

第十九章 曲面的形状调配和变形

曲线曲面的形状调配或融合(Shape blending), 也称为形状平均(Shape averaging), 形状插值(Shape interpolation), 形状演化(Shape evolving)或变形(Metamorphosis/Morphing), 一般指的是在称为关键帧(Key frame)的给定的两张(条)曲面(曲线)之间插入若干张(条)曲面(曲线), 来产生连续光滑的过渡^[1-3]. 这项技术应用于计算机动画, 可减轻美工师的手工劳动; 应用于产品造型设计, 可把不同造型特征相互交融, 产生新的系列造型. 此外它还可用于计算机绘图、计算机美术和计算机视觉等领域.

曲线曲面的变形, 往往转化为空间多边形或多面体的变形来实现. 这是因为曲面可用离散的空间四边形网格或三角网格来逼近, 曲线可用离散的空间多边形来逼近. 如果曲线曲面已经表为 NURBS 形式, 那么我们还可以更简单地把首末两帧的特征网格进行调配来产生中间过渡网格, 进而得到作为中间帧的曲线曲面.

多边形或多面体变形通常被分解为两个关键问题^[3], 一是确定首末两帧的网格中每个顶点的互相对应关系, 简称为顶点对应问题(Vertex correspondence problem); 二是建立一套插值算法, 由首末对应顶点产生一个中间点, 进而得到中间帧的网格, 简称顶点路径问题(Vertex path problem). 这两个问题是相互关联的, 因为不同的顶点对应将导致不同的顶点路径.

在顶点对应方面, 有基于能量的方法, 研究者是 Sederberg^[3](用于二维)和 Korfiatis 等^[4](用于三维); 有多面体投影到球面建立网格对应的方法, 研究者是 Kent^[5]等, 有 Harmonic 映照使多面体网格和平面网格对应的方法, 研究者是 Kanai^[6]等; 有用特征点把多面体网格分片再利用保形映射的方法, 研究者是 Gregory^[7]等; 有对简化网格先作顶点对应而后用逆过程返回原网格顶点对应的方法, 研究者是 Lee^[8]等; 有通过指定骨架曲线对类柱状物体的顶点建立对应的方法, 研究者是 Lazarus^[9]; 有利用“拓扑手术”对亏格不同的物体之间建立顶点对应关系, 研究者是 DeCarlo 等^[10]; 另有 Bethel^[11], Cohen^[12], Kaul^[13]等都作了许多研究. 我们将给出简单曲面之间的顶点对应方法.

在顶点路径方面, 一个最简单的方法——顶点线性插值法. 虽然速度快, 但在变形中会引起网格边界的不正常萎缩. Kent^[14]采用三次 Hermite 插值. Kaul 和 Riossignac^[13]提出基于 Minkowski 和的算法, 使得顶点对应问题和顶点路径问题同时得到解决, 但它很容易混淆物体的相似部分. Hughes^[15]利用隐式曲线曲面给出调配算法, 但它无法给出特征动画所需的细节. Chen 和 Parent^[16]利用三维物体的二维切片进行调配, 再把调配结果重构为三维中间体, 但其变形结果极大地依赖于切片剖分的恰当与否. 此外还有引入参数骨架^[17,18]或移动点^[19]的插值算法等. 上述多种算法的变形效果都不能尽如人意, 其根本原因在于未能体现多边形或多面体的内在结构特征.

基于这种思考和分析, 1993 年, Sederberg 和我们在洛杉矶 SIGGRAPH 国际图形会议上合作报告的一篇论文创新地提出关于平面多边形变形的一个内在解算法, 彻底地抛弃对多边形顶点坐标插值的传统做法, 代之以对于坐标系无关的多边形边长及邻边有向转角的几何内在量插值. 1997 年, 香港大学学者 Sun 等^[20]把边长、两面角和内角作为空间多面体变形的几何参数, 在把内在解算法向空间的推广中做了极为有益的工作, 不过它计算比较复杂, 且不能对空间多边形变形, 因而还不能说是以上方法在三维中的直接推广. 研究表明, 真正的几何内在量, 应当能体现顶点之间的相对位置关系, 可以自动重建出物体外形, 且能用于几何插值来体现物体变形的量. 1998 年初, 我们终于找到了空间点列的一组内在量, 并建立了用于空间多边形或多面体变形的算法 MSI (Moving Spherical coordinate Interpolation).

此外, 鉴于一般变形中只注意两相邻中间帧物体的动作平滑过渡和自然节奏, 而忽略每个中间帧内部几何元素之间的连续性, 我们提出了 Bézier 曲线变形中的平衡化几何连续条件及其算法.

本章系统地介绍我们的研究成果, 其中第 1 节内容取材于[CDR, WGJ, 98]; 第 2 节内容取材于[STW, GPS, WGJ, MH, 93], [LLG, WGJ, 99]; 第 3,4,6 节内容取材于[LLG, WGJ, 99]; 第 5 节内容取材于[LLG, WGJ, 98]; 第 7 节内容取材于[ZHX, WGJ, 2000].

19.1 简单曲面变形的顶点对应算法

三维关键帧中的动画实体，常可分解为由若干简单闭曲面包围而成的子实体。这些简单闭曲面不外乎是柱面，锥面，椭球面，圆环面，长方体表面，棱锥面，双曲抛物面，直纹面，旋转曲面，以及用 B ézier 或 B 样条方法所表示的自由曲面。这就启发我们，可以用先把曲面参数化，再对参数区间进行分割的方法来实行顶点间的对应。

第一步，考察关键帧子曲面的形状，选择适当的参数方程 $\mathbf{F} = \mathbf{F}(u, v) = (x(u, v), y(u, v), z(u, v))$ 和参数区间 $u^{(1)} \leq u \leq u^{(2)}, v^{(1)} \leq v \leq v^{(2)}$ ，使得当 (u, v) 在 $[u^{(1)}, u^{(2)}] \otimes [v^{(1)}, v^{(2)}]$ 中变化时，由方程 $\mathbf{F} = \mathbf{F}(u, v)$ 一次性所描绘出来的曲面为这一子曲面。第二步，在参数区间 $[u^{(1)}, u^{(2)}]$ 和 $[v^{(1)}, v^{(2)}]$ 中分别插入分点 $u_i (i=1, 2, \dots, m-1)$ 和 $v_j (j=1, 2, \dots, n-1)$ ，规定 $\mathbf{P}_{ij} = \mathbf{F}(u_i, v_j)$ 。这样，顶点对应问题就变成调整两关键帧对应的起始顶点问题了。

值得注意的是，对曲面施以参数化的方法不同，或者参数区间的选法不同，或者参数区间的分割方法不同(为方便，一般用等间隔法)，都直接影响到以后的形状调配效果。由于插值显示的实时性，我们很容易进行几种方案的试验比较，从中选择最满意的调配方案。在以下各节里，我们都假定两个关键帧图形中顶点对应关系业已建立。

19.2 平面多边形的内在量及其调配算法

定义 19.2.1 假定 $\mathbf{P}_i (i=0, 1, \dots, n)$ 是平面多边形 Γ 的顶点，记 $\rho_i = \|\mathbf{P}_{i-1}\mathbf{P}_i\|$ ， $i=1, 2, \dots, n$ ，并记由方向 $\mathbf{P}_{i-2}\mathbf{P}_{i-1}$ 到方向 $\mathbf{P}_{i-1}\mathbf{P}_i$ 的有向转角为 $\theta_i, i=2, 3, \dots, n$ ，这里规定 θ_i 为逆时针方向取正值，否则取负值，则称集合 $\Omega = \{\{\rho_i\}_{i=1}^n, \{\theta_i\}_{i=2}^n\}$ 为多边形 Γ 的内在集，称 ρ_i 和 θ_i 为内在变量。见图 19.2.1。

