

京剧脸谱表情建模与编辑

蔡飞龙^{1),2)} 于金辉¹⁾

¹⁾(浙江大学 CDA&CG 国家重点实验室, 杭州 310027) ²⁾(宁波大学艺术学院, 宁波 315211)

摘要 为了对京剧脸谱表情进行有效建模和编辑,提出了一个京剧脸谱表情分层驱动的矢量化建模与编辑方法,并首先分析了传统京剧脸谱的绘制过程、局部纹样形状特征以及纹样的分类;然后基于分层原理构造出了一个矢量化的脸谱纹样库。在进行京剧脸谱建模时,首先根据脸部运动编码系统(FACS)标准将脸谱表情分解成 40 个动作单元;然后在合成脸谱过程中用户只需按照脸谱绘制顺序和创作需要逐层选取所需的纹样即可得到一个京剧脸谱图案。对于每个表情动作单元,可利用自由形态变形(FFD)技术来驱动脸谱来生成不同的脸谱表情,另外,用户还可以通过系统提供的编辑工具对脸谱表情进行进一步的调控。

关键词 京剧脸谱 人脸表情 分层模型 FFD 变形

中图分类号: TP391.41 **文献标志码**: A **文章编号**: 1006-8961(2010)06-958-06

Modeling and Editing of Peking Opera Facial Makeup Expressions

CAI Feilong^{1),2)}, YU Jinhui¹⁾

¹⁾(State Key Lab of CAD/CG, Zhejiang University, Hangzhou 310027) ²⁾(Faculty of Arts, Ningbo University, Ningbo 315211)

Abstract In this paper we propose a vector-based hierarchical model for facial makeup's expressions in Peking opera. First we analyze the drawing process of facial makeup in the traditional Peking opera, the shape features of patterns as well as the classification of patterns, and then build a vector pattern bank based on the hierarchical structure of patterns. According to the standards of facial action coding system (FACS), we decompose facial makeup into 40 key action units. During the synthesis process of the facial makeup, users only need to select patterns one by one from the pattern bank to compose needed facial makeup. By using free form deformation (FFD) technique, each expression of action units is obtained. In addition, users may have further control over facial expressions through editing tools provided in the system.

Keywords Peking opera, expressions, hierarchic model, free form deformation

0 引言

表情对于人们表达感情、传递信息及相互交流是非常重要的,许多计算机研究人员曾想用电脑来生成真实的人脸表情。由于人脸表情的复杂性以及人类对表情相当熟悉和敏感,因此要生成高度真实的人脸表情目前还是很困难的。尽管人们可以利用动作捕捉技术来记录面部表情数据,但由于受到捕捉点数量的限制,很多细微的表情无法捕捉,所以捕捉到的表情数据一般都用于真实感要求比较低的动画中。

我国的国粹——京剧在其长达 200 多年的发展

过程中形成了独特的脸谱艺术。脸谱是为了表现京剧人物的性格,突破人物真实面目的局限,在勾画脸谱时,先通过选择其最有代表性的特征加以夸张,并恰当地画在面部上,同时将次要部分省略,这不仅使人远看性格突出、色彩主次分明,而且近看图案精细、美不胜收。脸谱主要用于京剧中的净(花脸)和丑(小丑)两种角色,通过几百年的发展,它在形式、色彩和类型上已形成一定的程式和规律。在传统脸谱分类中,通常将一张脸谱分成额头、眉、眼窝、嘴角等几个部分^[1](如图 1 所示)。

由于脸谱是绘制在京剧演员脸上,因此演员在表演时表情的变化会导致脸上的纹样也发生形状变

基金项目:国家自然科学基金重点项目(60933007);国家科技支撑计划项目(2007BAH11B02);浙江省科技计划项目(2007C21043)

收稿日期:2009-07-01;改回日期:2009-10-20

第一作者简介:蔡飞龙(1978—),男,博士研究生。主要研究方向数字化艺术、互动娱乐。E-mail: caifeilong@cad.zju.edu.cn



图1 京剧脸谱基本组成部分

Fig.1 A facial makeup of Peking opera

化。然而人们平时生活中经常看到的京剧脸谱却没有表情,给人感觉不够生动,缺乏活力。如果能绘制出有表情的京剧脸谱,则能使脸谱更加生动活泼和富有生命表现力,进而还可以生成脸谱表情动画。

