

# 基于线型草图的多源3维动画传输

卢涤非 张三元 叶修梓

(浙江大学计算机学院, 杭州 310027)

**摘要** 为了对3维动画进行鲁棒地传输,提出了一种基于线型草图的3维动画传输方法,该方法可以把多个源网格的动画传输到一个目标网格。该方法由以下几个步骤组成:(1)使用线型草图在源网格和目标网格间建立映射;(2)使用差分均值坐标(differential mean value coordinates),根据源动画来变形目标网格;(3)对齐各部分变形后的目标网格,通过求解最小化约束函数,以获取光滑的目标网格,从而生成目标动画的关键帧;(4)对由步骤(3)产生的结果进行插值,即可产生完整的目标对象的3维动画。通过对上述几个方面的深入研究,以及将这些成果集成在原型系统中的结果表明,该方法具有以下特点:①既不需要源网格和目标网格有相同的顶点数和三角面片数,也不需要类似的拓扑信息;②源动画既可以是3维的,也可以是2维视频或Flash;③目标对象可以是无结构化的点云数据。实践表明,该方法直观易用,可以生成逼真的3维动画。最后通过从狗、猫、蛇、人体的3维动画与2维车轮视频的合起来产生复杂的马车动画,对该方法进行了验证。另外,原型系统还通过产生其他几个例子来进一步验证其通用性和鲁棒性。

**关键词** 草图 变形 动画 差分均值坐标 交互式动画制作工具

中图法分类号:TN919.8 文献标识码:A 文章编号:1006-8961(2008)05-0958-14

## Sketch Based Multi-source 3D Animation Transfer

LU Di-fei, ZHANG San-yuan, YE Xiu-zi

(College of Computer Science, Zhejiang University, Hangzhou, 310027)

**Abstract** In this paper, we propose a sketch based synthesis approach to create 3D animation of a target character. Our approach consists of the following steps: (a) a sketch based mapping between the sources and the target; (b) deforming the target based on the deformations of the corresponding sources by means of differential mean value coordinates; (c) stitching the deformed portions of the target together smoothly with constrained minimization method to form a key frame of the target animation; and (d) interpolating between key frames created from (c) to produce a complete 3D animation of the target. We provide new research contributions on all these topics, and integrate them into our newly developed prototype animating system. Our approach does not require the sources and target to share the same number of vertices or triangles, or to have matching connectivity. In our system, source animations can be 3D animations as well as 2D videos or macromedia flashes; and the target can even be unstructured or point cloud. Our approach is intuitive and is able to produce highly authentic 3D animations. We demonstrate our approach by constructing a full cart animation from dog, cat, snake and human animations as well as applying a 2D video of the wheel of a car onto the wheels of the cart. Other examples produced from our prototype system are also given in the paper to illustrate the generality and robust of our approach.

**Keywords** sketch, deformations, Animation, differential mean value coordinates, interactive animating tools

基金项目:国家自然科学基金项目(602720601)

收稿日期:2006-08-24;改回日期:2006-11-06

第一作者简介:卢涤非(1972~),男。2007年6月获浙江大学计算机学院博士学位,现为浙江警察学院教师。主要从事与公安业务相关的图形图像技术研究。E-mail:10221039@zju.edu.cn

## 1 引言

尽管许多优秀的动画制作软件得以开发,并已投入商业应用,但这些系统的使用局限于一些经验丰富的专家,而简单易用的动画制作技术依然是一个具有挑战性的主题。另一方面,3维草图(sketch)虽非常适合初学者,但在实际应用中,其功能仅局限于简单的网格和动画处理。同时,尽管已有无数耗费大量时间创作出来的、具有高度艺术性和技巧性的动画作品,但几乎没有现成的技术可用来帮助重用(reuse)这些素材。基于这种情况,本文提出了一套方法,其能够方便地重用现有的动画作品,从而可快速容易地创建逼真的3维动画。

本方法不需要知道源动画是如何生成的,具有很大的通用性。该技术完全是样例驱动(by example)的,不需要源网格和目标网格拥有相同的顶点数和三角面片数以及相似的拓扑信息,甚至目标网格可以是非结构化的数据,如点云数据。再者,该方法也适合源网格和目标网格没有明确的语义(semantic)对应关系的情况,用户只需使用草图工具在源对象和目标对象两者上指定哪些部分应该以类似的方式运动。通过原型系统上的交互工具,就可以先让用户在多个源动画上指定拷贝区域,然后把这些区域的动画复制到对应的目标网格上。诚然,由于源对象和目标对象可能非常的不相似,因此在目标网格上生成与源动画一致的动画在艺术性上并不总是非常合意的(如一匹马如何像一个人体模型一样行走?),然而,可寄希望于用户合理地选择源动画来解决这一问题。本文研究的重点就是如何把源动画忠实地传输到目标对象上。

使用本文的方法,一个没有经过任何训练的用户都可以很容易地快速生成翔实的3维动画。用户只需在现有的动画基础上展开工作,而不需要像商业动画软件(如Maya,3DMax)那样,进行复杂的操作。利用该方法,用户几乎不需要从零开始创建新的动画,而且在草图工具的帮助下,用户可以利用差分均值坐标在源网格部件(part)与目标网格部件间建立对应关系。对目标网格上的每一个顶点,均可计算出与源网格顶点间的相应的均值坐标系数。源网格顶点的坐标变换可以先驱动目标网格以源动画展示的风格变形;然后汇集目标网格的各部分动画,并对目标网格的各部分进行对齐和光滑处理。光滑

处理是通过求解4个约束方程的最小值来完成的。

本文的主要贡献如下:(1)提出了一个新的在源网格和目标网格间基于线型草图建立关联(correspondence)的方法;(2)将仅适合闭合网格的均值坐标扩展为适合非闭合网格的差分均值坐标(differential mean value coordinates);(3)提出了从2维动画生成3维动画的方法;(4)提出了一组基于仿射变换矩阵的光滑处理约束函数,这些函数能够处理非结构化的数据,如点云数据。附录分析了求解这些方程的数值上稳定的方法;(5)最重要的贡献是通过组和这些不同的算法,提出了一个基于样例(by example)的完整的动画制作系统。

动画制作大致可以分为基于样例(by example)和基于控制(by controller)两大类。许多学者在基于控制方面做了大量研究<sup>[1-6]</sup>,但这些方法都需要用户有细致的输入,这对初学者是一个巨大的挑战。基于样例的动画合成是从现有的动画开始,而不是从零起步。有许多研究可以归类为基于样例的范畴<sup>[7]</sup>。文献[8]提出了一个利用在数据库中的现有网格构建一个新的模型的算法(modeling by examples)。文献[9]研究了基于网格的反向运动学(inverse kinematics),它依赖样例网格来指示各种有意义的变形分类。还有文献[10]~[14]分别研究了基于样例的动画制作、变形和建模等方法。

文献[15]介绍了一个变形传输(deformation transfer)算法,这个算法把一个网格的变形移植到另外一个网格上去。在这个方法中,针对网格的每个三角面片的变形都构造一个仿射变换矩阵,整个网格的变形可以通过一组矩阵来表示,这种表示方法包含了由网格三角面片变形引起的方向(orientation)、缩放(scale)以及偏斜(skew)等变化的信息,并将源网格三角面片的仿射变换矩阵应用于与其相对应的目标网格的三角面片上,但这种技术局限于大致相似的网格之间,并且只能把整个网格的变形传输另外一个网中。另外,文献[15]还提出了一套基于网格的控制光滑度和防止网格局部剧变的算法。

闭合三角网格的均值坐标(mean value coordinates)是在文献[16]中提出的,其主要用来对在网格顶点上定义的值进行连续光滑的插值。它主要的应用包括体纹理(volume texture)和曲面变形。本文把它扩展为能够应用于非闭合三角网格的差分均值坐标(differential mean value coordinates)。文献

[17] ~ [19] 提出了仿射矩阵的极分解 (polar decomposition), 它被用于高刚性 (as rigid as possible) 的曲面插值和操作, 在本文中用来控制变形的刚性。

文献[1]和文献[3]提出了基于草图的网格变形算法, 本文借鉴并扩展了该算法的一些思想, 提出了基于草图的创建关联 (correspondence) 的方法。本文使用线型草图来确定感兴趣区域 (regions of interest, ROI), 并把其作为控制点的源网格顶点映射到目标网格上。

## 2 算法概述

在用本文算法进行 3 维动画传输时, 输入的是若干个动画样例, 输出的是目标角色的一个新的 3 维动画。其操作的第 1 步是首先使用草图工具在源网格上选取希望得到的部分, 并与目标网格的相应部分关联, 而且在每个关键帧上, 目标网格以类似于源网格的运动方式变形; 然后把各部分目标动画对齐, 并进行光滑处理; 最后, 通过插值关键帧来产生完整的目标角色的动画。本文所使用的草图都是线型草图, 故在后面的描述中, 草图即指线型草图。

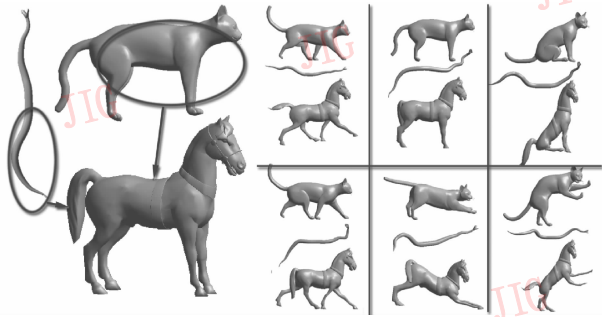


图 1 演示本算法主要特征的一个简短例子

Fig. 1 Brief instance demonstrating the main features of our system

图 1 是由本文算法的原型系统把蛇尾的动画和猫躯干和四肢的动画合成到一匹马上来的一个简洁例子。图中的蛇和猫是两个源动画, 马是目标角色。蛇尾的变形被复制到马尾上, 猫躯干和四肢的变形被复制到马躯干和其四肢上。用户所要做的就是画 12 条草图: 1 条在蛇尾上; 1 条在猫躯干上; 4 条在猫四肢上; 1 条在马尾上; 1 条在马躯干上; 4 条在马四肢上。完成这些后, 用户只需指定一些参数, 如插值频率, 就可以生成组合了蛇和猫动画的一个