算法 19.2.1 (由 $\Omega \cup \{\mathbf{P}_0, \mathbf{X}_0\}$ 求平面多边形 $\Gamma = \mathbf{P}_0\mathbf{P}_1\dots\mathbf{P}_n$)

设内在集为 Ω ，初始点为 \mathbf{P}_0 ，初始单位向量为 \mathbf{X}_0 ，则平面多边形 Γ 可按以下方式唯一确定：

Step 1. 计算顶点 $\mathbf{P}_1 = \mathbf{P}_0 + \rho_1 \mathbf{X}_0$ 。

Step 2. 对于 $i=2, 3, \dots, n$ ，递归地计算

$$\mathbf{P}_i = \mathbf{P}_{i-1} + \rho_i \frac{\mathbf{P}_{i-2}\mathbf{P}_{i-1}}{\|\mathbf{P}_{i-2}\mathbf{P}_{i-1}\|} \cdot \begin{pmatrix} \cos \theta_i & \sin \theta_i \\ -\sin \theta_i & \cos \theta_i \end{pmatrix}.$$

由定义 19.2.1 和算法 19.2.1 立即得到

定理 19.2.1 无重顶点的平面多边形 Γ 唯一地决定其内在集 Ω ；反之，具有相同内在集 Ω 但不同初始顶点 \mathbf{P}_0 ，不同初始单位边向量 \mathbf{X}_0

的平面多边形 Γ 必有相同的形状，只是位置和朝向有所不同(相差一个刚体运动)，它们分别由 \mathbf{P}_0 和 \mathbf{X}_0 决定。

下面导出平面多边形的调配算法。假定 $\Gamma^j = \mathbf{P}_0^j \mathbf{P}_1^j \dots \mathbf{P}_n^j$ 是首末两个关键闭多边形， $j=0, 1$ ； $\mathbf{P}' = \mathbf{P}_0' \mathbf{P}_1' \dots \mathbf{P}_n'$ 是欲求的相对于时间 $t \in (0, 1)$ 的中间多边形。首先根据位置和朝向适当选取点 \mathbf{P}_0' 和初始单位边向量 \mathbf{X}_0' ，并由 Γ^j 求得内在集 $\Omega^j = \{\{\rho_i^j\}_{i=1}^n, \{\theta_i^j\}_{i=2}^n\}$ ， $j=0, 1$ ；再用线性插值法求得 $\rho_i^t = (1-t)\rho_i^0 + t\rho_i^1$ ， $i=1, 2, \dots, n$ ； $\theta_i^t = (1-t)\theta_i^0 + t\theta_i^1$ ， $i=2, 3, \dots, n$ ；由此就可按算法 19.2.1 得到顶点 $\mathbf{P}_i^t, i=1, 2, \dots, n$ 。显然，按以上算法并不能保证从闭多边形调配得到闭多边形，即 \mathbf{P}_n^t 未必与 \mathbf{P}_0^t 重合。为此，我们对每个 ρ_i^t 赋以调整量 s_i ，令 $\rho_i^t = (1-t)\rho_i^0 + t\rho_i^1 + s_i, i=1, 2, \dots, n$ ，按多边形封闭条件求 s_i 。具体算法可由下节中令 $\beta_i \equiv 0, i=0, 1, \dots, n$ 来得到。

用多边形边长 ρ_i 及邻边有向转角 θ_i 这两个内在量进行调配，可得到相当好的变形效果。图 19.2.2 所示是马奔跑的过程，其中上排图采自一本权威的摄影资料书[21]，下排图的中

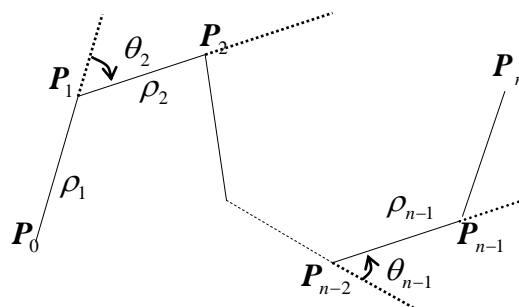


图 19.2.1 平面多边形的内在变量

间二幅由算法 19.2.1 产生.

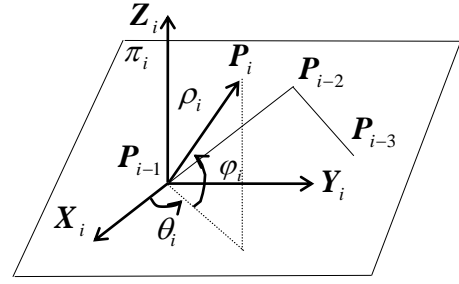


图 19.2.2 奔跑的马

图 19.3.1 空间多边形上顶点 P_{i-1} 处的局部移动坐标系 T_i

19.3 空间多边形的内在量及其调配算法 MSI

19.3.1 内在变量集的定义及其与空间多边形的关系

注意到上节中的 (ρ_i, θ_i) 实质上是在以 P_{i-1} 为原点, $P_{i-2}P_{i-1}$ 为 X 轴的局部坐标系里点 P_i 的极坐标, 我们可以用类似的思想沿着空间多边形移动一个局部坐标系来得到它的内在量集, 进而唯一决定这一空间多边形.

定义 19.3.1. 假定 Γ 是以 P_0, P_1, \dots, P_n 为顶点的空间多边形, X_i 是方向 $P_{i-2}P_{i-1}$ 上的单位向量, Z_i 是方向 $P_{i-3}P_{i-2} \times P_{i-2}P_{i-1}$ 上的单位向量, $Y_i = Z_i \times X_i$, $i = 3, 4, \dots, n$, 见图 19.3.1. 置 $T_i = \{P_{i-1}; X_i, Y_i, Z_i\}$, 并称其为顶点 P_{i-1} 处的局部直角坐标系. 记 $\rho_i = \|P_{i-1}P_i\|$, $i = 1, 2$; 记向量 P_0P_1 和 P_1P_2 间的有向转角为 θ_2 . 记顶点 P_i 在局部直角坐标系 T_i 中的球面坐标为 $(\rho_i, \theta_i, \varphi_i)$, $-\pi < \theta_i \leq \pi, -\pi/2 \leq \varphi_i \leq \pi/2, i = 3, 4, \dots, n$, 则称 $\rho_i, \theta_i, \varphi_i$ 是内在变量, 并称 $\Omega = \{\{\rho_i\}_{i=1}^n, \{\theta_i\}_{i=2}^n, \{\varphi_i\}_{i=3}^n \mid \rho_i > 0, -\pi < \theta_i \leq \pi, -\pi/2 \leq \varphi_i \leq \pi/2\}$ 为多边形 Γ 的内在集.

算法 19.3.1 (由 $\Omega \cup \{P_0, X_0, Z_0\}$ 求空间多边形 $\Gamma = P_0P_1\dots P_n$)

设初始顶点为 P_0 , 初始单位边向量为 X_0 , 初始平面单位法向量为 Z_0 , 又设内在集为 Ω , 则可按以下方式唯一确定空间多边形 Γ .

Step 1. 记过点 P_0 且以 Z_0 为法向的平面为 π_0 , 在 π_0 上按算法 19.2.1, 由集合 $\{\rho_1, \rho_2, \theta_2\} \cup \{P_0, X_0\}$ 求得平面多边形 $P_0P_1P_2$;

Step 2. 对于 $i = 3, 4, \dots, n$, 递归地计算顶点 P_{i-1} 处的局部直角坐标系 T_i ; 递归地计算顶点 P_i 使其关于局部直角坐标系 T_i 的球面坐标是 $(\rho_i, \theta_i, \varphi_i)$.

由定义 19.3.1 及算法 19.3.1 立即得到

定理 19.3.1 空间多边形 Γ 唯一地决定其内在集 Ω ; 反之, 具有相同内在集 Ω 但不同初始顶点 P_0 , 不同初始单位边向量 X_0 , 不同初始平面单位法向量 Z_0 的空间多边形 Γ 必有相同的形状, 只是位置和朝向有所不同 (相差一个刚体运动), 它们分别由 P_0 (称为锚点) 和 (X_0, Z_0) 决定.

