

## 形状插值算法综述

张智邦, 李桂清\*, 韦国栋, 王宇攀, 路慧娜, 袁 青

(华南理工大学计算机科学与工程学院 广州 510006)  
(ligq@scut.edu.cn)

**摘 要:** 给定 2 个以上的几何模型, 形状插值的目的是生成一系列新模型, 能够建立给定模型之间自然的形状过渡. 形状插值是计算机动画中的一个基本操作. 文中以插值量为主线, 对多边形和网格模型形状插值的相关工作进行分析梳理; 讨论插值结果应满足的一些重要性质, 提出分类方法, 对一些重要工作进行较详细的介绍与评述, 并分析它们的优缺点; 特别地, 深入探讨各种方法的大变形插值能力. 最后, 总结存在的问题并尝试提出一些新的研究方向.

**关键词:** 计算机图形学; 形状插值; 大变形; 插值量  
中图法分类号: TP391.41

## Survey on Shape Interpolation

Zhang Zhibang, Li Guiqing\*, Wei Guodong, Wang Yupan, Lu Huina, and Yuan Qing

(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006)

**Abstract:** Given two or more geometric models, the goal of shape interpolation is to create a sequence of intermediate models which establish natural smooth transition between given shapes. We review existing shape interpolating approaches from the viewpoint of quantities of intermediate shapes obtained by linearly interpolating those of the given shapes. Some important properties of shape interpolating approaches are summarized, classification methodology is discussed, some important interpolation methods are described and commented, and their pros and cons are analyzed. Particularly, we investigate the ability of these approaches to deal with interpolation with large deformation. Finally, we draw some open problems in shape interpolation.

**Key words:** computer graphics; shape interpolation; large deformation; interpolation quantity

形状插值是计算机动画的基础, 也是通过已有几何模型获取新模型的重要手段之一. 形状插值(shape interpolation)关注的是给定几何模型之间的形状过渡, 其结果反映如图 1 所示从一个形状到另一个形状的变形(deformation)过程中. 早期的工作更多地从插值序列是否兼具被插值关键帧形状

的角度来看待形状插值, 所以称之为形状混合(shape blending); 或者为了强调其变化过程, 而称之为变形(morphing). 当给定几何模型是同一物体的不同姿态时, 形状插值也被称为姿态插值(pose interpolation), 此时的插值序列反映的是物体的运动(motion)情况. 根据插值结果在序列中的时序

收稿日期: 2015-07-03; 修回日期: 2015-07-14. 基金项目: 国家自然科学基金青年科学基金(61300136); 中央高校基本科研业务费(2013ZM087); 教育部博士点基金(20130172120010); 广东省自然科学基金(S2013020012795). 张智邦(1984—), 男, 博士研究生, 主要研究方向为计算机图形学、数字几何处理; 李桂清(1966—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 论文通讯作者, 主要研究方向为计算机辅助几何设计、数字几何处理、动态几何处理; 韦国栋(1992—), 男, 博士研究生, 主要研究方向为几何建模与处理、计算机图形学; 王宇攀(1990—), 男, 博士研究生, 主要研究方向为几何建模与处理、计算机图形学; 路慧娜(1990—), 女, 硕士研究生, 主要研究方向为几何建模与处理、计算机图形学; 袁 青(1987—), 男, 博士研究生, 主要研究方向为计算机图形学、动态几何处理.



图 1 一个形状到另一个形状的变形过程

位置, 形状插值分为内插(interpolation)和外插(extrapolation) 2 种情况; 内插的结果建立关键帧之间的过渡; 而外插则是通过分析关键帧的变形或运动趋势, 估计所有关键帧之前或之后的姿态。

理论上, 形状插值中的关键帧可以是任意多个形状完全不同的模型集, 不过目前的方法大都以讨论 2 个形状之间的插值为主, 多个形状插值基本类似。此外, 插值之前需要建立被插值形状之间点的对应关系。本文主要讨论以多边形、平面网格、三维表面网格以及体网格所表示的形状之间的插值方法, 并且要求被插值网格具有相同的连接关系。

不同形状(表面)之间的对应问题本身独立于形状插值存在, 主要研究如何寻找不同形状之间的映射函数。事实上, 目前的形状插值所采用的关键帧, 来源主要有 2 种方式: 1) 对单个关键帧进行变形编辑获得其他关键帧; 2) 通过建立模型之间的点对应进行兼容剖分(compatible triangulation)<sup>[1-2]</sup>, 其往往也需要一定的交互。

对应问题解决后, 须要决定对应点的运动路径。在实际应用中, 人们希望形状插值的(路径)结果能够符合客观物理规律或主观视觉经验。因此, 为了追求更加逼真插值效果, 研究人员从不同的角度进行了大量的研究。本综述将主要讨论这方面的内容。

## 1 基本概念

在评述具体的插值方法之前, 我们首先引入一些记号, 然后描述判定插值方法优劣的一些性质, 并简要讨论形状插值方法的分类方式。

### 1.1 形状插值的形式化描述

前面已经提到, 我们考虑的形状由多边形或网格表示, 而且所有帧顶点数相同, 每一帧所有顶点的坐标连接在一起构成高维空间中的点, 我们把所有这样的点所构成的空间称为形状空间<sup>[3]</sup>, 记为  $\mathcal{S}$ 。给定关键帧  $S^0, S^1 \in \mathcal{S}$ , 形状插值(内插)可用函数  $\phi: \mathcal{S} \times \mathcal{S} \times [0, 1] \rightarrow \mathcal{S}$  来描述, 即

$$S^t = \phi(S^0, S^1, t), \quad t \in [0, 1] \quad (1)$$

其中,  $t$  是时间;  $S^t$  是时刻  $t$  的形状。计算  $t \in \mathbb{R} - [0, 1]$  时的形状  $S^t$  则是外插。在离散情形下,  $S^t$  的下一帧记为  $S^{t+\Delta t}$ 。我们总是用上标  $t$  表示与  $S^t$  有关的量, 用单个下标表示点, 2 个下标表示边, 3 个下标表示三角形, 如  $p_i^t$  为  $S^t$  中第  $i$  个点得坐标(列向量),  $e_{ij}^t$  为由点  $p_i^t$  指向  $p_j^t$  的有向边。

### 1.2 插值方法性质

尽管每个形状插值方法都会声称自己具有某些新特性, 但对插值方法应该满足什么样的性质并没有确切的定义和系统的研究, 多数情况下也没有提出量化新特性的方法。根据文献[4], 本文列举如下一些特性, 用以衡量一个形状插值算法的好坏:

- 插值. 插值序列经过 2 个关键帧, 即  $S^0 = \Phi(S^0, S^1, 0)$  和  $S^1 = \Phi(S^0, S^1, 1)$ 。
- 对称. 插值的结果与运动方向无关, 即  $\Phi(S_0, S_1, t) = \Phi(S_1, S_0, 1-t)$ 。
- 平滑. 即顶点路径是光滑的,  $\partial\Phi/\partial t$  存在且有界。
- 仿射变换重现(affinity transformation reproduction). 如果通过一个仿射变换能把  $S^0$  变为  $S^1$ , 那么中间帧也应由  $S^0$  作仿射变换得到。
- 有界扭曲(bounded distortion). 主要是度量网格曲面中的三角形或体网格中的四面体的变形情况。对关键帧中的 2 个对应三角形  $(p_i^0, p_j^0, p_k^0)$  和  $(p_i^1, p_j^1, p_k^1)$ , 把前者变换为后者的  $3 \times 3$  仿射变换矩阵为  $A = (p_i^1 \ p_j^1 \ p_k^1) (p_i^0 \ p_j^0 \ p_k^0)^{-1}$ 。分别记  $\lambda_{\max}$  和  $\lambda_{\min}$  为  $AA^T$  的最大和最小特征值, 则该变换的扭曲程度可用  $K = \sqrt{\lambda_{\max} / \lambda_{\min}}$  来衡量。因为被插值帧不相同, 变形不可避免。所谓有界扭曲是指三角形的扭曲程度在运动过程中随时间单调增长。
- 大变形(large deformation). 所谓大变形插值问题是指  $S^0$  要经过较大的变形才能得到  $S^1$ , 如何能构造出合理的、非退化的形状过渡序列。很多插值方法都声称能处理大变形问题, 但都没有对大变形进行明确的界定。对于插值方法来说, 大变形主要是指  $S^1$  中的局部区域

相对于  $S^0$  的位置有较大的旋转角.