全新的马的动画。

在这个例子上, 总共产生了 6 帧马的动画。马尾以类似于蛇尾的方式运动, 而马的四肢与躯干则以类似于猫的方式运动。产生马动画的关键步骤如下: (1) 在源网格和目标网格上绘出成对的草图 (sketch)。对于每对草图来说, 其中一条是源草图, 它用于指定源网格的感兴趣区域; 另一条是目标草图, 它用于指定相应的目标网格的感兴趣区域; (2) 基于最近距离原则, 找出关联到源草图的源网格顶点和关联到目标草图的目标网格顶点, 在计算最近距离时, 要考虑草图上的任意点, 而不仅仅是折点 (vertex); (3) 把源草图映射到目标草图上, 同时将与源草图关联的源网格顶点也一同映射到目标草图上; (4) 针对目标网格的每一个顶点, 计算出差分均值坐标的权值; (5) 在每个关键帧上, 使用非闭合三角网格的均值坐标来计算出目标网格顶点的新坐标; (6) 在新的位置对齐目标网格的各个部分, 然后对结果进行光滑处理; (7) 在关键帧间进行插值, 即生成目标对象的完整动画。

尽管上述例子比较简单, 但揭示了本文算法的如下主要特征: (1) 基于草图的映射把源网格和目标网格关联起来; (2) 非闭合三角网格的差分均值坐标能够使目标网格以源动画的风格进行变形; (3) 组合各部分变形后的目标网格, 经光滑处理形成目标动画的关键帧; (4) 经插值输出完整的目标动画。

## 3 基于草图的映射

### 3.1 绘制草图并生成相应 ROI

首先, 一个源网格和目标网格间的双向对应关系 (bi-jective mapping) 需要建立起来。文献[15]在一组用户选择的标记点 (marker point) 的基础上自动计算三角面片间的关联 (correspondence), 建立起源和目标的对应关系。但这种关联技术是一个迭代的最近点算法, 比较耗时, 并对错误比较敏感。在本文的方法中, 是使用一对草图来分别指定源 ROI 和目标 ROI, 并在网格表面上自由 (free-form) 绘制用来输入草图。有以下两类草图: (1) 在网格表面上的草图如图 2(a) 所示, 其是把视点与鼠标所在位置的连线与网格表面的第 1 个交点 (即靠近视点的交点) 作为草图的顶点; (2) 在网格内部的草图 (图 2(b)): 其是计算视点与鼠标所在位置的连线与

网格表面前两个交点(即靠近视点的两个交点),并把它们的中点作为草图顶点。但当使用像鼠标这样低精度的输入设备画草图时,原始草图会包含相当多的噪音(图 2(c))。原始草图需要以如下方式被光滑处理和归一化(regularize):首先对原始草图的顶点进行插值产生 3 次 B 样条曲线,然后光滑这条曲线,并把它离散化为草图的一组新的顶点(图 2(d))。经过这样处理可以提高映射的精度。

草图的第 1 个作用就是指定源网格和目标网格的 ROI,即找出草图与网格顶点间的联系。本文使用一个距离阈值和对一个隔离平面(在图 2 中,为了分开两条腿,在两腿间可设置一个隔离平面的可见性(visibility)来自动从草图中计算出 ROI。而任何到草图顶点或线段的最短距离小于这个阈值的顶点都被作为 ROI 的一部分。从图 2(b)中显示的草图可以计算出如图 3(a)所示的 ROI(右腿,被称为源 ROI)。

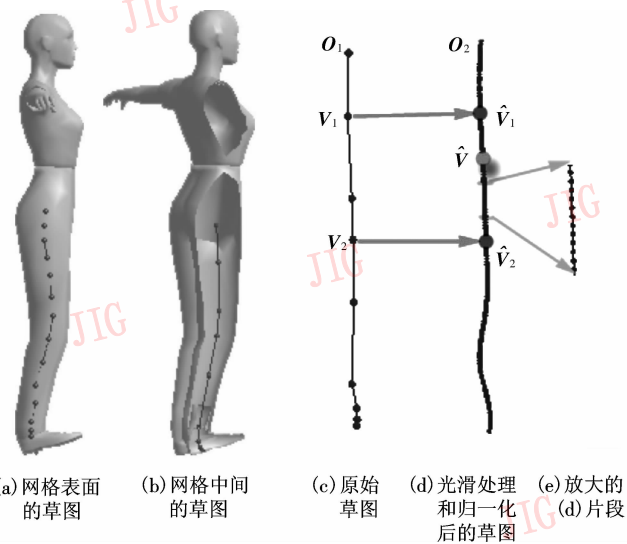


图 2 草图输入及处理

Fig.2 Sketches input and process

### 3.2 映射源 ROI 到目标 ROI

图 3 演示了一对草图把人腿模型映射到相应马腿上的过程。源草图和目标草图在图 3(b)和 3(c)上分别用折线显示。在图 3(d)上的右前腿区域是从目标草图中计算出的 ROI(称为目标 ROI)。在图 3 中, $O_s$ 和  $O_t$ 分别为源草图和目标草图的起始顶点。 $V_s$ 是与源 ROI 上的顶点  $V$  关联的源草图上的点; $V_t$ 是  $V_s$  基于弧长参数化(arc-length parameterization)映射在目标草图上的点。图 3(a)和图 3(d)分别显示了对应于图 3(b)和图 3(c)中

草图的 ROI。图 3(b)和图 3(c)上的浅色线连接了 ROI 上的顶点和它们在草图上对应的点。图 3(e)是源网格顶点被映射到目标 ROI 后的结果(框线框)。

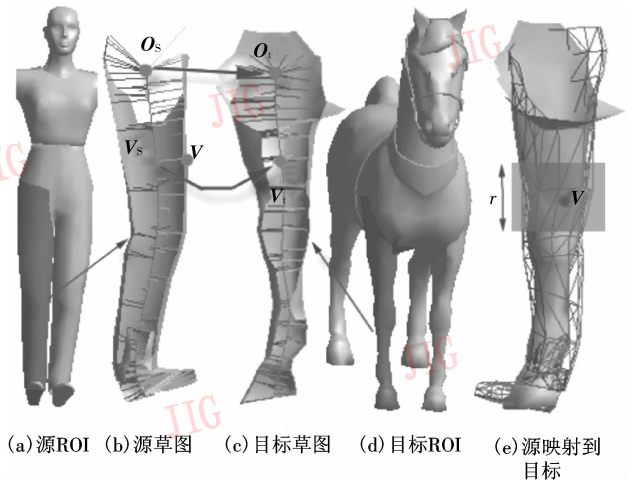


图 3 一对草图用来映射人体模型的腿到相应的马腿  
Fig.3 A pair of sketches used to map a woman leg to a horse leg

下一步要完成的工作就是根据目标 ROI 的尺寸和方向调整源网格相应的 ROI,即首先获取源 ROI 的中心点  $C$ ,然后相对于  $C$  缩放源 ROI 的各顶点。其缺省的缩放因子以如下方式求取:首先计算源 ROI 和目标 ROI 的主坐标轴;然后分别计算出相对于主坐标轴的源 ROI 和目标 ROI 包容盒的体积;最后获得两个体积间的比例(即缺省缩放因子)。用户可以根据需要并以此为基础调整缩放因子。方向的调整是基于源 ROI 和目标 ROI 间的语义(semantics)含义的,例如,马首的方向应该与猫头的方向一致。

接下来需要在源 ROI 的顶点与目标 ROI 的顶点间建立关联。对于在源 ROI 上的每个顶点  $V$ ,需在源草图上计算出离它最近的点  $V_s$ 。注意, $V_s$ 可以是草图的顶点,也可以是草图线段上的某一点。对于目标 ROI 上的顶点也可以用同样的方式求得目标草图上的最近点。为了把源 ROI 上的顶点映射到对应的目标 ROI 上去,本文采用了有些类似文献[1]中提出的方法。对于草图上的任一点(如  $V_s$ ),可用其标准化的弧长  $l$  来表示。也就是说,对草图上给定的点,可首先从起始点沿着草图计算得到此点的长度,然后用这个长度除以草图的总长度来获取标准化的弧长,这个标准化弧长的范围在 0 和 1 之间。每个在草图上的点  $V$  都有一个参数  $l(V)$  用于指示其在草图上的位置。对源草图上的点  $V_s$ ,源

草图上点  $V_s$  的参数是  $l_s$ , 可以在目标草图上找到一点  $V_t$  (图 3(c)), 它的参数也是  $l_s$ 。关联到  $V_s$  的源顶点  $V$  则可以用公式  $\hat{V} = V + (V_t - V_s)$  映射到对应的目标 ROI, 其新坐标为  $\hat{V}$ 。通过这种方法, 就可以把所有源顶点映射到相应的目标 ROI 上。图 3(e) 是映射后的结果。