由以上定理可知, 内在集刻划了一个空间多边形的基本几何特征. 这里有几个特殊情况必须加以说明. 首先, 当 $\varphi_i \equiv 0$, $i = 3, 4, \dots, n$ 时, 问题就转化为平面多边形的情况; 其次, 当出现共线顶点, 例如 $P_{i-3}, P_{i-2}, P_{i-1}$ 三点共线时, 可把失去定义的顶点 P_{i-1} 处的局部直角坐标系 T_i 取为 T_{i-1} ; 再次, 当出现重顶点, 例如顶点 $P_i, P_{i+1}, \dots, P_{i+k}$ 相重时, 可以用

$\mathbf{P}'_i = \mathbf{P}_i + \varepsilon(\mathbf{P}_{i-1} - \mathbf{P}_i)$, $\mathbf{P}'_{i+k} = \mathbf{P}_{i+k} + \varepsilon(\mathbf{P}_{i+k+1} - \mathbf{P}_{i+k})$, $\mathbf{P}'_{i+j} = (1-j/k)\mathbf{P}'_i + (j/k)\mathbf{P}'_{i+k}$ 分别代替 $\mathbf{P}_i, \mathbf{P}_{i+k}, \mathbf{P}_{i+j}$, $j=1,2,\dots,k-1$, $0 < \varepsilon < 1$. 这样重顶点处的角的插值仍有定义.

19.3.2 空间多边形调配的内在解

对顶点个数相同的首末两个关键多边形, 我们首先计算它们的内在变量, 然后对相应的内在变量插值以求得中间值, 最后重建中间多边形. 这一技术称为活动球面坐标插值, 简称 MSI.

算法 19.3.2. (由首末空间多边形 $\mathbf{F}^j = \mathbf{P}_0^j \mathbf{P}_1^j \dots \mathbf{P}_n^j$, $j=0,1$, 求中间多边形 $\mathbf{F}^t = \mathbf{P}_0^t \mathbf{P}_1^t \dots \mathbf{P}_n^t$)

Step 1. 计算 \mathbf{F}^j 的内在集 Ω^j , $j=0,1$. 若有重顶点, 按上一小节办法处理.

Step 2. 计算内在集 Ω^t :

$$\begin{cases} \rho_i^t = (1-f(t))\rho_i^0 + f(t)\rho_i^1, & i=1,2,\dots,n; \\ \theta_i^t = (1-f(t))\theta_i^0 + f(t)\theta_i^1, & i=2,3,\dots,n; \\ \varphi_i^t = (1-f(t))\varphi_i^0 + f(t)\varphi_i^1, & i=3,4,\dots,n; \end{cases} \quad (19.3.1)$$

这里 $f(t)$ 是调配函数, 满足 $f(0)=0, f(1)=1$. 为简单计, 可取 $f(t)=t$.

Step 3. 确定初始顶点 \mathbf{P}_0^t , 初始单位边向量 \mathbf{X}_0^t , 初始平面单位法向量 \mathbf{Z}_0^t .

Step 4. 利用算法 19.3.1, 由 $\Omega^t \cup \{\mathbf{P}_0^t, \mathbf{X}_0^t, \mathbf{Z}_0^t\}$ 计算顶点 $\mathbf{P}_i^t, i=0,1,\dots,n$.

当首末两个多边形 \mathbf{F}^0 和 \mathbf{F}^1 均为闭多边形时, 应用算法 19.3.2 所得到的中间多边形一般并不是闭的. 于是必须调整多边形的边长和角度使之成为闭. 为方便起见, 我们仅调整边长. 假设 $\mathbf{F}^t = \mathbf{P}_0^t \mathbf{P}_1^t \dots \mathbf{P}_n^t$ 为中间多边形, 其边向量 $\mathbf{P}_i^t \mathbf{P}_{i+1}^t$ 在世界直角坐标系 $T = \{\mathbf{O}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ 中的球面坐标是 $\{l_i, \alpha_i, \beta_i\}, i=0,1,\dots,n-1$. 现在把 l_i 调整为

$$L_i = l_i + s_i, \quad i=0,1,\dots,n-1; \quad (19.3.2)$$

多边形 \mathbf{F}^t 为闭的条件是

$$\begin{aligned} \phi_1(s_0, s_1, \dots, s_{n-1}) &= \sum_{i=0}^{n-1} (l_i + s_i) \cos \alpha_i \cos \beta_i = 0, \\ \phi_2(s_0, s_1, \dots, s_{n-1}) &= \sum_{i=0}^{n-1} (l_i + s_i) \cos \alpha_i \sin \beta_i = 0, \\ \phi_3(s_0, s_1, \dots, s_{n-1}) &= \sum_{i=0}^{n-1} (l_i + s_i) \sin \alpha_i = 0. \end{aligned}$$

我们的目标是确定 s_0, s_1, \dots, s_{n-1} 使得目标函数 $f(s_0, s_1, \dots, s_{n-1}) = \sum_{i=0}^{n-1} s_i^2 / l_i^2$ 在以上三个约束

条件之下为极小. 利用 Lagrange 乘子法, 取 $\lambda_1, \lambda_2, \lambda_3$ 为乘子, 置

$$\Phi(\lambda_1, \lambda_2, \lambda_3, s_0, s_1, \dots, s_{n-1}) = f + \lambda_1 \phi_1 + \lambda_2 \phi_2 + \lambda_3 \phi_3,$$

由

$$\begin{cases} \frac{\partial \Phi}{\partial \lambda_1} = \sum_{i=0}^{n-1} (l_i + s_i) \cos \alpha_i \cos \beta_i = 0, \\ \frac{\partial \Phi}{\partial \lambda_2} = \sum_{i=0}^{n-1} (l_i + s_i) \cos \alpha_i \sin \beta_i = 0, \\ \frac{\partial \Phi}{\partial \lambda_3} = \sum_{i=0}^{n-1} (l_i + s_i) \sin \alpha_i = 0, \\ \frac{\partial \Phi}{\partial s_i} = \frac{2s_i}{l_i^2} + \lambda_1 \cos \alpha_i \cos \beta_i + \lambda_2 \cos \alpha_i \sin \beta_i + \lambda_3 \sin \alpha_i = 0, \quad i=0,1,\dots,n-1, \end{cases}$$

我们得到

$$A\lambda_1 + B\lambda_2 + C\lambda_3 = U, \quad B\lambda_1 + E\lambda_2 + F\lambda_3 = V, \quad C\lambda_1 + F\lambda_2 + G\lambda_3 = W.$$

这里

$$(A, B, C, E, F, G, U, V, W) = \left(\sum_{i=0}^{n-1} l_i^2 (\cos \alpha_i \cos \beta_i)^2, \sum_{i=0}^{n-1} l_i^2 \cos^2 \alpha_i \sin \beta_i \cos \beta_i, \right. \\ \left. \sum_{i=0}^{n-1} l_i^2 \sin \alpha_i \cos \alpha_i \cos \beta_i, \sum_{i=0}^{n-1} l_i^2 (\cos \alpha_i \sin \beta_i)^2, \sum_{i=0}^{n-1} l_i^2 \sin \alpha_i \cos \alpha_i \sin \beta_i, \right. \\ \left. \sum_{i=0}^{n-1} l_i^2 \sin^2 \alpha_i, 2 \sum_{i=0}^{n-1} l_i^2 \cos \alpha_i \cos \beta_i, 2 \sum_{i=0}^{n-1} l_i^2 \cos \alpha_i \sin \beta_i, 2 \sum_{i=0}^{n-1} l_i^2 \sin \alpha_i \right). \quad (19.3.3)$$

