中图法分类号: TP391.41 **文献标志码**: A **文章编号**: 1006-8961(2011)09-1745-08
论文索引信息:高岩,陈敏刚,王长波,陈志华,马利庄. 基于层次约束的 2 维动画关键帧插值算法[J]. 中国图象图形学报,2011,16(9):1745-1752

基于层次约束的 2 维动画关键帧插值算法

高岩[1)],陈敏刚[2),3)],王长波[1)],陈志华[4)],马利庄[2)]
[1)](华东师范大学软件学院, 上海 200063) [2)](上海交通大学计算机系, 上海 221000)
[3)](上海市科技信息中心 上海市多媒体公共服务平台, 上海 200235) [4)](华东理工大学计算机科学与工程系, 上海 200237)

摘 要:2D 动画制作是一个劳动密集型的过程,如何自动进行关键帧插值是研究的难点。关键帧插值的核心环节是 2D 形体渐变,现有 2D 形体渐变算法大多针对闭合的单一多边形,在应用到包含多个区域的复杂形体时,可能出现区域之间的重叠。本文算法获取输入关键帧的封闭区域,并迭代计算区域间的匹配关系;对每个子区域建立相对于父区域的局部坐标系,施加层次间的约束关系进行插值,以消除父子区域插值过程中“运动”的不匹配现象。实验结果表明,该算法合成结果自然光滑,具有一定的实用价值。

关键词: 关键帧插值; 形体渐变; 区域匹配; 层次约束

Two-dimensional animation keyframes interpolation based on hierarchical constraints

Gao Yan[1)], Chen Mingang[2),3)], Wang Changbo[1)], Chen Zhihua[4)], Ma Lizhuang[2)]
[1)](*Institute of Software and Engineering, East China Normal University, Shanghai* 200063 *China*)
[2)](*Institute of Information and Computer Engineering, Shanghai Jiaotong University, Shanghai* 221000 *China*)
[3)](*Public Service for Multimedia Industry, Shanghai Science and Technology Information Center, Shanghai* 200235 *China*)
[4)](*Department of Computer Science & Engineering, East China University of Science & Technology, Shanghai* 200237 *China*)

Abstract: Two-dimensional animation is a labor-intensive process, where automatic keyframe in terpolation (inbetweening) is a research focus. Two-dimensional shape-blending plays a key role in keyframe inbetweening. Most of the existing 2D shape-blending algorithms are designed for a single closed polygon. When they are applied to complex shapes containing multiple regions, overlaps between different regions may occur. Our algorithm first obtains closed regions of input keyframes and calculates the matching relations between these regions. Then we iteratively construct a local coordinate system relative to its parent region for each child region. Then the hierarchical constraint relation can be used to get rid of “motion” mismatches between parent-child areas. Experimental results show that our algorithm can synthesize naturally smooth inbetweens and is efficient enough for practical demands.

Keywords: in-betweening; shape blending; region matching; hierarchical constraints

收稿日期:2011-01-17;**修回日期**:2011-05-04
基金项目:国家高技术研究发展计划(863)基金项目(2009AA01Z334);国家自然科学基金(61070128);浙江大学 CAD&CG 国家重点实验室资助项目(A0910);南京大学计算机软件新技术国家重点实验室资助项目(KFKT2009B18);中央高校基本科研业务费专项资金。

第一作者简介:高岩(1973—),男,讲师。2006 年于上海交通大学获计算机应用技术专业博士学位,主要研究方向为计算机动画、虚拟现实、图像处理等。E-mail: ygao@ sei.ecnu.edu.cn。

0 引 言

计算机动画技术被广泛地应用于商业广告,电影特技、动画片、娱乐以及虚拟现实等领域,根据其应用领域的不同,计算机动画可分为 3 维动画和 2 维动画。尽管计算机辅助技术在 3 维动画领域已取得巨大进展,在主流的商业动画软件(如 3DMax,Maya)中都集成了关键帧插值功能。但迄今为止 2D 动画制作仍是一个劳动密集型的过程,所有的动画序列都要由动画师手工完成,这一过程异常费时费力。如何自动实现关键帧的插值引起了众多研究者的关注。

在 2 维关键帧插值动画领域,Burtnyk 和 Wein[1] 利用骨架抽取算法实现中间帧生成,该方法需要大量的手工交互。Kort[2] 提出的关键帧插值算法将每帧分解为笔画和笔画链,并利用自定义规则寻找源和目标的匹配关系。南洋理工大学开发的 CACAni 系统[3-4] 实现了中间帧的自动上色和生成,他们利用 MFBA 算法来计算匹配关系。