### 3.3 通过目标草图调节目标顶点参数

目标草图的另一功能是对目标顶点赋参数。每个目标顶点  $V_t$  有两个参数相关联, 即目标顶点权重参数  $W_{V_t}$  和缩放系数  $s$ 。目标顶点权重参数  $W_{V_t}$  用于调整在第 5 节中的式(9)、(10)和式(11)中的权重, 参数  $s$  用于调整第 4 节中的变形后的坐标, 参数的赋值过程如图 2(c) 和图 2(d) 所示。在原始目标草图中(图 2(c)), 首先通过用户交互界面分别对顶点  $V_1$  和  $V_2$  赋参数  $t_1$  和  $t_2$ ; 然后在光滑处理后的目标草图(图 2(d))上找出与  $V_1$  和  $V_2$  对应的点  $\hat{V}_1$  和  $\hat{V}_2$  (寻找的规则是标准化的弧长相等, 即  $\widehat{O_1V_1}$  和  $\widehat{O_1V_2}$  的标准化弧长分别等于  $\widehat{O_1\hat{V}_1}$  和  $\widehat{O_1\hat{V}_2}$  的标准化弧长)。  $\hat{V}_1$  和  $\hat{V}_2$  的参数分别等于  $t_1$  和  $t_2$ 。对任何位于  $\hat{V}_1$  和  $\hat{V}_2$  之间的点  $\hat{V}$ , 可通过对  $t_1$  和  $t_2$  进行基于弧长的线性插值获取其参数  $\hat{t}$ 。所有与  $\hat{V}$  关联的顶点的参数等于  $\hat{t}$ 。各目标顶点  $V_t$  的权重参数  $W_{V_t}$  和缩放系数  $s$  的缺省值都设为 1.0。

## 4 非闭合三角网格的差分均值坐标

### 4.1 基于闭合三角网格的均值坐标

基于样例的方法的一个基本问题是如何按照源网格的变形来驱动目标网格。网格参数化和自由变形方法都需要任一点  $V$  被表示为围绕(enclosing)网格或图形顶点的仿射组合, 即

$$V = \sum_j w_j P_j / \sum_j w_j \quad (1)$$

其中,  $\{P_j\}$  是围绕网格的顶点集合,  $\{w_j\}$  是联系  $P_j$  和  $V$  的权重集。如果令  $\{P_j\}$  是闭合源网格的顶点集合,  $V_{target}$  是目标网格的顶点, 那么目标网格就会按照源动画的运动方式变形。现在关键问题是如何确定权重集  $\{w_j\}$ 。

文献[20]和文献[16]把均值坐标从 2 维多变形网格推广到 3 维闭合网格。对于一个顶点为  $\{\hat{V}, P_0, P_1\}$  的三角形, 其在以  $V$  为中心的单位球  $S_V$  上

的投影为球面三角形  $T_p$  (参见图 4(c)), 那么在一个闭合三角网格中权重  $w_j$  有如下形式:

$$w_j = \frac{n_j \cdot m}{n_j \cdot (\hat{V} - V)} \quad j=1,2,3 \quad (2)$$

其中,  $n_j$  是由点  $V$  和球面三角形  $T_p$  的第  $j$  条边确定的平面的向内单位法线向量,  $m$  是平均向量(mean vector, 见图 4(c)), 它可以由下面公式获得:

$$m = \sum_{j=1}^3 L_j \cdot n_j / 2$$

$\{L_j\}$  是球面三角形  $T_p$  的边长(见图 4(c))。

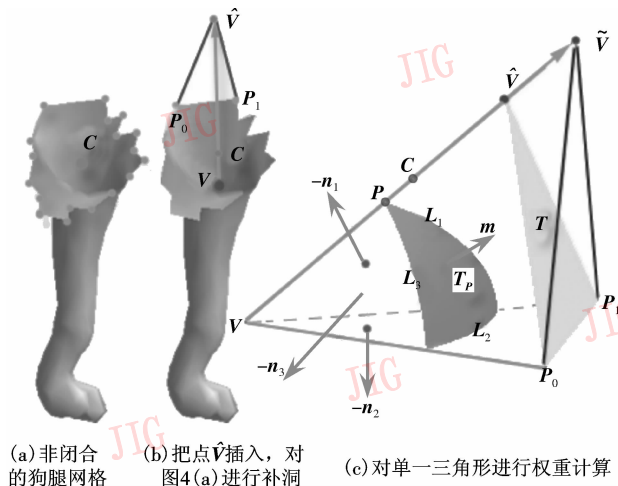


图 4 非闭合三角网格均值坐标

Fig. 4 Mean value coordinates for unclosed triangle mesh

注意式(2)只表示与  $\hat{V}$  相连的许多三角形中的一个。在本文算法中, 为了计算  $V$  相对于  $\hat{V}$  的权重, 需要遍历所有与顶点  $\hat{V}$  相连的三角形, 并对式(2)求和, 即

$$w_{\hat{V}} = \sum_{T_p \in T_p^s} n_{\hat{V}, T_p} \cdot m_{T_p} / (n_{\hat{V}, T_p} \cdot (\hat{V} - V)) \quad (3)$$

其中,  $T_p^s$  表示投影的球面三角形集合——投影的源是与  $\hat{V}$  相连的所有三角形;  $n_{\hat{V}, T_p}$  是对应于  $\hat{V}$  和  $T_p$  的向内单位法线向量(在图 4(c)中是  $n_2$ )。如图 4(c)所示, 如果沿着  $V\hat{V}$  方向把  $\hat{V}$  移到  $\tilde{V}$  的位置, 新三角形  $\Delta\tilde{V}P_0P_1$  在  $S_V$  上的投影仍旧是  $T_p$ , 式(3)的所有系数都维持不变。注意, 在式(3)中, 如果点  $\tilde{V}$  离点  $V$  足够远, 则在实际运算中可以忽略与  $\tilde{V}$  直接相关的权重。

### 4.2 推广——非闭合三角网格的差分均值坐标

式(2)和式(3)是针对闭合网格的。在本系统中, 由于要使用部分源网格来控制部分目标网格, 因

此必须把均值坐标从适用于闭合网格推广到非闭合网格。具体方法是通过插入额外的一点,使网格闭合,在此基础上再计算均值坐标权重。在图4(b)中,对目标网格上的任一顶点 $V$ ,可通过插入 $\hat{V}$ 来填补局部源网格的缺口。点 $\hat{V}$ 放置在点 $V$ 和缺口边界的重心坐标点 $C$ (见图4(a))的连线上,若 $\hat{V}$ 和 $V$ 足够远,则可以认为是 $+\infty$ (图4(b))。通过这种方法,在计算权重时,点 $\hat{V}$ 对点 $V$ 的直接影响力可以忽略不计,然而, $\hat{V}$ 仍然可通过缺口边界点在计算点 $V$ 的权重时起作用。例如,由于三角形 $T$ (图4(c))包含顶点 $\hat{V}$ ,点 $P_0$ 的权重还是受顶点 $\hat{V}$ 的影响。当顶点 $\hat{V}$ 在直线 $\overrightarrow{VC}$ 上时, $\hat{V}$ 和 $C$ 在单位球 $S_V$ 上的投影都是点 $P$ (图4(c))。因此,在计算缺口边界点(如 $P_0$ 和 $P_1$ )的权重和构造球面三角形 $T_p$ 时,就可以用 $C$ 来替换 $\hat{V}$ 。这样,在计算点 $V$ 的权重时, $\hat{V}$ 的影响就可以彻底消除。

本文感兴趣的是因样例网格的变形而引起的目标网格顶点坐标的改变量。可以把式(1)改写为

$$\Delta V = \sum_{j=1}^n w_j \Delta P_j / \sum_{j=1}^n w_j \quad (4)$$

其中, $V$ 为目标网格上的顶点(图3(e)), $\Delta V$ 是顶点 $V$ 的变化量; $P_j$ 为源网格上的顶点; $\Delta P_j$ 是 $P_j$ 相对于第1个关键帧的变化量; $n$ 是源网格ROI的顶点数。在具体实现时,为了加快处理速度和真实地复制源网格的局部动画风格,没有使用整个由草图指定的ROI。相反,如图3(e)所示,仅使用围绕顶点 $V$ 的很小一块区域(在图3(e)中矩形部分)内的源网格顶点。这个区域由一个因子 $r$ 来指定它的宽度范围。在具体计算中,这个宽度范围等于 $r$ 乘以整个目标草图的弧长。在本文的原型系统中, $r = 0.18$ 。从式(4)可以获得顶点的新位置:

$$V_{new} = \Delta V + V_0$$

其中, $V_0$ 为变形的目标网格顶点的原始坐标。

计算出 $V$ 变形后的新坐标后,还需要根据在第4节中描述的缩放系数 $s$ 和源草图与目标草图的弧长比 $R_{length}$ 来调整这个新坐标。 $C$ 为目标ROI的重心坐标,那么调整后的顶点 $V$ 的新坐标可以写为

$$V_{new} \leftarrow C + R_{length} s (V_{new} - C) \quad (5.1)$$

对一个目标ROI, $\{V_j\}$ 和 $\{\hat{V}_j\}$ 分别是变形前后的顶点集合。 $C$ 是顶点集合 $\{V_j\}$ 的重心坐标。 $\bar{d}_1$ 是 $C$

到 $\{V_j\}$ 的平均距离。同理可以计算出对应于 $\{\hat{V}_j\}$ 的平均距离 $\bar{d}_0$ 。为了防止在变形过程皱缩,还需要对变形后的坐标进行如下调整:

$$V_{new} \leftarrow C + \bar{d}_1 / \bar{d}_0 (V_{new} - C) \quad (5.2)$$

到这一阶段,在目标网格上的每个顶点 $V$ 可以分为以下3类:(1)由于点 $V$ 没有被任何ROI覆盖,其不涉及任何变形,从而顶点 $V$ 保持不变;(2)由于点 $V$ 被一个唯一的ROI覆盖,因此变形是唯一确定的;(3) $V$ 被多于一个ROI覆盖,即点 $V$ 在不同ROI上会得到几个不同的变形结果,这样在变形后的目标网格上会产生歧义的区域。在这种情况下,可求取各个新坐标值的加权平均坐标 $V_w$ 作为点 $V$ 变形后的新坐标。计算 $V_w$ 的权重等于相应草图的权重以及第3节中赋予点 $V$ 的权重的乘积。