目前已经有一些商用的人脸表情编辑软件^[2-4],但主要是基于位图变形技术来生成真实感表情。然而由于脸谱中的纹样形状作为重要的面部特征,这些纹样又具有明显的几何结构,因此如果对脸谱进行位图像素变形操作,则会给纹样带来不合理的变形(见图2)。若要生成有表情的脸谱还需要解决以下两个问题:一个是如何通过纹样的变化来表达不同的表情,另一个是纹样要在保持自己结构基础上能够合理过渡变化。



图2 基于位图变形导致纹样不合理变形

Fig.2 Unnatural deformation of patterns due to bitmap based deformation technique

本文提出了一个基于矢量对象的分层驱动表情控制模型用来解决上述两个问题。在实现过程中,首先分析京剧脸谱绘制过程、局部纹样形状特征以及纹样的分类方法;然后基于分层原理构造出矢量化纹样库;接着根据脸部运动编码系统(FACS)标准

将脸谱表情分解成40个动作单元^[5];最后在合成脸谱过程中,用户只需按照脸谱绘制顺序和创作需要进行逐层选取所需的纹样,即可得到一个脸谱图案,而脸谱表情则通过自由变形(FFD)技术来对脸谱纹样进行变形得到,用户还可以通过调控一些参数来对脸谱表情进行进一步的调控。与基于位图变形的表情编辑系统相比,本文方法既可以简单快捷地生成各种复杂的脸谱表情,又保持了纹样的几何特征。它在数字博物馆、京剧教学、卡通动画、装饰设计及旅游产品开发等领域有广泛应用前景。

1 相关工作

人脸表情的研究已有30多年的历史,Parke等人早在1972年就采用了一个3维模型^[6],其根据不同表情的照片,先通过调整模型顶点来得到不同表情的关键帧;然后进行插值得到动画效果。Ekman等人提出了一个肌肉模型^[5,7],他们定义了一个脸部运动编码系统(FACS),虽然只有50个运动单元,但组合起来可生产上千种表情。基于FACS标准,Waters定义了不同人脸的共同参数^[8],允许更方便灵活地控制人脸表情。在Waters工作的基础上,Terzopoulo等人改进了人脸绘制过程^[9],在早期的工作站上的绘制速度可以达到实时。

在非真实感表情动画研究中,Fabian等人做了一些基于2.5维矢量的卡通表情变形动画^[10-11],但是要手工绘制每个极值状态的矢量图,而且要求不同状态的每个点一一对应,这需要用户大量的交互和设定极值状态。笔者曾经用Bezier曲线构造出脸谱的矢量化纹样库并合成出不同的京剧脸谱^[12]。由于不同的脸谱具有不同的纹样,如果用Fabian等人提出的方法为每个脸谱中的每个纹样指定极值状态,则涉及的交互工作量太大,所以需要研究新的方法生成脸谱表情。

受Ekman的脸部动作编码与动作单元相关联的思想启发,通过对传统京剧脸谱的绘制过程、局部纹样形状特征以及纹样的分类方法的深入分析研究,本文提出了一个基于矢量对象的分层驱动表情模型,即首先将脸谱分层矢量化为相对独立的纹样,并以SVG格式存储到纹样库中;然后参考FACS的动作单元标准,将脸谱表情分解为40个动作单元;接着把纹样与FFD变形框进行关联得到动作单元库,其中某个表情动作的信息可通过无形变和有形变FFD变形

框控制点之间的差异来表示;最后将这些 FFD 变形信息分别作用于相应的矢量纹样上,用来驱动矢量纹样产生相应的变形,即可生成相应的表情变化;最后用户还可以进一步通过变化控制信息来调整脸谱表情的变化程度。图 3 为系统的流程图。本文将在下面的几节对模型的主要细节进行具体说明。

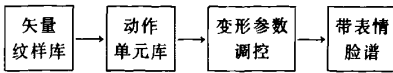


图 3 系统流程图

Fig.3 System framework

2 矢量纹样库

在手工绘制脸谱时一般是先在脸上打上红底色,然后画出眉和眼窝处的白色纹样,再在其上面分别画出黑色眉纹和眼窝纹。左右眉纹和眼窝纹完成之后,再添加额头纹以及嘴角纹(参见图 1)。本文按照传统脸谱的绘制顺序和局部纹样分类,将脸谱分为眉纹、眼窝纹、额头纹、嘴角纹等。由于每个纹样造型各异,其边缘弯曲且光滑,特别适合用 Bezier 曲线构造,因此本文先用 Bezier 曲线手工分层构造出各部分脸谱纹样(见图 4(a))。但由于每条 Bezier 曲线的参数有 4 个控制点,且每个纹样都需要好几条 Bezier 曲线去构造,而 Bezier 曲线的控制点的位置又比较自由,很多控制点的位置离其控制的 Bezier 曲线距离很远,因此不符合后面要采用的 FFD 变形的要求。