### 1.3 分类方法

按照插值的模型所嵌入的空间维度,可分为二维形状插值、三维形状插值. 而二维形状插值又可分为平面多边形插值和平面网格插值; 相应地, 三维形状插值分为网格曲面插值和体网格插值. 按照插值的优化能量来分则可分为线性插值和非线性插值. 按处理插值量所依附的体元来分, 可有以点为中心的插值、以边为中心的插值和以面为中心的插值.

插值的最终结果是计算出中间帧的顶点坐标. 因此, 最简单的形式是对模型的对应顶点作线性混合<sup>[5]</sup>, 即

$$p^t = (1-t)p^0 + tp^1 \quad (2)$$

线性混合其实就是认为模型上所有点遵循直线运动, 显然与绝大部分物体表面运动规律不符, 得到的结果往往会有严重变形. 因此需要寻找一些量, 其变化是线性的, 或者可由线性变化的量计算得到, 而由这些量可重构三维模型顶点坐标. 对已有形状插值方法进行归纳, 本文把插值量分为如下 5 类:

- 1) 几何内蕴量. 例如多边形或网格的夹角、边长、面积、体积、曲率等.
- 2) 几何量的运动变换. 2 个关键帧中间对应局部几何量之间的运动变换. 典型的如对应三角形之间的变换矩阵.
- 3) 连接映射的运动变换. 在模型内部相邻几何量之间存在空间变换关系, 对首尾帧的对应空间变换关系进行插值, 据此重构中间帧的顶点坐标.
- 4) 全局约束量. 这类方法不是从关键帧插值出中间帧的某种几何量逐帧进行重构, 而是加入相邻帧间的约束, 从而对所有帧进行整体求解. 这类方法一般都是要作非线性优化, 因此需要用前述方法进行初始化(给出初始序列).
- 5) 其他量.

下面几节将按照上述分类介绍一些具体方法. 考虑到文献[6-8]对 2002 年之前的工作有过很好的评述, 本文将侧重 2002 年之后的工作.

## 2 几何内蕴量形状插值

如 1.3 节提到, 常见的几何内蕴量有夹角、边长和体积这样的简单几何量, 也有曲率或更复杂的微分几何量, 下面分别进行讨论. 简单几何量和

复杂的微分几何量, 有些是互相关联的, 比如夹角跟曲率有关, 而边长也可看成近似的切向量长度. 尽管如此, 我们分别进行讨论.

### 2.1 简单几何内蕴量

最简单的几何内蕴量是诸如边长、夹角和体积之类的统计量. 如果我们强制要求  $S^t$  的边长、边相邻三角形面的法向夹角和体积都作线性变化<sup>[9]</sup>, 即

$$\begin{cases} e_{ij}^t = (1-t)e_{ij}^0 + te_{ij}^1 \\ \theta_{ij}^t = (1-t)\theta_{ij}^0 + t\theta_{ij}^1 \\ V^t = (1-t)V^0 + tV^1 \end{cases} \quad (3)$$

注意, 边  $(i, j)$  的长度  $\|e_{ij}^t\|$ , 共享边  $(i, j)$  的 2 个三角形法向量的夹角  $\theta_{ij}^t$ , 网格的总体体积  $V^t$  (如果考虑的是平面多边形形状, 则代表面积) 均可表示为  $S^t$  中顶点的函数. 因此最小化能量

$$\mathcal{E} = \lambda \mathcal{E}_{\text{edge}} + \mu \mathcal{E}_{\text{angle}} + \nu \mathcal{E}_{\text{volume}} \quad (4)$$

可以重构中间帧的形状<sup>[9]</sup>. 其中

$$\begin{cases} \mathcal{E}_{\text{edge}} = \sum_{(i,j) \in S} (e_{ij}^t - (1-t)e_{ij}^0 - te_{ij}^1)^2 \\ \mathcal{E}_{\text{angle}} = \sum_{(i,j) \in S} (\theta_{ij}^t - (1-t)\theta_{ij}^0 - t\theta_{ij}^1)^2 \\ \mathcal{E}_{\text{volume}} = (V^t - (1-t)V^0 - tV^1)^2 \end{cases} \quad (5)$$

基于上述模型, 不少工作围绕几何内蕴量展开. 这些方法的不同之处主要在于几何内蕴量的构造方式和非线性能量的优化方法.

Sederberg 等<sup>[10]</sup>最早针对平面多边形表示的形状研究这类插值方法, 其插值量是多边形边长和相邻边的外夹角. 将时刻  $t$  的多边形边长  $e_{ij}^t$  简化为  $e_{i+1}^t$ , 而时刻  $t$  的相邻三角形法向角  $\theta_{ij}^t$  由多边形中相邻边的外夹角  $\theta_{i+1}^t$  代替. Sun 等<sup>[11]</sup>将这种思想拓展到三维情形, 关键帧由(单个)多面体表示; 采用传播的方法遍历由多面体所确定的面图和点图, 计算比较复杂.

Liu 等<sup>[12]</sup>将类似思想与寻找平面多边形对应特征点问题相结合, 得到更好的二维平面多边形插值效果.

Liu 等<sup>[13]</sup>处理了三维封闭折线形状的插值问题, 对折线上的每个点建立微分标架, 这样每个顶点相对于其邻点的标架都有一个球面坐标, 球面坐标的半径即是边长. 该球面坐标事实上是一组内蕴量, 中间帧的球面坐标可用关键帧的线性插值得到; 他们还把该方法推广到多边形网格模型

插值问题.

Johan 等<sup>[14]</sup>针对图像变形过渡问题, 利用 Bézier 样条逼近用户指定的对应特征曲线, 所有样条端点构成连接图, 然后对连接图的边长和夹角进行插值. 类似地, Meng 等<sup>[15]</sup>和针对图像提出基于素描交互的变形过渡算法, 在用户通过若干笔画提取对应特征后, 将这些笔画转换为折线段并完成对应, 而折线段的拓扑关系使用连接图来表示. 该算法为连接图定义了内蕴量, 分别是图的边长、边之间的夹角; 通过选定锚点和旋转基线来表示连接图整体的平移和旋转, 并插值这些量来获得中间图像的连接图.

