

# 基于特征分解的 2-D 多边形渐变\*

杨文武<sup>+</sup>, 冯结青, 金小刚, 彭群生

(浙江大学 CAD&CG 国家重点实验室, 浙江 杭州 310027)

## 2-D Polygon Blending Based on Feature Decomposition

YANG Wen-Wu<sup>+</sup>, FENG Jie-Qing, JIN Xiao-Gang, PENG Qun-Sheng

(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: Phn: +86-571-87951045, E-mail: {wwyang,jqfeng,jin,peng}@cad.zju.edu.cn, <http://www.zju.edu.cn>

Received 2003-12-24; Accepted 2004-06-10

Yang WW, Feng JQ, Jin XG, Peng QS. 2-D polygon blending based on feature decomposition. *Journal of Software*, 2005,16(2):309–315. <http://www.jos.org.cn/1000-9825/16/309.htm>

**Abstract:** 2-D polygon blending has been widely used in 2-D character animation, pattern matching, and geometric modeling. Previous algorithms tend to use the polygon's geometric elements such as edge length, angle, area, skeleton to associate the regions on the two shapes that look alike. However they ignore the visual features of two polygonal shapes. This paper presents a new 2-D polygon blending method based on their visual features correspondence. By decomposing the two polygons into sub-polygon pairs with similar visual features compatibly, each sub-polygon in the source polygon can be smoothly transformed into the corresponding sub-polygon in the destination polygon. Since the feature-decomposition vertices are introduced, user can control the visual features correspondence flexibly and intuitively. Experimental results show that the two polygons can be blended as user prescribed whilst keeping the feature correspondence and feature preservation.

**Key words:** 2-D polygon blending; visual feature decomposition; local coordinate system; vertex correspondence; vertex interpolation path

**摘要:** 二维多边形渐变在二维角色动画、模式匹配、几何造型等领域有着重要的应用。已有的方法大多根据多边形的边长、角度、面积、骨架等几何属性来完成多边形之间的最佳对应和渐变,而忽略了多边形的内在视觉特征。提出了一种基于视觉特征对应的 2-D 多边形渐变方法,该方法把源多边形和目标多边形按照视觉特征进行同构特征分解,得到若干对对应的特征子多边形。在渐变过程中,每个源特征子多边形平滑地过渡到目标特征子多边形。通过引入特征分解点,用户可以灵活和直观地改变特征子多边形,按照预想的效果控制多边形的渐变。实验效果表明,基于特征分解的顶点对应和插值算法不仅可以得到光滑的多边形渐变序列,而且可以实现多

\* Supported by the National Natural Science Foundation of China under Grant Nos.60373036, 60333010 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312101 (国家重点基础研究发展规划(973))

**作者简介:** 杨文武(1981—),男,江西高安人,硕士生,主要研究领域为计算机图形学,计算机动画,计算机辅助几何设计;冯结青(1970—),男,博士,研究员,主要研究领域为几何造型,计算机动画,科学计算可视化;金小刚(1969—),男,博士,研究员,博士生导师,主要研究领域为计算机图形学,计算机动画,隐式曲面造型;彭群生(1947—),男,博士,教授,博士生导师,主要研究领域为真实感图形,虚拟现实,红外成像仿真,基于点的造型与绘制,科学计算可视化。

边形的特征对应和特征保留.

关键词: 2-D 多边形渐变;视觉特征分解;局部坐标系;顶点对应;顶点路径

中图法分类号: TP391 文献标识码: A

在二维空间中,一个源多边形平滑地过渡到一个目标多边形,称为 2-D 多边形渐变.它在二维角色动画、模式匹配以及几何造型中有着广泛应用.为了实现 2-D 多边形渐变,需要解决两个问题:顶点对应和顶点插值路径.顶点对应建立起两个多边形之间的对应关系;顶点插值路径确定渐变过程中产生的中间多边形.一个好的 2-D 多边形渐变算法应当满足:(1) 源多边形平滑地过渡到目标多边形;(2) 中间过渡多边形不发生自交、收缩、内部区域扭曲等不自然现象;(3) 中间多边形应当保留源和目标多边形上的视觉特征.