于是在正则条件下我们可得出

$$(\lambda_1, \lambda_2, \lambda_3) = \begin{bmatrix} A & B & C \\ B & E & F \\ C & F & G \end{bmatrix}^{-1} \begin{pmatrix} \begin{bmatrix} U & B & C \\ V & E & F \\ W & F & G \end{bmatrix} \begin{bmatrix} A & U & C \\ B & V & F \\ C & W & G \end{bmatrix} \begin{bmatrix} A & B & U \\ B & E & V \\ C & F & W \end{bmatrix} \end{pmatrix}, \quad (19.3.4)$$

$$s_i = (-l_i^2 / 2)(\lambda_1 \cos \alpha_i \cos \beta_i + \lambda_2 \cos \alpha_i \sin \beta_i + \lambda_3 \sin \alpha_i), \quad i = 0, 1, \dots, n-1. \quad (19.3.5)$$

进一步可计算顶点 \mathbf{P}_i^t 的坐标值

$$(x_i, y_i, z_i) = (x_{i-1} + l_{i-1} \cos \alpha_{i-1} \cos \beta_{i-1}, y_{i-1} + l_{i-1} \cos \alpha_{i-1} \sin \beta_{i-1}, z_{i-1} + l_{i-1} \sin \alpha_{i-1}), \\ i = 1, 2, \dots, n. \quad (19.3.6)$$

这样我们得到

定理 19.3.2 当首末关键多边形为闭时, 按(19.3.2)对中间多边形 \mathbf{F}^t 的边长 l_i 进行调整可使 \mathbf{F}^t 为闭, 调整量 s_i 可以由(19.3.3)–(19.3.5)算得. 最后, \mathbf{F}^t 的顶点可以由(19.3.6)算得. 这里 (l_i, α_i, β_i) 是调整前 \mathbf{F}^t 的边向量 $\mathbf{P}_i^t \mathbf{P}_{i+1}^t$ 在世界坐标系中的球面坐标, $i = 0, 1, \dots, n-1$.

19.4 空间四边形网格的形状调配算法

本节考察规则的空间四边形网格的变形, 这种网格的结点具有张量乘积的结构, 内部每个结点与四条边相连. 几乎所有曲面都可用这种网格来逼近.

定义 19.4.1 假设 Λ 是空间四边形网格 $\{\mathbf{P}_{ij}\}_{i=0, j=0}^{m, n}$, 见图 19.4.1. 记

$$\rho_{01} = \|\mathbf{P}_{00} \mathbf{P}_{01}\|, \quad \rho_{10} = \|\mathbf{P}_{00} \mathbf{P}_{10}\|, \quad \gamma = \angle \mathbf{P}_{01} \mathbf{P}_{00} \mathbf{P}_{10}. \quad (19.4.1)$$

根据定义 19.3.1, 记点 $\mathbf{P}_{i,0}$ 在由三点 $\mathbf{P}_{i-3,0}$, $\mathbf{P}_{i-2,0}$, $\mathbf{P}_{i-1,0}$ 决定的局部直角坐标系中的球面坐标是 $(\rho_{i0}, \theta_{i0}, \varphi_{i0})$, $i = 2, 3, \dots, m$, 这里 $\mathbf{P}_{-1,0} = \mathbf{P}_{01}$; 又记点 \mathbf{P}_{0j} 在由三点 $\mathbf{P}_{0,j-3}$, $\mathbf{P}_{0,j-2}$, $\mathbf{P}_{0,j-1}$ 决定的局部直角坐标系中的球面坐标是 $(\rho_{0j}, \theta_{0j}, \varphi_{0j})$, $j = 2, 3, \dots, n$, 这里 $\mathbf{P}_{0,-1} = \mathbf{P}_{10}$. 进一步地, 再记点 \mathbf{P}_{ij} 在由三点 $\mathbf{P}_{i-1,j}$, $\mathbf{P}_{i,j-1}$, $\mathbf{P}_{i,j-1}$ 决定的局部直角坐标系中的球面坐标是 $(\rho_{ij}, \theta_{ij}, \varphi_{ij})$, $i = 1, 2, \dots, m; j = 1, 2, \dots, n$; 则称 $\rho_{ij}, \theta_{ij}, \varphi_{ij}$ 是内在变量, 称 $\Theta = \{\rho_{ij}, \theta_{ij}, \varphi_{ij} \mid 0 \leq i \leq m, 0 \leq j \leq n, i+j \geq 2\} \cup \{\rho_{01}, \rho_{10}, \gamma\}$ 为空间四边形网格 Λ 的内在集.

算法 19.4.1. (由内在集 Θ 的位置和朝向求空间四边形网格 Λ)

Step 1. 按 Λ 的位置和朝向确定三个初始结点 $\mathbf{P}_{00}, \mathbf{P}_{01}, \mathbf{P}_{10}$ 使之适合(19.4.1).

Step 2. 令 $\mathbf{P}_{-1,0} = \mathbf{P}_{01}$, $\mathbf{P}_{0,-1} = \mathbf{P}_{10}$, 利用算法 19.3.1,

2.1 从三点 $\mathbf{P}_{-1,0}, \mathbf{P}_{00}$ 和 \mathbf{P}_{10} 开始, 由内在集 $\{\rho_{i0}, \theta_{i0}, \varphi_{i0}\}_{i=2}^m$ 计算结点 $\mathbf{P}_{i0}, i = 2, 3, \dots, m$;

2.2 从三点 $\mathbf{P}_{0,-1}, \mathbf{P}_{00}$ 和 \mathbf{P}_{01} 开始, 由内在集 $\{\rho_{0j}, \theta_{0j}, \varphi_{0j}\}_{j=2}^n$ 计算结点 $\mathbf{P}_{0j}, j = 2, 3, \dots, n$;

Step 3. 对于 $j = 1, 2, \dots, n$, 和 $i = 1, 2, \dots, m$, 递归地计算

3.1 由三点 $\mathbf{P}_{i-1,j}, \mathbf{P}_{i-1,j-1}$ 和 $\mathbf{P}_{i,j-1}$ 决定的点 $\mathbf{P}_{i,j-1}$ 处的局部直角坐标系 T_{ij} ,

3.2 点 \mathbf{P}_{ij} , 使其在局部坐标系 T_{ij} 中的球面坐标为 $(\rho_{ij}, \theta_{ij}, \varphi_{ij})$.

内在集 Θ 刻划了网格 Λ 的基本几何性质并唯一地决定了其形状. 对已给的两个空间四边形网格, 我们首先计算它们的内在集, 然后对相应的内在变量插值, 最后按算法 19.4.1 重建中间网格, 就完成了整个调配过程.

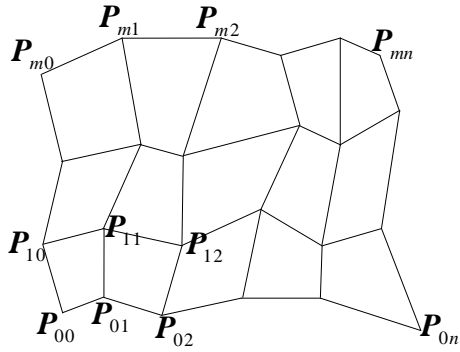


图 19.4.1 空间四边形网格 Λ 的结构

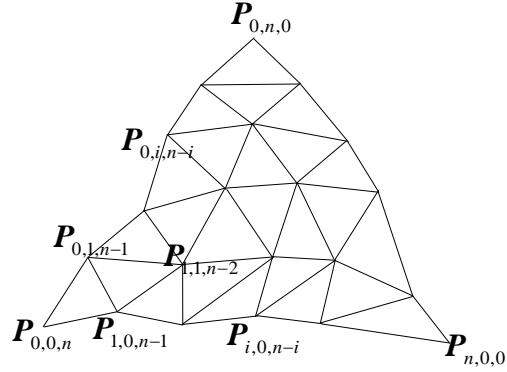


图 19.5.1 n 次 B-B 曲面的 Bézier 网

19.5 空间三角网格的形状调配算法

19.5.1 空间 n 次 Bézier 三角网格的情形

定义 19.5.1 假设空间 n 次 Bézier 三角网格 $\Lambda = \{\mathbf{P}_{i,j,k} \mid i, j, k \geq 0, i + j + k = n\}$ 已经给定,