图5显示了由图3(e)中人体模型的两条腿控制的两条马腿的动画。在下一节将对目标网格要进行缝合和光滑处理,以期获得理想的变形结果。

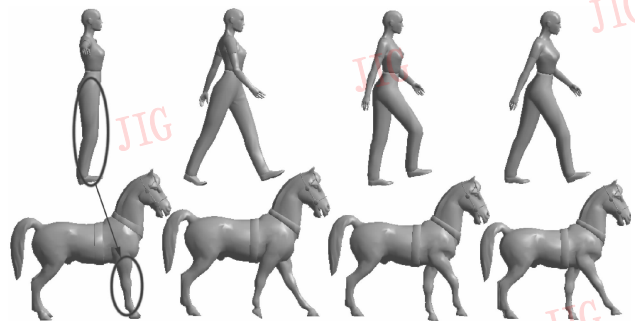


图5 用人体模型(第1行)两条腿的动画驱动马的两条腿运动。

具体的变形是通过式(4)计算出来的

Fig. 5 Animate the two front legs of a horse according to the animation of the two legs of a woman. The deformation is computed based on Equation (4)

## 5 对齐和光滑处理

### 5.1 变形后目标网格各部分的对齐

当多个源网格控制一个目标网格的各个不同部分时,则目标网格上相邻的两部分在变形后会产生不连续,甚至相互分离(图6(a))。本文使用了一个简单的策略来大致对齐目标网格的相邻部分,即对用户输入的目标草图,从最重要的到不重要的进行排序。具体实现时,则按照与这些草图对应的ROI区域的面积来排序(从面积最大的到最小的)。排序完成后,先找出排在最后(如图6中的马尾)的目标ROI的边界点集 $\{V_j\}$ (图6(c)中边上的点),然

后计算出这些点的重心位置(图 6(c)中点  $C$ );接着对每个边界顶点  $V_j$  找出变形前的位置,另外,还要找出此点在其他未经对齐处理的目标 ROI(见图 6 中马躯干)上的变形后的坐标,最后对这些坐标求加权平均值,记为  $P_j$ (如果  $V_j$  仅仅被最后一个 ROI 本身覆盖,那么  $P_j = V_j$ )。用于计算的权重等于相应草图的权重和在第 3 节中赋予  $V_j$  的权重 ( $W_{V_j}$ ) 的乘积。 $\hat{C}$  为  $\{P_j\}$  的重心坐标,再把向量  $\hat{C} - C$  加到这个排在最后的 ROI(图 6 中马尾)的所有顶点上,然后在排好序的目标 ROI 队列中,依次向前对每个 ROI 进行这样处理,直到遇到对列中的第 1 个成员。通过这种方法,重叠或分离的变形后的目标 ROI 就被大致缝合到一起(如图 6(b)所示)。

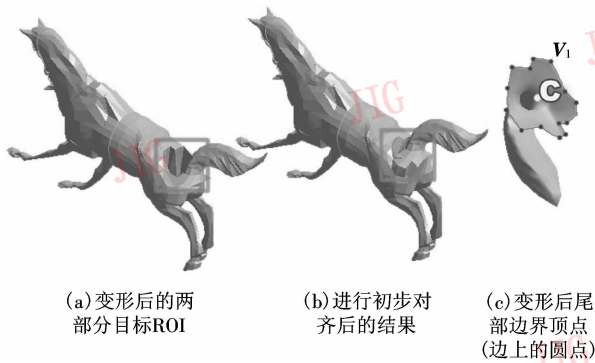


图 6 使用图 2 所示的蛇控制马尾、猫控制马躯干和四肢变形的最初结果

Fig. 6 Deforming the horse using a snake to control the tail of the horse and a cat to control the rest of the horse, as shown in Fig. 2

### 5.2 光滑处理目标网格

从图 6(b) 可以看到,经过初步对齐后的马远未达到理想的光滑程度。下面再介绍对初步对齐后的结果(图 6(b))进行光滑处理的技术,这个技术基于以下几个考虑:①坐标变换的一致性(uniformity of the transform);②坐标变换的刚性(rigidity of the transform);③最小化最终坐标的偏离;④平滑性(flatness)。

为了准确地衡量网格的变形,本文采用仿射变换矩阵来描述变形。因为一个  $3 \times 4$  的仿射变换矩阵的自由度是 12,所以可以使用 4 个不共面的顶点的坐标来表达这个矩阵<sup>[15]</sup>。另外,一个  $3 \times 4$  的仿射变换矩阵又可以表示为一个  $3 \times 3$  的线性矩阵  $M$  和一个位移矢量  $d$ 。本文采用类似于文献[15]但更有效的方法构建仿射变换矩阵。该方法不同于文献

[15] 之处在于不采用变形梯度(deformation gradient),而是采用网格上 4 个在同一个邻域内的不共面的顶点来产生仿射变换矩阵。

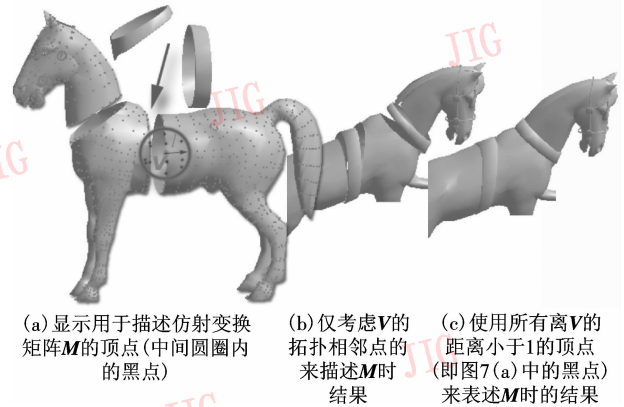


图 7 仿射变换矩阵  $M$  的计算和不同方法计算  $M$  的效果对比

Fig. 7 Calculating the affine matrix  $M$  and Showing the results with different methods of calculating  $M$

具体是先给定一个顶点  $V$  和一个用户指定的距离阈值  $l$ ,然后找出所有与顶点  $V$  的距离小于  $l$  的顶点(图 7(a) 中圆圈内的黑色顶点),并把这些顶点存放到一个队列  $a_v$  中。对于包含拓扑信息的网格,则还需要把不包含于  $a_v$  中的与顶点  $V$  直接相邻(first ring)的顶点加入到  $a_v$  中。对数组中的每个元素  $P$ ,可使用  $V, P$  以及  $P$  在数组里的上一个元素和下一个元素(基于循环原则)来构建仿射变换矩阵。 $A_i = (x_i, y_i, z_i)$  和  $\hat{A}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$  ( $i = 1, 2, 3, 4$ ) 分别表示变形前、后的目标网格的顶点,它们的数学关系可以写成

$$MA_i + d = \hat{A}_i \quad i = 1, 2, 3, 4$$

当  $\{A_i\}$  非共面时,则可以消除上式中的  $d$ ,并可将上式改写成如下矩阵形式:

$$M = \Delta Q \times \Delta Q^{-1} \quad (6)$$

其中,

$$\Delta \hat{Q} = \begin{bmatrix} \hat{x}_2 & \hat{x}_3 & \hat{x}_4 \\ \hat{y}_2 & \hat{y}_3 & \hat{y}_4 \\ \hat{z}_2 & \hat{z}_3 & \hat{z}_4 \end{bmatrix} - \begin{bmatrix} \hat{x}_1 & \hat{x}_1 & \hat{x}_1 \\ \hat{y}_1 & \hat{y}_1 & \hat{y}_1 \\ \hat{z}_1 & \hat{z}_1 & \hat{z}_1 \end{bmatrix}$$

$$\Delta Q = \begin{bmatrix} x_2 & x_3 & x_4 \\ y_2 & y_3 & y_4 \\ z_2 & z_3 & z_4 \end{bmatrix} - \begin{bmatrix} x_1 & x_1 & x_1 \\ y_1 & y_1 & y_1 \\ z_1 & z_1 & z_1 \end{bmatrix}$$

因为  $\Delta Q$  只与已知的变形前的目标网格顶点有关,所以  $M$  是变形后顶点坐标的函数。由于  $M$  与顶

点  $V$  和  $P$  相关,因此可以把  $M$  改写为  $M(V, ind)$ , 其中  $ind$  是  $P$  在数组  $a_v$  中的索引值。图 7(b) 显示了仅考虑顶点  $V$  的拓扑相邻点来描述  $M$  时的结果。如果使用所有离  $V$  的距离小于  $l$  的顶点(即图 7(a) 中的黑点)来表示  $M$  时,可以得到更好的结果(图 7(c))。

基于式(6),本文定义了一系列优化变形结果的约束函数。在整个优化过程中总共有 4 个约束函数,这些函数都是用仿射变换矩阵来表示的。第 1 个约束函数类似于文献[15]中的光滑目标函数和文献[21]中的模板适应进程。记变形光滑度为  $E_s$ , 它表示目标网格中在同一邻域内的四面体的变换应该尽量相等:

$$E_s(\hat{V}_1, \dots, \hat{V}_n) = \sum_{j=1}^n \sum_{k=1}^{|a_v|} \|w_j^s(M(\hat{V}_j, k) - M(\hat{V}_j, k+1))\|_F^2 \quad (7)$$