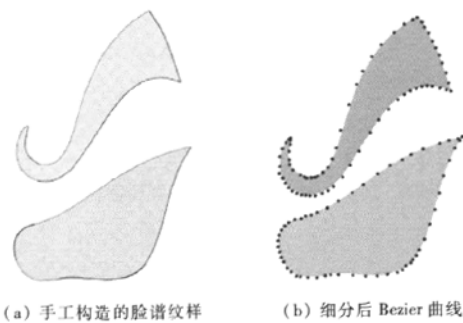


图 4 Bezier 构造的纹样和细分点构成的纹样

Fig.4 Patterns represented by Bezier curve and subdivided points

本文用 De Casteljau 细分算法^[13]对 Bezier 曲线先进行细分,即根据曲率变化将 Bezier 曲线细分为一定数量的点;然后用直线连接这些点用来代替原

来的 Bezier 曲线(见图 4(b))。鉴于 De Casteljau 细分是一种比较常用的算法,本文不做具体描述。对于任意一个纹样,可先用若干条这样细分后的曲线来构造;然后将这个线段构成的纹样储存为 SVG 格式的矢量图;最后按区域位置和纹样类型分类保存到纹样库中。图 5 为纹样库中的一些纹样。在纹样构造过程中,用户只需用 Bezier 曲线设定纹样的轮廓,接下来的细分和存储过程则由系统自动完成。



图 5 纹样库中的一些纹样

Fig.5 Some patterns in the pattern bank

3 FFD 动作单元库

参考 FACS 的动作单元标准,本文首先将脸谱表情分解为 40 个动作单元。在驱动纹样变形时,本文采用了 FFD 方法,其优点是既能实现纹样的变形,又保持了纹样的几何特征。

3.1 FFD 变形

FFD (free form deformation) 是 Sederberg 等人提出的一种对任意形状进行自由变形的技术^[14],其主要思想是把要变形的物体嵌入一个变形空间内,先建立起变形物体和变形空间的联系,然后对变形空间进行变形,嵌入其内的物体随之变形。FFD 既可以在 3 维空间实现,也可以在 2 维空间实现。本文主要讨论 2 维平面内的 FFD 变形。2 维空间上的 FFD 就是一个从 $\mathbf{R}^2 \rightarrow \mathbf{R}^2$ 的映射,它为每个点在一个区域内定义了一个新的位置。

变形时,首先通过指定点 (X_{\min}, Y_{\min}) 和 (X_{\max}, Y_{\max}) 定义了一个初始变形区域(见图 6);然后根据变形框位置生成相应的初始化控制点。变形可以通过一个二变量张量积伯恩斯坦多项式的 Bezier 平面控制,可用如下公式表示为

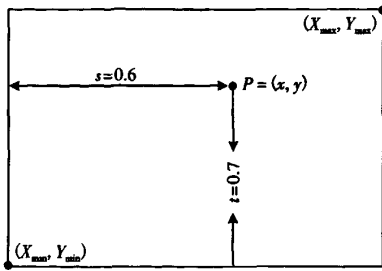


图 6 FFD 变形区

Fig. 6 FFD deformation region

$$X(s, t) = \frac{\sum_{j=0}^m \sum_{i=0}^n w_{i,j} B_i^n(s) B_j^m(t) P_{ij}}{\sum_{j=0}^m \sum_{i=0}^n w_{i,j} B_i^n(s) B_j^m(t)} \quad (1)$$

由于伯恩斯坦多项式的性质,致使移动一个控制顶点将影响变形框内的整个空间。也就是说,当变形对象全部包含在控制框内时,移动一个控制顶点,变形对象的所有点都受到影响。这里定义 m 和 n 分别为 FFD 控制点的横栏和纵栏的数目, $B_i^n(s)$ 和 $B_j^m(t)$ 是 Bezier 混合函数, s 和 t 为一个点在变形区的局部坐标,对于在变形区的一个点,其中 s 和 t 的范围在 $[0, 1]$ 之间。

$$s = \frac{x - X_{\min}}{X_{\max} - X_{\min}}, t = \frac{y - Y_{\min}}{Y_{\max} - Y_{\min}} \quad (2)$$

