



JOURNAL OF IMAGE AND GRAPHICS

主办：中国科学院遥感与数字地球研究所
中国图象图形学学会
北京应用物理与计算数学研究所

中国图象学报 图形学报

2020
05
VOL.25

ISSN1006-8961
CN11-3758/TB



智能疫
“小珈”
助力



第25卷第5期（总第289期）

2020年5月16日

中国精品科技期刊
中国国际影响力优秀学术期刊
中国科技核心期刊
中文核心期刊

版权声明

凡向《中国图象图形学报》投稿，均视为同意在本刊网站及CNKI等全文数据库出版，所刊载论文已获得著作权人的授权。本刊所有图片均为非商业目的使用，所有内容，未经许可，不得转载或以其他方式使用。

Copyright

All rights reserved by Journal of Image and Graphics, Institute of Remote Sensing and Digital Earth, CAS. The content (including but not limited text, photo, etc) published in this journal is for non-commercial use.

主管单位 中国科学院

主办单位 中国科学院遥感与数字地球研究所

中国图象图形学学会

北京应用物理与计算数学研究所

主 编 顾行发

编辑出版《中国图象图形学报》编辑出版委员会

通信地址 北京市海淀区北四环西路19号

邮 编 100190

电子信箱 jig@radi.ac.cn

电 话 010-58887035

网 址 www.cjig.cn

广告发布登记号 京朝工商广登字20170218号

总 发 行 北京报刊发行局

订 购 全国各地邮局

海外发行 中国国际图书贸易集团有限公司

（邮政信箱：北京399信箱 邮编：100048）

印刷装订 北京科信印刷有限公司

Journal of Image and Graphics

Title inscription: Song Jian

Monthly, Started in 1996

Superintended by Chinese Academy of Sciences

Sponsored by Institute of Remote Sensing and Digital Earth, CAS

China Society of Image and Graphics

Institute of Applied Physics and Computational Mathematics

Editor-in-Chief Gu Xingfa

Editor, Publisher Editorial and Publishing Board of Journal of
Image and Graphics

Address No. 19, North 4th Ring Road West, Haidian District,
Beijing, P. R. China

Zip code 100190

E-mail jig@radi.ac.cn

Telephone 010-58887035

Website www.cjig.cn

Distributed by Beijing Bureau for Distribution of Newspapers and Journals

Domestic All Local Post Offices in China

Overseas China International Book Trading Corporation

(P.O.Box 399, Beijing 100048, P.R.China)

Printed by Beijing Kexin Printing Co., Ltd.

CN 11-3758/TB

ISSN 1006-8961

CODEN ZTTXFZ

国外发行代号 M1406

国内邮发代号 82-831

国内定价 60.00元



多相图像分割Vese-Chan模型连续最大流方法(第0926页)



保持区域平滑的3维动画形状高效编辑(第1019页)



树形结构卷积神经网络优化的城区遥感图像语义分割(第1043页)

学者观点

多源空一谱遥感图像融合方法进展与挑战

肖亮, 刘鹏飞, 李恒 0851

综述

中国图像工程: 2019

章毓晋 0864

图像处理和编码

相同编码参数HEVC视频重压缩检测

潘鹏飞, 姚晔, 王慧 0879

密集连接卷积网络图像去模糊

吴迪, 赵洪田, 郑世宝 0890

密集网络图像哈希检索

王亚鸽, 康晓东, 郭军, 李博, 张华丽, 刘汉卿 0900

图像分析和识别

域自适应城市场景语义分割

张桂梅, 潘国峰, 刘建新 0913

多相图像分割Vese-Chan模型连续最大流方法

王洁, 潘振宽, 魏伟波, 徐子森 0926

结合注意力机制和多属性分类的行人再识别

郑鑫, 林兰, 叶茂, 王丽, 贺春林 0936

多分辨率特征注意力融合行人再识别

沈庆, 田畅, 王家宝, 焦珊珊, 杜麟 0946

全卷积网络电线识别方法

刘嘉玮, 李元祥, 龚政, 刘心刚, 周拥军 0956

双核压缩激活神经网络艺术图像分类

杨秀芹, 张华熊 0967

图像理解和计算机视觉

高效检测复杂场景的快速金字塔网络SPNet

李鑫泽, 张轩雄, 陈胜 0977

结合自底向上注意力机制和记忆网络的视觉问答模型

闫茹玉, 刘学亮 0993

统筹图像变换与缝合线生成的无参数影像拼接

高炯笠, 吴军, 刘祺昌, 徐刚 1007

计算机图形学

保持区域平滑的3维动画形状高效编辑

冼楚华, 杨煜, 黄俊贤, 李桂清 1019

医学图像处理

肌骨超声图像特征检测及拼接

颜焕欢, 张培镇, 王伊依, 李璇, 阳维, 王青 1032

遥感图像处理

树形结构卷积神经网络优化的城区遥感图像语义分割

胡伟, 高博川, 黄振航, 李瑞瑞 1043

融合累积变异比和集成超限学习机的高光谱图像分类

尹玉萍, 魏林, 刘万军 1053

CONTENTS

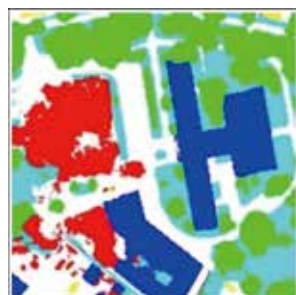
JOURNAL OF IMAGE AND GRAPHICS



Continuous max-flow method for multiphase segmentation Vese-Chan model(P0926)



Efficient 3D animation shape manipulation with region smoothness preservation(P1019)