在 2 维空间中,一个源形体光滑地过渡到一个目标形体,称为 2D 形体渐变,它是关键帧插值的核心。为了实现 2D 形体渐变,需要解决两个问题:顶点对应和顶点插值路径。Sederberg 等人[5] 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,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[6] 提出的方法中,插值路径被分成两部分: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,通过在仿射标架之间的刚体变换获得。杨文武等人[7] 提出一种基于视觉特征对应的 2D 多边形渐变方法,由用户交互地对多边形进行同构的特征分解建立特征对应来实现变形。

现有 2D 形体渐变算法大多针对闭合的单一多边形,忽略了复杂形体中包含的多个区域间的在插值过程中的内部约束关系,从而导致不自然的结果。本文实现了一个 2 维自动关键帧插值动画系统,首先对输入图像进行矢量化,在此基础上,提取源和目标的封闭区域并作匹配,最后提出带层次约束的顶点路径插值算法,生成平滑的中间帧序列。

1 系统框架

系统的输入是动画序列中用轮廓线表达的起始和终止帧,输出的是从起始帧到终止帧过渡变化的中间帧。系统的工作流程如图 1 所示,系统的主体主要包括:

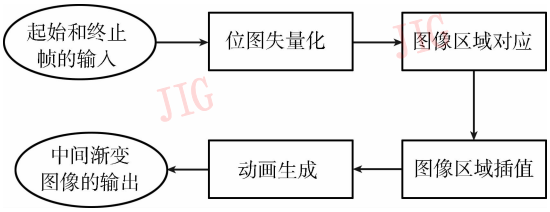


图 1 系统流程图

Fig. 1 System flow chart

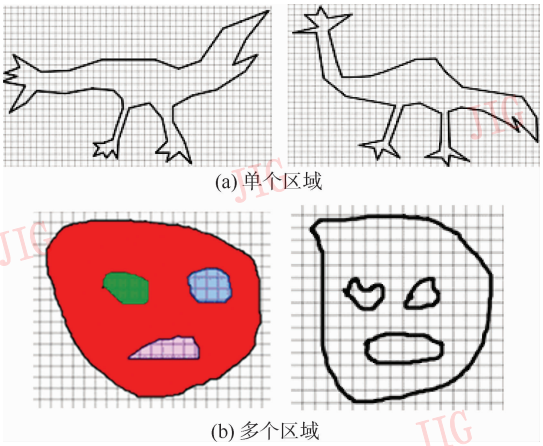
1)位图矢量化模块 该模块将用户输入的轮廓线表示的动画角色分解成各个区域部分,然后对每个部分分别进行矢量化处理;

2)区域匹配模块 该模块将起始帧和终止帧分解出来的区域对应起来,以便后面分别对这些对应区域插值生成中间帧区域;

3)对应区域插值模块 在该模块中,使用基于视觉特征的平面形体渐变算法,插值生成每对区域中间渐变区域;

4)动画生成模块 这一模块将在前一模块中生成的各个中间帧渐变过程进行播放。

系统的输入是动画制作过程中角色轮廓的关键帧,输入的图像可以是着色的,也可以是没有着色的(对未着色的图像可以用上色子模块对其填充颜色)。输入图像的每个区域必须是封闭的,这样才能利用填充算法将每个区域识别出来。系统输入的角色形体,可能只包含单个区域(如图 2(a)),也有可能由多个关联区域组成(如图 2(b))。



(a) 单个区域

(b) 多个区域

图 2 系统输入的图像

Fig. 2 The input image

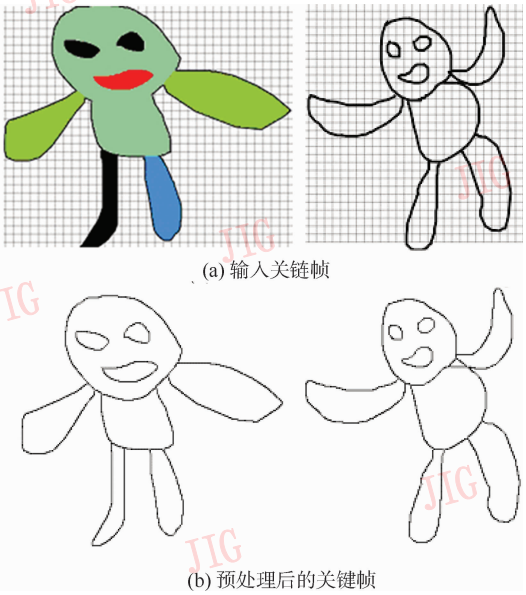
现有的大多数插值算法都是针对单个区域的形体。我们采用分而治之的方法,将每个封闭的区域分开,分别对每个区域进行插值,使问题得以简化。