在顶点对应方面,Sederberg 等人引入了基于物理模型的顶点对应方法<sup>[1]</sup>,其核心思想是将两个多边形看作线框,当把源多边形变形到目标多边形时做功最小;Zhang 提出了基于最相似三角片的方法建立顶点对应关系<sup>[2]</sup>;Carmel 等人根据多边形上的特征点<sup>[3]</sup>,通过变形建立源多边形和目标多边形的顶点对应.前两种方法忽视了多边形的视觉特征,仅根据多边形的几何属性,如边长、角度、三角形面积等自动寻找源、目标多边形的最佳顶点对应,难以得到多边形视觉特征的对应.Carmel 的方法中存在一个多边形变形过程,计算量较大且变形过程难以控制.

相对于顶点对应问题,顶点插值路径问题的相关研究则较多.线性插值是最简单的顶点插值方法,但用此方法得到的中间多边形可能会发生自相交或收缩,效果不自然.当多边形进行刚体运动时,此问题尤为明显.Sederberg 等人提出了一种基于多边形内在几何属性(边长和角度)的插值方法<sup>[4]</sup>,该方法可以得到较好的视觉效果.但中间多边形容易产生内部区域扭曲.在 Carmel 等人和 Zhang 提出的方法中<sup>[2,3]</sup>,将插值路径分成两部分:刚性部分和弹性部分.刚性部分通过全局的旋转和平移实现.对于弹性部分,Carmel 等人通过顶点间的变形过程实现.而在 Zhang 方法中,先找到满足目标函数最佳的一对三角片,并基于此建立对应仿射标架,然后将源和目标多边形变换到两个仿射标架中.通过在仿射标架之间进行刚体变换,并对对应顶点的局部坐标进行线性插值,得到中间多边形.他们方法的不足之处在于,单个的旋转和平移通常并不能反映出两个多边形的朝向及位置变化趋势,不一定能够得到平滑的渐变效果.

上述顶点插值方法仅考虑了多边形的边界,而没有考虑多边形的内部区域.Shapria 和 Rapport 提出了一种基于多边形内部区域的插值方法<sup>[5]</sup>,先将源和目标多边形进行同构星形骨架分解,然后再把源和目标多边形以极坐标形式表示到相应骨架中,通过插值对应的骨架和极坐标得到顶点位置.该方法能够很好地避免中间多边形的自交.但当两个多边形较复杂和奇异时,星形骨架的同构分解是不确定的.Alexa 等人把源和目标多边形进行同构三角化<sup>[6]</sup>,他们定义了一对三角片之间的最佳变换,通过逼近每对三角片之间的局部最佳变换,得到所有对应三角片之间的全局最佳变换.该方法能得到较好的插值效果,并能实现多边形内部纹理的平滑过渡.由于在计算每一对三角片的最佳变换时需要进行 QS 分解,考虑到分解带来的误差,若同构三角化中存在不规则三角片,则会产生不光滑的渐变效果.Gotsman 和 Surazhsky 也通过插值同构三角化来完成顶点插值路径<sup>[7]</sup>,他们的方法可以保证中间多边形不发生自交.但在插值之前,源和目标多边形需要被两凸多边形封闭起来,当源和目标多边形差别较大时,中间多边形不可避免地会发生扭曲.

本文提出了一种基于特征分解的 2-D 多边形渐变方法.在新方法中,首先基于视觉特征交互地指定特征分解点,将两个多边形同构分解为多个特征子多边形,每一个特征子多边形具有各自的局部坐标系.通过插值对应的局部坐标系,以及特征子多边形顶点在局部坐标系中的局部坐标,从而实现两多边形对应特征的逐渐过渡.本文的第 1 节解释多边形的特征分解以及同构特征分解的概念.第 2 节给出顶点对应以及插值路径算法.第 3 节给出实验结果和讨论.第 4 节总结全文并指出将来工作.

## 1 多边形的特征分解及同构特征分解

特征分解就是把多边形分解成若干个特征子多边形,每个特征子多边形对应于原多边形的一个视觉特征.

如图 1 所示.