要计算一个点变形后的位置,可先通过式(2)计算它的 (s, t) 坐标,如果它们的值在 $[0, 1]$ 之间,则用式(1)即可以计算得到新的位置。

由于在脸谱中某类纹样具有不同的形状(参见图 5 中的眉纹),因此在系统中为了实现对同类纹样的统一处理,本文将纹样库中所有同类纹样放到一起进行比较,首先找到在前面构造它们的经细分后点的坐标最大值和最小值,作为初始化 FFD 变形框的 (X_{\min}, Y_{\min}) 和 (X_{\max}, Y_{\max}) 的位置;接下来根据局部纹样的变形幅度不同设定不同的控制点数,对于图 7 的眼窝纹,

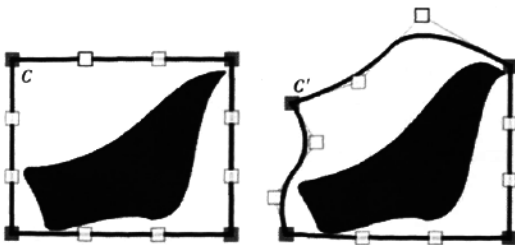


图 7 纹样在 FFD 变形前后比较

Fig. 7 A comparison of the results before and after FFD

本文先设定了 12 个控制点;然后根据人脸肌肉运动参考图,移动控制点到相应的位置。每个控制点 $C(i)$, $(i = 1, \dots, 12)$ 的相对位移就可以通过下式计算得到:

$$d_x(i) = \frac{\hat{C}_x(i) - C_x(i)}{X_{\max} - X_{\min}}, d_y(i) = \frac{\hat{C}_y(i) - C_y(i)}{Y_{\max} - Y_{\min}} \quad (3)$$

这样最后 12 个控制点的相对位移就作为眉纹的一个动作数据被存储到动作库中。

3.2 FFD 动作单元库

FACS 是通过分析人脸的生理结构而获得的人脸面部肌肉、下巴和舌头动作的描述。FACS 包含 50 个基本动作单元(Action units, AUs),不同动作单元的组合产生不同的脸部表情。Ekman 等人已经编制出用基本动作单元表示的常见人脸编码供使用者查阅。

由于脸谱仅用一些纹样表现面部特征,因此真实感表情中的一些细微肌肉变化可以省略掉,本文在系统中为脸谱设定了 40 个动作单元。以图 8 的抬右眉内侧动作单元(AU1)为例,其涉及的纹样包括右眉纹、右眼窝纹、额头纹和右眼底纹,还需建立相应的右眉纹-AU1-FFD、右眼窝纹-AU1-FFD、额头纹-AU1-FFD 和右眼底纹-AU1-FFD 的变形动作。对右眉纹有影响的运动单元包括抬右内眉端动作单元(AU1)、抬右外眉端动作单元(AU3)、抬右主眉动作单元(AU5)和压右主眉动作单元(AU7)。

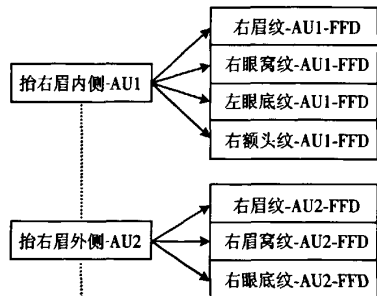


图 8 动作单元库

Fig. 8 Action units bank

综合起来,本文系统中的 40 个动作单元一共包含 152 个纹样 FFD 变形动作,此外还有 10 个眼睛和牙齿的配套动作的运动单元库。

图 9 为将同一动作作用于不同纹样对象的结果,其中上面 4 张图为由直眉纹和方角直眼窝纹(表示人物性情较刚烈)构成的眼部表情,下面 4 张图为由棒槌眉纹和尖角直眼窝纹(表示人物性情较含蓄)构成的眼部表情。其中图 9(a)对应的是基本状态;图 9(b)对应的是内眉端下压的动作状态;



图 9 同一动作对不同纹样的影响结果比较
Fig.9 A comparison of two patterns in one action

图 9(c) 对应的是外眉端上抬的动作状态;图 9(d) 是将图 9(b) 和图 9(c) 两个动作混合的表情状态, 表示了脸谱愤怒下的眼部的表情状态。

4 脸谱表情生成

在生成脸谱表情时,用户可以直接通过选择一个系统预置的表情模板直接作用于脸谱来生成相应的表情(见图 10),还可以对其进行简单的整体表情幅度调控,以便产生同一表情的不同幅度结果。用户也可以根据创作需要,对每个局部运动单元的参数进行