2　位图矢量化

位图矢量化模块将用户在屏幕上绘制的图像自动转换为矢量形式。系统自动记录用户光标的移动轨迹,当光标离开屏幕后,一段 stroke 即完成。考虑到图像不同区域的线条间存在公共边界,我们在用户完成每一条 stroke 时,将其作为一个图形组件保存下来,直到绘图过程完成后,再对整幅图像矢量化处理。

2.1　图像预处理

图像预处理包括 3 个步骤:二值化、细化和补断。绘图区中每一线条被作为一个绘图组件单独记录。首先利用二值化算法将线条颜色值转化为黑色,其他区域的颜色值转变为白色。接着,图像细化算法被用来将图像中的线条转换为单像素宽度的线条,这里采用数字形态学的方法进行图像细化[8]。由于用户在绘图时可能出现断笔,采用 Seah 的算法[9]对线条进行补断处理。经过这样的处理后得到单像素、二值表示的封闭区域表示的图像,如图 3 所示。



(a) 输入关链帧

(b) 预处理后的关键帧

图 3　图像预处理

Fig. 3　Image preprocessing

2.2　区域获取

区域获取的目的是为了得到关键帧中的每个封闭区域,以便建立正确的匹配关系。系统使用扫描线种子填充算法给输入图像中的每个区域赋一个唯一的标签。系统从上到下、从左到右扫描每个像素,如果某个像素未被贴上标签,则以它为种子,并给它一个新的未被使用过的标签来填充。以这个像素为种子的填充结束后,从原来的位置继续扫描,直到扫描完整幅图像。每个区域的最左上角的像素最先被检测到,并被作为种子赋予一个标签,然后整个区域被一行一行地填充。在同一个区域内部,所有的像素被赋予相同的标签号。这样,一幅图片 $\boldsymbol{I}$ 一定可以被分解成有限的区域的集合:

$$\boldsymbol{I} = \{R(i),\ i = 1,2,\cdots,N\} \tag{1}$$

式中 N 是区域的总数,i 是区域的标号。进一步计算每个区域的面积(即区域内像素的数目)A_i 和每个区域质心的位置 P_i 以供后续使用。将这些数据组合成一个结构,将其作为区域的描述。

图 4 为进行区域获取结果。起始和终止帧的每个封闭区域都被标识出来,并给出标号。

图 4　区域获取后的起始帧和终止帧

Fig. 4　The keyframes after region obtaining

2.3　矢量表示

图 4 提取的封闭区域仍然是基于像素表示的形式,为便于插值,进一步将区域表示为矢量形式。通过提取特征点来产生多边形,选取的特征点包括:比划的交点,比划的端点,曲率过大点以及弧长过大点。其中曲率的计算方法采用 k-cosine 测度[10]近似计算,首先定义点 p_n 的 k 邻域向量

$$\begin{aligned}\boldsymbol{a}_{nk} &= (x_n - x_{n-k}, y_n - y_{n-k})\\ \boldsymbol{b}_{nk} &= (x_n - x_{n+k}, y_n - y_{n+k})\end{aligned} \tag{2}$$

点 p_n 的 k-cosine 测度值为

$$\boldsymbol{\kappa}_{nk} = \frac{\boldsymbol{a}_{ik} \cdot \boldsymbol{b}_{ik}}{|\boldsymbol{a}_{ik}|\,|\boldsymbol{b}_{ik}|} \tag{3}$$

相应的，$\theta = \cos^{-1}(\kappa_{nk})$。对 θ 设置阈值，小于阈值的点即被认为是曲率过大的特征点。同样，对轮廓线走向的像素数目计数，超过阈值的即增加一个特征点。进一步的细节可参见文献[11]。