Winkler 等<sup>[16]</sup>考虑文献[15]的特殊情形( $v=0$ ), 使用了基于多配准的层次化插值技术, 相比文献[15], 其显得更加复杂; 另一方面, 层次化使用了自底向上的合并策略, 而该策略涉及种子点的选取和扩散, 与插值结果密切相关. 因此, 需用特殊的方法选取种子点以确保结果的质量. 由于该方法通过层次拼接的方法构造中间帧, 因此能较好地处理大变形插值. Baek 等<sup>[17]</sup>的工作与之类似, 不过除了边长外, 还线性插值相邻三角形标架之间的 2 个转角, 如图 2<sup>[17]</sup>所示.

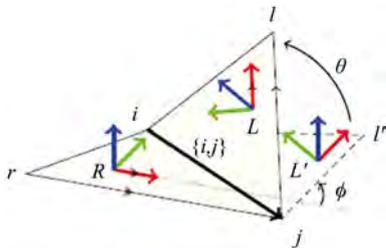


图 2 插值 2 个三角形标架的旋转角  $\theta$  和  $\phi$

相比起上述直接在几何体表面或边界提取内蕴量外, 也有一些方法尝试在几何物体内部构造内蕴量. Shapirat 等<sup>[18]</sup>观察到早期的插值方法只考虑二维几何对象边界, 在大变形情况下, 有时会导致局部扭曲和边相交; 于是提出在几何对象内部建立星型骨架的方法. 随后, Blanding 等<sup>[19]</sup>将此思想推广至三维.

## 2.2 微分几何内蕴量插值

从微分几何的角度看, 内蕴量可以由微分度量(metrics)确定. 因此, 如果能插值微分度量, 就相当于插值内蕴量. Lipman 等<sup>[20]</sup>考虑网格顶点一邻域的情况, 与直接插值与网格朝向有关微分坐标不同, 其建立从嵌入空间到局部空间之间的映射, 局部空间由附着于顶点的离散局部标架来描

述, 如图 3 所示; 并利用标架编码一邻域的信息, 以旋转不变的第一离散型和第二离散型表示<sup>[21]</sup>. 图 3 中紫色向量分别投影至 2 个切平面, 投影用红色的切向量表示;  $\mathcal{F}_0$  和  $\mathcal{F}_1$  分别是 2 个切平面上定义的局部标架, 它们之间的局部朝向关系通过紫色向量联系. 这 2 种离散型可以看作空间度量的一种离散化结果, 第一离散型描述一邻域边向量在切平面上投影的位置关系, 包括长度和夹角; 第二离散型则描述了沿法向的长度. 相邻标架之间通过上述编码信息互相表征彼此的相对位置, 进而能够在给定目标插值量后, 获得标架在内嵌空间中的坐标表示. 最后, 其通过线性插值长度和夹角重构中间网格. Baran 等<sup>[22]</sup>扩展了上述方法, 考虑由多三角面片组成的子网格之间的情形. Chen 等<sup>[14]</sup>插值了由二维网格之间双射映射所诱导出的度量

$$g^t = (1-t)I + tF^T F \quad (6)$$

其中  $F$  是由映射函数确定的变形梯度. 其随后证明了这种插值方式能限制中间帧每个三角形的扭曲程度, 并使之随时间单调改变.

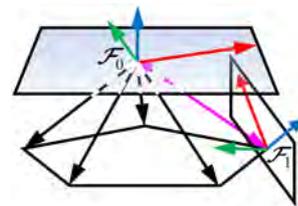


图 3 顶点一邻域示意图

## 2.3 重心坐标插值

Floater 等<sup>[23]</sup>研究了针对二维平面上边界相同和不同的 2 种凸包 tiling 形状之间的插值. 对于相同边界的 2 个形状, 中间帧的顶点坐标用邻域的平均凸组合进行表示, 而非简单地用首尾 2 帧对应点的线性组合. 若边界不同, 则用极坐标表示边界上的点, 先对边界进行凸组合插值, 然后再对内部点采用邻域凸组合进行插值. Gotsman 等<sup>[24]</sup>随后扩展了这种思路, 提出了一种基于全局范围(global scale)平面多边形插值方法, 以保证中间帧不会自相交; 并对关键帧增加少量的虚拟点进行兼容三角剖分, 将这 2 个三角化之后的形状置于同一凸包中, 用平均重心坐标方法<sup>[23]</sup>进行中间形状的插值.

## 2.4 混合插值

Yang 等把类刚性运动(as-rigid-as-possible, ARAP)<sup>[25]</sup>应用到二维多边形的形状插值<sup>[26]</sup>. 他们采用类似多分辨率的思路, 将平面多边形插值

问题分解为粗糙轮廓插值和细节特征插值 2 部分. 首先, 构造与关键帧形状相近但顶点少得多的所谓帧多边形, 对帧多边形中所有顶点与其相邻的 2 个顶点构成的三角形, 利用类似的 ARAP 插值构造中间三角形, 从而得到中间帧的帧多边形. 其将细节附着于帧多边形的每条边上, 并在由该边所定义的局部坐标系下用折线段来表示; 于是, 中间帧的细节便能通过折线段的边长和夹角进行插值获得, 而其朝向由局部坐标轴的朝向决定. Marras 等<sup>[27]</sup>针对具有关节的模型提出基于混合形状空间的高效插值算法. 首先, 分割关键帧获得刚性块和关节; 接着, 对刚性块进行多配准, 即为对应的刚性块寻找刚性变换使之对齐后, 对顶点位置进行线性插值; 然后, 采用文献[16]方法对关节进行插值; 最后, 重构整个中间网格.

### 3 几何内蕴量形状插值

直接建立关键帧之间的外蕴几何量变化关系, 然后对此关系进行插值则是另外一类常见的方法. 其中以 ARAP 为代表衍生出大量相关工作. ARAP 类插值量是局部旋转矩阵的旋转角, 其原理是局部区域的变换矩阵尽可能遵循刚性运动<sup>[28]</sup>

$$\mathcal{E}(\varphi) = \int_{S^0} \min_{R \in SO(3)} \|d\varphi - Rp\|^2 dp \quad (7)$$

微分量  $d\varphi$  可以理解为  $\varphi$  的线性近似, 即  $d\varphi = Fp$ ,  $p \in S^1$ .

$$\mathcal{E}(S^0, S^1) = \sum_{k=1}^m \min_{R_k \in SO(3)} \sum_{(i,j) \in E_k} c_{ijk} \|e_{ij}^0 - R_k e_{ij}^1\|^2 \quad (8)$$

能量式(8) 可理解为  $S^1$  中的局部边集  $E_k$  上的点尽可能遵循相同的刚性运动  $R_k$ . 如果  $E_k$  取遍  $S^1$  中的所有三角形, 这是通常的 ARAP 方法<sup>[29-30]</sup>. 由于没有对模型内部进行约束, 这类方法会由于挤压而导致不平滑过渡. 为此, Chao 等<sup>[28]</sup>对模型内部也进行刚性运动约束, 此时式(7)是对  $S^0$  的体积分, 离散求解时  $E_k$  取遍体网格  $S^0$  的所有四面体.