Semantic segmentation of urban remote sensing image based on optimized tree structure convolutional neural network (P1043)

Scholar View

Progress and challenges in the fusion of multisource spatial-spectral remote sensing images
Xiao Liang, Liu Pengfei, Li Heng 0851

Review

Image engineering in China: 2019
Zhang Yujin 0864

Image Processing and Coding

Detection of double compression for HEVC videos with the same coding parameters
Pan Pengfei, Yao Ye, Wang Hui 0879

Motion deblurring method based on DenseNets
Wu Di, Zhao Hongtian, Zheng Shibao 0890

Image Hash retrieval with DenseNet
Wang Yage, Kang Xiaodong, Guo Jun, Li bo, Zhang Huali, Liu Hanqing 0900

Image Analysis and Recognition

Domain adaptation for semantic segmentation based on adaption learning rate
Zhang Guimei, Pan Guofeng, Liu Jianxin 0913

Continuous max-flow method for multiphase segmentation Vese-Chan model
Wang Jie, Pan Zhenkuan, Wei Weibo, Xu Zisen 0926

Improving person re-identification by attention and multi-attributes
Zheng Xin, Lin Lan, Ye Mao, Wang Li, He Chunlin 0936

Multi-resolution feature attention fusion method for person re-identification
Shen Qing, Tian Chang, Wang Jiabao, Jiao Shanshan, Du Lin 0946

Power line recognition method via fully convolutional network
Liu Jiawei, Li Yuanxiang, Gong Zheng, Liu Xingang, Zhou Yongjun 0956

Art image classification with double kernel squeeze-and-excitation neural network
Yang Xiuqin, Zhang Huaxiong 0967

Image Understanding and Computer Vision

Snap Pyramid Network: Real-time complex scene detecting system
Li Xinze, Zhang Xuanxiong, Chen Sheng 0977

Visual question answering model based on bottom-up attention and memory network
Yan Ruyu, Liu Xueliang 0993

Stitching of parametric-free images based on coordinated image transformation and seam-line generation
Gao Jiongli, Wu Jun, Liu Qichang, Xu Gang 1007

Computer Graphics

Efficient 3D animation shape manipulation with region smoothness preservation
Xian Chuhua, Yang Yu, Huang Junxian, Li Guiqing 1019

Medical Image Processing

Feature detection algorithm of musculoskeletal ultrasound image and its application of image stitching
Yan Huanhuan, Zhang Peizhen, Wang Yinong, Li Xuan, Yang Wei, Wang Qing 1032

Remote Sensing Image Processing

Semantic segmentation of urban remote sensing image based on optimized tree structure convolutional neural network
Hu Wei, Gao Bochuan, Huang Zhenhang, Li Ruirui 1043

Ensemble extreme learning machine with cumulative variation quotient for hyperspectral image classification
Yin Yuping, Wei Lin, Liu Wanjuan 1053

中图法分类号: TP37 文献标识码: A 文章编号: 1006-8961(2020)05-1019-13

论文引用格式: Xian C H, Yang Y, Huang J X and Li G Q. 2020. Efficient 3D animation shape manipulation with region smoothness preservation. Journal of Image and Graphics, 25(05): 1019-1031 (洗楚华, 杨煜, 黄俊贤, 李桂清. 2020. 保持区域平滑的3维动画形状高效编辑. 中国图象图形学报, 25(05): 1019-1031) [DOI:10.11834/jig.190389]

保持区域平滑的3维动画形状高效编辑

洗楚华¹, 杨煜¹, 黄俊贤², 李桂清¹

1. 华南理工大学计算机科学与工程学院, 广州 510006; 2. 英伟达半导体(深圳)有限公司, 深圳 518057

摘要: 目的 基于控制单元的形状编辑效果受各个控制单元对应权重的影响, 而计算闭合形式的控制点权重方法难以有效地处理控制骨骼权重。针对3维空间的控制骨骼提出了一种虚拟控制单元插入算法和骨骼关节点标架变换方法, 以保持骨骼控制区域的形状, 从而得到过渡平滑、形状保持的良好编辑效果。方法 选择 C^2 连续的线性权值计算方法, 在用户输入相应的控制单元后, 根据控制单元的支持度插入满足条件的虚拟控制点, 实现了对动画形状平滑高效的编辑。首先采用离散化的方式, 近似求解输入形状构成的封闭域中任意两点之间的内部距离, 然后进行Voronoi区域分解, 初步获得每个控制单元的控制区域。如果控制点的支持度约束不符合要求, 则通过插入虚拟控制点的方式进行调整, 并根据邻接关系计算实控制点对虚拟控制点的权重实现实控制点对虚拟控制点的控制。由于算法计算权值和编辑更新顶点可以并行, 因此引入图形处理器(graphics processing unit, GPU)实行并行化处理。结果 实验对比了算法在编辑细节以及对不同网格模型的适应性和编辑效率方面的表现, 结果表明本文算法在局部细节处不发生过度形变且保持平滑, 对非三角网格和多个封闭区域叠加的网格模型依然适用, 且本文算法不需要迭代, 又有GPU并行计算, 编辑时间显著下降。结论 本文算法易于实现, 编辑效果过渡平滑, 保留细节特征; GPU并行计算极大提高效率, 达到实时交互效果。