如图 19.5.1. 记定义 19.3.1 中由三点 $\mathbf{P}_{i-3}, \mathbf{P}_{i-2}, \mathbf{P}_{i-1}$ 决定的局部直角坐标系 $T_i = \{\mathbf{P}_{i-1}; \mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i\}$ 为 $\langle \mathbf{P}_{i-3}, \mathbf{P}_{i-2}, \mathbf{P}_{i-1} \rangle$; 并记

$$\rho_{10} = \|\mathbf{P}_{0,0,n} \mathbf{P}_{1,0,n-1}\|, \rho_{01} = \|\mathbf{P}_{0,0,n} \mathbf{P}_{0,1,n-1}\|, \text{有向角 } \gamma = \angle \mathbf{P}_{0,1,n-1} \mathbf{P}_{0,0,n} \mathbf{P}_{1,0,n-1}; \quad (19.5.1)$$

再设点 $\mathbf{P}_{i,0,n-i}$ 在局部直角坐标系 $\langle \mathbf{P}_{i-3,0,n-i+3}, \mathbf{P}_{i-2,0,n-i+2}, \mathbf{P}_{i-1,0,n-i+1} \rangle$ 下的球面坐标为

$$(\rho_{i,0,n-i}, \theta_{i,0,n-i}, \varphi_{i,0,n-i}), \quad i = 2, 3, \dots, n, \quad \text{其中 } \mathbf{P}_{-1,0,n+1} = \mathbf{P}_{0,1,n-1}; \quad (19.5.2)$$

点 $\mathbf{P}_{0,j,n-j}$ 在局部直角坐标系 $\langle \mathbf{P}_{0,j-3,n-j+3}, \mathbf{P}_{0,j-2,n-j+2}, \mathbf{P}_{0,j-1,n-j+1} \rangle$ 下的球面坐标为

$$(\rho_{0,j,n-j}, \theta_{0,j,n-j}, \varphi_{0,j,n-j}), \quad j = 2, 3, \dots, n, \quad \text{其中 } \mathbf{P}_{0,-1,n+1} = \mathbf{P}_{1,0,n-1}; \quad (19.5.3)$$

点 $\mathbf{P}_{i,j,k}$ 在局部直角坐标系 $\langle \mathbf{P}_{i-1,j,k+1}, \mathbf{P}_{i-1,j-1,k+2}, \mathbf{P}_{i,j-1,k+1} \rangle$ 下的球面坐标为

$$(\rho_{i,j,k}, \theta_{i,j,k}, \varphi_{i,j,k}), \quad i + j \geq 1, \quad i + j + k = n; \quad (19.5.4)$$

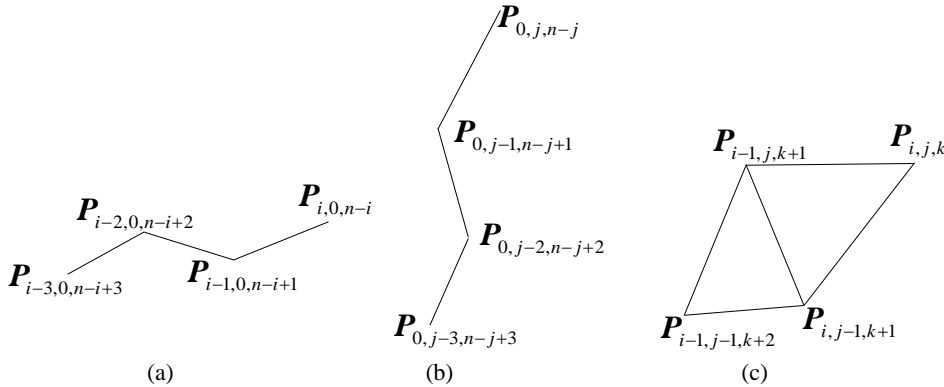


图 19.5.2 Bézier 三角网格中的点与其附近三点的相对位置

分别参考图 19.5.2 (a), (b), (c). 由上述(19.5.1) – (19.5.4)构成的集合

$$\Theta = \{\rho_{i,j,k}, \theta_{i,j,k}, \varphi_{i,j,k} \mid i+j+k=n, i+j \geq 2\} \cup \{\rho_{10}, \rho_{01}, \gamma\}$$

称为 Λ 的内在集, 其元素称为 Λ 的内在变量.

显然, 由 B ézier 网 Λ 可唯一地确定其内在集 Θ ; 反之, 若给定内在集 Θ 及 Λ 的初始位置和方向(锚点 $P_{0,0,n}$ 的空间位置, 边 $P_{0,0,n}P_{1,0,n-1}$ 或 $P_{0,0,n}P_{0,1,n-1}$ 的方向及 $\Delta P_{0,0,n}P_{1,0,n-1}P_{0,1,n-1}$ 所在平面的法向), 我们可求得三角网格 Λ .

算法 19.5.1 (由内在集 Θ 及 Λ 的位置和朝向, 求空间三角网格 Λ)

Step 1. 由 Λ 的初始位置和方向, 确定三点 $P_{0,0,n}, P_{0,1,n-1}$ 和 $P_{1,0,n-1}$, 使得

$$\|P_{0,0,n}P_{1,0,n-1}\| = \rho_{10}, \quad \|P_{0,0,n}P_{0,1,n-1}\| = \rho_{01}, \quad \angle P_{0,1,n-1}P_{0,0,n}P_{1,0,n-1} = \gamma;$$

Step 2. 令 $P_{-1,0,n+1} = P_{0,1,n-1}$, $P_{0,-1,n+1} = P_{1,0,n-1}$, 利用算法 19.3.1, 从三点 $P_{-1,0,n+1}, P_{0,0,n}$ 和 $P_{1,0,n-1}$ 开始, 由内在集 $\{\rho_{i,0,n-i}, \theta_{i,0,n-i}, \varphi_{i,0,n-i}\}_{i=2}^n$ 计算结点 $P_{i,0,n-i}, i=2,3,\dots,n$; 再从三点 $P_{0,-1,n+1}, P_{0,0,n}, P_{0,1,n-1}$ 开始, 由内在集 $\{\rho_{0,j,n-j}, \theta_{0,j,n-j}, \varphi_{0,j,n-j}\}_{j=2}^n$ 计算结点 $P_{0,j,n-j}, j=2,3,\dots,n$.

Step 3. 对于 $j=1,2,\dots,n-1; i=1,2,\dots,n-j$, 递归地计算

3.1 局部直角坐标系 $T = \langle P_{i-1,j,k+1}, P_{i,j-1,k+2}, P_{i,j-1,k+1} \rangle$;

3.2 点 $P_{i,j,k}$, 使其在坐标系 T 之下有球面坐标 $(\rho_{i,j,k}, \theta_{i,j,k}, \varphi_{i,j,k})$.

易知内在集 Θ 唯一确定了三角网格 Λ 的形状. 对于给定的两个空间三角网格, 可用类似于对四边形网格的方法进行调配.

19.5.2 一般空间三角网格的情形

对任一三角网格来说, 其拓扑结构往往不像 B ézier 网那样规则, 可能是很复杂的, 因而要找到一种普遍的内在定义是困难的, 甚至是不可能的. 这里我们利用数据结构的原理引入其 Marching 形式的内在定义.