通过线性插值  $R_k$  的旋转角  $\theta_k$  就可以得到中间帧的旋转矩阵  $R_k^t$ . 但显然纯刚性的插值是不存在的, 拉伸变换部分直接对变换矩阵进行插值, 最后计算出  $E_k^t$ , 再利用  $E_k^t$  对网格的所有顶点进行优化可得  $\varphi(S^0, S^1, t)$ . Chao 等则直接优化非线性能量

$$\mathcal{E}_{\text{spring}}(S^0, S^1, t) = (1-t)\mathcal{E}(S^0, S^t) + t\mathcal{E}(S^t, S^1) \quad (9)$$

其中包含 3 组未知量:  $S^0$  到  $S^t$  的旋矩阵,  $S^t$  到  $S^1$  的旋转矩阵和  $S^t$  的三维坐标. 式(9)是在体网格上计算, 复杂度很高. 因此, Levi 等<sup>[30]</sup>在网格面上考虑, 引入刚性运动的平滑约束

$$\mathcal{E}_{\text{smooth}}(S^0, S^1, t) = \sum_{k=1}^m \sum_{l \in N(E_k)} \|R_k^t - R_l^t R_l^t\|^2 \quad (10)$$

其中,  $R_k^t$  则由  $R_k^0 = R_k^0 (R_l^0)^T$  和  $R_k^1 = R_k^1 (R_l^1)^T$  作球面线性插值得到,  $R_k^t$  其实可以看成是相邻块的运动变换矩阵之间的弱连接映射(weak connection map). 称之为“弱”, 是由于  $R_k^t$  并不是旋转不变量, 即目标网格发生旋转后,  $R_k^t$  也随之改变. 基于 ARAP 的原理, Chen 等<sup>[31]</sup>对插值路径进行了旋转补偿, 并通过 GPU 并行求解关键帧网格每个顶点的路径; Yang 等<sup>[32]</sup>研究了由多个独立网格组成的几何对象之间的插值.

Xu 等<sup>[33]</sup>考虑关键帧微分量之间的变化关系, 给定曲面和定义于其上的函数、梯度和散度等微分量能刻画曲面自身的几何, 因此, 一旦建立关键帧之间运动变换关系, 就能将该关系作用到微分量上, 达到插值微分量的目的, 最终通过微分量重构中间网格. 运动变换关系由关键帧对应三角形之间的变形梯度得到, 与文献[26]类似. 在求解方法上, Xu 等引入域网格的概念, 使得泊松方程的系数矩阵固定不变, 因此, 重构网格顶点的效率非常高; 另一方面, 域网格的选取在理论上对结果构成影响, 但其认为这种影响并不显著. 在适用性方面, 该方法不能处理非流形和退化网格(微分量基于微分流形假设, 退化三角形微分量没有定义). 为了更加逼近真实运动, Sumner 等<sup>[34]</sup>利用关键帧之间的变形梯度构造非线性特征空间来约束帧间的变形过渡. 具体通过对变形梯度进行极分解分别插值旋转矩阵和缩放矩阵实现; 旋转矩阵插值采用矩阵对数插值的方式, 因而能够支持多个旋转矩阵的混合. Gao 等<sup>[35]</sup>借助已有的同类模型数据库引导基于 ARAP 的插值. 由于图形数据库所包含的模型数量一般不会太多(约 200 个), 文中采用上采样技术增加形状空间的数据点, 而且通过聚类尽量使得新增的数据点能够位于若干个线性形状子空间中; 最后, 对每个顶点一邻域(的顶点坐标)

进行 ARAP 插值. 由于该方法充分利用了大量的(接近)真实情况的姿态, 因此其能够处理文献[33]所无法处理的问题, 如避免自相交、使运动符合物理规律等等.

经典的 ARAP 方法刻画的是不同帧对应三角形之间的变换关系, 其最大的缺点在于: 当三角形发生超过  $2\pi$  的旋转, 那么直接插值旋转角将导致不合理的结果. 这是因为旋转矩阵本身无法记录旋转的绕数<sup>[36]</sup>. 为此, 考虑周边三角形的旋转一致性, 而不只是单独考虑每个三角形的情况, 这是一种比较直接的解决办法. Fu<sup>[37]</sup>等利用基于平面多边形的拉普拉斯坐标, 提出一种简单的能保证旋转一致性的算法. 该算法中, 每个多边形顶点对应的局部标架的运动依然采用 ARAP 方式刻画, 但并非同时计算它们的旋转角, 而是以任意选定的某个局部标架的旋转角作为基准向外传播, 使得与相邻标架旋转角的差位于区间  $[-\pi, \pi)$ .

最后, Rossignac 等<sup>[38]</sup>探讨了 ARAP 的平稳性问题, 同时提出插值仿射变换矩阵对数以达到平稳过渡目的之方法, 其中还介绍了计算矩阵对数解析解的方法.

#### 4 连接映射之间的运动变换插值

除了直接表示物体元素在起始和结束之间的运动关系外, 还有一种间接描述物体运动的方式, 即通过物体内部元素的相对位置关系表达其运动过程的变化. 物体自身几何元素的相对空间位置变换称为连接映射(connection map)<sup>[39]</sup>. 采用连接映射的插值技术一般需要经过 2 个基本步骤: 连接映射线性插值和网格重构.

已有的连接映射方法一般都是建立同一帧中相邻局部区域标架之间的表示关系. 假设  $P_i^t$  和  $P_j^t$  是某帧上相邻的 2 个局部区域, 如图 4 所示, 可以是顶点、面、三角形或更大的区域, 提取标架的算子为  $F^t$ , 连接映射为  $T_{ij}^t$ , 则有示意性的形式形式化表示为

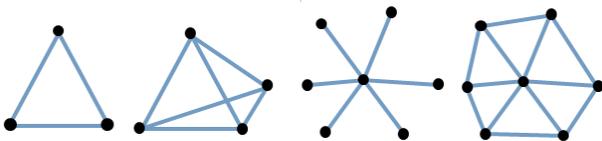


图 4 网格插值的基本图形元素.

$$T_{ij}^t \circ F_i^t(P_i) = F_j^t(P_j), P_j \in N(P_i), t \in [0, 1] \quad (11)$$

一般来说, 可以建立关键帧之间的连接映射的变换关系  $F^t(P_i^t)$ . 与 ARAP 方法类似, 也可以把  $T_{ij}^t$  分解为旋转和缩放 2 个部分, 旋转部分对旋转角线性插值, 缩放部分直接对变换矩阵线性插值, 这样就得到中间帧的连接映射. 由于连接映射之间的旋转角一般都不会太大, 不会导致角度不连续变化的情况.

重构阶段一般分为 2 步: 1) 通过  $T^t$  重构中间形状  $S^t$  的局部标架在世界坐标中的朝向(至少需要一个已知标架); 2) 重构网格顶点坐标.

首先,  $F^t(P_i^t)$  的值通过求解由式(11)所组成的线性方程组得到. 值得注意的是, 式(11)所描述的相对关系并不紧凑, 拥有大量的冗余信息, 而且, 这些信息并非一致, 需要使用最小二乘法来求解; 另一方面, 即使其方程个数多于未知数个数(约 3~6 倍), 但它的系数矩阵并不满秩. 这是因为式(11)建立在  $F^t(P_i^t)$  的像空间中, 缺乏源内嵌空间的信息. 为了解决这个问题, 可以引入锚点(anchor)集合, 这些区域的标架可从给定形状的相应标架线性插值得到. 最后, 通过求解  $F^t(P_i^t)$  的逆函数, (在最小二乘的意义下)重构中间网格的顶点位置. 同样地, 锚点的坐标位置也要预先从给定形状线性插值得到.

基于连接映射的方法, 最重要的步骤是寻找合适的连接映射. 这里“合适”是指连接映射应该具备刚性不变性, 即旋转不变和平移不变, 并且独立于目标网格所嵌入空间的坐标系. 这是判断插值量是否基于连接映射的最重要依据.