3 区域匹配

在起始帧和终止帧被分解成各个区域后，为了对每个区域单独进行插值，需要将起始帧的各个区域与终止帧的各个区域相对应，即区域匹配。每个区域有自己的属性，如面积、中心位置等，形成树的叶子节点；对于独立单线条，则链接首位端点，以形成封闭区域。从叶子节点开始，逐层合并生成树：对当前层，计算出所有节点间的包容关系。如果某些叶子节点同处于一个封闭区域内，则构造一个上层节点代表这个外围的封闭区域。这一过程迭代进行，直至形成一个树状结构。

区域匹配算法对代表初始帧和结束帧的树 T_A 和 T_B 中的区域进行匹配。匹配过程从根节点开始逐层进行：

1）在树 T_A 和 T_B 当前层任选一节点 M，计算 M 的面积 A_M 和质心 P_M；

2）对树 T_B 同层所有节点遍历，每次任选一节点 N，计算 N 的面积 A_N 和质心 P_N；

3）计算 M 和 N 的相似性

$$S(M,N) = w_1 \frac{|A_M - A_N|}{\max(A_M, A_N)} + w_2 |P_M - P_N| \tag{4}$$

w_1 和 w_2 为权重，本文取 0.7 和 0.3；

4）按照相似性大小排序，最相似的 M、N 为互相匹配区域；

5）迭代直至结束。

4 区域插值

在图像区域分解和区域匹配之后，区域之间的对应关系已经建立，系统要完成的是对区域的轮廓进行插值以生成中间区域形状。

4.1 顶点对应

该步骤为平面多边形渐变建立特征点之间的对应。

设源多边形和目标多边形分别表示为 $\boldsymbol{P}^0 = \{a_0^0, a_1^0, \cdots, a_m^0\}$，$\boldsymbol{P}^1 = \{a_0^1, a_1^1, \cdots, a_n^1\}$，其中 m 和 n 分别是源和目标多边形的顶点数目。如图 5 所示，采用文献[12]的方法来定义源多边形特征点 a_1^0 与目标多边形特征点 a_1^1 的相似性。

$$D(a_1^0, a_1^1) = w_1\left(1 - \frac{|e_1^0 \times e_2^1 - e_1^1 \times e_2^0|}{|e_1^0 \times e_2^1 + e_1^1 \times e_2^0|}\right) + w_2\left(1 - \frac{|\alpha_1^0 - \alpha_1^1|}{360°}\right) \tag{5}$$

式中 w_1 和 w_2 为权重系数，$0 < w_1, w_2 < 1$ 且 $w_1 + w_2 = 1$。

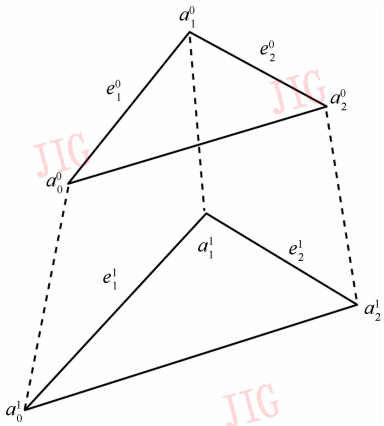


图 5 两个相似三角形间的关系

Fig. 5 Relation between two similar tangibles

根据特征点之间相似代价函数来定义所有源和目标特征点对应的整体相似代价函数。假定 F 是源特征点到目标特征点的一种对应，$F: \{\boldsymbol{P}^0\} \rightarrow \{\boldsymbol{P}^1\}$，定义这一整体对应的相似代价函数为

$$C(\boldsymbol{P}^0, \boldsymbol{P}^1, F) = \sum_{i=0}^{m-1} D(a_i^0, a_{i'}^1) \tag{6}$$

这一最小化问题可以通过动态规划（dynamic programming）技术来解决，由于我们不知道起始对应点如何选取，该算法需调用多次，每次使用不同的初始对应点，最后选择总体对应代价最小的那个对应。

4.2 顶点路径插值

经过以上的处理，可以得到源和目标形体之间的特征点的对应关系。首先采用文献[12]的方法，对每个多边形形体产生中间渐变序列。

由于简单的线性插值一般会产生扭曲和缩减，系统采用极分解算法[13]对刚体变化矩阵 $\boldsymbol{M}$ 进行分解，使得变换后的平移部分与旋转、缩放部分相互分离，以取得更好的插值结果。算法的主要思想如下：