由于三角网格一般是空间曲面的逼近表示, 可假定三角网格是正则的, 即其中任两个三角形或者互不相交, 或者只有一个公共顶点, 或者只有一条公共边, 且任一顶点的去心领域是连通区域, 即不能出现如图 19.5.3 所示的情况:

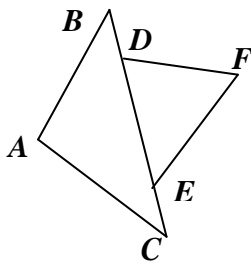


图 19.5.3 三角网格中的奇异情形

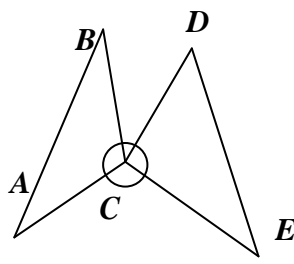


图 19.5.4 Marching 内在定义

由于内在量 (ρ, θ, φ) 体现某一点在其相邻三点中的相对位置关系, 即三点决定一点, 为简单直观起见, 可从某一已知三角形来定义与其具有公共边的三角形的另一顶点. 如图 19.5.4, 若点 A, B, C 已被定义, 则点 D 在局部直角坐标系 $\langle A, B, C \rangle$ 下的球面坐标 $(\rho_D, \theta_D, \varphi_D)$ 就可作为点 D 的内在变量. 我们不妨把点 A, B, C 称为点 D 的前继结点, 点 D 称为点 A, B, C 的后继结点, 可见点 D 的内在变量 $(\rho_D, \theta_D, \varphi_D)$ 体现了点 D 与其前继结点 A, B, C 三点之间的相对位置关系. 这样, 对任一正则三角网格, 从某一初始已知的边界三角形开始, 沿着各三角形的相互连接关系可逐步“爬行”定义其他顶点的内在量 (ρ, θ, φ) , 按遍历顺序依次记录各点的内在变量及其前继结点, 直至无重复地遍历三角形网格的所有顶点. 这种从一个三角形爬行到另一个三角形来决定顶点内在量的方式, 我们称之为三角网格的 Marching 内在定义. 由

三角网格的正则性可知其任一结点均能被遍历定义到。

由已知的三角网格可以 **Marching** 定义出所有顶点的内在变量；反之，若给定某三角网格的 **Marching** 内在变量，则从初始三角形开始，按顺序由已知的三个前继结点和相应的内在变量便可确定它们的后继结点，于是可以逐步重建出整个三角网格。我们利用堆栈遍历^[22]来生成三角网格的 **Marching** 内在变量，使得拓扑结构相同的两个三角网格具有对应的 **Marching** 内在变量。

19.6 自由曲线曲面的调配算法

为了实现两条自由曲线的形状调配，我们可以首先对曲线做分段线性逼近。最简单的方法是在曲线的参数域内作均匀分割，由此得到曲线上一批离散样本点及其依次相连而成的空间多边形。对此多边形进行调配，就得到中间曲线的一个逼近。样本点的取法对调配的效果有很大的影响。

为了实现两张自由曲面的形状调配，可首先将它们用四边形网格或三角网格剖分^[23-25]，再用离散得到的网格按前两节的方法进行调配，其结果就是中间曲面的一个逼近。

用 **MSI** 技术对曲线曲面进行调配的实例见本书彩图 19.6.1 和 19.6.2。彩图 19.6.1 表示了一个葫芦的外壳曲面变形到一张环形圆纹曲面(汽车胎)的过程。彩图 19.6.2 表示了一张三维数字化雕塑的人脸曲面变到一张“S”状曲面的过程。其中每个彩图的首末两幅图都是给定的，中间三幅图是应用 **MSI** 技术在 **SGI IRIX6.2 UNIX** 工作站上生成的，时间参数分别是 0.25, 0.50, 0.75。理论分析与实际试验表明，**MSI** 技术具有以下一些显著的优点：

1. 虽然其原始思想产生于空间多边形的形状调配，但它可以十分容易地推广到空间多面体与自由曲线曲面的变形。
 2. 算法简单直观，计算速度快，能在动画制作中到达实时交互。
 3. 中间多边形或多面体的形状不依赖于首末关键图形的位置与朝向。
 4. 所有的内在几何量，例如边长及有向转角均单调地变化，避免了中间帧形状的扭曲萎缩，图形过渡平滑，动画画面流畅。
 5. 对更复杂的物体，如广义圆柱体或用骨架曲线来定义的普通结构的任意物体也适用。
- 值得提及的是，曲面的调配和变形一直是近几年 **SIGGRAPH** 会议的一个重要议题。例如 1999 年 Turk 等人^[26]在会上报告了基于隐函数的变形方法，2000 年 Alexa 等人^[27]在会上提出了同时考虑区域内部的调配方法，使得在混合过程中尽量保持刚性变形。

19.7 Bézier 曲线形状调配中的平衡化几何连续条件

这一节研究 **Bézier** 样条曲线在调配过程中连续阶衰减的可能性及恢复的方法。设 Z_+ 为非负整数集， Z_- 为非正整数集，我们定义控制顶点的向前差分 and 向后差分：

$$\Delta P_j = P_{j+1} - P_j, \quad j \in Z_+; \quad \nabla P_j = P_j - P_{j-1}, \quad j \in Z_-; \quad (19.7.1)$$

再设初始帧为曲线 $P_0(t)$ ，终了帧为曲线 $P_1(t)$ ，并假设 $P_i(t) \in G^n[0,2]$ ， $i = 0,1$ 。

$$P_i(t) = \begin{cases} u_i(t) = \sum_{j=0}^m B_j^m(t) P_{j-m}^i & , \quad t \in [0,1], i = 0,1. \\ v_i(t) = \sum_{j=0}^{\bar{m}} B_j^{\bar{m}}(t-1) P_j^i & , \quad t \in [1,2], i = 0,1. \end{cases} \quad (19.7.2)$$

这里 $\{P_j^i\}_{j=-m}^{\bar{m}}$ 表示曲线 $P_i(t)$ 的控制顶点，顶点的上角标 i 表示它属于曲线 $P_i(t)$ ， $i = 0,1$ 表示关键帧曲线， $0 < i = \omega < 1$ 则表示中间帧曲线；顶点的非正（负）下角标表示相应曲线的前（后）段。定义调配曲线为 $P_\omega(t) = \text{Blend}(P_0(t), P_1(t); \omega)$ ，如图 19.7.1。相应地有

$$\begin{cases} u_\omega(t) = \sum_{j=0}^m B_j^m(t) P_{j-m}^\omega = \text{Blend}(u_0(t), u_1(t); \omega), & t \in [0,1], \\ v_\omega(t) = \sum_{j=0}^{\bar{m}} B_j^{\bar{m}}(t-1) P_j^\omega = \text{Blend}(v_0(t), v_1(t); \omega), & t \in [1,2]. \end{cases} \quad (19.7.3)$$

为简便起见, 取

$$\text{Blend}(\mathbf{P}_0(t), \mathbf{P}_1(t); \omega) = (1-\omega)\mathbf{P}_0(t) + \omega\mathbf{P}_1(t), \quad 0 \leq \omega \leq 1. \quad (19.7.4)$$

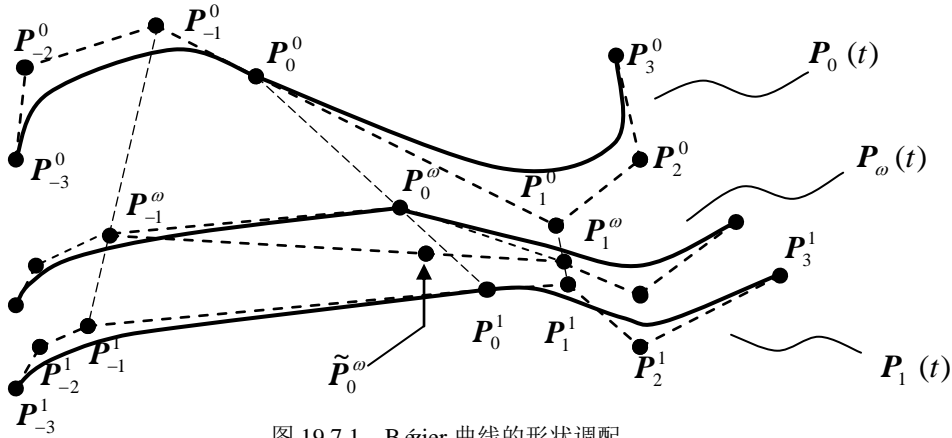


图 19.7.1. Bézier 曲线的形状调配

19.7.1 保持 G^1 的曲线调配

若 $\mathbf{P}_i(t) \in G^1$ ($i=0,1$), 则由定理 10.3.2 和性质 1.3.4 知必有实数 $\beta_1^i > 0, i=0,1$, 使得

$$\bar{m}\Delta\mathbf{P}_0^i = m\beta_1^i\nabla\mathbf{P}_0^i, \quad i=0,1. \quad (19.7.5)$$

