

## 基于局部单射的平面形状插值形变

齐 飞, 陈中贵\*

(厦门大学信息科学与技术学院 厦门 361005)  
(chenzhonggui@xmu.edu.cn)

**摘 要:** 在只给出用简单多边形表示的两输入形状的情况下, 实现一种简单易用、自然高效的形状插值方法. 首先利用基于形状感知的特征匹配算法生成源形状和目标形状之间的匹配; 之后在源形状上构造三角剖分, 并通过求解映射到目标形状上的尽量刚体的局部单射得到同构三角剖分; 最后利用扭曲有界的插值方法得到中间序列. 实验结果表明, 该方法构造的形变结果能较好地体现源形状和目标形状的特征对应信息, 形变过程自然, 扭曲较小.

**关键词:** 特征匹配; 同构剖分; 三角网格; 插值形变  
**中图法分类号:** TP391.41

## Planar Shape Interpolation Based on Local Injective Mapping

Qi Fei and Chen Zhonggui\*

(School of Information Science and Technology, Xiamen University, Xiamen 361005)

**Abstract:** This paper presents an efficient and easy-to-use planar shape interpolation method, given two input shapes represented by simple polygons. We firstly used a perception-based feature matching algorithm to match the feature points in the source shape with the target shape, then built compatible triangulations by constructing a locally injective mapping between the source and target shapes. Finally, an interpolation method with bounded distortion was adopted to get intermediate frames. Experimental results show that the interpolation results by our method can well reflect the feature correspondences between the source and the target shapes, and the resultant deformation is visually pleasing with less distortion.

**Key words:** feature matching; compatible triangulations; triangular mesh; shape interpolation

形状插值在 CAD 中扮演着重要的角色, 如何设计并实现一个高效准确的形状插值序列一直是被广泛研究的热点问题<sup>[1-2]</sup>. 形状插值最简单的应用便是通过输入的 2 个不同形状来生成计算机动画序列的中间帧. 在不同的应用下对形状插值的要求也不同, 所以对形状插值的方法好坏也没有一个统一的标准. 大多数情况下, 能够尽可能地保持图形的基本几何特征是极其重要的, 因为输入的形状各异, 扭曲肯定是无法避免的, 所以只能尽可能地减少扭曲与误差.

通常, 形状插值形变可以分为 2 个子问题: 一是输入形状之间特征值匹配问题, 二是对插值方法的设计和使用. 现有的方法要么着眼于对输入多边形之间的特征匹配或是生成输入多边形之间的同构剖分, 要么是对已知同构剖分的输入进行处理生成形变序列<sup>[3-5]</sup>, 因此这些方法都只完成了平面形状插值序列生成的部分工作, 并且对用户输入有较高的要求. 本文提出一种用户只需要输入 2 个形状, 就能得到从一个形状形变到另外一个形状的中间过程序列的方法. 首先采用基于感知

收稿日期: 2016-05-13; 修回日期: 2016-07-22. 基金项目: 国家自然科学基金(61472332); 中央高校基本科研业务费专项基金(20720140520). 齐 飞(1991—), 男, 硕士研究生, 主要研究方向为计算机图形学; 陈中贵(1982—), 男, 博士, 副教授, CCF 会员, 论文通讯作者, 主要研究方向为计算机图形学、数字几何处理.

的方法对 2 个形状进行特征点匹配, 并对源图形进行采点和构造 Delaunay 三角剖分; 之后利用用于形变的局部单射方法将源图形变形为目标图形, 从而生成同构的网格剖分; 最后采用高效的扭曲有界的插值方法得到插值的中间形状序列。