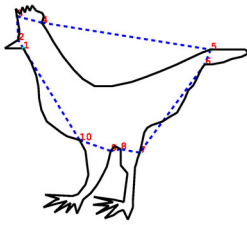


Fig.1 The dashed line polygon is called the frame polygon

图 1 虚线多边形为对应的轮廓多边形

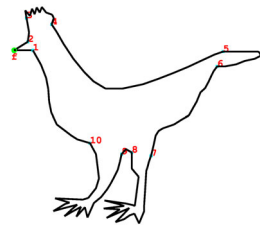


Fig.2 The vertex  $f$  is a feature vertex, but not a feature decomposition vertex.

The vertex 1 and 2 are feature decomposition vertices  
图 2 顶点  $f$  为一个特征点,但不是特征分解点,顶点 1 和 2 为特征分解点

图 1 中的公鸡多边形被分解成了 10 个特征子多边形,分别代表了公鸡的 10 个特征.依次为:嘴,头部,鸡冠,背部,尾巴,躯干,后脚,腹部,前脚和胸部.由于视觉特征具有复杂性,所以难以自动生成.为了能够给用户提供的控制手段,本文通过交互方式实现多边形的特征分解,即在多边形上指定一系列特征分解点.图 1 中顶点 1,2,...,10 即为多边形的特征分解点.这里,相邻特征分解点之间的多边形顶点组成特征子多边形.1 和 2 以及它们之间的多边形顶点构成了“鸡嘴”特征子多边形.值得提出的是,特征分解点不完全等同于特征点:特征点对应于多边形上明显的几何特征,如尖点、拐点等;而特征分解点则是多边形视觉特征区域的边界点.如图 2 所示,圆形顶点  $f$  为特征点,对应着鸡嘴处的一个突出几何特征,而方形顶点 1 和 2 为特征分解点,为鸡嘴特征段的边缘顶点.

顺序连接多边形上的特征分解点,可以得到一个封闭多边形,称为轮廓多边形.轮廓多边形代表着多边形视觉特征区域的大致轮廓,它的每一条边对应着一个特征子多边形.图 1 中的虚线多边形即为轮廓多边形.

为了实现源和目标多边形视觉特征区域对应,需要对两多边形进行“同构特征分解”,即使得源和目标多边形具有相同个数的特征子多边形,如图 3 所示,边界点为相同标号特征分解点的特征子多边形为一对对应的特征子多边形,对应着一对相似视觉特征.需要指出的是,两个多边形上特征分解点的排列顺序可以相同,也可以不同,这也是本文方法灵活性的一个体现.同构分解体现的是源和目标多边形上的视觉特征的对应.通过移动、修改特征分解点,用户可以方便、直观地控制特征子多边形的大小、顺序、个数,从而直观地编辑和控制渐变效果.

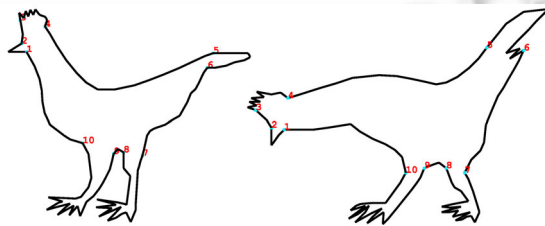


Fig.3 The source and destination polygons are decomposed into ten pairs of corresponding sub-polygons by choosing ten corresponding feature-decomposition vertices interactively

图 3 分别在两个多边形上选择了 10 个特征分解点,按序号一一对应,源和目标多边形被同构分解成 10 个一一对应的特征子多边形

由于视觉特征不同于简单的几何特征,难以完全用几何度量描述,所以难以自动地进行多边形的同构特征分解.另一方面,当两个多边形代表不同类型的物体,它们的视觉特征可能并不一致,即两个多边形上的视觉特征个数或类型不同(如图 6 所示,圆是各项同性的,不具备任何视觉特征),此时,同构特征分解就完全取决于用户的交互,即用户指定源多边形上的区域与目标多边形区域之间的对应关系.

我们曾尝试首先根据多边形的几何特征,由系统自动地生成一些特征点,如尖点、拐点等,然后用户从这些特征点中选择、编辑特征分解点.但通过实验发现,在大多数情况下,由于自动生成的特征分解点较多,而且不一

定能够覆盖特征分解点,反而需要更多的交互工作以实现同构特征分解.因此,我们采用交互的方式实现同构特征分解.由实验结果可知,即使对两个比较复杂的多边形物体,也可以较快地交互完成同构特征分解.

