲线（左：不封闭；右：封闭）

2.3 四点插值型细分 (补角)[1]

拓扑规则：

- 保留原有顶点
- 对每条边，增加一个新顶点

几何规则：

- $\mathbf{Q}'_{2i+1} = \frac{\mathbf{Q}_i+\mathbf{Q}_{i+1}}{2} + \alpha(\frac{\mathbf{Q}_i+\mathbf{Q}_{i+1}}{2} - \frac{\mathbf{Q}_{i-1}+\mathbf{Q}_{i+2}}{2})$

下图展示了重复 5 次的四点插值细分得到的细分曲线：

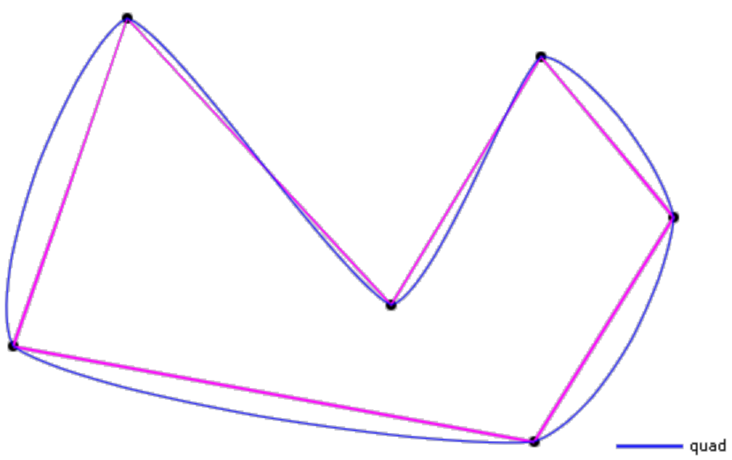

图 3: 封闭控制多边形的四点插值细分曲线

3 主要代码

——更多示例结果可参见附带的视频文件——

3.1 code

Chaikin 细分：

```
std::vector<Ubpa::pointf2> Chaikin_subdivision(std::vector
 <Ubpa::pointf2>* p, bool close) {
  std::vector<Ubpa::pointf2> newP;
  newP.clear();
  int n=p->size();
  if (close) {
    for (int i=0;i<p->size();++i) {
      newP.push_back(p->at((i-1+n)%n)*0.25f+p->at(i)*0.75f);
      newP.push_back(p->at(i)*0.75f+p->at((i+1)%n)*0.25f);
    }
  }
  if (!close) {
    newP.push_back(p->at(0)*0.75f+p->at(1)*0.25f);
    for (int i=1;i<p->size()-1;++i) {
      newP.push_back(p->at((i-1+n)%n)*0.25f+p->at(i)*0.75f);
      newP.push_back(p->at(i)*0.75f+p->at((i+1)%n)*0.25f);
    }
    newP.push_back(p->at((2*n-2)%n)*0.25f+p->at(n-1)*0.75f);
  }
  return newP;
}
```

三次 B-spline 细分曲线：

```
std::vector<Ubpa::pointf2> cubic_subdivision(std::vector
 <Ubpa::pointf2>* p, bool close) {
  std::vector<Ubpa::pointf2> newP;
  newP.clear();
  int n=p->size();
  if (close) {
    for (int i=0;i<p->size();++i) {
      newP.push_back(p->at((i-1+n)%n)*0.125f+p->at(i)*0.75f
       +p->at((i+1)%n)*0.125f);
      newP.push_back(p->at(i)*0.5f+p->at((i+1)%n)*0.5f);
    }
  }
  if (!close) {
```

```
    newP.push_back(p->at(0)*0.5f+p->at(1)*0.5f);
    for (int i=1;i<p->size()-1;++i) {
      newP.push_back(p->at((i-1+n)%n)*0.125f+p->at(i)*0.75f
       +p->at((i+1)%n)*0.125f);
      newP.push_back(p->at(i)*0.5f+p->at((i+1)%n)*0.5f);
    }
  }
  return newP;
}
```

四点插值型细分曲线：

```
std::vector<Ubpa::pointf2> quad_subdivision(std::vector
 <Ubpa::pointf2>* p, bool close, float alpha=0.075) {
  std::vector<Ubpa::pointf2> newP;
  newP.clear();
  int n=p->size();
  if (close) {
    for (int i=0;i<p->size();++i) {
      newP.push_back(p->at(i));
      newP.push_back((p->at(i)+p->at((i+1)%n))/2.0f+((p->
       at(i)+p->at((i+1)%n))/2.0f-(p->at((i-1+n)%n)+p->
        at((i+2)%n))/2.0f)*alpha);
    }
  }
  if (!close) {
    for (int i=0;i<p->size()-1;++i) {
      newP.push_back(p->at(i));
      newP.push_back((p->at(i)+p->at((i+1)%n))/2.0f+((p->
       at(i)+p->at((i+1)%n))/2.0f-(p->at((i-1+n)%n)+p->
        at((i+2)%n))/2.0f)*alpha);
    }
  }
  return newP;
}
```

参考文献

[1] GAMES-102 Courses. 细分曲线. [EB/OL]. http://staff.ustc.edu.cn/~lgliu/Courses/GAMES102_2020/PPT/GAMES102-8_SubdivisionCurves.pdf 2020-11-16.