图 10 脸谱表情调控界面
Fig.10 The user interface of expression control

调节,而系统则根据用户的调节先对每个纹样所用动作的 FFD 变形进行累加得到其最终的 FFD 变形,然后驱动相应纹样变形得到用户希望的变形结果。

例如要得到一个悲伤的表情,需调节的动作单元有:抬右内眉端、抬左内眉端、压低右眉、压低左眉、下拉右嘴角、下拉左嘴角。图 11 为利用本文系统所创作的脸谱表情,其中上面两栏的图 11(a)一



图 11 本文系统生产的脸谱表情
Fig.11 Facial makeup's expressions created by the system

(d)分别为正常、高兴、悲伤和愤怒的表情状态,最上面的红色脸谱和其下边对应的黄色脸谱采用了相同的参数设定。由于在悲伤和愤怒表情中嘴部张开,因此本文添加了牙齿以保证脸谱形象的完整。图11第3栏对应的为不同程度愤怒的表情,用户通过滑杆调节表情幅度百分比即可,图11(a)~(d)对应的幅度百分比分别为0,30%,60%,100%。

5 结 论

本文提出了一个由计算机辅助生成矢量化京剧脸谱表情的方法。该方法简单直观,整个交互过程都是面向用户需求的,它并不需要用户具有专业的京剧脸谱方面的专业知识或者进行复杂的交互操作。用户只需先按顺序从纹样库中选取所需的纹样,接着再通过调节脸谱各个部分的表情参数,即可得到各种表情的京剧脸谱矢量图。相对于基于位图的表情编辑系统,该方法可以处理大幅度纹样变形,而不会产生不自然的变形或者锯齿现象。虽然该系统可快速生成大多数类型的脸谱表情,但对于整体结构复杂的脸谱,如在某些花脸、歪脸等脸谱中由于有些纹样犬牙交错,难以用FFD变形框对它们进行独立调控,因此希望能在接下来的工作能对此系统进行改进。

参考文献 (References)

- [1] Zhao Menglin. Chinese Facial Make-ups in Peking Opera [M]. Beijing: Zhaohua Press, 2003. [赵梦林. 中国京剧脸谱 [M]. 北京: 朝华出版社, 2003.]
- [2] CrazyTalk 5 [EB/OL]. <http://www.reallusion.com/crazytalk/>, 2009-05-16/2009-05-16.
- [3] Zealsoft Fun Morph [EB/OL]. <http://www.funmorph.com/fun-morph.html>, 2009-05-16/2009-05-16.
- [4] Abrosoft FantaMorph [EB/OL]. <http://www.fantamorph.com/overview.htm>, 2009-05-16/2009-05-16.
- [5] Ekman P, Friesen W V, Hager J C. The Facial Action Coding System (2nd edn) [M]. London, UK: Weidenfeld & Nicolson, 2002.
- [6] Parke F I. Computer generated animation of faces [C] // Proceedings of ACM National Conference on Computer Graphics, Boston, MA, USA: ACM Press 1972: 451-457.
- [7] Ekman P. Emotion in the Human Face [M]. Cambridge, UK: Cambridge University Press, 1982.
- [8] Waters K. A Muscle model for animating three-dimensional facial expression [C] // Proceedings of ACM Conference SIGGRAPH '87, New York, ACM Press, 1987: 17-24.
- [9] Terzopoulou D, Waters K. Physically-based facial modelling, analysis, and animation [J]. The Journal of Visualization and Computer Animation, 1990, 1(2): 73-80.
- [10] Fabian D F, Philip S, Koen E, et al. Automatic in-betweening in computer assisted animation by exploiting 2.5D modelling techniques [C] // Proceedings of Computer Animation 2001, Seoul, Korea: IEEE Press, 2001: 192-200.
- [11] Fabian D F, Frank V R. Multi-level performance-driven stylised facial animation [C] // Proceedings of Computer Animation and Social Agents 2005, Hong Kong, China: IEEE Press, 2005: 73-78.
- [12] Cai Feilong, Yu Jinhui, Peng Ren. Analysis and synthesis of peking opera facial make-ups [J]. Journal of Computer-Aided Design & Computer Graphic, 2009, 21(8): 1092-1097. [蔡飞龙, 于金辉, 彭韧. 京剧脸谱分析与合成 [J]. 计算机辅助设计与图形学学报, 2009, 21(8): 1092-1097.]
- [13] Farin G, Hansford D. The Essentials of CAGD [M]. Natick, Massachusetts, USA: A K Peters, Ltd. ISBN 1-56881-123-3. 2000.
- [14] Sederberg T W, Parry S R. Free-form deformation of solid geometric models [J]. ACM SIGGRAPH Computer Graphics, 1986, 20(4): 151-160.