对任一刚体变换矩阵 $\boldsymbol{M}=\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$,可分解为矩阵 $\boldsymbol{BC}$ 的乘积,其中 $\boldsymbol{B}$ 为旋转矩阵,$\boldsymbol{B}=\boldsymbol{M}+\mathrm{sgn}(\det(\boldsymbol{M}))\begin{pmatrix} m_{22} & -m_{21} \\ -m_{12} & m_{11} \end{pmatrix}$,代表 $\boldsymbol{M}$ 的旋转分量,$\boldsymbol{C}=\boldsymbol{B}^{-1}\boldsymbol{A}=\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$。由于 $\boldsymbol{B}$ 是旋转矩阵,可表示为 $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$,故刚体变换矩阵 $\boldsymbol{M}$ 的插值公式为

$$(1-t)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}+\begin{pmatrix} \cos(t\times\theta) & -\sin(t\times\theta) \\ \sin(t\times\theta) & \cos(t\times\theta) \end{pmatrix}\times\begin{pmatrix} t\times c_{11} & t\times c_{12} \\ t\times c_{21} & t\times c_{22} \end{pmatrix}$$

平移向量的插值采用最简单的线性插值即可。

由于 3 对顶点确定一个仿射变换,因此在源多边形中任意选取 3 个顶点,计算它与目标多边形中对应顶点间的刚体变换,并计算出变换代价。变换代价函数综合考虑了三角形的相似性,最小旋转角和三角形面积等因素。最后,选择变换代价最小的刚体变换作为全局仿射变换。具体可参考文献[12]。图 6 给出了一个插值结果的实例。

4.3　层次约束插值算法

从图 6 中可以看出,生成的中间帧序列在左眼珠处出现了脱离左眼约束的现象。这是由于现有的形体变形算法大多考虑的是单一多边形,虽然能生成光滑、特征吻合的结果,但在存在区域嵌套关系的情况下,它们并未考虑区域间彼此的约束关系,由于内层多边形中间帧的“运动”与外层“运动”不一定相吻合,从而导致可能出现约束违背的情形。

从图 6 和图 7 的结果可以看出,如果把左眼珠作为一个单独的形体,则其插值后的结果轨迹符合要求。但将它作为眼睛的一部分,它的变形轨迹不合理。可以这样看待这一问题:每一形体可以单独插值,但在将它们组合起来时必须考虑周围几何形体间的约束关系。

利用相对坐标系来解决问题:

1)对最外层节点,选用绝对坐标值来进行插值计算;

2)对子节点,计算出其父节点质心 C;

3)利用 PCA 分解计算父节点的主轴[14],沿两惯性主轴方向以 C 为原点建立局部坐标系;

4)对子节点,计算其代表的多边形每个顶点在局部坐标系中的相对坐标;

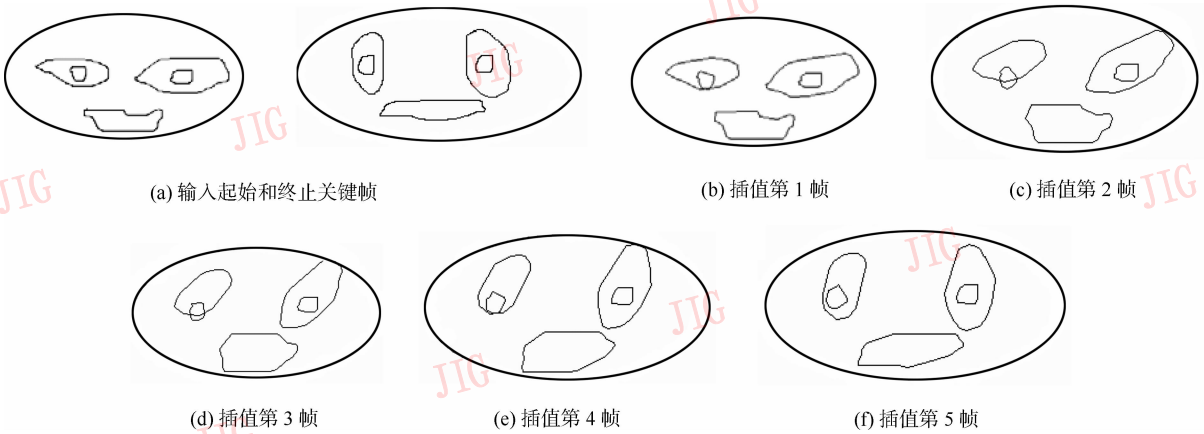
5)对源和目标形体按照相对坐标进行路径插值;