## 2 顶点对应和插值路径

通过源和目标多边形之间的同构特征分解,实现了两个多边形之间的视觉特征对应.下面,通过弧长参数化与顶点合并,建立特征子多边形顶点之间的对应;然后,将每一个特征子多边形进行局部坐标变换;最后,通过插值轮廓多边形和特征子多边形顶点的局部坐标,实现基于特征的多边形过渡.

### 2.1 建立特征子多边形之间的顶点对应

首先对特征子多边形之间的顶点进行弧长参数化,然后线性映射到单位线段上,并在单位线段上合并两个特征子多边形的顶点,再把合并后的顶点线性映射回两个特征子多边形.这样,就实现了特征子多边形之间的顶点对应.具体算法如下.

记对应的两特征子多边形  $P_f^s$  和  $P_f^d$  如下:

$$\begin{cases} P_f^s = \{p_{f1}^s, p_{f2}^s, \dots, p_{fm}^s\} \\ P_f^d = \{p_{f1}^d, p_{f2}^d, \dots, p_{fm}^d\} \end{cases} \quad (1)$$

其中,  $p_{f1}^s, p_{fm}^s$  和  $p_{f1}^d, p_{fm}^d$  分别为源和目标多边形上相邻的 4 个特征分解点.对于  $P_f^s$  中的每个顶点进行弧长参数化,然后将弧长参数映射到  $[0, 1]$  区间,则每个顶点对应参数  $t_i^s (i=1, 2, \dots, m)$  为

$$t_1^s = 0; t_i^s = \frac{\sum_{j=2}^i l^j}{\sum_{j=2}^m l^j} (i=2, 3, \dots, m-1); t_m^s = 1 \quad (2)$$

公式中  $l^j$  为  $p_{f,j-1}^s, p_{f,j}^s$  的边长.上述公式按照各个顶点对应的弧长参数,把源特征子多边形  $P_f^s$  的顶点映射到单位区间  $[0, 1]$  上.按相同的方法,把目标特征子多边形  $P_f^d$  的顶点也映射到单位区间  $[0, 1]$  上.把单位区间  $[0, 1]$  上的所有映射点合并在一起,再分别映射回两个特征子多边形,从而实现特征子多边形之间的顶点对应.记建立对应的特征子多边形  $P_f^s$  和  $P_f^d$  为

$$\begin{cases} P_f^s = \{p_{f1}^s, p_{f2}^s, \dots, p_{fm}^s\} \\ P_f^d = \{p_{f1}^d, p_{f2}^d, \dots, p_{fm}^d\} \end{cases} \quad (3)$$

### 2.2 顶点插值路径

经过同构特征分解,可以得到源和目标轮廓多边形.在两个轮廓多边形中,分别取对应的顶点和边,为每个特征子多边形建立局部坐标系;然后将特征子多边形顶点变换到相应的局部坐标系,得到其顶点的局部坐标.对于两个轮廓多边形,采用 Sederberg 等人提出的边、角插值方法<sup>[4]</sup>,得到中间轮廓多边形;再对特征子多边形顶点的局部坐标进行线性插值,就可以得到中间多边形.

#### 2.2.1 构造局部标架

对于每个特征子多边形,以对应轮廓多边形中的边为参考,建立局部坐标系.下面以源特征子多边形  $P_f^s$  为例说明局部标架的构造.记  $P_f^s = \{p_{f1}^s, p_{f2}^s, \dots, p_{fm}^s\}$ , 取其起始顶点  $p_{f1}^s$  (即一个特征分解点)为原点,相邻的两个特征分解点  $p_{f1}^s, p_{f2}^s$  为方向的单位矢量为  $u^s$  轴,将  $u^s$  轴逆时针旋转  $90^\circ$  所得的矢量作为  $v^s$  轴,建立一个局部坐标系  $(p_{f1}^s, u^s, v^s)$ .对于特征子多边形内的每一顶点  $p_{fi}^s$ ,可以在局部坐标系  $(p_{f1}^s, u^s, v^s)$  内表示为  $(u_i^s, v_i^s)$ , 计算公式如下:

$$p_{fi}^s = p_{f1}^s + u_i^s u^s + v_i^s v^s \quad (i=2, 3, \dots, M-1) \quad (4)$$

对于目标特征子多边形  $P_f^d = \{p_{f1}^d, p_{f2}^d, \dots, p_{fm}^d\}$ , 类似地进行上述局部坐标变换.