Kircher 等<sup>[39]</sup>首次给出基于三角形对之间的连接映射的形式化定义,  $T_{ij}^t$  对应于相邻三角形局部仿射标架之间的变换关系. 其中对变换关系进行极分解, 对旋转矩阵进行球面插值, 对缩放矩阵的缩放因子进行线性插值. 该方法的结果依赖锚点的数量: 在较少锚点的情况下, 会如图 5 所示出现明显的全局错切现象(也可参考文献[22]的视频), 其中上排是仅设定一个锚点(anchor)的情况, 中间帧有明显的收缩和错切; 下排是设定了较大面积的锚点区域(黄色区域). Zhang 等<sup>[40]</sup>的初始化部分采用了基于边的连接映射, 其方法与文献[39]类似, 但  $T_{ij}^t$  建立在附着于边向量的正交局部标架之上, 类似地, 也不可避免地出现错切; 而进入迭代阶段后, 错切得以矫正.

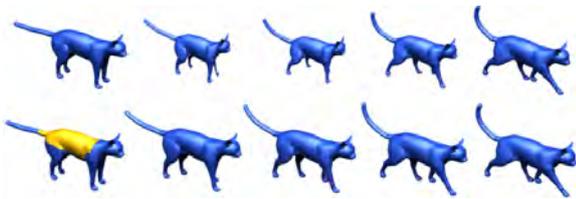


图 5 全局错切现象

基于映射的方法具有解决大变形的能力. 尽管如此, 这类方法一般至少需要求解 2 次大型稀疏矩阵来重构每一帧, 对于生成网格序列而言, 计算量比较大. 因此, 如何提高计算效率成为此类技术的关键; 其次, 此类方法一般无法赋予重构的网格某些特定性质, 需要其他后处理技术支持.

### 5 全局优化

这里的全局优化指的是同时建立所有插入帧能量函数的方法. 本文分基于几何约束和基于物理约束 2 类来加以讨论.

#### 5.1 基于几何约束的方法

比较典型的是一类 As-isometric-as-possible (AIAP)方法, 该方法假设模型遵循等距运动, 即表面上任意 2 点在运动过程中的测地距离保持不变. 从流形的角度来说, 等距运动的曲面  $\varphi(S^0, S^1, t)$  上每点处切向量的长度保持不变, 即最小化能量

$$\int_0^1 \int_{\varphi} \int_{T_p(\varphi)} \langle d\tau(t, p)/dt, \tau(t, p) \rangle^2 d\tau dp dt \quad (12)$$

其中,  $T_p(\varphi(S^0, S^1, t))$  为曲面  $\varphi(S^0, S^1, t)$  在点  $p$  处切平面上的所有切向量的集合,  $\tau(t, p)$  则是该集合上的任何一个切向量. 具体到网格序列模型, Kilian 等<sup>[3]</sup>用网格顶点的边集近似切向量集, 相邻帧的的切向量之差近似切向量的导数, 就得到如

下能量<sup>[3]</sup>

$$\mathcal{E}_{AIAP} = \sum_{t=0}^1 \sum_{p^t \in S^t} \sum_{q^t \in N(p^t)} \langle q^{t+\Delta t} - p^{t+\Delta t} - (q^t - p^t), q^t - p^t \rangle^2 \quad (13)$$

其最小化的解不唯一, 因此 Kilian 等<sup>[3]</sup>另外引入了一个切向量的正则化能量. 此外, 计算能量中的近似向量时采用了向前差分, 为了使得插值结果具有对称性, 应该再加上从最后一帧到第一帧的一个类似能量. 由于是非线性优化, 求解时需要预先(通过线性插值顶点位置来)初始化, 因此无法处理图 6<sup>[40]</sup>所示大变形问题. 最后所有中间帧顶点在一起优化, 使得复杂度大增, 为此 Kilian 等<sup>[3]</sup>还引入了多分辨率表示<sup>[41-43]</sup>, 使实现变得相当复杂.

Li 等<sup>[44]</sup>考虑了平面形状的 AIAP 插值问题, 不过其根据  $\langle d\tau(t, p)/dt, \tau(t, p) \rangle = 0$  把切向量的变化量表示成切向量的旋转形式, 则在离散情况下有

$$q^{t+\Delta t} - p^{t+\Delta t} - (q^t - p^t) = c_{pq} R_{pq} (q^t - p^t) \quad (14)$$

据此得到另一种离散形式

$$\mathcal{E}_{AIAP} = \sum_{t=0}^1 \sum_{p^t \in S^t} \sum_{q^t \in N(p^t)} \left\| q^{t+\Delta t} - p^{t+\Delta t} - (q^t - p^t) - c_{pq} R_{pq} (q^t - p^t) \right\|^2 \quad (15)$$

其中,  $R_{pq}$  是  $\pi/2$  旋转矩阵, 其在二维情形是已知的; 标量  $c_{pq}$  是缩放系数; 所有中间帧的顶点也是未知的. 该方法用块坐标下降(block-coordinate descent)方法优化能量式(15). 为了能处理大变形, 该方法初始化时考虑了邻接三角形旋转角的平滑过渡<sup>[45]</sup>, 但无法将其向三维推广. 为了进一步提高效率, 该方法还引入了贪心方法, 分别用能量式(15)对每条边进行 AIAP 优化. Zhang 等<sup>[40]</sup>将此思想推广至三维情况, 同样需要优化式(15)的非线性

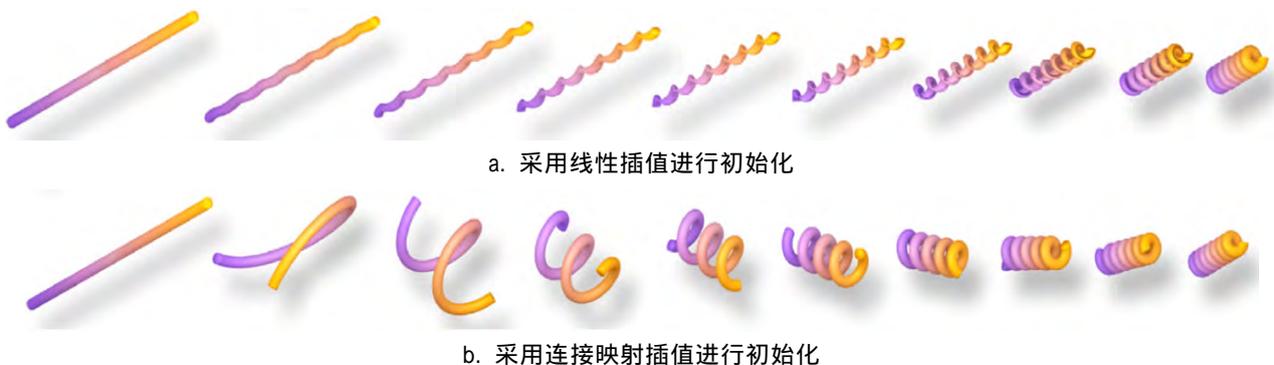


图 6 文献[3]方法生成的结果

问题. 与文献[44]不同, 其初始化方法采用了基于边向量的连接映射技术, 能统一地处理三维和二维情形, 且  $R_{pq}$  不再是单一平面上的  $\pi/2$  旋转矩阵; 在迭代阶段,  $c_{pq}R_{pq}$  采用投影法获得解析解.