6)对质心轨迹进行线性插值;

7)在每一帧插值出的多边形序列的每个顶点,加上新质心的坐标值,使之转换回绝对坐标;

8)迭代直至所有节点都被访问。

图 7 给出了新的插值结果。

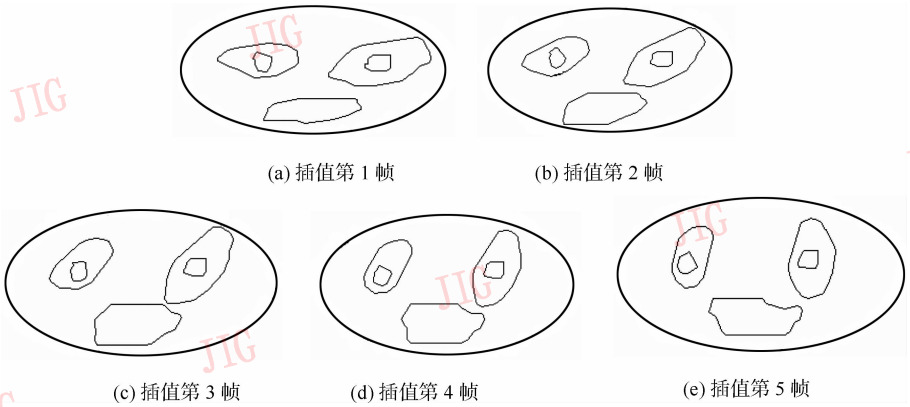


(a) 输入起始和终止关键帧　(b) 插值第 1 帧　(c) 插值第 2 帧

(d) 插值第 3 帧　(e) 插值第 4 帧　(f) 插值第 5 帧

图 6　插值 5 帧后的序列

Fig. 6　Animation sequences after in-betweeing of 5 frames



(a) 插值第 1 帧 (b) 插值第 2 帧

(c) 插值第 3 帧 (d) 插值第 4 帧 (e) 插值第 5 帧

图 7 插值 5 帧后的新序列

Fig. 7 New results after in-betweeing of 5 frames

5 实验结果

对本文算法进行实现。系统环境是 CPU Pentium 4 1.8 GHz,内存 512 M,开发环境是 VC++ 6.0。我们系统的动画模块界面如图 8 所示。图 9—11 给出了部分插值结果的实例。

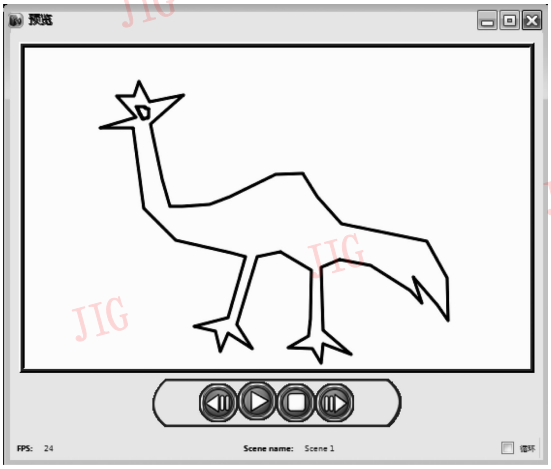
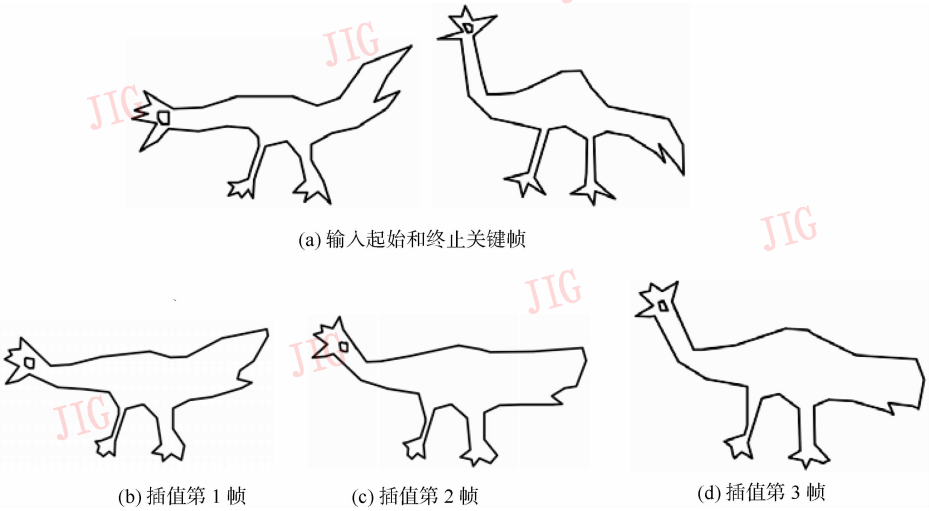