## 1 相关工作

在研究形变问题中, 首要的问题是在 2 个形状之间建立合适的匹配。在 2 幅图中找到特征点的对应关系是计算机图形学与计算机视觉领域中的一个重要问题。在找到匹配的基础上, 便可以将其用于形状或模型之间的形变。如何识别出图片之间点与点的关系无疑是困难的, 因为同一个形状在不同图片中的形态是可变的, 因此应对不同姿态的同一个形状进行比较; 更复杂的, 还可以去对同一类的不同目标进行处理。在这些情况下, 待匹配特征点的出现与它周围的环境都有着显著差异。近年来出现了很多针对图形中特征点匹配的算法<sup>[3-4]</sup>, 这些方法都是在每张图片中识别出兴趣点, 并且利用它们自身局部的一些形状特征来描述生成之后的候选匹配点。与正确的匹配相比, 这些方法都容易产生一定数量错误的匹配, 如 SIFT 算法<sup>[6]</sup>。还有一些基于优化的自动特征点匹配算法通过物理方法来最小化目标函数去检测匹配<sup>[7]</sup>, 通过点之间最大化相似函数的模糊点匹配, 以及通过找到形状中合理的骨架匹配来建立点之间的匹配<sup>[8]</sup>。此外, 对于弯曲形状的形变, 已有的方法大都是采取在形状上生成均匀分布的点集来表示形状; Sederberg 等<sup>[9]</sup>扩展了在 B 样条曲线间使用基于物理方法寻找匹配对的算法, Cohen 等<sup>[10]</sup>使用在 2 条曲线上的离散点及点上切向量场的动态规划算法求得近似解。最近, Lipman 等<sup>[11]</sup>提出一种限制失真的特征匹配算法。本文方法采用基于感知的形状特征匹配<sup>[12]</sup>, 不需要骨架的建立与手动操作, 通过检测特征点与利用协方差计算出特征点的几何特性来自动完成匹配。

多边形的同构三角剖分在形变中有着重要的应用。2 个多边形同构剖分质量的好坏将直接影响形变最后结果的好坏, 因此, 如何得到高质量的同构三角剖分也是需要研究的问题。Aronov 等<sup>[13]</sup>最先提出同构三角剖分的问题, 并且证明在允许加入内部控制点的条件下, 两多边形是存在同构三角剖分的。Surazhsky 等<sup>[14]</sup>利用重新网格化和网格光顺

的方法, 提出一种高质量同构三角剖分生成算法。由于在形变过程中对同构三角剖分质量要求较高, 因此本文希望使用一种完全避免三角形在同构剖分中有反转情况的方法, 把求解同构剖分问题看作是已知一网格剖分后, 根据另一个多边形形状求内部同构映射的问题, 并采用构造局部单映射方法<sup>[15]</sup>来求得这个同构映射。

对插值形状的生成也有很多方法, 一般形状插值方法可以分为 3 步: 首先使用某种几何特征量去描述输入的源形状, 之后对这些量进行插值, 最后通过这些插值量来重建出需要的中间形状。不同的方法有着不同插值量的计算方法。最直接的方法是用控制点的位置来表示这些几何结构, 这种全局的方法对于输入形状的形变并不是一致的, 并且会因为发生自交而导致结果错误。为了保证中间结果是无交错的, Surazhsky 等<sup>[16-17]</sup>设计了一种特殊的方法, 但是必须要求输入形状的边界多边形是凸的, 这一点在实际场景中是很难得到保证的。Alexa 等<sup>[5]</sup>从源形状到目标形状的映射中, 使用每个几何元素(可以是控制点也可以是面)的雅克比矩阵表示每个目标的几何特性, 通过对雅克比矩阵中旋转与错切的部分进行插值来求得中间序列。但是这些方法在旋转比较剧烈或源图形到目标图形是非对称变换的情况下, 生成是连续序列往往是不理想的<sup>[18-19]</sup>。基于微分坐标表示的插值方法<sup>[20-22]</sup>都有一个缺点: 它们都是基于元素集的, 最后都是去解决一个全局最优的问题。因为这些方法都是通过最小二乘求解的, 所以无法控制生成形状的最大局部扭曲; 同样在 Hu 等<sup>[23]</sup>、Bao 等<sup>[24]</sup>和 Chao 等<sup>[25]</sup>提出的根据连续力学与弹性能量等基于物理角度的插值方法也有类似的缺点。Poranne 等<sup>[26]</sup>提出的基于控制点的双射形变是扭曲可控的, 但是却需要用户手动的操作与控制。本文采用一种扭曲有界插值的方法<sup>[27]</sup>, 在得到源形状的三角剖分后求出目标形状的局部单射映射, 从而避免了三角网格的翻转; 之后利用扭曲有界的插值方法, 生成一个对称的、光滑的形变序列。

## 2 本文方法

### 2.1 概述

本文方法主要包含特征点匹配、构造同构三角剖分和生成中间插值形状三部分, 具体步骤如下:

Step1. 输入 2 个形状  $S$  和  $T$ , 对它们的边缘进行

重新点采样, 用得到的边缘计算出边缘上的候选特征点, 利用基于感知的方法对两个形状的特征点进行特征匹配<sup>[12]</sup>.