不失一般性, 以下假定 $\mathbf{P}_{-1}^i \neq \mathbf{P}_0^i, \mathbf{P}_0^i \neq \mathbf{P}_1^i, i=0,1$. 易知

$$\mathbf{P}_j^\omega = (1-\omega)\mathbf{P}_j^0 + \omega\mathbf{P}_j^1, \quad j=-m, -(m-1), \dots, 0, 1, \dots, \bar{m}, \quad 0 \leq \omega \leq 1. \quad (19.7.6)$$

由(19.7.5),(19.7.6)可知

$$\begin{aligned} G_1 &= \bar{m}\Delta\mathbf{P}_0^\omega - m\beta_1^\omega\nabla\mathbf{P}_0^\omega = \bar{m}[(1-\omega)\Delta\mathbf{P}_0^0 + \omega\Delta\mathbf{P}_0^1] - m\beta_1^\omega[(1-\omega)\nabla\mathbf{P}_0^0 + \omega\nabla\mathbf{P}_0^1] \\ &= m[(1-\omega)(\beta_1^0 - \beta_1^\omega)\nabla\mathbf{P}_0^0 + \omega(\beta_1^1 - \beta_1^\omega)\nabla\mathbf{P}_0^1]. \end{aligned} \quad (19.7.7)$$

但是, $\mathbf{P}_\omega(t)$ 要达到 G^1 , 即要满足 $G_1 \equiv 0$ ($0 < \omega < 1$) 且 $\beta_1^\omega > 0$, 其充要条件是

$$\beta_1^0 = \beta_1^1 = \beta_1^\omega \quad (0 < \omega < 1) \quad (19.7.8)$$

或

$$\begin{cases} \nabla\mathbf{P}_0^1 = \lambda\nabla\mathbf{P}_0^0, \quad (\lambda \neq 0), \\ \beta_1^\omega = \frac{1-\omega}{(1-\omega)+\lambda\omega}\beta_1^0 + \frac{\lambda\omega}{(1-\omega)+\lambda\omega}\beta_1^1, \quad (0 < \omega < 1). \end{cases} \quad (19.7.9)$$

再由已知条件(19.7.5), 可知(19.7.8)又等价于

$$\begin{cases} \|\Delta\mathbf{P}_0^0\|:\|\nabla\mathbf{P}_0^0\| = \|\Delta\mathbf{P}_0^1\|:\|\nabla\mathbf{P}_0^1\|, \\ \beta_1^\omega = \beta_1^0, \quad (0 < \omega < 1). \end{cases} \quad (19.7.10)$$

这样我们就得出

定理 19.7.1 按(19.7.2)定义的两条 G^1 拼接的曲线 $\mathbf{P}_0(t)$ 和 $\mathbf{P}_1(t)$, 其线性混合曲线 $\mathbf{P}_\omega(t)$ ($0 < \omega < 1$) 始终达到 G^1 的充要条件是(19.7.9)成立或(19.7.10)成立.

容易看出, 上述条件是比较苛刻的. 不论如何选取 β_1^ω ($0 < \omega < 1$) 常有 $G_1 \neq 0$, 即 $\mathbf{P}_{-1}^\omega, \mathbf{P}_0^\omega$ 和 \mathbf{P}_1^ω 三点不共线. 由此我们想到, 把顶点 \mathbf{P}_0^ω 调整到直线段 $\mathbf{P}_{-1}^\omega\mathbf{P}_1^\omega$ 上的一点(见图 19.7.1)

$$\tilde{\mathbf{P}}_0^\omega = [m\beta_1^\omega/(\bar{m} + m\beta_1^\omega)]\mathbf{P}_{-1}^\omega + [\bar{m}/(\bar{m} + m\beta_1^\omega)]\mathbf{P}_1^\omega, \quad (19.7.11)$$

式中 β_1^ω 仍按(19.7.9)的第二式计算, 其中

$$\lambda = \|\nabla\mathbf{P}_0^1\|:\|\nabla\mathbf{P}_0^0\| > 0. \quad (19.7.12)$$

这时必有 $\tilde{G}_1 = \bar{m}(P_1^\omega - \tilde{P}_0^\omega) - m\beta_1^\omega(\tilde{P}_0^\omega - P_{-1}^\omega) = 0$. 这个思想可总结为

算法 19.7.1 (保持 G^1 的曲线调配方法)

1. 计算控制顶点 $P_j^\omega = (1-\omega)P_j^0 + \omega P_j^1$, $-m \leq j \leq \bar{m}$, $j \neq 0$;
2. 按(19.7.9)的第二式和(19.7.12)求得 $\beta_1^\omega > 0$, 按(19.7.11)计算 \tilde{P}_0^ω ;
3. 以 $\{P_j^\omega\}_{j=-m, j \neq 0}^{\bar{m}}$, \tilde{P}_0^ω 为控制顶点绘制调配曲线 $\tilde{P}_\omega(t)$, 则 $\tilde{P}_\omega(t) \in G^1[0,2]$.

19.7.2 保持 G^2 的曲线调配

类似 G^1 情况的分析, 我们容易得到

定理 19.7.2 假设参数曲线 $P_0(t)$ 和 $P_1(t)$ 按定理 19.7.1 所定义, 其线性混合曲线 $P_\omega(t)$ ($0 < \omega < 1$) 始终达到 G^2 的充要条件是(19.7.10)成立, 且(19.7.9)的第一式或 $\beta_2^0 = \beta_2^1 = \beta_2^\omega$ 成立.

由此可见 G^2 的条件更苛刻. 一般需要调整 P_j^ω , $j = -1, 0, 1$. 易知 $\{P_j^i\}_{j=-2}^2$ 五点共平面 π^i ($i = 0, 1$), 因而可在平面 π^i 上按形状参数 β_1^i 和 β_2^i 求得多边形 $P_{-2}^i P_{-1}^i P_0^i P_1^i P_2^i$ 的补角多边形 $P_{-2}^i T_{21}^i P_2^i = T_{20}^i T_{21}^i T_{22}^i$, $i = 0, 1$ (见图 19.7.2). 若对点列 T_{2j}^i ($i = 0, 1$) 进行线性调配得到位于平面 π^ω 上的三点 $\tilde{P}_{-2}^\omega = T_{20}^\omega, T_{21}^\omega, \tilde{P}_2^\omega = T_{22}^\omega$, 则按类似于上述补角多边形产生的逆过程, 由割角产生的多边形 $\tilde{P}_{-2}^\omega \tilde{P}_{-1}^\omega \tilde{P}_0^\omega \tilde{P}_1^\omega \tilde{P}_2^\omega$ 必共面且满足 $\tilde{G}_2 = 0$. 这一思想可推广到保持 G^n ($n > 2$) 的曲线调配, 我们不妨称由此得到的条件为平衡化几何连续条件. 图 19.7.3 给出了不使用或使用上述调整方法对旋转体母线进行形状调配的情形.