关键词: 线性混合; 平滑处理; 区域分解; 并行计算; 网格变形

Efficient 3D animation shape manipulation with region smoothness preservation

Xian Chuhua¹, Yang Yu¹, Huang Junxian², Li Guiqing¹

1. School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China;

2. NVIDIA Semiconductor (Shenzhen) Co., Ltd., Shenzhen 518057, China

Abstract: **Objective** Shape editing has been extensively researched for its applications in the fields of computer graphics and computer animation. Shape editing methods based on control units are widely used because they are intuitive, effective, and easy to implement during operation on animation characters. Generally, for an input 3D animation shape, the editing method first establishes the relationship between the shape and control units. It then calculates the weights of the points on the shape affected by each control unit and completes shape editing by applying transformations on the control units and updating the points on the shape by a linear combination of the control units with the pre-computed weights. Usually, the editing result depends seriously on weight calculation. Although numerous methods of weight computation have been introduced, most of them cannot effectively or efficiently handle control skeletons. In addition, the linear combination of rotation

收稿日期: 2019-08-14; 修回日期: 2019-09-24; 预印本日期: 2019-09-30

基金项目: 广东省自然科学基金项目(2017A030313347)

Supported by: Natural Science Foundation of Guangdong Province, China(2017A030313347)

is no longer a rotation. Thus, the transformation of 3D skeletons is very complicated, and the transformation of local frame at bone joint points is difficult to determine. In this study, our proposed algorithm is aimed at editing 2D images, from 2D domain to 3D domain. In our editing method, the control units are classified into real units (or user-defined units) and virtual units. The virtual control units are generated by our virtual unit insertion algorithm. A novel insertion algorithm of virtual control unit is developed for the control skeletons in 3D space. An improved transformation method of skeleton joint point frame is also proposed to maintain the volume of the shape around the joint points of the skeletons, resulting in naturally smoother editing results. **Method** To achieve a smooth editing result, we utilize the Bezier function with 5° as the basic function for weight computation because of its C^2 weighting properties. The virtual control points based on the C^2 continuous weight calculation function are inserted into the input model based on the pre-input control units. The points of the shape are then controlled by the virtual and real control points, thereby realizing a smooth and efficient shape editing algorithm. The proposed method mainly includes several key steps: 1) calculating the internal distance for region decomposition, 2) inserting virtual control units, and 3) calculating control weights. The input 3D shape is voxelized. Next, the internal distances of the points on the shape and the control units are computed in the closed 3D space. The 3D space is decomposed into different parts according to the internal distances to obtain initial control region of each control unit. In our method, the support domain of a control point is also defined to achieve intuitive editing results. No other control point should be in a determined support region. If the regions do not satisfy the constraint of control point support domain, the insertion of virtual control points is required. In this work, the control units are divided into two classes: control points and control skeletons. The virtual control points inserted according to the pre-input control points become the first class of virtual control points, whereas the ones inserted according to the control skeletons are called the second class of virtual control points. When editing a 3D shape, the virtual control points cannot be directly operated by the user. Therefore, the transformation rules of the virtual control points following the real control points are also defined. 1) For the first type of virtual control points, the transformation is general translation and rotation. 2) For the second class of virtual control points, the local frame must be constrained to have an axis along the skeleton direction during transformation. When all support domains of the control units fulfill the constraints, the transformation of the real and virtual control points is then determined by solving a sparse linear system. Finally, the weights of all control points (including real and virtual control points) on the points of the 3D shape are computed using the C^2 weighting function. During the editing process, users can conduct transformations on the real control units, and the input shape will be updated by setting new coordinates of the points on the shape with the transformation propagations of the control units. In our pipeline, the distance computation, weight computation, and shape updating steps can be computed in parallel. Thus, we incorporate graphics processing unit (GPU) to conduct the computations in these steps, which accelerate the computation significantly. **Result** In this study, we compare our method with existing methods: the SR-ARAP (smooth rotation as-rigid-as-possible) method, and Enhanced-ARAP (smooth rotation enhanced as-rigid-as-possible) method. Our experiments mainly compare these algorithms according to four aspects: editing details, adaptability of the algorithm to the non-triangular mesh models, multiple closed-domain overlay models, and computation time of the editing process. Experimental results show that the proposed algorithm never over-deforms and maintains smooth local details of the model, especially the shape around the control skeleton region. For the non-triangular mesh model and multiple closed-domain overlay models, the proposed algorithm is still applicable. However, the other two algorithms are not adaptive. Importantly, our algorithm requires no iteration, and due to GPU parallel computation, the editing efficiency is significantly improved. **Conclusion** Our proposed method is easy to understand and implement. The editing effect is natural, intuitive, and smooth, and the model details can be preserved quite well. Thus, the proposed method satisfies the general requirements of 3D shape editing. At the same time, the algorithm efficiency is greatly improved given the GPU parallel computing, which delivers feedback to the user operation faster. It basically achieves the effect of real-time interaction and thus improves user experience.

Key words: linear blending; smooth manipulation; region decomposition; parallel computing; mesh deformation

0 引言

形状编辑技术在计算机动画中应用广泛。与其他模型编辑方法相比,基于控制单元的方法因其对动画角色的操作直观有效、易于实现而备受关注,并广受学者研究。此类方法首先显式地建立模型和控制单元之间的关系,然后计算各个控制单元对模型顶点的影响权重,通过移动控制单元,再利用相应的权重进行顶点位置的更新来编辑模型。经典方法包括嵌入变形(Sumner等,2007),该方法允许用户编辑由均匀分布的图形节点和指示模型上节点之间重叠影响的边缘组成的预定义变形图;有界双调和权重(Jacobson等,2011),该方法通过最小化双调和能量获得线性混合的权重;内部径向基函数(interior radial basis functions, IRBF)方法则采用局部或全局方法计算系数(Levi和Levin,2014);优化旋转中心方法提出了计算控制骨骼的最佳旋转中心策略(Le和Hodgins,2016);高效率的 C^2 加权方法则引入显式计算权重的方法来获得基于线性混合的图像变形方法的高质量权重(Xian等,2018)。另一类方法是基于将用户操作编码为约束的模型,通过求解预设能量最小化来确定蒙皮结果,例如spokes and rims(Chao等,2010),该方法基于弹性差分能量来计算变形结果;尽可能刚性(as-rigid-as-possible, ARAP)法则提出了近似刚性方式评估模型变形的办法(Levi和Gotsman,2015);局部子空间设计提出了一种一致的变分形式来实现局部线性变形(Wang等,2015)。此外,还有一些研究人员提出利用并行计算的强大能力来提高计算性能。例如Schäfer等人(2015)提出直接在图形处理器(graphics processing unit, GPU)上进行形状编辑和渲染以避免预计算以及与中央处理器(central processing unit, CPU)进行数据传输上的耗时。