其中,  $\hat{V}$  是相对于目标网格顶点  $V$  的新位置,它是一组需要求解的未知数。 $|a_v|$  是数组  $a_v$  的长度。计算索引  $(k+1)$  时需应用循环规则,即,  $(k+1) \text{ mode } |a_v|; n$  是目标网格中顶点的数目;  $w_j^s$  是平滑作用权重。

根据式(6),先给定 4 个不共面的顶点  $\{V\}$ , 然后就可以使用上一节中求得的对应的顶点坐标  $\{V_w\}$  计算出一个过渡性的仿射变换矩阵  $M_w$ 。按照文献[19]的矩阵极分解(polar decomposition)算法,  $M_w$  可以分解为一个直交(orthogonal)因子  $Q_w$  和一个对称正定(symmetric positive definite)因子  $S$ , 即  $M_w = Q_w S$ 。当  $M_w$  的行列式值为正时,  $Q_w$  是一个纯旋转矩阵; 否则它还包含一个反射。如果  $M_w$  等于  $Q_w$ , 则网格没有局部扭曲,完全是一个刚体变换。第 2 个目标函数  $E_r$  是用来在变形过程中尽量保持整个网格的刚性( $w_j^r$  是刚性保持权重):

$$E_r(\hat{V}_1, \dots, \hat{V}_n) = \sum_{j=1}^n \sum_{k=1}^{|a_v|} \|w_j^r(M(\hat{V}_j, k) - Q_w(V_{j,w}, k))\|_F^2 \quad (8)$$

目标函数式(7)和式(8)基于以下一个假设:4 个用于计算仿射变换矩阵的顶点都是非共面的,但和这个假设不总是成立的。为了处理共面的情况,增加第 3 个目标函数  $E_l$  用来保证在同一个邻域内各顶点间的距离在变形前和变形后尽量保持不变。

$$E_l(\hat{V}_1, \dots, \hat{V}_n) = \sum_{j=1}^n \sum_{k=1}^{|a_v|} \|w_j^l((\hat{V}_j - \hat{V}_{a[k]}) - (V_j - V_{a[k]}))\|_F^2 \quad (9)$$

其中,  $a[k]$  表示数组  $a_v$  中的第  $k$  个元素,  $w_j^l$  是距离保持权重。

最后一个目标函数,也是最重要的一个控制变形函数  $E_d$ 。这个函数表示最终的变形结果要尽可能地接近上一节中的初步变形结果:

$$E_d(\hat{V}_1, \dots, \hat{V}_n) = \sum_{j=1}^n \|w_j^d(\hat{V}_j - V_j^{new})\|^2 \quad (10)$$

其中,  $V_j^{new}$  是上一节中的初步变形结果。 $w_j^d$  是权重。

为了方便地指定上述 4 个目标函数的权重,根据各个目标草图确定的 ROI 的重要性,本文对各个目标草图设置一个全局权重  $w_{sketch}$ ; 4 个权重分量  $W_{sketch}^S$ 、 $W_{sketch}^R$ 、 $W_{sketch}^L$  和  $W_{sketch}^d$  分别对应于 4 个目标函数(上角 S 代表 smooth, R 代表 rigid, L 代表距离保持阈值, d 代表 deformation)。目标函数的权重等于草图权重分量  $W_{sketch}^z$  ( $z = S, R, L, d$ ) 和顶点权重(即第 3 节中的权重参数,计算  $E_d$  和  $E_l$  时,顶点权重为 1.0)的乘积。当顶点被多个目标 ROI 覆盖时,  $W_{sketch}^z$  ( $z = S, R, L, d$ ) 是各个草图的相应的权重的加权平均值,权重是各草图的  $w_{sketch}^o$ 。

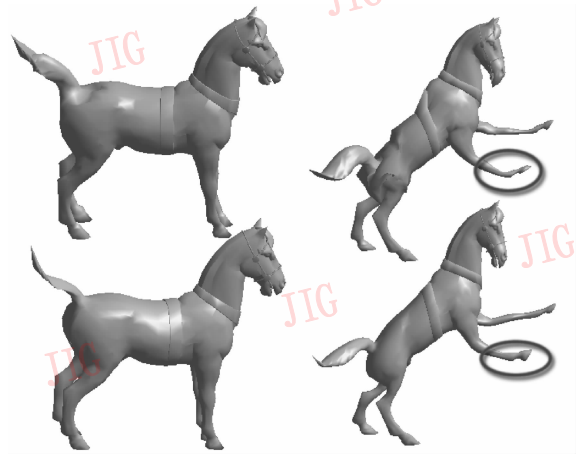


图 8 光滑处理前(第 1 行)和后(第 2 行)的变形结果  
Fig. 8 Deformation results before (the upper row) and after (the lower row) smoothing

由于全局目标函数是上述 4 个目标函数的加权总和,因此,这个全局最小值问题就变成使得下边的公式取得最小值,从而计算出目标网格各顶点对应于  $\{V_1, V_2, \dots, V_n\}$  的新坐标值  $\{\hat{V}_1, \hat{V}_2, \dots, \hat{V}_n\}$  的问题:

$$E(\hat{V}_1, \dots, \hat{V}_n) = E_s + E_r + E_l + E_d = \min \quad (11)$$

通过对全局目标函数(式(11))进行仔细全面地分析,可以发现以下一些特点:首先,由于各个变量的坐标分量是独立的,因此可以直接使用向

量来表达式(11),也就是说,式(11)是个向量值(vector-valued)公式;其次,这个目标函数是变量的二次多项式,这样最小值问题就可转化为求解一个线性方程组问题。这个线性方程组可以通过式(11)对向量值变量求偏导数来获得。这是个标准的数学过程。附录详细分析了这个推导过程。为了高效地求解线性方程组,本文使用了一个稀疏LU软件包<sup>[22]</sup>。图8显示了光滑处理前、后的变形结果。

在原型系统的实现中,缺省的草图权重分量分别是: $W_{sketch}^S = 0.0005$ ,  $W_{sketch}^R = 0.0005$ ,  $W_{sketch}^L = 0.001$  和  $W_{sketch}^d = 500$ ,在此基础上针对具体不同的草图做进一步细调。

### 6 关键帧插值

在两个关键帧间插值是动画创建的一个重要内容。令  $P_1$  和  $P_2$  是两个由同一源网格顶点  $P$  变形而来的在两个相邻源关键帧上的顶点;  $Q_1$  和  $Q_2$  是两个由同一个目标网格顶点  $Q$  变形而来的在两个相邻目标关键帧的顶点。对任意给定的一个时刻  $t(0 \leq t \leq 1)$ , 分别进行前向和后向插值(见图9),即

$$V_1(t) = Q_1 + t \sum_j w_j (P_2 - P_1) / \sum_j w_j$$

$$V_2(t) = Q_2 + (1 - t) \sum_j w_j (P_1 - P_2) / \sum_j w_j \quad (12)$$

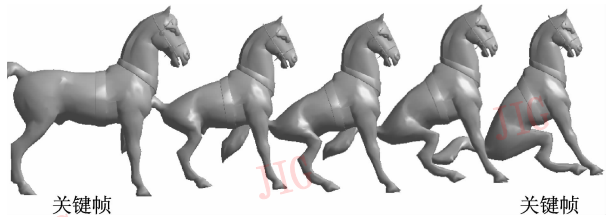


图9 在关键帧间插值

Fig. 9 Interpolating between key frames Between Frame

插值时,首先取  $V_1(t)$  和  $V_2(t)$  的线性组合作为新的坐标点,即  $V(t) = t \cdot V_1(t) + (1 - t) \cdot V_2(t)$ ;然后使用式(5.1)和式(5.2)调整  $V(t)$ ;最后使用第5节的方法对结果进行对齐和光滑处理。图9展示了在两个关键帧间插值的结果:在关键帧1和关键帧2之间,3个帧被插值产生,分别对应于时间点0.25、0.50和0.75。

### 7 从2维动画生成3维动画

现实生活中有大量的视频和Flash,而把这些2维动画传输到3维角色是非常有意义的。如今通过2维动画来产生3维动画的方法已在电影和游戏中得到应用。这种应用工作的第1步就是从2维动画中产生控制网格。本文使用视频跟踪技术<sup>[23,24]</sup>来获取各视频序列中ROIs的轮廓线。图10是一个从视频序列的轮廓线构造3维网格的例子。图中第1列的两张图是2维动画中的两个关键帧,第2列显示了相应嘴唇的轮廓线。图中标出各个轮廓线的起始顶点(图10(a)和图10(b)中的顶点  $V_1$  和  $V_2$ ),这些点先用来对齐轮廓线;然后将每条轮廓线按长度参数化到0(较大标记点)和1(一圈后回到较大标记点)的范围内;接着,使用第1条轮廓线的顶点来对齐其余轮廓线的顶点。对于在第1条轮廓线上的每一点  $V$ ,先算出其对应的参数  $t$ (具体方法与处理草图线时一样),然后在其余轮廓线上找出参数为  $t$  的所有点,并把这些点作为轮廓线的新顶点。通过这种方法,不仅其余轮廓线与第1条轮廓线有相同的顶点数,而且还把不同轮廓线上的顶点的对应关系(correspondence)建立了起来。为了构造一个网格,需增加两个虚拟点(图10(c)中的  $\hat{V}$  和  $\tilde{V}$ )。这两个虚拟点是分别沿着轮廓线所在平面的法线方向通过抬升和降低轮廓线的中心点一定高度而产生的,但具体移动的高度对结果没有影响。