Step2. 对其中一个形状  $S$ , 进行 Delaunay 三角剖分, 这里将  $S$  的特征点都设为三角剖分的顶点; 得到三角剖分后, 通过  $S$  与  $T$  中的匹配特征点对来构造局部单射<sup>[15]</sup>, 将  $S$  中的剖分映射到形状  $T$  的轮廓中, 从而得到  $S$  和  $T$  的同构三角剖分.

Step3. 将  $S$  和  $T$  的同构剖分作为输入, 采用扭曲有界的插值方法<sup>[27]</sup>, 得到从  $S$  形变到  $T$  中的任意时刻的中间形状.

本文方法的流程图如图 1 所示.

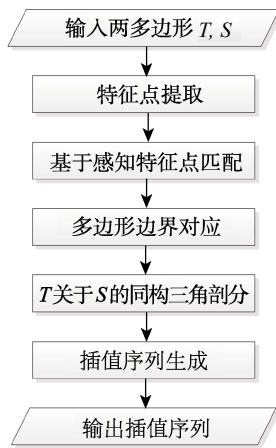


图 1 本文方法流程图

### 2.2 基于感知的形状特征匹配

本文采用基于感知的形状特征匹配方法<sup>[12]</sup>, 建立 2 个输入多边形之间的特征点对应关系. 该方法简述如下:

给定由多边形表示的形状  $P$ , 在多边形的轮廓上重新采样, 并保证多边形的顶点在采样点集中, 将采样点集记为  $\{P_i | i = 0, 1, \dots, n\}$ . 在点集中自动提取出特征点, 特征点一般为曲率极值点、尖点和拐点. 对每个特征点定义一个支撑域作为它的局部邻域, 记为  $R(P_i) = \{P_j | j = i - h, \dots, i, \dots, i + h\}$ , 其中  $h$  是某个整数, 表示邻域的范围. 计算所有特征点在它邻域的协方差矩阵, 并得到这个二阶矩阵的特征向量和特征值. 本文把和  $P_i$  切方向接近的特征向量对应的特征值记为  $\lambda_T$ , 垂直方向对应的特征值记为  $\lambda_N$ . 再把两特征点中的所有采样点定义为一个特征元素, 一个特征点左右两边的特征元素分别记作  $R_L(P_i)$  和  $R_R(P_i)$ .

对于每个特征点, 定义 3 个几何特征量:

$$\sigma(P_i) = \sigma(R(P_i)) = \xi \frac{\lambda_N}{\lambda_N + \lambda_T},$$

$$\tau(P_i) = \frac{\sigma(R_L(P_i)) + \sigma(R_R(P_i))}{2},$$

$$\rho(P_i) = \frac{\rho^L(P_i) + \rho^R(P_i)}{2}.$$

其中, 若  $P_i$  点局部是凸的,  $\xi = 1$ , 反之  $\xi = -1$ ;  $\rho^L(P_i)$  和  $\rho^R(P_i)$  分别表示特征点  $P_i$  两侧的特征元素占整个形状长度的比例. 令点集  $S = \{S_i | i = 0, 1, \dots, m\}$  作为源形状的特征点集,  $T = \{T_i | i = 0, 1, \dots, n\}$  为形变后目标形状的特征点集. 若两特征点相似, 则它们的几何特征量也应该是相近的. 形状  $S$  中的特征点  $S_i$  与形状  $T$  中特征点  $T_i$  的相似性

$$C_{\text{sim}}(S_i, T_j) = \psi(S_i, T_j) \sum_{q=\sigma, \tau, \rho} \omega_q \Delta_q(S_i, T_j).$$

其中,  $\psi$  表示不同几何特征量对特征点相似性的贡献,

$$\Delta_\sigma(S_i, T_j) = |\sigma(S_i) - \sigma(T_j)|,$$

$$\Delta_\tau(S_i, T_j) = \left| \sigma(R_L(S_i)) - \sigma(R_L(T_j)) \right| + \left| \sigma(R_R(S_i)) - \sigma(R_R(T_j)) \right| / 2,$$

$$\Delta_\rho(S_i, T_j) = \left| \rho^L(S_i) - \rho^L(T_j) \right| + \left| \rho^R(S_i) - \rho^R(T_j) \right| / 2.$$

