几何建模与处理基础
GAMES102
作业 3：参数曲线拟合

Qingjun CHANG

2020 年 10 月 31 日

1 目标

- 学习参数曲线拟合
- 使用各种参数化方法，并进行比较

2 曲线参数化

问题：对于给定 $\mathbb{R}^2$ 域内一组有序点 $\{\mathbf{P}_i = (x_i, y_i)\}_{i=1}^n$，找到一组合适的实数 $\{t_i\}_{i=1}^n$，使得 t_i 与点 $\mathbf{P}_i$ 一一对应。

下面是几种常用的参数化算法：

- 均匀参数化
$$t_{i+1} - t_i = C \qquad (C\text{是常数})$$
- 弦长参数化
$$t_{i+1} - t_i = \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2$$
- 中心参数化
$$t_{i+1} - t_i = \sqrt{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2}$$
- Foley-Nielsen 参数化 [1]
$$t_{i+1} - t_i = \begin{cases} d_i \left[1 + \frac{3\hat{\alpha}_{i+1} d_{i+1}}{2(d_i + d_{i+1})}\right] & i = 1 \\ d_i \left[1 + \frac{3\hat{\alpha}_i d_{i-1}}{2(d_{i-1} + d_i)} + \frac{3\hat{\alpha}_{i+1} d_{i+1}}{2(d_i + d_{i+1})}\right] & i = 2, 3, \dots, n-2 \\ d_i \left[1 + \frac{3\hat{\alpha}_i d_{i-1}}{2(d_{i-1} + d_i)}\right] & i = n-1 \end{cases}$$
$$d_i = \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2 \qquad \alpha_i = angle(\mathbf{P}_{i-1}, \mathbf{P}_i, \mathbf{P}_{i+1}) \qquad \hat{\alpha}_i = min\{\alpha_i, \frac{\pi}{2}\}$$

3 拟合参数曲线

问题：对于给定 $\mathbb{R}^2$ 域内一组点 $\{\mathbf{P}_i = (x_i, y_i)\}_{i=1}^n$，找到一个拟合这组点的向量值函数 $\mathbf{P} = f(t) : \mathbb{R} \longmapsto \mathbb{R}^2$，使得 $f(t)$ 的轨迹尽可能靠近这组点。

3.1 拉格朗日型参数曲线拟合

$$\begin{bmatrix} x \\ y \end{bmatrix} = f(t) = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} t + \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} t^2 + \cdots + \begin{bmatrix} a_{n-1} \\ b_{n-1} \end{bmatrix} t^{n-1} \tag{1}$$

将 $\{t_i, \mathbf{P}_i\}_{i=1}^n$ 代入 (1) 式得到如下线性系统：

$$\begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_{n-1} & b_{n-1} \end{pmatrix} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix} \tag{2}$$

通过求解 (2) 式线性系统可得到 $\boldsymbol{a} = [a_0, a_1, \dots, a_{n-1}]^{\mathrm{T}}$ 和 $\boldsymbol{b} = [b_0, b_1, \dots, b_{n-1}]^{\mathrm{T}}$

3.2 高斯基函数型参数曲线拟合

$$\begin{bmatrix} x \\ y \end{bmatrix} = f(t) = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} g_1(t) + \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} g_2(t) + \cdots + \begin{bmatrix} a_{n-1} \\ b_{n-1} \end{bmatrix} g_n(t) \tag{3}$$

$$g_i(t) = e^{-(t-t_i)^2/(2\sigma^2)}$$

将 $\{t_i, \mathbf{P}_i\}_{i=1}^n$ 代入 (3) 式得到如下线性系统：

$$\begin{pmatrix} 1 & g_1(t_1) & g_2(t_1) & \dots & g_n(t_1) \\ 1 & g_1(t_2) & g_2(t_2) & \dots & g_n(t_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & g_1(t_n) & g_2(t_n) & \dots & g_n(t_n) \end{pmatrix} \begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_{n-1} & b_{n-1} \end{pmatrix} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix} \tag{4}$$