#### 2.2.2 插值轮廓多边形——变换局部标架

根据第 2.2.1 节的叙述可知,轮廓多边形实际上决定了特征子多边形的局部坐标系.因此,对轮廓多边形的

插值也就等价于对特征子多边形的局部坐标系的插值.由于轮廓多边形反映了物体的大致形状,并具有简单、顶点少的特点.所以,当采用 Sederberg 等人的边角插值方法对其进行插值时<sup>[4]</sup>,可以取得较好的效果.图 4 所示为图 3 所确定的轮廓多边形的插值结果.根据中间轮廓多边形的顶点和边,就可以建立特征子多边形的中间局部坐标系.

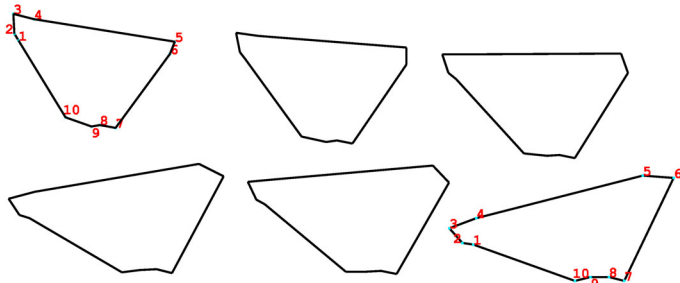


Fig.4 Interpolating results of the frame polygons of the two polygons in Fig.3

图 4 由图 3 确定的轮廓多边形的插值结果

### 2.2.3 插值特征子多边形

对于对应的两特征子多边形  $P_f^s$  和  $P_f^d$ , 它们的局部坐标系分别为  $(p_{f1}^s, u^s, v^s)$  和  $(p_{f1}^d, u^d, v^d)$ , 而对应顶点  $p_{fi}^s$  和  $p_{fi}^d$  在初始局部坐标系中的局部坐标分别为  $(u_i^s, v_i^s)$  和  $(u_i^d, v_i^d)$ . 在进行多边形过渡的某一时刻  $t$ , 通过插值轮廓多边形可以得到中间局部坐标系  $(p_{f1}^t, u^t, v^t)$ , 则在时刻  $t$  处, 对应顶点  $p_{fi}^s$  和  $p_{fi}^d$  的插值结果为  $p_{fi}^t$ .

$$p_{fi}^t = p_{f1}^t + u_i^t u^t + v_i^t v^t \tag{5}$$

其中  $p_{f1}^t, u^t, v^t$  通过插值轮廓多边形得到, 而  $u_i^t$  和  $v_i^t$  通过下面的公式计算得到:

$$\begin{cases} u_i^t = (1-t)u_i^s + tu_i^d \\ v_i^t = (1-t)v_i^s + tv_i^d \end{cases} \tag{6}$$

## 3 实验结果和讨论

下面给出了多边形进行渐变的一些例子, 并将新方法所得结果与 Sederberg 等人的方法<sup>[4]</sup>以及 Zhang 的方法<sup>[2]</sup>所得结果进行了比较.

图 5 给出了使用本文方法所得的公鸡渐变效果图. 两多边形中特征分解点交互给出. 图 6(a)是由 Sederberg 等人的边角插值方法得到的结果, 图 6(b)和图 6(c)是由基于特征分解的顶点插值方法得到的结果. 图 6(a)和图 6(b)使用相同的同构特征分解建立顶点对应关系, 见图 6(b)中的源和目标多边形的特征分解点. 可以看出, 边角插值路径得到的中间多边形产生了内部区域扭曲, 并且中间多边形丢失了源与目标原始多边形的特征对应信息, 而基于特征分解的插值方法产生了光滑的、特征保留的中间多边形序列. 图 6(c)中的特征分解点通过简单地编辑图 6(b)中的特征分解点得到, 显然, 图 6(b)和图 6(c)产生了不同的中间多边形光滑渐变序列. 这说明, 用户可以通过交互地指定特征分解点实现多边形渐变过程的控制.