可以把形状  $S$  中特征点到  $T$  中特征点的匹配看作一个映射  $J: \{S_i\} \rightarrow \{T_i\}$ , 这个相似性函数定义为

$$C_{\text{sim}}(S, T, J) = \sum_{i=0}^{m-1} C_{\text{sim}}(S_i, T_{J(i)}).$$

当此函数的值为最小值时, 表示  $(S_i, T_j)$  是一个比较好的匹配. 所以本文需要求解一个优化问题

$$\min_J C_{\text{sim}}(S, T, J).$$

该优化问题可以通过动态规划算法有效求解, 详细求解过程见文献<sup>[12]</sup>. 图 2 所示为本文采用方法对输入的源形状和目标形状的特征匹配结果.

### 2.3 基于局部单射的同构三角剖分生成

第 2.2 节描述的特征匹配方法只是将 2 个输入形状的边界对应起来, 形状内部的区域映射关系的建立对最终的形变效果有重要的影响. 对于输入

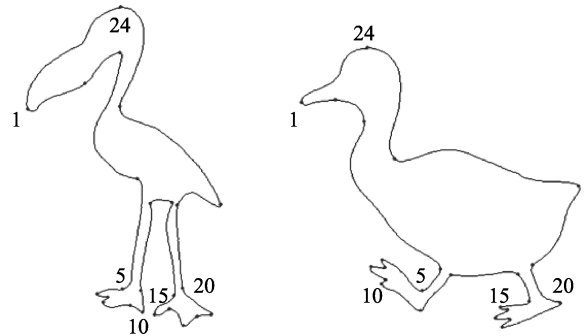


图 2 特征匹配结果

的 2 个任意形状的区域, 通常很难获得解析的光滑映射, 对 2 个区域构造同构三角剖分是建立 2 个区域分片光滑映射的常用方法. 之前的方法构造同构三角剖分时, 通常需要在源形状和目标形状的内部插入 Steiner 点<sup>[13-14]</sup>. 与此不同, 本文采用一种基于构造局部单射的方法<sup>[15]</sup>来生成同构三角剖分, 使得在形变比较剧烈的情况下也能保证三角形不发生翻转.

基于构造局部单射的方法首先对源形状  $S$  进行 Delaunay 三角剖分, 将表示形状  $S$  的多边形的每一条边都设定为三角剖分的约束边, 之后根据这些约束边求得带约束 Delaunay 三角剖分. 当然,  $S$  中的所有特征点都是得到的三角网格结构中的顶点. 在得到  $S$  的三角网格结构之后, 将  $S$  的边界根据之前匹配好的特征点对均匀地映射到形状  $T$  上,  $S$  的两特征点之间的顶点数与映射到  $T$  上的对应特征点之间的顶点数是相同的.

本文采用局部单射映射的方法<sup>[15]</sup>得到  $S$  与  $T$  中的内部同构剖分结构. 将  $S$  到  $T$  的同构映射看作是由  $S$  的剖分变形到形状  $T$ , 而边界点正好就是约束顶点. 因此这个问题就变成了求解网格形变后新的网格内部顶点位置的问题, 可以看作求一个平面映射的过程. 设目标形状  $T$  上的所有网格顶点为  $v$ , 利用网格尽量刚体<sup>[28-29]</sup>的形变能量函数  $E(v)$ , 并引入一个边界点位置约束  $Cv-d$ , 其中,  $C$  为边界点对应的系数矩阵,  $d$  为边界点的约束位置, 则目标函数可以写成

$$\arg \min_v E(v) + \alpha \|Cv - d\|^2,$$

其中  $\alpha > 0$  是指定的约束权参数. 将这个问题看作网格求形变后的位置后, 为了防止在形变中网格中的元素发生翻转, 可以在上述目标函数上引入一个惩罚项. 考虑网格中第  $j$  个三角形的有向面积, 显然它是关于顶点坐标的函数, 记为  $\lambda_j(v)$ . 则将约束函数定义为  $c_j(v) = \lambda_j(v) - \varepsilon$ , 其中  $\varepsilon$  作为计算误差, 把它设为一个很小的常量. 只要让  $c_j(v) > 0$ , 便能保证所有三角形没有翻转. 所以本文定义惩罚函数  $\varphi(x)$ , 让其满足  $\lim_{x \rightarrow 0} \varphi(x) = \infty$ , 当  $x \leq 0$  时  $\varphi(x) = \infty$ ; 当  $x$  大于某个常数  $s_j$  时,  $\varphi(x) = 0$ . 常数  $s_j$  对于每个三角形都不同, 取为第  $j$  个三角形初始面积乘以一定的比例因子. 另外函数  $\varphi(x)$  需要在  $[0, s_j]$  有个光滑的过渡. 考虑所有这些因素, 定义惩罚函数