本文设计了一个 C^2 连续的局部权值计算函数获得线性混合权重的算法来编辑3维动画形状。给定一个动画形状和预定义的控制单元,首先对整个形状区域进行基于Voronoi的区域分解,将形状分解为多个区域,然后计算控制单元对网格顶点的权值,最后通过移动预定义控制单元驱动基于线性混合的形状编辑。其中,控制单元主要包括控制点和控制骨骼。Xian等人(2018)提出了无网格 C^2 连续

算法来编辑2维图像。本文将该方法从2维扩展到3维。对于3维形状而言,旋转的线性组合变换是复杂的,不能使用一次旋转变换来代替(Alexa, 2002),所以控制骨骼的变换非常复杂。

本文方法主要有3方面改进:1)针对3维空间控制骨骼提出了一种虚拟控制单元插入算法;2)提出了一种骨骼关节点标架变换方法以保持两个或多个骨骼周围的形状体积;3)拓展了编辑算法,使其可以在GPU上并行执行,大大提高编辑的效率。

1 相关研究

形状编辑作为计算机图形学和计算机动画中的重要技术,已有大量研究成果,可将现有编辑方法分为3类:基于微分属性、基于空间和线性混合蒙皮的方法。

20世纪初,用于离散几何建模和处理的基于微分属性的方法已广泛应用于网格编辑。经典方法包括泊松编辑(Yu等,2004)和基于拉普拉斯坐标的编辑方法(Sorkine等,2004)。Igarashi等人(2005)提出引入体积信息和刚度属性能更好地保持形状,Lipman等人(2005)基于离散几何的微分属性提出了网格旋转不变坐标,从而实现旋转不变性的编辑方法。这些方法利用网格模型的分属性来求解超定稀疏线性系统去重构网格,一旦移动控制单元则需要重新计算网格模型所有顶点的变换。如果模型包含大量顶点即意味着需要求解一个庞大的线性系统,这将比较耗时。此外,如果用户在网格模型上插入新控制单元,则必须重建相对应的线性系统。

Sederberg和Parry(1986)首次提出了基于空间的形状编辑方法。随后也出现了许多典型的基于空间的方法,包括均值坐标(Floater等,2005),调和坐标(Joshi等,2007)和格林坐标(Lipman等,2008)。首先构造封闭的控制网格或笼子来包围目标模型,然后将笼子的内部点定义为封闭笼子顶点的线性组合,顶点的权重可以预先计算并被视为坐标。用户可以移动或旋转笼子顶点的局部标架,然后使用预先计算的权重更新目标形状顶点位置来编辑目标形状。在基于空间的编辑方法中笼子起着重要的作用,Le和Deng(2017)提出了一个为具有任意拓扑复杂性的3维模型生成高质量笼子的高效的方法,

即使没有明显的骨架结构,也可以处理各种 3 维模型。Casti 等人(2019)提出了一种用户辅助的、由模型骨架指导的笼子生成方法。但是通过笼子编辑目标形状是间接的,而且笼子的建造相当繁琐,需要熟练的专家数小时精心设计才能得到一个比较好控制的笼子。

线性混合蒙皮技术,也称为骨骼子空间变形,因其原理简单、可预测且效率高,广泛用于动画形状编辑。Lewis 等人(2000)介绍了线性混合蒙皮的数学描述。这些技术的基本思想是在动画形状上定义一系列骨骼,并将形状区域表示为骨骼的线性变换,编辑结果严重依赖于蒙皮权重和骨骼变换。Kavan 等人(2007)利用球形蒙皮和四元数来避免在骨骼上进行旋转时产生的伪影。Eisemann 和 Thiery (2017)综合了尽可能刚性的方法来优化大规模旋转,从而避免大变形中的伪影。然而除了避免旋转产生的伪影,还应考虑旋转的线性组合变换不能用一次旋转变换代替(Alexa, 2002),骨骼关节的变换也很难确定。Mukai 和 Kuriyama(2016)提出了一种通过构建辅助骨骼控制器控制由骨骼姿势和次级动力学效果引起的变形的办法,有效地将骨架运动映射到辅助骨骼的动态运动,从而合成合理的蒙皮变形;Eisemann 和 Thiery (2017)则采用骨骼参数化方法,利用统计权重优化技术对 3 维网格进行线性混合蒙皮。另外,控制单元对网格顶点的权重计算也很重要,它直接影响控制单元的仿射变换。上述基于空间的方法可以应用于计算闭合形式的控制点权重,但是无法有效地处理控制骨骼。虽然 Jacobson 等人(2011)使用有界双调和函数来计算平滑权重并生成形状感知变换(Jacobson 等, 2012),从而更好地保持形状,但是仍然无法避免大型线性系统的问题。Xian 等人(2018)提出了一种有效的无网格 C^2 加权方法,引入一种显式方法来计算权重,但它只关注 2 维控制点和控制骨骼。本文将该方法从 2 维扩展到 3 维,因为 3 维骨骼的变换非常复杂,为处理控制骨骼旋转的问题,本文还提出一种新的算法来处理控制骨骼关节的旋转。