可以发现上式左边系数矩阵维数为 $n \times (n+1)$，未知数个数多余线性无关的方程个数，方程解不唯一。可添加单位 1 约束：$\sum a_i = 1$ 和 $\sum b_i = 1$ 使得方程有唯一解，则得到如下方程：

$$\begin{pmatrix} 1 & g_1(t_1) & g_2(t_1) & \dots & g_n(t_1) \\ 1 & g_1(t_2) & g_2(t_2) & \dots & g_n(t_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & g_1(t_n) & g_2(t_n) & \dots & g_n(t_n) \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_{n-1} & b_{n-1} \end{pmatrix} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \\ 1 & 1 \end{pmatrix} \tag{5}$$

通过求解 (5) 式线性系统可得到 $\boldsymbol{a} = [a_0, a_1, \dots, a_{n-1}]^{\mathrm{T}}$ 和 $\boldsymbol{b} = [b_0, b_1, \dots, b_{n-1}]^{\mathrm{T}}$

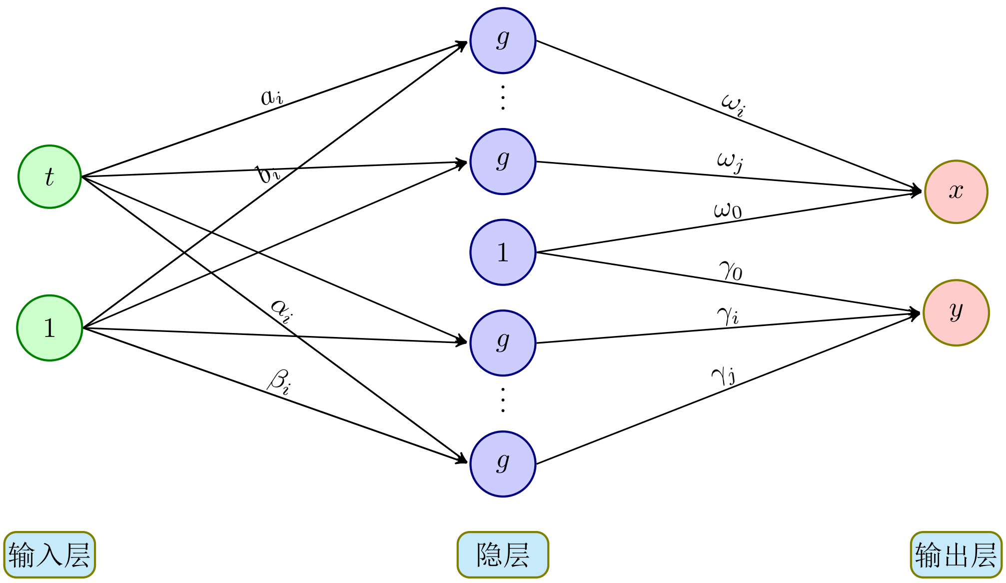

图 1: 以高斯函数为基函数的 $f(t)$ 的网络图

3.3 RBF 算法拟合参数曲线

$$\begin{bmatrix} x \\ y \end{bmatrix} = f(t) = \begin{bmatrix} \omega_0 + \sum_{i=1}^{m} \omega_i g_{0,1}(a_i t + b_i) \\ \gamma_0 + \sum_{i=1}^{m} \gamma_i g_{0,1}(\alpha_i t + \beta_i) \end{bmatrix} \tag{6}$$

参照作业 2 的方法分别对 (6) 式中两个分量运用 RBF 算法，如图 3.3网络所示。

3.4 最小二乘型参数曲线拟合

在拉格朗日型函数的基础上，考虑使用最高次数不高于 $m(m < n)$ 的多项式作为基函数。

$$\begin{bmatrix} x \\ y \end{bmatrix} = f(t) = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} t + \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} t^2 + \cdots + \begin{bmatrix} a_m \\ b_m \end{bmatrix} t^m \tag{7}$$

将 $\{t_i, \mathbf{P}_i\}_{i=1}^{n}$ 代入 (7) 式得到如下线性系统：

$$\mathbf{AX} = \mathbf{P} \tag{8}$$

其中

$$\mathbf{A} = \begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^m \\ 1 & t_2 & t_2^2 & \dots & t_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^m \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_m & b_m \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix}$$

通过对方程 (8) 做最小二乘得到

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})\mathbf{X} = \mathbf{A}^{\mathrm{T}}\mathbf{P} \tag{9}$$