$$\varphi_j(x) = \begin{cases} \infty, & x \leq 0 \\ \frac{1}{\frac{1}{s_j^3}x^3 - \frac{3}{s_j^2}x^2 + \frac{3}{s_j}x} - 1, & 0 < x < s_j \\ 0, & x \geq s_j \end{cases}.$$

于是, 这个优化问题变成如下形式

$$\arg \min_v E(v) + \alpha \|Cv - d\|^2 + \beta \sum_j \varphi_j(c_j(v)),$$

其中  $\beta > 0$  是惩罚强度的权. 只要最小化这个目标函数便能保证没有一个三角形会发生翻转, 得到与源形状上三角剖分尽量相似的同构三角剖分.

图 3 所示为在形状的轮廓特征匹配基础上, 通过构造优化目标函数得到的同构三角剖分. 可以看出, 2 个输入形状区别较大, 局部三角形发生剧烈的扭曲, 但本文方法不需要在网格上添加额外的顶点, 并保证得到的同构三角剖分没有翻转的三角形.

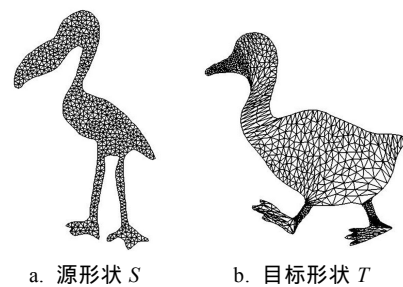


图 3 两形状的同构三角剖分

#### 2.4 扭曲有界的形状插值

本文采用扭曲有界的插值方法<sup>[27]</sup>得到中间插值序列. 该方法假设输入的源形状和目标形状之间的映射关系是已知的, 第 2.3 节的方法求出的同构三角剖分提供了 2 个区域间的映射关系, 为该方法的使用提供了前提条件. 下面简述扭曲有界的插值方法<sup>[27]</sup>.

对于输入的 2 个简单连通域  $S, T \subset \mathbb{R}^2$ , 以及从  $S$  到  $T$  的映射  $\hat{\phi}$  和时间参数  $t \in [0, 1]$ . 此时, 求中间插值序列的过程就变成了求插值映射  $\hat{\phi}: S \rightarrow \mathbb{R}^2$  与  $S$  的对应映射域  $\hat{S}_t = \hat{\phi}_t(S)$  的问题, 其中,  $S = \hat{\phi}_0(S)$ ,  $T = \hat{\phi}_1(S)$ .

引入度量张量  $g = J^T J$  来描述由映射  $\hat{\phi}$  引起的形状扭曲, 其中  $J$  是  $\hat{\phi}$  的雅克比矩阵. 对于中间映射  $\hat{\phi}_t$ , 可以先通过线性插值得到中间度量张量  $g_t = (1-t)I + tg$ , 其中,  $I$  是  $S$  上的欧拉度量张量,  $g$  是映射  $\hat{\phi}$  的度量张量, 再求出中间映射  $\hat{\phi}_t$ .

根据第 2.3 节描述的同构三角剖分, 把问题

离散化到三角网格，并计算源网格中的每个三角面上的度量张量，再通过线性插值得到中间度量张量。根据中间度量张量计算中间三角形每条边的边长，最终可以简化为

$$\|\tilde{e}\| = \sqrt{(1-t)\|e_0\|^2 + t\|e_1\|^2},$$

其中， $e_0$  和  $e_1$  源网格和目标网格上对应边。由此，已知中间插值网格每条边的边长，只要再限定网格的位置和旋转自由度，就可以唯一确定该网格。图 4 所示为得到的插值结果。