2 方法概述

一般地,对于一个输入的网格模型 M ,其上有 m 个控制单元(控制点和控制骨骼),可以用线性混

合的方法更新模型顶点 p 的位置进行形状编辑,即

$$p' = \sum_i^m w_i(p) T_i p \quad (1)$$

式中, $w_i(p)$ 表示控制单元 i 对顶点 p 的权重, T_i 表示该控制单元上的变换, p' 表示更新后顶点 p 的位置。本文方法中的控制单元包括模型输入时用户定义的实控制单元和由本文方法生成的虚拟控制点(详见 3.3 节)。

对模型进行线性混合编辑,最重要的是计算模型上的顶点受控制点影响的权值,以及该点相对于控制点变换的关系。本文采用 Xian 等人(2018)提出的 C^2 连续的权值算法来实现控制点对网格顶点影响权值的计算。对于控制骨骼,则采用离散化方法,将其离散化成一系列控制顶点,并利用新的变换方式实现利用控制骨骼对形状进行平滑编辑。

算法框架如图 1 所示,主要步骤是:1) 计算内部距离并进行区域分解;2) 插入虚拟控制单元和计算控制权重。采用 Xian 等人(2018)定义的内部距离来度量网格顶点和控制点的距离,对于预输入的控制骨骼,将它离散化为两个加约束的骨骼端控制点,依据控制点内部距离计算的规则进行处理,根据内部距离进行区域分解从而获得初始的控制点控制区域。当存在不满足控制点支持度约束的区域时,执行插入虚拟控制单元的步骤;在插入虚拟控制单元时,对预输入的控制点和控制骨骼采取不同的插入策略。当所有区域都满足控制点支持度约束后,将 C^2 连续的权值计算方法从 2 维推广到 3 维,用于计算每个控制点对网格顶点的影响权重。最后通过 GPU 并行计算和更新顶点坐标完成形状编辑。

3 方法实现细节

3.1 输入和初始化

在编辑模型前,需要输入相应的控制单元来编辑网格。控制单元包括编辑网格时用到的控制顶点和控制骨骼。

初始化的目的是对于每个控制顶点以及控制骨骼的两个端点生成相应的局部标架。一般地,初始化的标架方向可以是任意的,但为了简化计算,采用标准右手坐标系,并选择其 X 轴为控制点的初始化标架方向,如图 2(a) 所示。对于控制骨骼,先对骨骼进行离散化处理,然后用相应的离散点作为控制

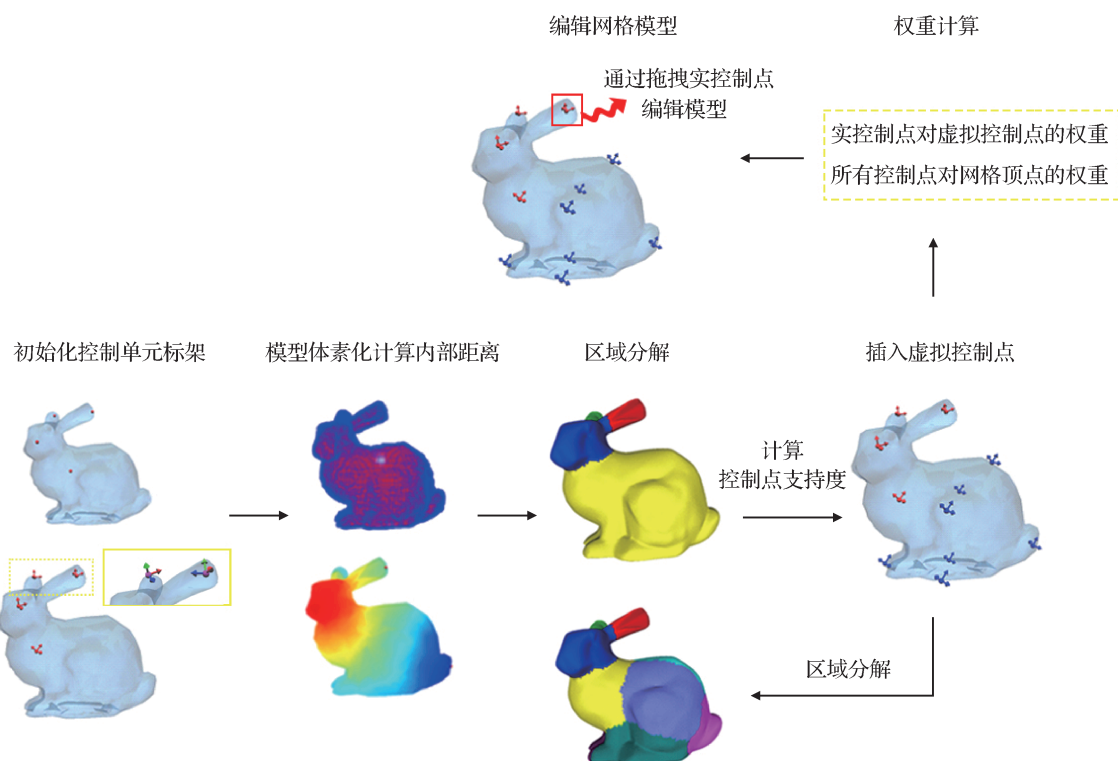


图1 本文方法框架

Fig. 1 Algorithm framework

点来逼近骨骼的控制效果。骨骼的端点可分为两种类型:叶子点和关节点。叶子点只属于一个骨骼,关节点则是两个或多个骨骼的公共端点。在初始化叶子节点的标架时,把标架的 X 轴方向约定为骨骼延伸的方向,如图 2(b)。关节点标架初始化则需要考虑到多个骨骼的综合影响。先根据连接关系建立好骨骼端点的连接关系图,然后来确定关节点标架方向,即

$$F_i = \frac{1}{N} \sum_j F_{ij} \quad (2)$$

式中, F_i 表示关节点 i 的标架, F_{ij} 表示该关节点的邻居 j 的标架, N 表示邻居的数量。将所有连接关系图的叶子点的初始化标架作为约束,而其他所有关节点利用式(2)展开,则构成一个线性系统,通过求解该线性系统即可得到所有关节点标架初始化方向,图 2(c)是一个关节点标架示例。