图 7 比较了 Sederberg 等人的方法、Zhang 的方法和基于特征分解的插值方法所产生的中间多边形序列. 图 7(a)~图 7(c)均使用本文所述基于特征分解点的对应方式建立顶点对应关系. 图 7(a)为边角插值方法得到的渐变结果<sup>[4]</sup>, 图 7(b)和图 7(c)则分别使用 Zhang 的插值方法<sup>[2]</sup>和基于特征分解的插值方法. 从效果来看, 图 7(a)中间多边形在树的顶部以及枝干底部发生了较大扭曲, 丢失了源和目标多边形的特征信息, 图 7(b)中间多边形在树的顶部以及树干的右边树叶处发生了自交, 并且枝干上部扭曲严重, 产生了不自然的多边形过渡序列. 而在图 7(c)中, 使用基于特征分解的插值方法得的中间多边形不仅避免了自交、扭曲情况, 并且较好地保留了源和目标多边形的特征信息, 保证了光滑的过渡效果. 图 8~图 10 是其他基于特征分解的多边形渐变结果.

由实验结果可以看出, 本文提出的基于特征分解的插值方法能够很好地避免多边形内部扭曲, 产生光滑、特征保留的渐变效果. 此外, 新方法为每个特征子多边形构造了局部坐标系, 由多对局部坐标系的旋转和位置变

换来描述整体多边形的旋转和位置变换,相对于 Zhang 提出的由一个旋转和平移来描述整体多边形的变换,渐变效果更光滑、自然.

此外,在建立特征子多边形顶点之间的对应时,我们采用的是基于弧长参数的顶点合并方法.由于特征子多边形具有某种视觉上的相似性,所以采用弧长对应是合理的.此外,这种对应方法还具有简单的优点.对于大多数特征子多边形,该方法可以大致实现特征子多边形之间的细节对应.但对两个相差较大的特征子多边形来说,则还要考虑更多的几何属性(边长,角度,面积,骨架等)来实现顶点对应,如采用 Sederberg 的基于物理的顶点对应方法<sup>[1]</sup>;或者通过在特征子多边形内进一步细分特征来实现细节对应.

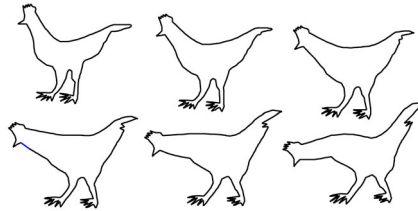


Fig.5 Blending of two cock-shape polygons, the compatible feature decomposition is shown in the Fig.3

图 5 鸡形状的渐变,使用图 3 所示的同构特征分解

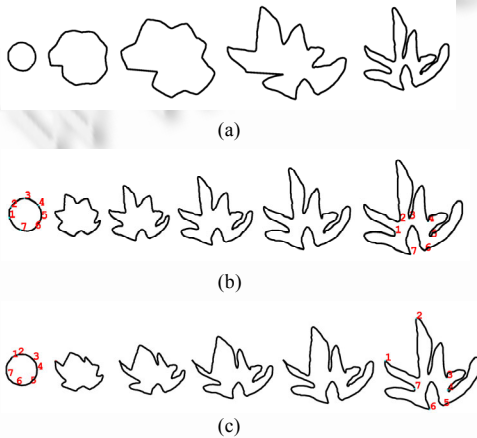


Fig.6 Blending of a circle to a star-shaped object

图 6 圆到星状物的渐变

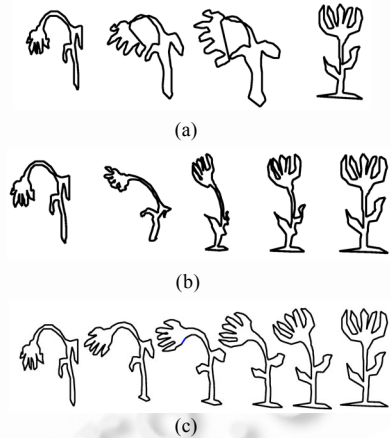


Fig.7 Blending of trees

图 7 树的渐变

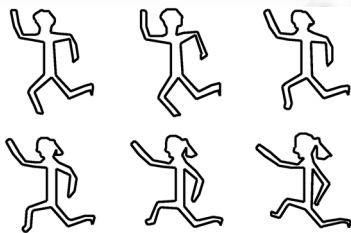


Fig.8 Blending of the runners, seven pairs of feature-decomposition vertices are used