Huang 等<sup>[46]</sup>提出了与文献[3]思想相似的全局优化方法, 但使用了 ARAP 能量项取代了后者的 AIAP 能量项. Huang 等<sup>[46]</sup>基于关键帧的配准结果提取刚性块, 认为对于待求序列的任意相邻 2 帧, 其对应刚性块之间满足相同的刚性变换. 由于该建模方法为非线性的, 需要通过线性插值顶点坐标来初始化待求序列, 因此, 也无法处理大变形问题.

Heeren 等<sup>[47]</sup>基于弹性理论, 通过最小化能量损耗求解帧间的变形路径. 变形能由薄膜能(反映拉伸程度, 需要 Lamé 参数)和弯曲能(反映曲率变化程度) 2 部分组成, 分别对应于微分几何的第一基本型和第二基本型. 通过控制 Lamé 参数、2 种能量之间的比例, 能模拟不同材料的变形过渡. 由于在高维形状空间进行全局优化, 该方法采用了多分辨率技术<sup>[48]</sup>提高效率; 另外, 文中采用非线性模型, 初始化使用线性插值顶点坐标的方法. 值得注意的是, 虽然 Heeren 等<sup>[47]</sup>提及该方法的数值解基本满足对称性, 即交换首尾帧后(视觉)效果相同, 但理论上单向的能量建模不能确保对称性.

全局优化算法可以认为是寻找高维黎曼流形上测地线的过程. 高维度往往造成计算量大的问题, 为了提高初始化和迭代的效率, 文献[40, 44]皆使用加速算法. 在初始化阶段, 启发式的边向量、标架传播策略保持旋转一致性, 同时保证优化变量尽可能少; 在迭代阶段, 采用 Cholesky 分解求解线性方程组, 由于方程组的矩阵系数与网格拓扑相关, 与网格几何无关, Cholesky 分解只需执行一次, 便可多次回代重构所有中间帧.

全局优化算法能赋予插值结果具备某些性质, 在文献[3,40,44,47]中, 该性质为拟等距 或满足弹性变形规律. 图 7 所示为对于狮子模型插值序列 MSGI<sup>[16]</sup>与 AIAP<sup>[40]</sup>方法拟等距性质的比较; 横轴表示插值的帧数, 纵轴表示边长相对线性插值的最大误差; 图例百分比表示去掉质量最差的边后保留下来参与统计的边数百分比. 尽管如此, 全局优化的代价除了较高的计算量外, 一般只能处理 2 个关键帧的插值问题. 因为现有的算法利用速度

场作为引导, 令由中间形状构成的高维折线段逼近黎曼流形测地线, 涉及大型非线性优化问题, 而多帧插值必然与多测地线相关, 使得插值问题难以推广至多帧情况.

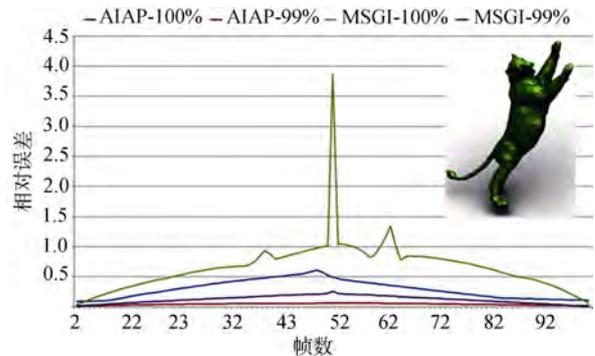


图 7 MSGI 与 AIAP 方法对于狮子模型插值序列拟等距性质的比较

## 5.2 基于物理约束的方法

除了直接考虑欧氏空间的变换关系外, 也有文章从其他空间描述变形, 而且这些空间往往具有一定的物理涵义. Sederberg 等<sup>[49]</sup>最早考虑了基于物理的平面多边形形状插值. 他们认为关键帧之间的运动过程受拉伸(stretching)和弯曲(bending) 2 种力的作用, 为此定义了边的拉伸模型和内角的弯曲模型进行中间帧求解, 但比较复杂. Whitted 等<sup>[50]</sup>将该方法应用于卡通图像插值中的笔画(strokes)形状插值.

考虑到应变场能反映物体的变形情况, Bao 等<sup>[51]</sup>考虑基于薄壳理论的非网格(点云)插值问题, 利用第一基本型和第二基本型导出薄膜张量和弯曲张量, 并对两者进行线性插值; Yan<sup>[52]</sup>等构造中间帧的应变场进行形状插值. 中间帧的应变场可简单地由关键帧的插值得到, 中间帧的顶点坐标则用有限元方法从应变场计算得到. Adams 等<sup>[53]</sup>把 2 个形状的插值看作一个基于物理的变形问题, 其中一个关键帧起到抓手(handle)的作用. 其变形模型包含点的位置约束、刚性约束、体积约束、关键帧速度约束(用户指定)以及加速度平滑约束等, 最后建立关于所有帧的一个全局优化函数.

Huang 等<sup>[54]</sup>考虑给定体网格的参考姿态, 通过模态分析的方法计算振动模式, 并将其转换成“旋转-应变”空间里的基. 于是, 插值量随之转化为此空间的模态坐标. Martin 等<sup>[55]</sup>从连续介质

关键帧对应边的长度一般不等, 而假设网格在变化过程中边长度线性变化, 称这种情况为拟等距.

力学和弹性理论出发,用格林应变张量表征形状(体网格)变化,通过线性插值该张量来引导中间帧的变形;与纯几何方法类似,线性插值后的张量(集)并不能保证对应一个真实存在的形状,因此需要把结果投影至符合物理假设的高维流形上.

一般而言,物理方法,尤其是基于固体力学的方法需要较大的计算量;另一方面,这类方法因为兼顾物体内部和边界,能很好地避免自相交.但求解这些方法时需要进行初始化,其结果是否可用受初始化影响很大.

## 6 其他插值方法

基于变分原理,Turk 等<sup>[56]</sup>提出利用隐式函数升维插值的方法,即对  $N$  维对象边界采用  $N+1$  维对象切面的水平集来表示.其指出,变分方法与有向距离函数(signed distance function)不同,无须知道高维对象完整的边界,仅通过低维(被插值)对象嵌入高维空间后所形成的切面,即能构造隐式函数.与上述升维思想类似,Ohbuchi 等<sup>[57]</sup>使用细分曲面技术,将二维平面多边形插值转化为基于三维曲面等值面抽取,将三维多面体网格插值转化为基于四面体的四维等值面抽取.由于没有使用多分辨率计算框架,该方法容易使得插值结果出现过度平滑的情形,无法较好地保持细节.

针对二维平面多边形,Weber 等<sup>[58]</sup>提出基于复映射的形状表示方式,考虑多边形边缘并于其上定义所谓的角因子.于是,多边形过渡通过不同形状角因子的线性插值获得.

Yang 等<sup>[59]</sup>提出一种基于 T 样条水平集的插值方法,其基本思想是通过构造包裹关键帧的 T 样条控制网格,以及关键帧的符号距离场获得所谓的 T 样条水平集.其插值问题等价于水平集的演

化过程,这种方法特别适用于不同亏格模型,而且使得插值结果  $C^2$  连续.对于具有尖锐特征的模型,该方法需要插入额外的帧来防止水平集走样.