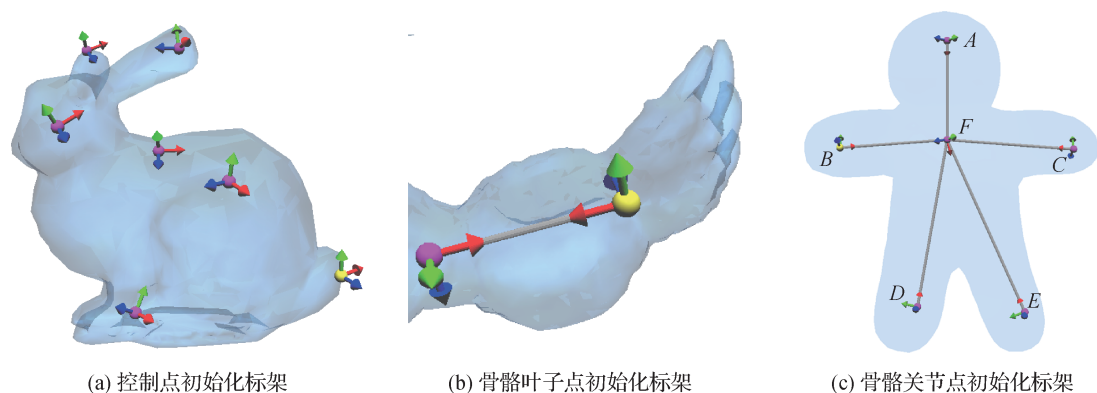


图2 控制点和控制骨骼标架初始化示意图

Fig. 2 Local coordinate axis (LCA) initialization of control points and control skeletons

((a) LCA of control points; (b) LCA of skeleton leaf control point; (c) LCA of skeleton joint control point)

3.2 计算内部距离和区域分解

为了进行区域分解,首先要计算由形状包围起来的所有点相对于控制顶点的内部距离。采用 Xian 等人(2018)的定义,即由输入形状构成的封闭域的最短距离作为区域内两点的内部距离。然而,在3维空间任意封闭域内找出严格意义上的任意两点的内部距离,计算量非常大。为加快求解过程,本文实行并行计算,并采用离散化的方式近似求解输入形状封闭域内两点的内部距离。

3.2.1 内部距离估算

在计算前,先对输入形状进行体素化,即将形状内部和表面离散为一系列的体素。显然,在对网格模型进行体素化后,形状内部和表面上任意一点,都会位于任意一个体素中。这样,形状表面和内部任意两个点内部距离则可以近似为这两个点对应的两个体素间的距离。假设待计算的两个顶点分别为 p_i 和 p_j ,那么这两个点的内部距离 d_{ij} 计算步骤如下:

1) 若 p_i 和 p_j 在同一个体素内,则内部距离 d_{ij} 定义为两个顶点的欧氏距离;

2) 假设 p_i 和 p_j 所在的体素分别为 t_i 和 t_j ,则 $d_{ij} = D_{ij} + d_i + d_j$ 。这里, D_{ij} 表示 t_i 和 t_j 中心的距离,采用最短路径方法求得。 d_i 和 d_j 分别是 p_i 和 p_j 与体素 t_i 和 t_j 的中心的欧氏距离。

此算法计算距离的时间复杂度为 $O(n_{\text{vox}})$, 这里 n_{vox} 表示网格模型体素化后的体素个数。图3是一个 Dragon 模型控制点分布及其中一个控制点距离场分布的例子。图中显示对于指定的某一控制点,距离该控制点越近的点颜色偏红,越远的点颜色偏蓝。图3的示例表明上述计算内部距离方法的效果较好。

3.2.2 区域分解

在计算控制顶点对形状空间的点的权值前,需要对形状进行区域分解。以控制点为种子点对用户输入的 shape 空间进行 Voronoi 区域分解。传统3维空间的 Voronoi 区域划分方法复杂度比较高,而且由于模型的复杂性,内部距离不是定义在欧氏距离上,因此给直接划分带来了较大的难度。

前文已经计算了封闭区域内点之间的内部距离,即空间封闭域内的点到所有控制点的距离已知,因此不需要求解控制点“邻近”关系,也不需要求解垂直平分面,问题转化为求解区域划分边界点。

假设控制点集合为 $H(\cdot)$,对于空间给定一点



(a) 模型上的控制点



(b) 头部某一控制点距离场色带分布

图3 Dragon 控制顶点及头部某一控制点的距离场
Fig. 3 Control points of the Dragon model and the distant field of one control point ((a) control points of the Dragon model; (b) distant field of one control point)

p 到控制点 $H(j)$ 的距离表示为 $d(p, H(j))$ 。这样,记录点 p 到每一个控制点的距离,并选取这些控制点中距离 p 点最近的一个,由 Voronoi 区域划分的定义可知, p 点即属于该控制点所属区域。然后遍历形状内部和表面的每一个点,重复以上过程,即可完成对输入形状的 Voronoi 划分,从而对形状进行区域分解。