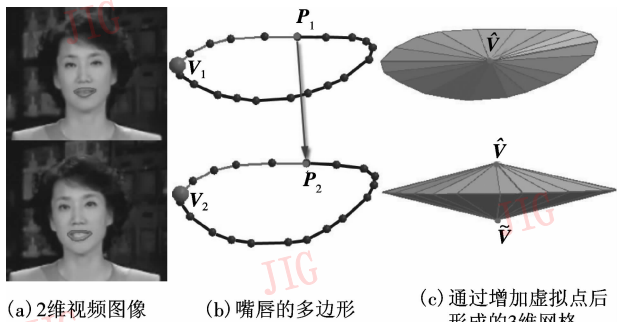


图10 从视频序列的轮廓线构造3维网格

Fig. 10 Constructing 3D meshes from contour outlines of a video sequence

在图10(a)中的曲线是播音员嘴唇的轮廓线,图10(b)和图10(c)显示了3维网格的构造过程。控制网格产生后,就可以通过前边介绍过的方法来

生成图 11 中头像中嘴唇的 3 维动画。在这个例子中使用了以下两对草图线(图 11 中第 1 列中嘴唇上曲线):一条在上嘴唇,另一条在下嘴唇。为了使头像的整个脸动起来,还需要把整个头的上部(图 11 左下角图片的上边部分)映射到第 1 条目标草图线,而将整个头的下部(图 11 左下角图片的下边部分)映射到另一条目标草图线。图 11 是一幅从 2 维视频动画产生的 3 维动画。

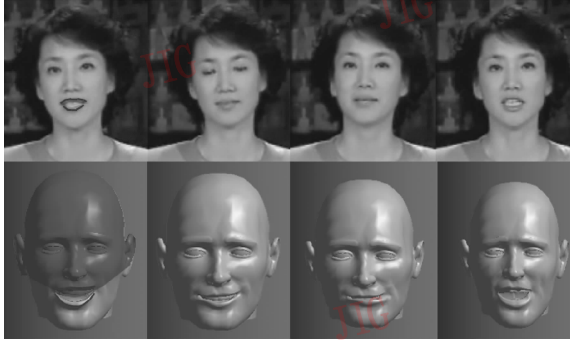


图 11 利用 2 维视频生成 3 维头像口型动画

Fig. 11 Animating the mouth of a 3D head by a 2D video clip

## 8 结 论

本文研究了基于草图的 3 维动画传输,这是一个从现有动画中抽取感兴趣的部分来创建动画的范例。本文首先提出了基于草图的交互式建立映射的算法;然后使用均值坐标来变形目标网格;最后开发了对齐和光滑处理目标网格的方法。在原型系统上进行的实验表明,本文所述的是一个容易学习的非常有用的基于样例的 3 维动画创建方法,它可以方便地建立新角色复杂有趣的动画。该方法既可以把一个完整的

源动画传输到一个目标角色中,也可以把多个源动画传输到一个目标角色中。图 12 是一个把一个雪球融化的动画过程传输到一个 Venus 模型上的实例,这是一对一传输的例子。图 12(a)是剖面图,而草图是在网格中间的,图 12(c)第 1 行是一个雪球融化过程的动画模拟,可利用此动画驱动 Venus 产生类似动画。用户所要做的就是源网格和目标网格中分别绘制两条线型草图(sketch,图 12(a)上的黑色折线)。图 12(b)显示的是雪球网格在草图的基础上映射到 Venus 网格上的结果。图 12(c)第 2 行其余的图片显示了 5 帧通过本文算法产生的 Venus 动画。图 13 汇集了本文所有动画的合成结果,即把 4 个源动画传输到一辆马车上,生成一个完整的马车动画的例子。图 13(a)中画圈的源动画控制马车的相应部分,即 2 维视频上的轮子控制马车的轮子;狗控制完整的 1 号马;蛇的尾巴控制 2 号和 4 号马的尾巴;猫的躯干控制 2 号马的躯干;人体模型的两条腿控制 3 号和 4 号马的腿;人体模型的双臂控制两个乘客的双臂;人体模型的躯干控制两个乘客的躯干。在图 13(b)中,使用现有动画产生了 3 个马车动画的关键帧。图 14 是例子(图 13)用到的所有草图,所有这些草图都是绘制在网格的中间。草图的顶点由小黑点指出。因为有些源草图控制多条目标草图,所以源草图的总数要小于目标草图总数。所有显示在图 14 中的草图都是未经光滑和插值处理的原始草图。图 15 展示了本文算法在传输细微动画时的处理结果,即把一个水波的模拟动画映射到一个小天使的脸上。其草图是图 15(a)上画在网格表面上的黑色曲线,图 15(b)显示了源网格映射到目标网格上的结果,图 15(c)显示了由原型系统产生的 4 个关键帧。

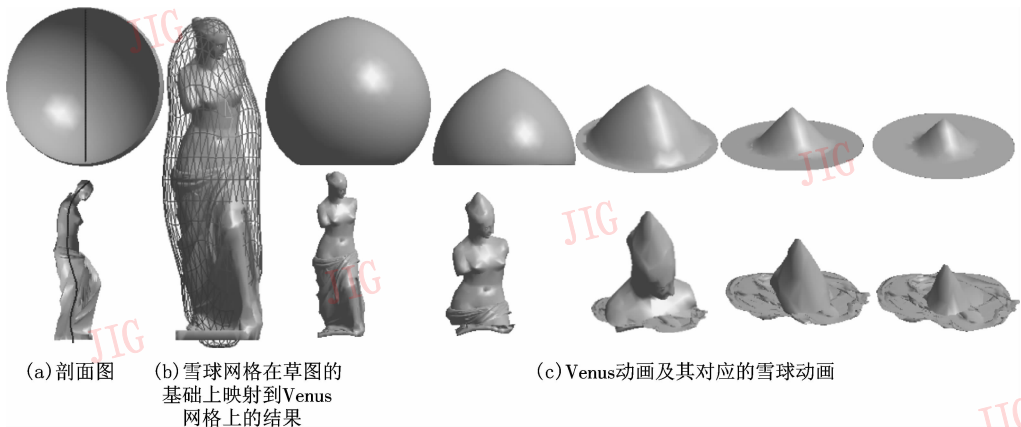
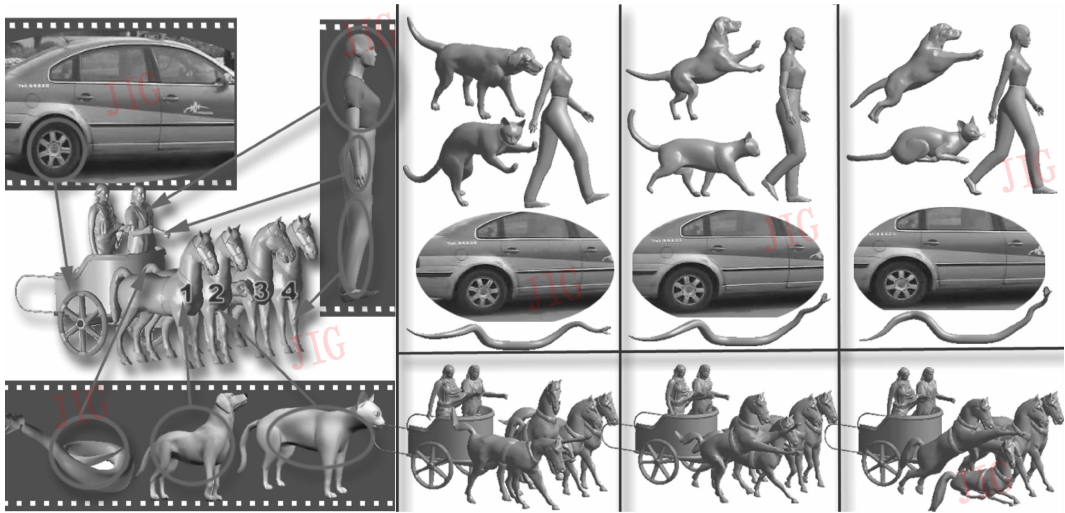


图 12 把雪球的融化过程传输到 Venus 上

Fig. 12 Transferring animation of a melting ball to the Venus

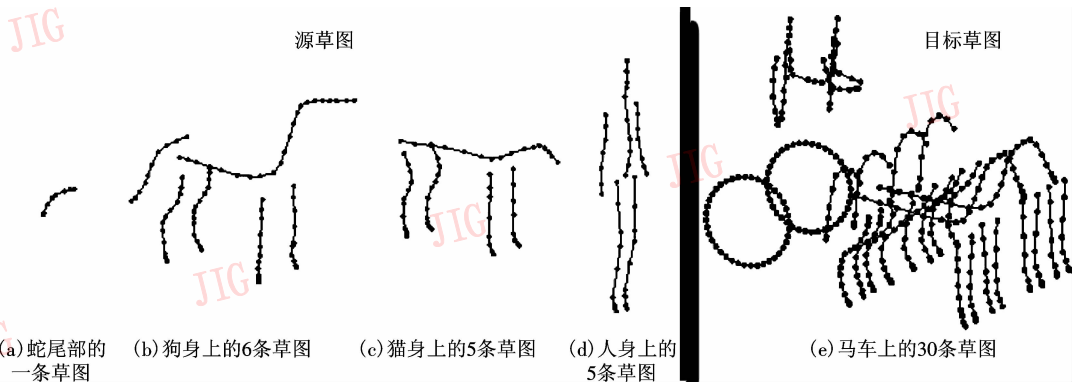


(a) 多个源动画驱动马车

(b) 3个源动画关键帧和对应的马车动画

图 13 使用狗、猫、蛇和人体模型的 3 维动画和小车的 3 维视频驱动马车

Fig. 13 Using 3D animations of dog, cat, snake, woman and 3D video of car to animate the cart