图 8 动画模块界面

Fig. 8 Interface of animation module

6 结 论

本文提出了带层次约束的 2 维动画关键帧插值算法。该算法针对现有形体插值算法大都针对单一多边形而忽略了复杂形体中多个区域间的约束关系,利用区域间的嵌套包含关系,以父节点作为包围空间,构建相对坐标系对子区域顶点进行插值。这样可以消除父子区域插值过程中“运动”的不匹配。



(a) 输入起始和终止关键帧

(b) 插值第 1 帧 (c) 插值第 2 帧 (d) 插值第 3 帧

图 9 插值 3 帧后的序列

Fig. 9 Animation sequences after in-betweeing of 3 frames

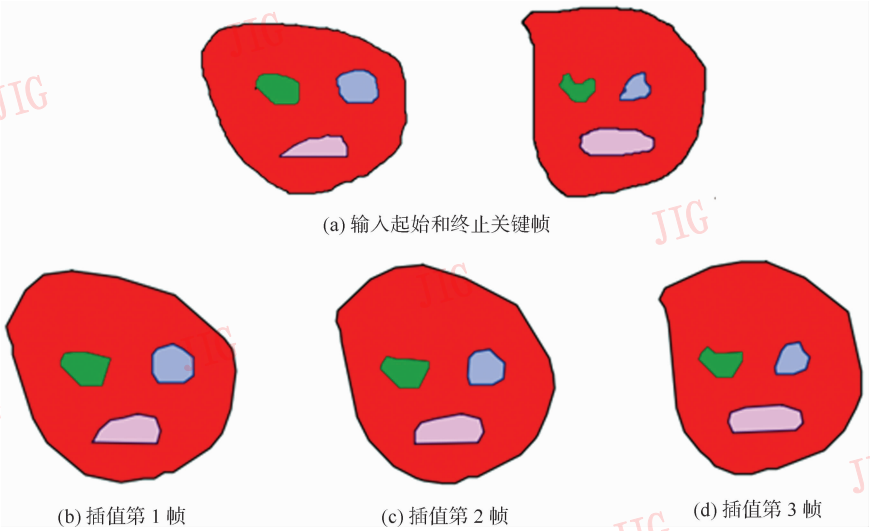


图 10　插值 3 帧后的序列

Fig. 10　Animation sequences after in-betweeing of 3 frames

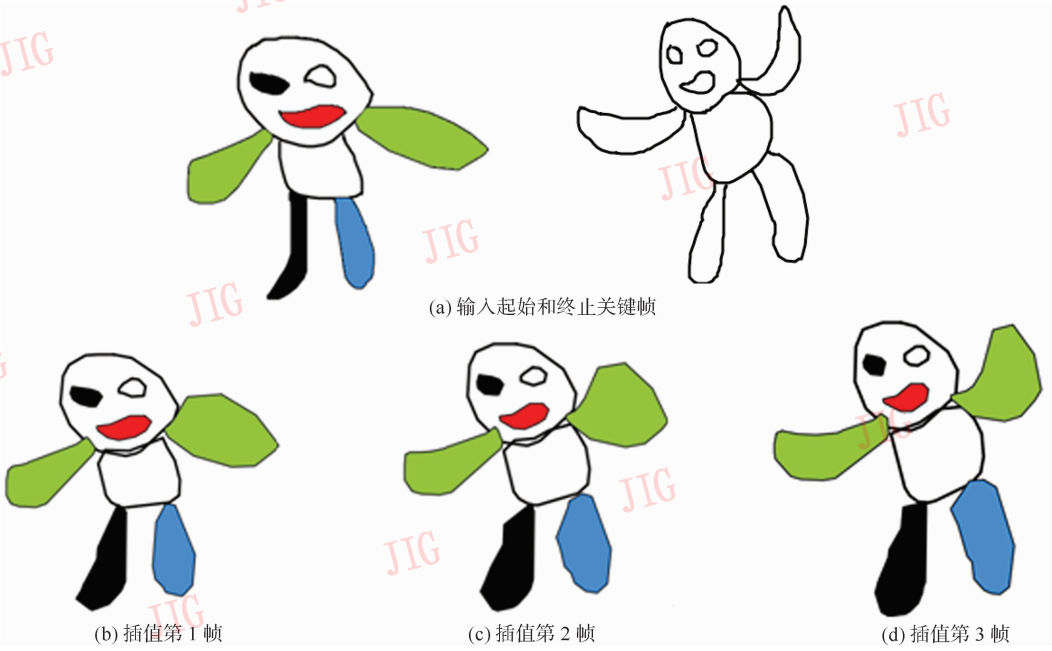


图 11　插值和上色 3 帧后的序列

Fig. 11　Animation sequences after in-betweeing and auto-coloring of 3 frames

对提出算法进行大量实验,实验结果表明,该算法合成结果自然光滑,具有一定的实用价值。当然,该算法也存在不足之处,如顶点路径插值算法对外多边形外形特征保持不够好,造成插值中间帧多边形面积出现较大变动,这一点可以通过选用更好的顶点路径插值算法来改善。

参考文献(References)

[1] Burtnyk N, Wein M. Interactive skeleton techniques for enhancing motion dynamics in key frame animation [J]. Communications of the ACM, 1976, 19(10):564-569.

[2] Kort A. Computer aided inbetweening [C]//Proceedings of the Second International Symposium on Non Photorealistic Rendering. Annecy, France: ACM Press, 2002:125-132.

[3] Seah H S, Lu J. Computer-assisted in-betweening of line drawings: image matching [C]//Proceedings of the 7th International Conference on Computer Aided Design and Computer Graphics. Kunming, China: IEEE Computer Society, 2001: 193-200.

[4] Qiu J, Seah H S, Tian F, et al. Enhanced auto coloring with hierarchical region matching [J]. Computer Animation and Virtual Worlds, 2005, 16(3):463-473.

[5] Sederberg T W, Gao P, Wang G, et al. 2D shape blending: an intrinsic solution to the vertex path problem[J]. ACM Computer Graphics, 1993, 27(4): 15-18.