通过求解 (9) 式线性系统可得到 $\mathbf{X} = [\boldsymbol{a}, \boldsymbol{b}]$，$\boldsymbol{a} = [a_0, a_1, \dots, a_{n-1}]^{\mathrm{T}}$ 和 $\boldsymbol{b} = [b_0, b_1, \dots, b_{n-1}]^{\mathrm{T}}$

3.5 岭回归型参数曲线拟合

在最小二乘型拟合中添加正则约束，等式 (9) 变为：

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{E})\mathbf{X} = \mathbf{A}^{\mathrm{T}}\mathbf{P} \tag{10}$$

通过求解 (10) 式线性系统可得到 $\mathbf{X} = [\boldsymbol{a}, \boldsymbol{b}]$,$\boldsymbol{a} = [a_0, a_1, \ldots, a_{n-1}]^{\mathrm{T}}$ 和 $\boldsymbol{b} = [b_0, b_1, \ldots, b_{n-1}]^{\mathrm{T}}$

4 数值实验

4.1 code

以下是四种参数化算法在 c++ 中的实现。
均匀参数化：

```cpp
Eigen::VectorXf uniformParameterization(int numOfPoints) {
  Eigen::VectorXf y = Eigen::VectorXf::Zero(numOfPoints);
  for (int i=0; i<numOfPoints; ++i)
    y[i] = i;
  y /= y[numOfPoints-1];
  return y;
}
```

弦长参数化：

```cpp
Eigen::VectorXf chordParameterization(std::vector<
 Ubpa::pointf2> points) {
  int n = points.size();
  Eigen::VectorXf y = Eigen::VectorXf::Zero(n);
  for (int i=1; i<n; ++i)
    y[i] = y[i-1]+(points[i]-points[i-1]).norm();
  y /= y[n-1];
  return y;
}
```

中心参数化：

```cpp
Eigen::VectorXf centripetalParameterization(std::vector<
 Ubpa::pointf2> points) {
  int n = points.size();
  Eigen::VectorXf y = Eigen::VectorXf::Zero(n);
  for (int i=1; i<n; ++i)
    y[i] = y[i-1]+sqrt((points[i]-points[i-1]).norm());
  y /= y[n-1];
  return y;
}
```

Foley-Nielsen 参数化：

```
Eigen::VectorXf FoleyParameterization(std::vector<
 Ubpa::pointf2> points) {
  int n = points.size();
  Eigen::VectorXf y = Eigen::VectorXf::Zero(n);
  if (n == 2) { y[1] = 1; return y; }
  Eigen::VectorXf dist = Eigen::VectorXf::Zero(n-1);
  Eigen::VectorXf alpha = Eigen::VectorXf::Zero(n-1);
  for (int i=0; i<n-1; ++i)
    dist[i] = (points[i+1]-points[i]).norm();
  for (int i=1; i<n-1; ++i) {
    float cosvalue = (points[i-1]-points[i]).cos_theta(
     points[i+1]- points[i]);
    alpha[i] = std::min(PI-acos(cosvalue), PI/2);
  }
  y[1] = dist[0]*(1+1.5*alpha[1]*dist[1]/(dist[0]+dist[1]));
  for (int i=2; i<n-1; ++i) {
    y[i] = y[i-1]+dist[i-1]*(1+1.5*alpha[i-1]*dist[i-2]/
     (dist[i-2]+dist[i-1])+1.5*alpha[i]*dist[i]/
     (dist[i-1]+dist[i]));
  }
  y[n-1] = y[n-2]+dist[n-2]*(1+1.5*alpha[n-2]*dist[n-3]/
   (dist[n-3]+dist[n-2]));
  y /= y[n-1];
  return y;
}
```

4.2 实验结果

RBF 算法在 python 平台下实现，其他算法在 Ubpa 平台实现。(暂时不知道如何实现 python 和 c++ 之间的数据传输，故采用了在 c++ 中调用 python 命令行的“笨”方式实现。)

```
...
std::string cmd = "␣../src/hw3/RBF.py";
std::ostringstream fullcmd;
fullcmd<<data->python_interpreter<<cmd<<"␣"<<
 data->learning_rate<<"␣"<<data->step_num<<"␣"<<data->layer;
system(fullcmd.str().c_str());
...
```