(a) 蛇尾部的一条草图

(b) 狗身上的6条草图

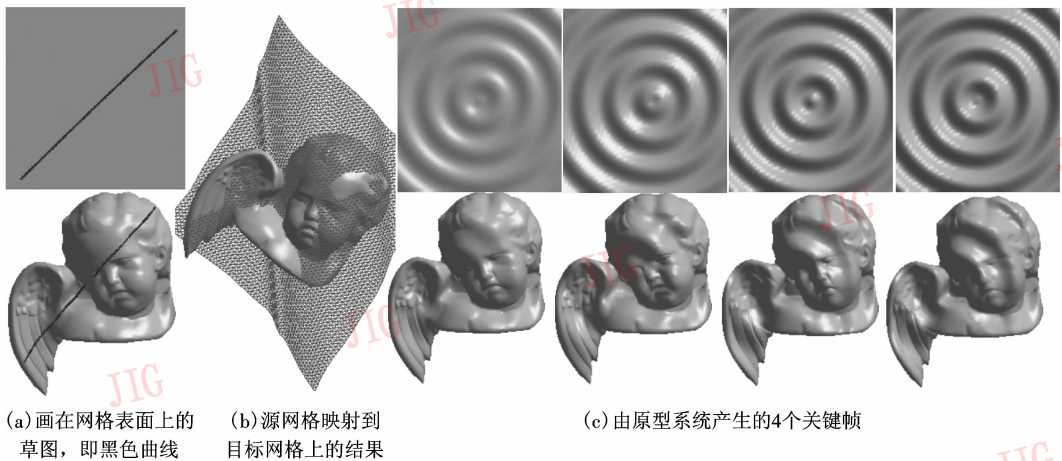
(c) 猫身上的5条草图

(d) 人身上的5条草图

(e) 马车上的30条草图

图 14 所有用于产生图 13 的 3 维动画的草图

Fig. 14 Sketches used to produce the 3D animation in Fig. 13



(a) 画在网格表面上的草图，即黑色曲线

(b) 源网格映射到目标网格上的结果

(c) 由原型系统产生的4个关键帧

图 15 用一个水波的动画控制 Angel 的脸部

Fig. 15 Using an animation of ripple to control the face of Angel

表1列出了本文所用到网格的一些几何属性。表2列出本文例子的计算时间。测试的计算机配置为:Windows XP 操作系统、1024M 内存、2.4GHz 的 CPU。

表1 本文各个例子中网格的几何属性

Tab.1 Geometric properties of meshes used in the examples

| 网格    | 顶点数    | 三角面片数  |
|-------|--------|--------|
| 马车    | 24,034 | 45,911 |
| 猫     | 9,634  | 19,098 |
| 狗     | 6,650  | 13,176 |
| 人体模型  | 2,208  | 4,352  |
| 蛇     | 11,137 | 22,250 |
| 雪球    | 642    | 1,280  |
| Venus | 5,693  | 11,563 |
| 马     | 1,232  | 3,564  |

表2 各个例子用到的草图数和计算时每个关键步骤的消耗时间

Tab.2 The number of sketch pairs used in each example, and time used in each steps to produce a key frame

| 例子  | 草图数 | 草图计算时间(s) | 网格映射时间(s) | 网格变形时间(s) | 光滑时间(s) |
|-----|-----|-----------|-----------|-----------|---------|
| 图11 | 2   | 2.07      | <0.01     | 0.56      | 0.51    |
| 图12 | 1   | 1.73      | 0.28      | 7.24      | 3.53    |
| 图13 | 30  | 22.99     | 18.95     | 90.64     | 20.41   |
| 图15 | 1   | 2.77      | 0.36      | 63.01     | 49.00   |

本文将 Summer 等人的算法<sup>[15]</sup>用来进行比较。通过比较可见,在以下几个方面本文的算法有更好的表现,即

(1) 高效性 本算法不需要像文献[15]一样,通过多次迭代来计算出源网格和目标网格的对应关系;另外,本算法没有使用虚拟顶点(在文献[15]中,对每个三角面片要构造一个虚拟顶点),针对相同的网格,本算法的方程组规模要小很多。且局部变形可重用。本算法既可以从源网格上复制部分动画到目标网格,也可以从多个源网格向一个目标网格复制动画。

(2) 易用性 在用户操作上,本算法只需要比较少的输入。例如图12,用户只需输入一对草图即可。

另外,本算法还允许目标对象是非结构化的数

据,而文献[15]还要求源对象和目标都必须是三角网格。

尽管本文的研究在新的道路上迈出了一小步,可是几个不足仍需在未来工作中解决。最明显的不足就是,当源网格和目标网格非常不相似时,其结果不是非常自然。这需要有更加智能的方法以便能够自动调整均值坐标的权重。另外,当多个源运动传输到目标对象后,运动之间的协调性问题也是非常重要的,其处理不好也会产生不自然的动画效果,这个问题也需要进一步研究。

## 参考文献 (References)

- 1 Kho Y, Garland M. Sketching mesh deformations [A]. In: Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games[C]. Washington, District of Columbia, USA, 2005: 147~154.
- 2 Lipman Y, Sorkine O, Levin D, et al. Linear rotation-invariant coordinates for meshes [J]. ACM Transactions on Graphics, 2005, 24(3):479~487.
- 3 Nealen A, Sorkine O, Alexa M, et al. A sketch-based interface for detail-preserving mesh editing [J]. ACM Transactions on Graphics, 2005, 24(3):1142~1147.
- 4 Thorne M, Burke D, van de Panne M. Motion doodles: an interface for sketching character motion [J]. ACM Transactions on Graphics, 2004, 23(3):424~431.
- 5 Yoshizawa S, Belyaev A G, Seidel H. Free-form skeleton-driven mesh deformations [A]. In: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications [C], Seattle, Washington, USA, 2003:247~253.
- 6 Yu Y, Zhou K, Xu D, et al. Mesh editing with poisson-based gradient field manipulation [J]. ACM Transactions on Graphics, 2004, 23(3):644~651.
- 7 Cohen M. Everything by example [EB/OL]. <http://research.microsoft.com/~cohen/EverythingByExample.pdf>.
- 8 Funkhouser T, Kazhdan M, Shilane P, et al. Modeling by example [J]. ACM Transactions on Graphics, 2004, 23(3):652~663.
- 9 Summer R W, Zwicker M, Gotsman C, et al. Mesh-based inverse kinematics [J]. ACM Transactions on Graphics, 2005, 24(3):488~495.
- 10 Arikian O, Forsyth D A. Interactive motion generation from examples [A]. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques [C], San Antonio, Texas, USA, 2002:483~490.
- 11 Bregler C, Loebe L, Chuang E, et al. Turning to the masters: motion capturing cartoons [J]. ACM Transactions on Graphics, 2002, 21(3):399~407.
- 12 Gleicher M. Retargetting motion to new characters [A]. In: Proceedings of the 25th Annual Conference on Computer Graphics and interactive Techniques [C], New York, USA: ACM Press, 1998:33~42.

- 13 Hsu E, Pulli K, Popović J. Style translation for human motion[J]. ACM Transactions on Graphics, 2005, 24(3):1082~1089.
- 14 Monzani J S, Baerlocher P, Boulic R, et al. Using an intermediate skeleton and inverse kinematics for motion retargeting[J]. Computer Graphics Forum, 2000, 9(3):115~223.
- 15 Summer R W, Popović J. Deformation transfer for triangle meshes [J]. ACM Transactions on Graphics, 2004, 23(3):399~405.
- 16 Ju T, Schaefer S, Warren J. Mean value coordinates for closed triangular meshes [J]. ACM Transactions on Graphics, 2005, 24(3):561~566.
- 17 Alexa M, Cohen-Or D, Levin D. As-rigid-as-possible shape interpolation [A]. In: Proceedings of the ACM 27th Annual International Conference on Computer Graphics and interactive Techniques [C], New York, USA: Addison-Wesley Publishing Co, 2000:157~164.
- 18 Igarashi T, Moscovich T, Hughes J F. As-rigid-as-possible shape manipulation[J]. ACM Transactions on Graphics, 2005, 24(3):1134~1141.
- 19 Shoemake K, Duff T. Matrix animation and polar decomposition [A]. In: Proceedings of the Conference on Graphics Interface'92 [C]. Vancouver, British Columbia, Canada, 1992:258~264.
- 20 Floater M S, Hormann K. Surface parameterization: a tutorial and survey [A]. In: Dodgson N A, Floater M S, Sabin M A eds. Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization [C], Heidelberg German, Berlin: Springer, 2005:157~186.
- 21 Allen B, Curless B, Popović Z. The space of human body shapes: Reconstruction and parameterization from range scans [J]. ACM Transactions on Graphics, 2003, 22(3):587~594.
- 22 Davis T A. Umfpack Version 4.1 User Guide [R]. Technique report, TR-03-008, University of Florida, Orlando, FL, USA, 2003.
- 23 Agarwala A, Hertzmann A, Salesin D H, et al. Keyframe-based tracking for rotoscoping and animation [J]. ACM Transactions on Graphics, 2004, 23(3):584~591.
- 24 Wang J, Bhat P, Colburn R A, et al. Interactive video cutout[J]. ACM Transactions on Graphics, 2005, 24(3):585~594.