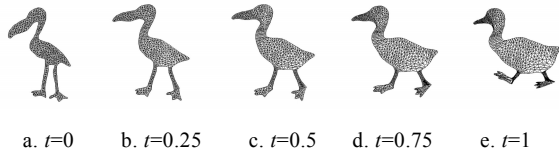


图 4 插值结果

### 3 实验结果及分析

本节展示采用本文方法得到的形变结果。本文中，所有方法都是通过 C++ 与 Matlab 实现的，实验的电脑配置是 Intel 酷睿 i5 处理器，主频 2.8 GHz，内存 14 GB。实验结果表明，本文的算法效率是比较高的，其中特征匹配花费的时间不到 1 s，而同构剖分的生成结果也可以时实得到，中间插值序列

生成一帧的时间如表 1 所示。

表 1 文中例子运行时间比较

输入模型	顶点数	运行时间/ms		
		Laplacian	Dirichlet	本文方法
鸭子	700	23.79	45.75	40.24
“横杠”变到“S”	641	16.25	33.97	42.05
“横杠”变到“U”	1243	16.20	79.43	54.37
青蛙	919	24.43	73.80	40.69
马变到象	1264	29.43	136.78	55.84
马变到狼	736	21.37	41.49	38.38
跳舞小人	1397	36.85	200.05	55.91

本文实验中参数设置如下：在形状匹配时，设置式(1)中  $\omega_q = 1/3$ ，以及一次最多忽略的顶点数为 2；在构造同构剖分时，最大迭代次数为 1000 次，并设置式(2)中  $\beta = 0.01$ ；在生成中间插值形状时，指定生成 100 帧的序列。在生成同构网格的过程中，本文方法只要得到匹配正确的特征点对，便能保证生成的同构剖分网格是局部无自交的；而扭曲有界的方法只能保证生成的中间插值序列网格中的三角面片是无翻转的，对于不同的输入，得到的中间插值网格可能会出现重叠。

本文方法的实验结果与线性插值、Laplacian 能量<sup>[30]</sup>、Dirichlet 能量<sup>[31]</sup>方法对比结果如图 5~6 所示，是将“横杠”形变为“S”和“U”的结果对比。

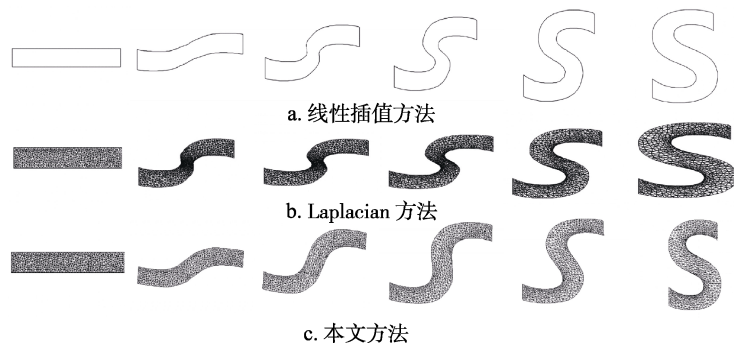


图 5 3 种方法生成的形变序列 1

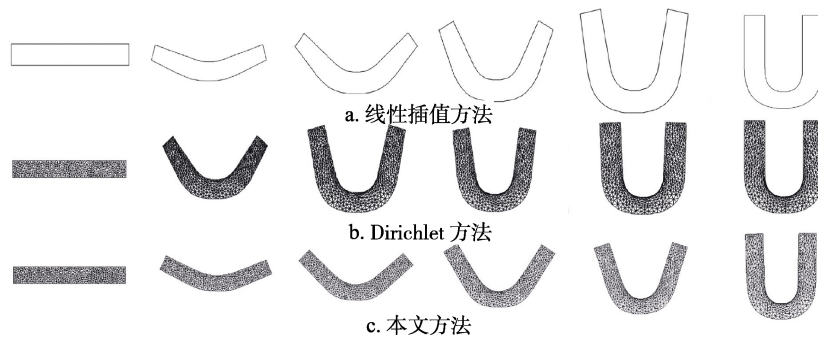


图 6 3 种方法生成的形变序列 2

可以看出, 虽然线性插值方法是光滑的, 但是生成的插值序列中图形的某些部分明显发生了缩小现象, 导致生成的中间序列不自然; 另外 2 种方法的插值过程则不是光滑的, 形变初期模型的变换很剧烈, 到了最后几帧发生的变化又是很小的, 这样的插值

序列也明显不能让人满意; 而本文采用扭曲有界的方法生成每一帧的形变都较为均匀的, 因此形变看起来是一个更加自然的过程. 图 7 所示本文方法的更多实验结果. 表 1 所示为不同插值方法生成一帧所需的计算时间.