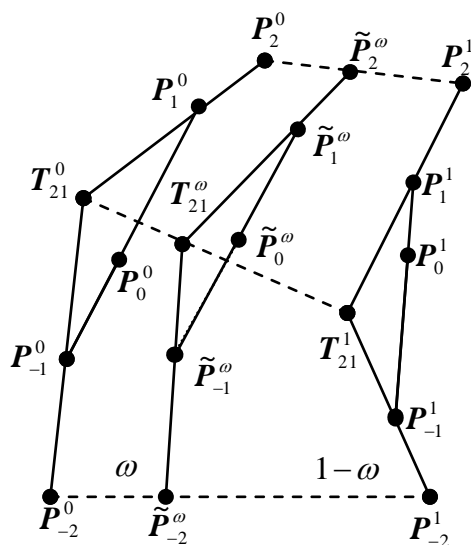


图 19.7.2 保持 G^2 的调整

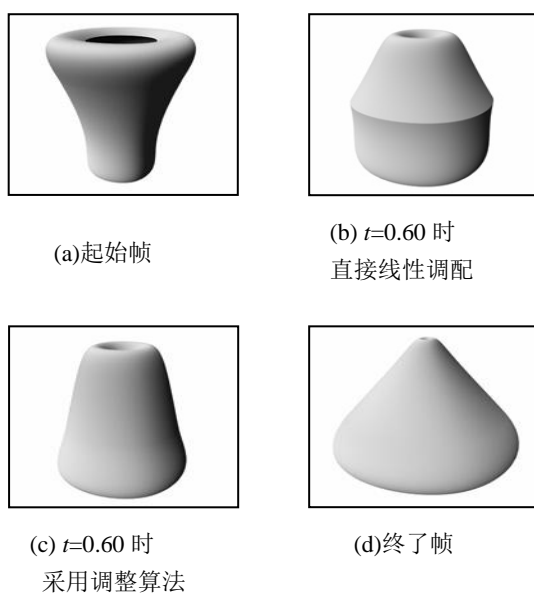


图 19.7.3 对旋转体的母线进行曲线调配比较

主要文献

- [CDR, WGJ, 98] 陈动人, 王国瑾, 简单曲面的顶点对应算法及其关键帧动画实现, 高校应用数学学报, 1998, 13A(增刊): 111-116
- [STW, GPS, WGJ, MH, 93] Sederberg, T.W., Gao Peisheng, Wang Guojin, Mu Hong, 2-D Shape blending: an intrinsic solution to the vertex path problem, Proceedings of SIGGRAPH'93, Computer Graphics, 1993, 27(4): 15-18

-
- [LLG, WGJ, 99] Liu Ligang, Wang Guojin, Three-dimensional shape blending: intrinsic solutions to spatial interpolation problems, *Computers and Graphics*, 1999, 23(4): 535-545
- [LLG, WGJ, 98] 刘利刚, 王国瑾, 基于内在变量的空间三角网格的形状混合, *计算机学报*, 1998, 21(9): 850-854
- [ZHX, WGJ, 2000] 张宏鑫, 王国瑾, 保持几何连续性的曲线形状调配, *高校应用数学学报*, 即将发表

参考文献

- 1 Watt, A., Watt, A., *Advanced animation and rendering techniques*, Addison-Wesley, 1992
- 2 Catmull, E., The problems of computer-assisted animation, *Computer Graphics*, 1978, 12: 348-353
- 3 Sederberg, T.W., Greenwood, E., A physically based approach to 2-D shape blending, *Proceedings of SIGGRAPH 92*, *Computer Graphics*, 1992, 26(2): 25-34
- 4 Korfiatis, I., Paker, Y., Three-dimensional object metamorphosis through energy minimization, *Computers and Graphics*, 1998, 22(2-3): 195-202,
- 5 Kent, J.R., Parent, R.E., Carlson, W.E., Establishing correspondences by topological merging: a new approach to 3-D shape transformation, *Proceedings of Graphics Interface '91*, Calgary, Canadian Info. Processing Society, 1991, 271-278
- 6 Kanai, T., Suzuki, H., Kimura, F., 3D Geometric metamorphosis based on harmonic map, *The Visual Computer*, 1998, 14(4): 166-176
- 7 Gregory, A., State, A., Lin, M., Manocha, D., Livingston, M., Interactive surface decomposition for polyhedral morphing, *The Visual Computer*, 1999, 15(9): 453-470
- 8 Lee, W.F., Dobkin, D., Sweldens, W., Schröder, P., Multiresolution mesh morphing, *Proceedings of SIGGRAPH '99*, 1999, 343-350
- 9 Lazarus, F., Verroust, A., Metamorphosis of cylinder-like object, *International Journal of Visualization and Computer Animation*, 1997, 8:131-146
- 10 DeCarlo, D., Gallier, J., Topological evolution of surfaces, *Proceedings of Graphics Interface '96*, Toronto, Canada, 1996, 194-203
- 11 Bethel, E.W., Uselton, S.P., Shape distortion in computer-assisted key frame animations, in: Thalmann, N.M. and Thelmann, D. eds., *State of the Art in Computer Animation 89*, Springer-Verlag, New York, 1989, 215-224
- 12 Cohen, S., Elber, G., Yehuda, R.B., Matching of freeform curves, *Computer Aided Design*, 1997, 29(5): 369-378
- 13 Kaul, A., Riessnig, J., Solid interpolating deformations: construcion and animation of PIPs, in: Post, F.H. and Barth, W., eds., *Proceedings of Eurographics '91*, Elsevier Science Pulishers, B.V., 1991
- 14 Kent, J.R., Carlson, W.E., Parent R E., Shape transformation for polyhedral objects, *Proceedings of SIGGRAPH'92*, *Computer Graphics*, 1992, 26(2): 47-54
- 15 Hughes, J.F., Scheduled Fourier volume morphing, *Proceedings of SIGGRAPH'92*, *Computer Graphics*, 1992, 26(2): 43-46
- 16 Chen, S.E., Parent, R.E., Shape averaging and its applications to industrial design, *IEEE Computer Graphics and its Applications*, 1989, 9(1): 47-54
- 17 Burtnyk, N., Wein, M., Interactive skeleton techniques for enhancing motion dynamics in key frame animation, *Communications of the ACM*, 1976, 19(10): 564-569
- 18 Shapira, M., Rappoport, A., A shape blending using the star-skeleton representation, *IEEE Computer Graphics and its Application*, 1995, 15(2): 44-50
- 19 Reeves, W.T., Inbetweening for computer animation utilizing moving point constrain, *Proceedings of SIGGRAPH'81*, *Computer Graphics*, 1981, 15(3): 263-269
- 20 Sun, Y.M., Wang, W.P., Chin, Y.L., Interpolating polyhedral models using intrinsic shape parameters, *The Journal of Visualization and Computer Animation*, 1997, 8: 81-96
- 21 Eadweard, M., *Animals in Motion*, Dover Publications, Inc., New York, 1957
- 22 Mauren, H.A., *Data structures and Programming Techniques*, Prentice-Hall, Englewood, Cliffs N J, 1977
- 23 Fjälström, P.O., Evaluation of a Delaunay-based method for surface approximation, *Computer Aide Design*, 1993, 25(11): 711-719
- 24 陈永府, 张华, 陈兴, 李德群, 任意曲面的三角网格剖分, *计算机辅助设计与图形学学报*, 1997, 9(5): 396-401
- 25 周晓云, 刘慎权, 实现约束 Delaunay 三角剖分的健壮算法, *计算机学报*, 1996, 19(8): 615-624
- 26 Turk, G., O'Brien, J.F., Shape transformation using variational implicit functions, *Proceedings of SIGGRAPH'99*, *Compuer Graphics*, 1999, 335-342
- 27 Alexa, M., Cohen-Or, D., Levin, D., As-rigid-as-possible shape interpolation, *Proceedings of SIGGRAPH'2000*, *Computer Graphics*, 2000, 157-163