如图 2所示，在拟合较多数据点时，RBF 可以达到较好的拟合结果。在图 3和图 4中分别展示了不同参数化对于最小二乘和高斯基函数拟合结果的影响。可以看出在图 3中，当数据点分布比较均匀时，四种参数化类型对结果的影响不大；在图 4中，当数据点分布不均时，均

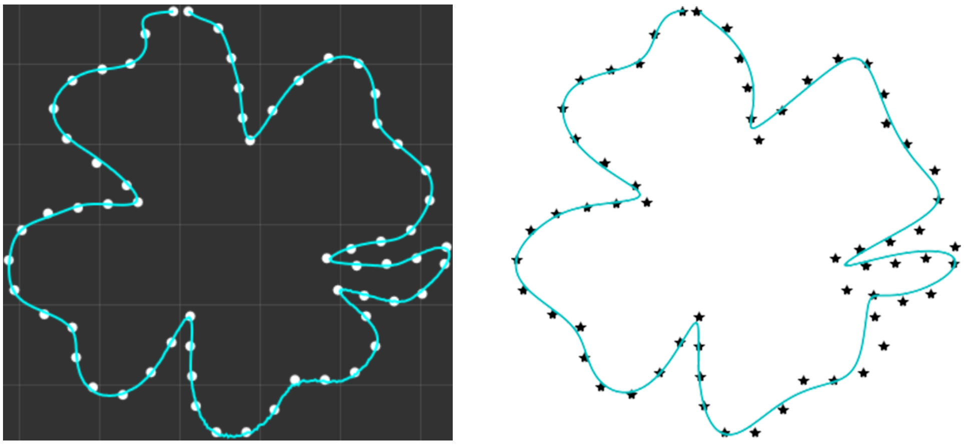

图 2: 高斯基函数型拟合结果 (左) 与 RBF 拟合结果 (右) 对比

匀参数化在间隔较近的点附近会产生较大的缠绕，例如图 4第三幅图中中下方和右上角，拟合曲线出现了缠绕。（更多结果可参见附件录像）

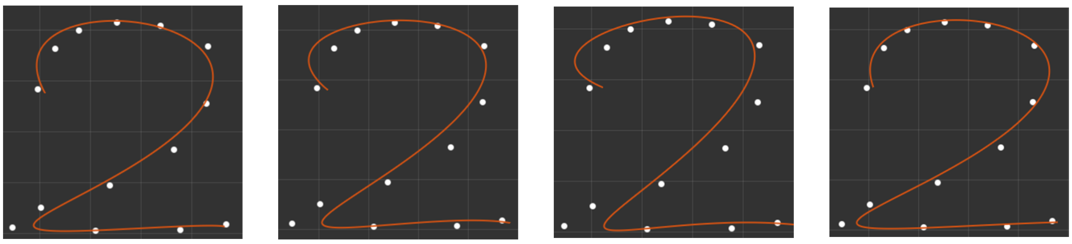

图 3: 不同参数化对于 5 阶最小二乘拟合结果的影响。从左到右参数化类型依次为弦长参数化、中心参数化、均匀参数化和 Foley-Nielsen 参数化

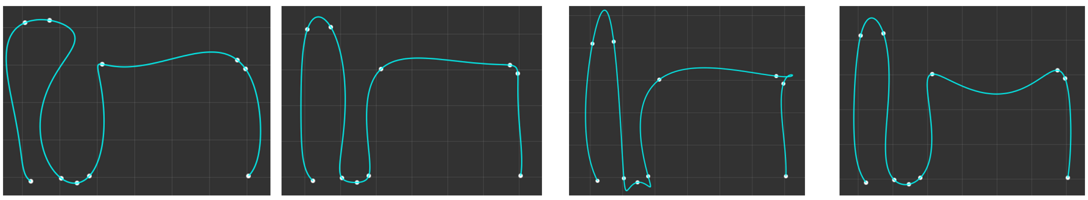

图 4: 不同参数化对高斯基函数拟合结果的影响。从左到右参数化类型依次为弦长参数化、中心参数化、均匀参数化和 Foley-Nielsen 参数化

参考文献

[1] Sıtkı Öztürk, Cengiz Balta, and Melih Kuncan. Comparison of parameterization methods used for b-spline curve interpolation. *European Journal of Technic*, 7:21–32, 06 2017.