## 附录 A: 求解光滑处理目标网格的最小化问题

在第 5 节中已经讨论过,若要光滑处理变形后的目标网格,则需要求解式(11)中的最小化问题,这个可以通过使全局目标函数  $E$  对所有向量值变量(目标网格顶点变形后的位置)的偏导数为 0 来计算。对式(6)的矩阵  $M(\hat{V}_j, k)$ ,可令矩阵  $\Delta Q^{-1}$  的元素为  $C^{(j,k)}$ ,则可使用向量来展开式(7)和式(8):

$$E_S = \sum_{j=1}^n \sum_{k=1}^{|a_{V_j}|} \sum_{i=1}^3 \|w_j^S [C_{1,i}^{(j,k)} (\hat{V}_{a_{V_j}[k-1]} - \hat{V}_j) +$$

$$(C_{2,i}^{(j,k)} - C_{1,i}^{(j,k+1)}) (\hat{V}_{a_{V_j}[k]} - \hat{V}_j) + (C_{3,i}^{(j,k)} - C_{2,i}^{(j,k+1)}) (\hat{V}_{a_{V_j}[k+1]} - \hat{V}_j) - C_{3,i}^{(j,k+1)} (\hat{V}_{a_{V_j}[k+2]} - \hat{V}_j) \|^2 \quad (A1)$$

$$E_R = \sum_{j=1}^n \sum_{k=1}^{|a_{V_j}|} \sum_{i=1}^3 \|w_j^R [C_{1,i}^{(j,k)} (\hat{V}_{a_{V_j}[k-1]} - \hat{V}_j) + C_{2,i}^{(j,k)} (\hat{V}_{a_{V_j}[k]} - \hat{V}_j) + C_{3,i}^{(j,k)} (\hat{V}_{a_{V_j}[k+1]} - \hat{V}_j) - Q_i^{(j,k)}] \|^2 \quad (A2)$$

其中,  $Q_i^{(j,k)}$  是以向量表示的式(8)中的矩阵  $Q_w(V_j^w, k)$  的第  $i$  列向量。令  $N_S, N_R, N_L, N_d$  分别为式(A1)、(A2)、式(9)和式(10)的项数:

$$N_R = N_S = 3 \times \sum_{j=1}^n |a_{V_j}|, \\ N_d = n, N_L = \sum_{j=1}^n |a_{V_j}|.$$

把  $E_z (z=S, R, L, d)$  (其中  $S$  代表 smooth,  $R$  代表 rigid,  $d$  代表 deformation,  $L$  代表距离保持阈值)的项以线性形式表达:

$$E_z = \sum_{\xi=1}^{N_z} \|A_{\xi}^z \hat{V} + D_{\xi}^z\|_F^2 \quad (A3)$$

其中,  $A_{\xi}^z$  是一个  $N_z \times n$  大小的矩阵,  $D_{\xi}^z$  是一个  $N_z$  维列向量。它们都是稀疏矩阵,其非零元素可以用以下伪代码计算出:对于  $E_S$  和  $E_R$

$$\text{for} \left( \begin{array}{l} i = 1; i \leq 3; i++ \\ j = 1; j \leq n; j++ \\ k = 1; k \leq |a_{V_j}|; k++ \end{array} \right) \\ \{ \xi_{i,j,k} = 3 \sum_{\tau=1}^{j-1} |a_{V_{\tau}}| + 3(k-1) + i; \eta_{j,k} = a_{V_j}[k]; \\ A_{\xi_{i,j,k}}^S (\xi_{i,j,k}, \eta_{j,k-1}) = w_j^S C_{[1],[i]}^{(j,k)}; \\ A_{\xi_{i,j,k}}^R (\xi_{i,j,k}, \eta_{j,k-1}) = w_j^R C_{[1],[i]}^{(j,k)}; \\ A_{\xi_{i,j,k}}^R (\xi_{i,j,k}, \eta_{j,k}) = w_j^R C_{[2],[i]}^{(j,k)}; \\ A_{\xi_{i,j,k}}^R (\xi_{i,j,k}, \eta_{j,k+1}) = w_j^R C_{[3],[i]}^{(j,k)}; \\ A_{\xi_{i,j,k}}^R (\xi_{i,j,k}, j) = -w_j^R (C_{[3],[i]}^{(j,k)} + C_{[2],[i]}^{(j,k)} + C_{[1],[i]}^{(j,k)}); \\ D_{\xi_{i,j,k}}^R (\xi_{i,j,k}) = -w_j^S Q_{[i]}^{(j,k)}; A_{\xi_{i,j,k}}^S (\xi_{i,j,k}, \eta_{j,k+2}) = -w_j^S C_{[3],[i]}^{(j,k+1)}; \\ A_{\xi_{i,j,k}}^S (\xi_{i,j,k}, \eta_{j,k}) = w_j^S (C_{[2],[i]}^{(j,k)} - C_{[1],[i]}^{(j,k+1)}); \\ A_{\xi_{i,j,k}}^S (\xi_{i,j,k}, \eta_{j,k+1}) = w_j^S (C_{[3],[i]}^{(j,k)} - C_{[2],[i]}^{(j,k+1)}); \\ A_{\xi_{i,j,k}}^S (\xi_{i,j,k}, j) = w_j^S (C_{[3],[i]}^{(j,k+1)} + C_{[2],[i]}^{(j,k+1)} + C_{[1],[i]}^{(j,k+1)} - C_{[3],[i]}^{(j,k)} - C_{[2],[i]}^{(j,k)} - C_{[1],[i]}^{(j,k)}); \}$$

对于  $E_L$

$$\text{for} \left( \begin{array}{l} j = 1; j \leq n; j++ \\ k = 1; k \leq |a_{V_j}|; k++ \end{array} \right) \{ \xi_{j,k} = \sum_{\tau=1}^{j-1} |a_{V_{\tau}}| + k +$$

$$\begin{aligned}
 & i; \boldsymbol{\eta}_{j,k} = \mathbf{a}_{V_j}[k]; \mathbf{D}_{\xi_{j,k}}^L(\xi_{j,k}) = -w_j^L(\mathbf{V}_j - \mathbf{V}_{\eta_{j,k}}); \\
 & \mathbf{A}_{\xi_{j,k}}^L(\xi_{j,k}, \boldsymbol{\eta}_{j,k}) = -w_j^L \mathbf{I}; \mathbf{A}_{\xi_{j,k}}^L(\xi_{j,k}, j) = w_j^L \mathbf{I}; \} \\
 & \text{其中, } \{\mathbf{V}_j\} \text{ 为目标网格原始顶点集。对于 } E_d \\
 & \text{for}(j = 1; j \leq n; j++) \{ \xi_j = j; \\
 & \mathbf{A}_{\xi_j}^d(\xi_j, j) = w_j^d \mathbf{I}; \mathbf{D}_{\xi_j}^d(\xi_j) = -w_j^d \mathbf{V}_j^{\text{new}}; \}
 \end{aligned}$$

则式(A3)可以改写为

$$\begin{aligned}
 E_z &= \sum_{\xi}^{N_z} \left\| \mathbf{A}_{\xi}^z \hat{\mathbf{V}} + \mathbf{D}_{\xi}^z \right\|_F^2 \\
 &= \sum_{\xi}^{N_z} \left( (\mathbf{A}_{\xi}^z \hat{\mathbf{V}})^T + (\mathbf{D}_{\xi}^z)^T \right) \cdot (\mathbf{A}_{\xi}^z \hat{\mathbf{V}} + \mathbf{D}_{\xi}^z) \\
 &= \sum_{\xi}^{N_z} \left( \hat{\mathbf{V}}^T (\mathbf{A}_{\xi}^z)^T \mathbf{A}_{\xi}^z \hat{\mathbf{V}} + \hat{\mathbf{V}}^T (\mathbf{A}_{\xi}^z)^T \mathbf{D}_{\xi}^z + \right. \\
 & \quad \left. (\mathbf{D}_{\xi}^z)^T \mathbf{A}_{\xi}^z \hat{\mathbf{V}} + (\mathbf{D}_{\xi}^z)^T \mathbf{D}_{\xi}^z \right) \tag{A4}
 \end{aligned}$$

从  $\mathbf{A}_{\xi}^z$  和  $\mathbf{D}_{\xi}^z$  的非零元素的分布容易推导出

$$(\mathbf{A}_i^z)^T \mathbf{D}_j^z = 0 \quad i \neq j \tag{A5}$$

因此,式(11)的最小化问题就转化为求解以下向量值线性方程组的问题:

$$\begin{aligned}
 \frac{\partial E}{\partial \hat{\mathbf{V}}} &= 2 \left( \sum_{\xi=1}^{N_S} (\mathbf{A}_{\xi}^S)^T \mathbf{A}_{\xi}^S + \sum_{\xi=1}^{N_L} (\mathbf{A}_{\xi}^L)^T \mathbf{A}_{\xi}^L + \right. \\
 & \quad \left. \sum_{\xi=1}^{N_R} (\mathbf{A}_{\xi}^R)^T \mathbf{A}_{\xi}^R + \sum_{\xi=1}^{N_d} (\mathbf{A}_{\xi}^d)^T \mathbf{A}_{\xi}^d \right) \hat{\mathbf{V}} + \\
 & \quad 2 \left( \sum_{\xi=1}^{N_R} (\mathbf{A}_{\xi}^R)^T \sum_{\xi=1}^{N_R} \mathbf{D}_{\xi}^R + \sum_{\xi=1}^{N_d} (\mathbf{A}_{\xi}^d)^T \sum_{\xi=1}^{N_d} \mathbf{D}_{\xi}^d + \right. \\
 & \quad \left. \sum_{\xi=1}^{N_L} (\mathbf{A}_{\xi}^L)^T \sum_{\xi=1}^{N_L} \mathbf{D}_{\xi}^L \right) = 0
 \end{aligned}$$

这是一个标准的向量值线性方程组。本文使用一个稀疏 LU 算子<sup>[22]</sup>来有效求解这个方程。