[6] Carmel E, Cohen D. Warp-guided object-space morphing [J]. The Visual Computer, 1997, 13(9):465-478.

[7] Yang Wenwu, Feng Jieqing, Jin Xiaogang, et al. 2D polygon blending based on feature decomposition [J]. Journal of Software, 2005, 16(2):309-316. [杨文武,冯结青,金小刚,等. 基于特征分解的2D多边形渐变[J]. 软件学报, 2005, 16(2): 309-316.]

[8] Li Ying, Duan Shan. Study of binary image thinning based on mathematical morphology [J]. Journal of South-central College for Nationalities, 2005, 24(4):68-71. [李迎,段汕. 基于数学形态学的二值图像细化算法研究[J]. 中南民族大学学报, 2005, 24(4):68-71.]

[9] Seah H S, Chua B C. A skeletal line-joining algorithm [C]// Proceedings of the Computer Graphics International (CGI'94). Melbourne, Australia: IEEE Computer Society, 1994:62-73.

[10] Teh C H, Chin R T. A scale-independent dominant point detection algorithm [C]//Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition. Ann Arbor, MI, USA: IEEE Computer Society, 1988:229-234.

[11] Seah H S, Feng T. Computer-assisted coloring by matching line drawings [J]. The Visual Computer, 2000, (16):289-304.

[12] Zhang Y. A Fuzzy approach to digital image warping [J]. IEEE Computer Graphics and Applications, 1996, 16(4):33-41.

[13] Shoemake K, Duff T. Matrix animation and polar decomposition [C]//Proceedings of the Graphics Interface 92. Toronto. Ontario, Canada: Canadian Information Processing Society, 1992:258-264.

[14] Cui Chenyang, Shi Jiaoying. Analysis of feature extraction in 3D model retrieval [J]. Journal of Computer Aided Design & Computer Graphics, 2004, 16(7):882-889. [崔晨旸,石教英. 三维模型检索中的特征提取技术综述[J]. 计算机辅助设计与图形学学报, 2004, 16(7):882-889.]