图4是采用上述方法进行区域分解的示例,不同的颜色表示不同的区域。从图4可以看出,本文方法能对输入形状进行有效地区域划分。本文区域划分的时间复杂度为 $O(n_v \cdot N_H)$, n_v 表示网格顶点数, N_H 表示控制点的个数。本文方法比传统的划分方法更高效,且可以进行并行化处理。

3.3 虚拟控制顶点的生成

Xian 等人(2018)提出,编辑形状的控制点 h_i 的支持度 r_i 应该满足约束为

$$r_d(h_i) \leq r_i \leq r_h(h_i) \quad (3)$$

式中, $r_d(h_i)$ 表示控制点 h_i 所在 Voronoi 区域内所有点与控制点 h_i 的最大内部距离(称为 h_i 的 Voronoi 距离); $r_h(h_i)$ 则表示控制点 h_i 与所有其他控制点的最小内部距离(称为 h_i 的隔离距离)。在

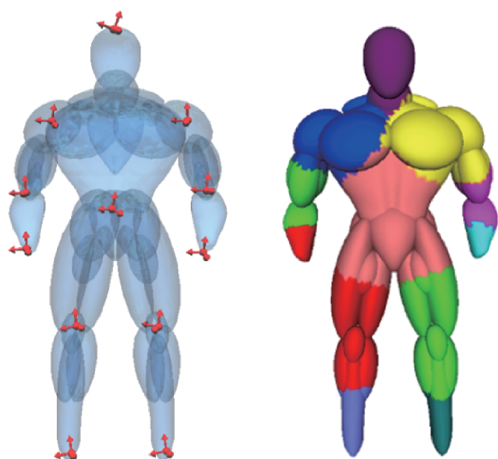


图4 3维模型实例的区域划分

Fig. 4 Region decomposition on a 3D model

编辑模型前,先检查所有的控制点是否满足式(3)。如果对于某个控制点,出现隔离距离小于该控制点的 Voronoi 距离,则需要插入一个新的虚拟控制点,以避免该控制点控制一个过大的区域,产生不直观的编辑效果。

把虚拟控制点分为两种类型,一种基于用户输入控制点(后面称为实控制点)生成,称为第Ⅰ类虚拟控制点;另一种基于用户输入的控制骨骼生成,称为第Ⅱ类虚拟控制点。第Ⅰ、Ⅱ类控制点分别使用不同的方法生成。第Ⅰ类虚拟控制点采用类似 Xian 等人(2018)方法生成。图5是一个插入第Ⅰ类虚拟控制点的例子。第Ⅰ类虚拟控制点的标架方向初始化的要求与实控制点相同。

由于本文方法首先处理实控制点,因此在开始时,先使用骨骼端点控制点代替预输入控制骨骼,然后对这些控制点执行第Ⅰ类虚拟控制点插入算法。

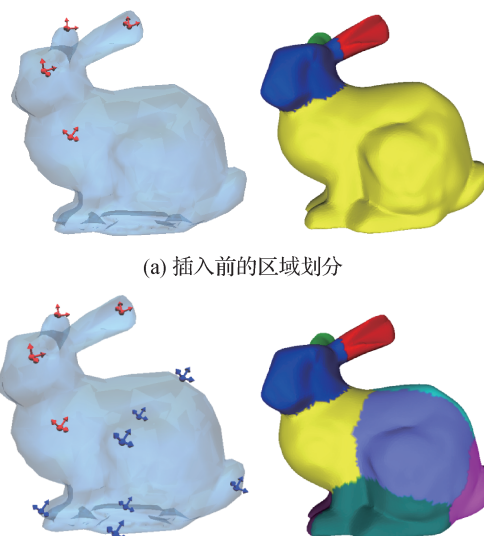
对于控制骨骼,仅用骨骼的两个端点代替骨骼既不合理也不直观。若单纯地只插入第Ⅰ类虚拟控制点也难以完全满足形状保持的要求。为了使用式(1)统一进行顶点更新,要对骨骼进行离散化逼近,则需插入第Ⅱ类虚拟控制点,从而更好地保证控制骨骼影响区域刚性形变和保持局部形状。

假设对控制骨骼两端控制点初始化支持度分别为 r_s^1 和 r_s^2 , 令

$$d' = \min \{r_s^1, r_s^2\} \quad (4)$$

则有

$$n = \frac{L_s}{d'} \quad (5)$$



(a) 插入前的区域划分

(b) 插入后的区域划分

图5 插入第Ⅰ类虚拟控制点前后区域分解结果对比图

Fig. 5 Comparison of region decomposition before and

after virtual control points insertion

((a) before virtual control points insertion;

(b) after virtual control points insertion)

式中, L_s 表示控制骨骼 S 的长度, n 表示将该控制骨骼分成 n 段。在操作时,以其中一个端点为参考,在该控制骨骼每隔 L_s/n 处插入一个新控制点即为第Ⅱ类虚拟控制点,局部标架 X 轴方向始终沿着控制骨骼延伸的方向。

然而,执行第Ⅱ类虚拟控制点插入算法后,有可能无法保证所有控制点都满足式(3)的支持度约束,此时需要将已经生成的第Ⅰ、Ⅱ类虚拟控制点连同原来输入的实控制点一起作为输入,再次执行第Ⅰ类和第Ⅱ类虚拟控制点插入算法,直到所有控制点满足支持度约束为止。图6展示的是一个平面模型虚拟控制点的完整生成流程,图中所有控制点的标架都已经初始化。