## 7 结 语

本文以插值量为主线,对形状插值的基本概念、相关理论与具体方法进行梳理.同时讨论了形状插值方法的分类方式、插值结果应该满足性质;对具体算法进行分析归类,并评述各自的优缺点;揭示一些不同方法内在的共同优化模型及其形式化描述;同时利用实例对一些重要问题进行解析;特别地,分析了大变形插值方法的共性.一般来说这些方法均满足“源-目标模型完全重构”和“平滑过渡”的性质.表 1 所示为其他插值性质的一览,有些方法因没有明确提到相关性,表中暂空,留待以后完善.

虽然目前已提出很多形状插值方法,但仍然存在一些未解决的问题.此外,随着形状概念的扩展,形状插值方法也会遇到新的挑战.而对所谓自然过渡理解的不同,也会导致新的问题.本文列出一些问题供读者参考:

- 如何对自然过渡进行形式化的刻画?
- 什么是大变形插值?是否有明确的界定方式?
- 拓扑插值:一方面,2 个具有相同拓扑的形状离散成网格形式时,可能会有不同的连接关系,如何建立 2 个连接关系的“自然”过渡,并且其中间帧在形状上仍然能保持同构网格形状插值的效果<sup>[60]</sup>?
- 流形上的形状插值:在二维流形表面如何直接建立 2 个形状之间的插值方法?
- 结构插值:如何在形状插值的过程中保持细节的结构?

表 1 各种代表性插值方法性质一览

文献	插值量	内蕴	对象	对称性	有界变形	支持非流形
[4]	$I$ 与 $F^T F$	Yes	二维平面网格			
[25]	$R$ 的旋转角, $S$	No	二维平面网格			
[10, 49]	边长, 夹角	Yes	二维多边形			
[11]	边长, 二面角	Yes	三维多面体			
[18]	边长, 夹角	Yes	二维多边形			
[19]	边长, 二面角	Yes	三维多面体			
[24]	顶点位置	No	二维多边形			
[16]	边长, 二面角	Yes	三维面网格			
[45]	变形梯度	No	二维网格			

续表

文献	插值量	内蕴	对象	对称性	有界变形	支持非流形
[3]	顶点位置	No	三维面网格			
[9]	边长, 二面角, 体积	Yes	三维面网格			
[21-22]	局部坐标, 投影边夹角	Yes	三维面网格			
[39]	三角形连接映射	Yes	三维面网格			
[44]	边向量旋转矩阵	No	二维网格			
[40]	边向量连接映射	No	三维网格			
[33]	曲面梯度场	No	三维网格			
[38]	变换矩阵对数	No	三角网格			
[13]	边长, 折线段夹角/ 网格局部球面坐标	Yes	四边形网格, 三角网格, 折线段			
[36]	变形梯度	No	二维平面网格			
[26]	变形梯度, 边长, 夹角	Mixed	二维多边形			
[32]	变形梯度	No	二维多网格			
[58]	复数表示的边界描述子	No	二维多边形			
[15]	折线段的边长和夹角	Yes	二维平面图像			
[31]	固定转轴, 插值旋转角	No	三维面网格			
[14]	边长和角度	Yes	二维平面图像			
[34]	变形梯度	No	三维面网格			
[47]	顶点位置	No	三维面网格			
[30]	顶点位置、相邻三角形的相对旋转矩阵	No	三维网格			
[27]	顶点位置、边长、二面角	Mixed	三维面网格			
[54]	变形矩阵经变换后的模态坐标	No	体网格			
[53]	时空位移场	No	非网格			
[55]	格林应变张量	No	体网格			
[17]	边长, 局部标架	Yes	三维面网格			
[20]	微分坐标	No	三维面网格			
[51]	薄膜张量, 弯曲张量	Yes	非网格			
[19]	骨骼边长, 骨骼夹角	Yes	三维多面体			
[36-37]	边长, 旋转角(带一致性检测)	No	二维多边形			
[23]	重心坐标	Yes	二维平面网格			
[46]	$R$ 的旋转角, $S$	No	三维面网格			

## 参考文献(References):

- [1] Aronov B, Seidel R, Souvaine D. On compatible triangulations of simple polygons[J]. Computational Geometry: Theory and Applications, 1993, 3: 27-35
- [2] Chazelle B, Palios L. Triangulating a non-convex polytope[J]. Discrete and Computational Geometry, 1990, 5(2): 505-526
- [3] Kilian M, Mitra N, Pottmann H. Geometric modeling in shape space[J]. ACM Transactions on Graphics, 2007, 26(3): Article No. 64
- [4] Chen R J, Weber O, Keren D, *et al.* Planar shape interpolation with bounded distortion[J]. ACM Transactions on Graphics, 2013, 32(4): Article No. 108
- [5] Zockler M, Stalling D, Hege H-C. Fast and intuitive generation of geometric shape transitions[J]. The Visual Computer, 2000, 16(5): 241-253.
- [6] Alexa M. Recent advances in mesh morphing[J]. Computer Graphics Forum, 2002, 21(2): 173-196
- [7] Lazarus F, Verroust A. Three-dimensional metamorphosis: a survey[J]. The Visual Computer, 1998, 14(8/9): 373-389
- [8] Wolberg G. Recent advances in image morphing[C] //Proceedings of Computer Graphics International Conference. IEEE Computer Society, 1996: 64-71
- [9] Fröhlich S, Botsch M. Example-driven deformations based on discrete shells[J]. Computer Graphics Forum, 2011, 30(8): 2246-2257.
- [10] Sederberg T W, Gao P, Wang G, *et al.* 2D shape blending: an intrinsic solution to the vertex path problem[C] //Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1993: 15-18
- [11] Sun Y M, Wang W, Chin F Y L. Interpolating polyhedral models using intrinsic shape parameters[J]. The Journal of Visualization and Computer Animation, 1997, 8(2): 81-96
- [12] Liu L G, Wang G P, Zhang B, *et al.* Perceptually based approach for planar shape morphing[C] //Proceedings of 12th Pa-