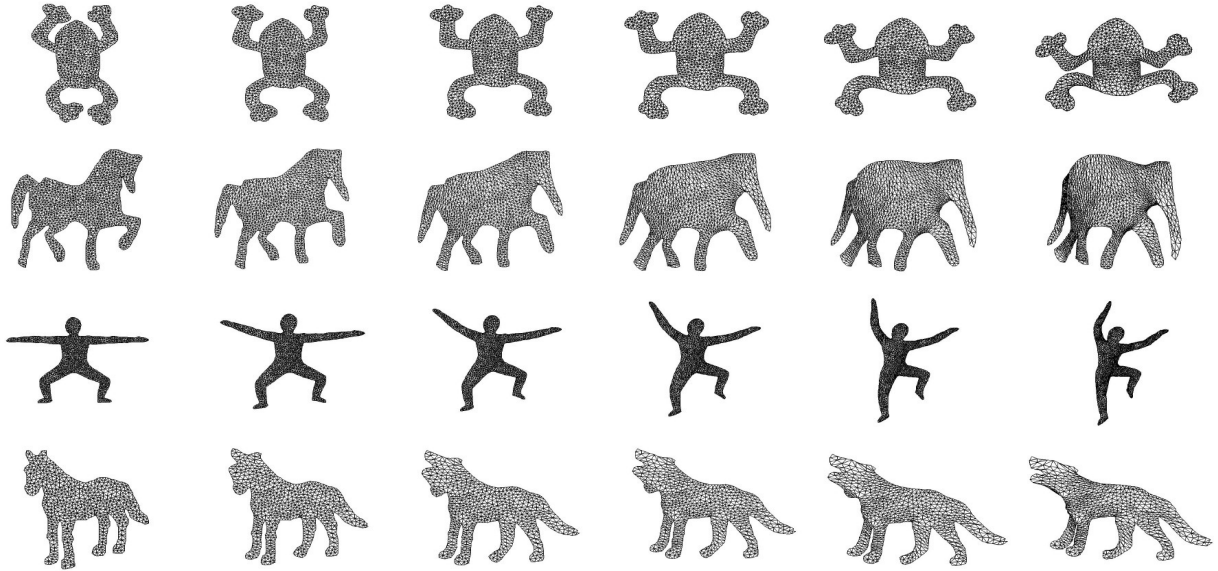


图 7 本文方法更多实验结果展示

#### 4 结 语

本文考虑从输入形状到插值中间序列生成的全过程, 首先通过形状的几何特征量利用基于感知的方法对两输入形状的特征进行匹配; 之后利用局部单射的方法将源形状的三角剖分映射到目标形状多边形上, 得到两输入形状之间的同构剖分; 最后使用扭曲有界的插值方法生成自然的插值形状序列.

但是, 本文方法存在以下一些问题: 在特征匹配上, 准确率和速度都比较理想, 但是结果依赖于两形状特征点的选取. 如果输入两形状特征点数目差异太大, 最后的匹配结果往往太不理想, 从而影响同构剖分的生成. 而在生成同构剖分时, 由于严格限制三角形翻转, 在生成同构剖分过程中, 需要对形变路径进行启发式指导; 否则在很多极端情况下, 无法得到正常的形变后的同构剖分.

今后, 我们希望在特征点的检测和匹配上采用一种鲁棒性更高的方法, 并且在同构剖分的问题上, 加入合理启发路径, 从而优化剖分质量. 将本文方法拓展到曲面或三维体上, 也是我们未来研究工作的重点.

#### 参考文献(References):

- [1] Lazarus F, Verroust A. Three-dimensional metamorphosis: a survey[J]. *The Visual Computer*, 1998, 14(8/9): 373-389
- [2] Mokhtarian F, Mackworth A K. A theory of multiscale, curvature-based shape representation for planar curves[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, 14(8): 789-805
- [3] Heider P, Pierre A, Li R S, *et al*. Local shape descriptors, a survey and evaluation[C] // *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*. Aire-la-Ville: Eurographics Association Press, 2011: 49-56
- [4] Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey[J]. *Foundations and Trends® in Computer Graphics and Vision*, 2008, 3(3): 177-280
- [5] 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
- [6] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. *International Journal of Computer Vision*, 2004, 60(2): 31-110
- [7] Sederberg T W, Greenwood E. A physically based approach to 2-D shape blending[J]. *ACM SIGGRAPH Computer Graphics*, 1992, 26(2): 25-34
- [8] Mortara M, Spagnuolo M. Similarity measures for blending polygonal shapes[J]. *Computers & Graphics*, 2001, 25(1): 13-27
- [9] Sederberg T W, Greenwood E. Shape blending of 2-D piece-