图 8 奔跑的渐变序列,使用了 7 组特征分解点

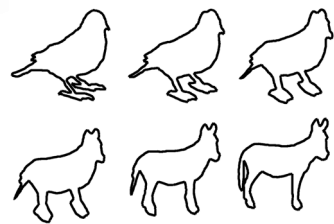


Fig.9 Blending of a bird to a donkey, eight pairs of feature-decomposition vertices are used

图 9 小鸟到驴的渐变,使用了 8 组特征分解点

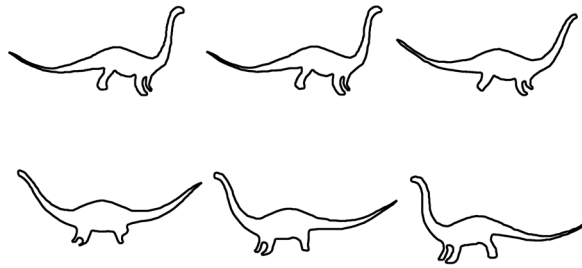


Fig.10 Blending of dinosaurs, six pairs of feature-decomposition vertices are used

图 10 恐龙的渐变序列,使用了 6 组特征分解点

#### 4 结 论

本文提出了一种基于视觉特征分解的 2-D 多边形渐变方法,首先由用户交互地对多边形进行同构特征分解,这样的同构分解同时建立了视觉特征之间的对应,然后将特征子多边形变换到局部坐标系,并对局部坐标系和顶点的局部坐标进行插值,这样,源多边形的特征区域就平滑地过渡到目标多边形的对应特征区域.在交互时,用户通过移动源、目标多边形的特征分解点,可以直观、方便地控制变形区域和效果,这在角色动画编辑中,有很好的应用价值.与已有的方法相比,该方法在进行多边形的特征对应时,方法简单且计算量小,并且我们的顶点插值路径方法保证了源多边形的特征区域平滑地过渡到目标多边形的特征区域.

我们认为进一步的研究内容包括:如何自动地生成满足用户需要的、较好的多边形的特征分解点.当然,完全自动地实现多边形的特征分离是十分困难的事情,这里只是希望能够通过一些自动对应以减少用户的交互工作量.此外,将本文方法推广到具有不同拓扑的多连通多边形情形,以解决复杂多边形的渐变问题,也是值得研究的.

**致谢** 感谢邵瑾等人对本文提出的建议.

#### References:

- [1] Sederberg TW, Greenwood E. A physically based approach to 2D shape blending. *Computer Graphics (Proc. of the SIGGRAPH)*, 1992,26(2):25-34.
- [2] Zhang Y. A fuzzy approach to digital image warping. *IEEE Computer Graphics & Applications*, 1996,16(4):34-41.
- [3] Carmel E, Cohen-Or E. Warp-Guided object-space morphing. *The Visual Computer*, 1997,13(9,10):465-478.
- [4] Sederberg TW, Gao P, Wang G, Mu H. 2D shape blending: An intrinsic solution to the vertex path problem. In: James TK, ed. *Proc. of the SIGGRAPH'93*. New York: ACM Press, 1993. 15-18.
- [5] Shapira M, Rappoport A. Shape blending using the star-skeleton representation. *IEEE Computer Graphics & Applications*, 1995,15(2):44-50.
- [6] Alexa M, Cohen-Or D, Levin D. As-Rigid-As-Possible shape interpolation. In: Marisa C, ed. *Proc. of the SIGGRAPH 2000*. New York: ACM Press, 2000. 157-164.
- [7] Gotsman C, Surazhsky V. Guaranteed intersection-free polygon morphing. *Computers & Graphics*, 2001,25(1):67-75.