- cific Conference on Computer Graphics and Applications. Los Alamitos: IEEE Computer Society Press, 2004: 111-120
- [13] Liu L G, Wang G J. Three-dimensional shape blending: intrinsic solutions to spatial interpolation problems[J]. *Computer & Graphics*, 1999, 23(4): 535-545
- [14] Johan H, Koiso Y, Nishita T. Morphing using curves and shape interpolation techniques[C] //Proceedings of the 8th Pacific Conference on Computer Graphics and Applications. Los Alamitos: IEEE Computer Society Press, 2000: 348-454
- [15] Meng M, Liu L G. Sketching image morphing using moving least squares[OL]. [2015-07-03]. [http://www.math.zju.edu.cn/liganliu/publications/papers/2007\\_kcjc\\_morphing\\_paper.pdf](http://www.math.zju.edu.cn/liganliu/publications/papers/2007_kcjc_morphing_paper.pdf)
- [16] Winkler T, Drieseberg J, Alexa M, *et al.* Multiscale geometry interpolation[J]. *Computer Graphics Forum*, 2010, 29(2): 309-318.
- [17] Baek S Y, Lim J, Lee K. Isometric shape interpolation[J]. *Computers & Graphics*, 2015, 46: 257-263
- [18] Shapira M, Rappoport A. Shape blending using the star-skeleton representation[J]. *IEEE Computer Graphics Application*, 1995, 15(2): 44-50
- [19] Blanding R, Turkiyyah G, Storti D, *et al.* Skeleton-based three-dimensional geometric morphing[J]. *Computational Geometry: Theory and Applications*, 2000, 15(1-3): 129-148
- [20] Alexa M. Local control for mesh morphing[C] //Proceedings of the International Conference on Shape Modeling and Applications. Los Alamitos: IEEE Computer Society Press, 2001: 209-215
- [21] Lipman Y, Sorkine O, Levin D, *et al.* Linear rotation invariant coordinates for meshes[J]. *ACM Transactions on Graphics*, 2005, 24(3): 479-487
- [22] Baran I, Vlastic D, Grinspun E, *et al.* Semantic deformation transfer[J]. *ACM Transactions on Graphics*, 2009, 28(3): Article No. 36
- [23] Floater M S, Gotsman C. How to morph tilings injectively[J]. *Journal of Computational and Applied Mathematics*, 1999, 101: 117-129.
- [24] Gotsman C, Surazhsky V. Guaranteed intersection-free polygon morphing[J]. *Computers & Graphics*, 2001, 25(1): 67-75
- [25] Alexa M, Cohen-Or D, Levin D. As-rigid-as-possible shape interpolation[C] //Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 2000: 157-164
- [26] Yang W W, Feng J Q. 2D shape morphing via automatic feature matching and hierarchical interpolation[J]. *Computers & Graphics*, 2009, 33(3): 414-423
- [27] Marras S, Cashman T J, Hormann K. Efficient interpolation of articulated shapes using mixed shape spaces[J]. *Computer Graphics Forum*, 2013, 32(8): 258-270
- [28] Chao I, Pinkall U, Sanan P, *et al.* A simple geometric model for elastic deformations[J]. *ACM Transaction on Graphics*, 2010, 29(4): Article No. 38
- [29] Sorkine O, Alexa M. As-rigid-as-possible surface modeling[C] //Proceedings of the 5th Eurographics Symposium on Geometry Processing. Aire-la-Ville: Eurographics Association Press, 2007: 109-116.
- [30] Levi Z, Gotsman C. Smooth rotation enhanced as-rigid-as-possible mesh animation[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2015, 21(2): 264-277
- [31] Chen L, Huang J, Zhang H X, *et al.* GPU-friendly shape interpolation based on trajectory warping[J]. *Computer Animation and Virtual Worlds*, 2011, 22(2-3): 285-294
- [32] Yang W W, Feng J Q, Wang X. Structure preserving manipulation and interpolation for multi-element 2D shapes[J]. *Computer Graphics Forum*, 2012, 31(7): 2249-2258
- [33] Xu D, Zhang H X, Wang Q, *et al.* Poisson shape interpolation[J]. *Graphical Models*, 2006, 68 (3): 268-281
- [34] Sumner R W, Zwicker M, Gotsman C, *et al.* Mesh-based inverse kinematics[J]. *ACM Transaction on Graphics*, 2005, 24(3): 488-495
- [35] Gao L, Lai Y K, Huang Q X, *et al.* A data-driven approach to realistic shape morphing[J]. *Computer Graphics Forum*, 2013, 32(2): 449-457
- [36] Choi J, Szymczak A. On coherent rotation angles for as-rigid-as-possible shape interpolation[C] //Proceedings of 15th Canada Conference on Computational Geometry. Halifax, 2003: 111-114
- [37] Fu H, Tai C L, Au O. Morphing with Laplacian coordinates and spatial-temporal texture[C] //Proceedings of the 13th Pacific Conference on Computer Graphics and Applications. Berlin: Springer, 2005: 100-102
- [38] Rossignac J, Vinacua A. Steady affine motions and morphs[J]. *ACM Transactions on Graphics*, 2011, 30(5): Article No. 116
- [39] Kircher S, Garland M. Free-form motion processing[J]. *ACM Transactions on Graphics*, 2008, 27(2): Article No. 12
- [40] Zhang Z B, Li G Q, Lu H N, *et al.* Fast as-isometric-as-possible shape interpolation[J]. *Computers & Graphics*, 2015, 46: 244-256
- [41] Guskov I, Sweldens W, Schroder P. Multiresolution signal processing for meshes[C] //Proceedings of Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1999: 325-334
- [42] Kobbelt L, Campagna S, Vorsatz J, *et al.* Interactive multi-resolution modeling on arbitrary meshes[C] //Proceedings of the 25th annual conference on Computer graphics and interactive techniques. New York: ACM Press, 1998: 105-114
- [43] Lounsbery M, DeRose T D, Warren J. Multiresolution analysis for surfaces of arbitrary topological type[J]. *ACM Transactions on Graphics*, 1997, 16(1): 34-73
- [44] Li G Q, Yang L, Wu S H, *et al.* Planar shape interpolation using relative velocity fields[J]. *Computers & Graphics*, 2013, 37(5): 364-375
- [45] Baxter W, Barla P, Anjyo K. Rigid shape interpolation using normal equations[C] //Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering. New York: ACM Press, 2008: 59-64
- [46] Huang Q X, Adams B, Wicke M, *et al.* Non-rigid registration under isometric deformations[J]. *Computer Graphics Forum*, 2008, 27(5): 1449-1457
- [47] Heeren B, Rumpf M, Wardetzky M, *et al.* Time-discrete geodesics in the space of shells[J]. *Computer Graphics Forum*, 2012, 31(5): 1755-1764
- [48] Zorin D, Schroder P, Sweldens W. Interactive multiresolution mesh editing[C] //Proceedings of Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Tech-

- niques. New York: ACM Press, 1997: 259-268
- [49] Sederberg T W, Greenwood E. A physical based approach to 2d shape blending[C] // Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1992: 25-34
- [50] Whitted B, Noris G, Simmons M, *et al.* BetweenIT: an interactive tool for tight inbetweening[J]. Computer Graphics Forum, 2010, 29 (2): 605-13
- [51] Bao Y F, Guo X H, Qin H. Physically based morphing of point-sampled surfaces[J]. Computer Animation and Virtual Worlds, 2005, 16(3-4): 509-518
- [52] Yan H B, Hu S M, Martin R R. 3D morphing using strain field interpolation[J]. Journal of Computer Science and Technology, 2007, 22(1): 147-155
- [53] Adams B, Ovsjanikov M, Wand M, *et al.* Meshless modeling of deformable shapes and their motion[C] // Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Aire-la-Ville: Eurographics Association Press, 2008: 77-86
- [54] Huang J, Tong Y Y, Zhou K, *et al.* Interactive shape interpolation through controllable dynamic deformation[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(7): 983-992
- [55] Martin S, Thomaszewski B, Grinspun E, *et al.* Example-based elastic materials[J]. ACM Transactions on Graphics, 2011, 30(4): 72-79
- [56] Turk G, O'Brien J. Shape transformation using variational implicit functions[C] // Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1999: 335-342
- [57] Ohbuchi R, Kokojima Y, Takahashi S. Blending shapes by using subdivision surfaces[J]. Computers & Graphics, 2001, 25(1): 41-58
- [58] Weber O, Gotsman C. Controllable conformal maps for shape deformation and interpolation[J]. ACM Transactions on Graphics, 2010, 29(4): Article No. 78
- [59] Yang H, Jüttler B. 3D shape metamorphosis based on T-spline level sets[J]. The Visual Computer, 2007, 23(12): 1015-1025
- [60] Lin C H, Lee T Y. Metamorphosis of 3D polyhedral models using progressive connectivity transformations[J]. IEEE Transactions on Visualization and Computer Graphics, 2005, 11(1): 2-12