- wise curves[J]. *Mathematical Methods in CAGD*, 1995, 3(1): Article No.3
- [10] Cohen S, Elber G, Bar-Yehuda R. Matching of freeform curves [J]. *Computer-Aided Design*, 1997, 29(5): 369-378
- [11] Lipman Y, Yagev S, Poranne R, *et al.* Feature matching with bounded distortion[J]. *ACM Transactions on Graphics*, 2014, 33(3): Article No.26
- [12] Liu L G, Wang G, Zhang B, *et al.* Perceptually based Approach for planar shape morphing[C] //Proceedings of the 12th Pacific Conference on Computer Graphics and Applications. Los Alamitos: IEEE Computer Society Press, 2004: 111-120
- [13] Aronov B, Seidel R, Souvaine D. On compatible triangulations of simple polygons[J]. *Computational Geometry*, 1993, 3(1): 27-35
- [14] Surazhsky V, Gotsman C. Controllable morphing of compatible planar triangulations[J]. *ACM Transactions on Graphics*, 2001, 20(4): 203-231
- [15] Schüller C, Kavan L, Panozzo D, *et al.* Locally injective mappings[C] //Proceedings of the 11th Eurographics ACM SIGGRAPH Symposium on Geometry Processing. Aire-la-Ville: Eurographics Association Press, 2013: 125-135
- [16] Surazhsky V, Gotsman C. Intrinsic morphing of compatible triangulations[J]. *International Journal of Shape Modeling*, 2011(2): 191-201
- [17] Surazhsky V, Gotsman C. High quality compatible triangulations[J]. *Engineering with Computers*, 2004, 20(2): 147-156
- [18] Fu H, Tai C L, Au K C. Morphing with Laplacian coordinates and spatial-temporal texture[OL]. [2016-05-13]. [http://sweb.cityu.edu.hk/hongbofu/morphing\\_pg05.pdf](http://sweb.cityu.edu.hk/hongbofu/morphing_pg05.pdf)
- [19] 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
- [20] Lipman Y, Sorkine O, Levin D, *et al.* Linear rotation-invariant coordinates for meshes[J]. *ACM Transactions on Graphics*, 2005, 24(3): 479-487
- [21] Sheffer A, Kraevoy V. Pyramid coordinates for morphing and deformation[C] //Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission. IEEE Computer Society Press, 2004: 68-75
- [22] Xu D, Zhang H, Wang Q, *et al.* Poisson shape interpolation[J]. *Graphical Models*, 2006, 68(3): 268-281
- [23] Hu S M, Li C F, Zhang H. Actual morphing: a physics-based approach to blending[C] //Proceedings of the 9th ACM Symposium on Solid Modeling and Applications. Aire-la-Ville: Eurographics Association Press, 2004: 309-314
- [24] Bao Y, Guo X, Qin H. Physically based morphing of point-sampled surfaces[J]. *Computer Animation & Virtual Worlds*, 2005, 16(3/4): 509-518
- [25] Chao I, Pinkall V, Sanan P, *et al.* A simple geometric model for elastic deformations[J]. *ACM Transactions on Graphics*, 2010, 29(4): Article No.38
- [26] Poranne R, Lipman Y. Provably good planar mappings[J]. *ACM Transactions on Graphics*, 2014, 33(4): Article No.76
- [27] Chen R, Weber O, Keren D, *et al.* Planar shape interpolation with bounded distortion[J]. *ACM Transactions on Graphics*, 2013, 32(4): Article No.108
- [28] Liu L G, Zhang L, Xu Y, *et al.* A local/global approach to mesh parameterization[J]. *Computer Graphics Forum*, 2008, 27(5): 1495-1504
- [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] Sorkine O, Cohen-Or D, Lipman Y, *et al.* Laplacian surface editing[C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 2004: 175-184
- [31] Botsch M, Sorkine O. On linear variational surface deformation methods[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14(1): 213-230