3.4 权重计算和控制点的变换

计算控制单元对网格顶点的影响权重是形状编辑的关键。采用 Xian 等人(2018)提出的连续权值计算方法,也就是采用度为5的贝塞尔曲线作为控制点对顶点权重的基函数,即

$$\phi_i(t) = (1-t)^5 + 5t(1-t)^4 + 10t^2(1-t)^3 \quad (6)$$

式中, t 代表实参,是函数的自变量, $\phi_i(t)$ 是函数的因变量。

● 为实控制点或者控制骨骼离散化后的控制点 ● 为第 I 类虚拟控制点 ● 为第 II 类虚拟控制点

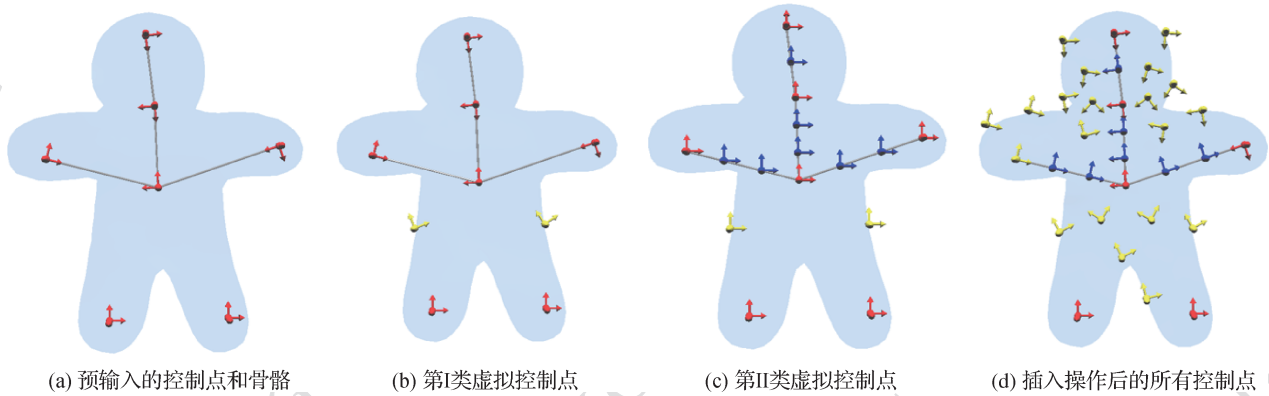


图6 插入虚拟控制点算法的整体流程

Fig. 6 Insertion of virtual control points ((a) initial control points and skeletons; (b) insert the first type of virtual control points; (c) insert the second type of virtual control points; (d) all control points after the insertion operation)

该基函数保证在待形变的控制区域内是连续的,使得用户可以通过变换控制点的标架来得到 3 维模型的理想平滑变形结果。另外由于虚拟控制点无法由用户直接操控,它跟随实控制点变化而进行相应的变换,因此还要计算实控制点对虚拟控制点的权重。同样采用 Xian 等人 (2018) 提出的 Delaunay 图来分配实控制点的调和场值计算。但只计算了控制点权重后还不能直接进行网格顶点坐标的编辑,还需要计算相应的虚拟控制点变换方式。

在编辑模型的时候,通过拖动输入的控制点以及对该控制点的局部标架施加一个变换,然后将相应的变换传播到虚拟顶点和形状上的各个顶点去,从而来完成形状编辑。每次用户操作完实控制点的变换后,虚拟控制点则根据规则进行相应的变换。由于本文有两类不同的虚拟控制点,不同类别的控制点采用的变换规则不一样,下面分别描述这两类虚拟控制点的变换规则。

3.4.1 第 I 类虚拟控制点的变换规则

令第 I 类虚拟控制点 $H_1^v(i)$ 的位置 $O(i) = (o_x, o_y, o_z)$, 则

$$O(i) = \sum_{j=1}^m \bar{\omega}(H_1^v(i)) \cdot [O(j) + \Delta(i, j)] \quad (7)$$

式中, $\bar{\omega}(H_1^v(i))$ 表示实控制点 $H(j)$ 对虚拟控制点 $H_1^v(i)$ 的权重。 $O(j)$ 表示当前实控制点 $H(j)$ 的坐标, $\Delta(i, j)$ 表示其他第 I 类虚拟控制点 $H_1^v(i)$ 对实控制点 $H(j)$ 的相对坐标。 $H_1^v(i)$ 的标架 $N_1(i) = (n_x, n_y, n_z)$ 定义为

$$N_1(i) = (n_x, n_y, n_z) = \sum_{j=1}^m \bar{\omega}(H_1^v(i)) \cdot N(j) \quad (8)$$

式中, $N(j)$ 表示实控制点 $H(j)$ 的标架。

由于第 I 类虚拟控制点 $H_1^v(i)$ 的标架由用户输入的实控制点标架线性组合而成,不能保证该标架满足右手定则并且正交。为得到正确的局部标架,采用奇异值分解来修正式(8)。根据奇异值分解的方法可知 $N_1(i) = U \Sigma V^T$, $N_1(i)$ 是正交矩阵,则 U 和 V^T 也是 3×3 的正交矩阵, Σ 是 3×3 对角矩阵, Σ 对角线元素是 $N_1(i)$ 的奇异值。令对角矩阵 Σ 的对角线元素全为 1, 则 $\Sigma = I$, 即单位矩阵。修正后标架

$$N_1(i)' = U I V^T = U V^T \quad (9)$$

修正后的标架 $N_1(i)'$ 是最接近式(8)中的 (n_x, n_y, n_z) 的坐标系。图 7 给出了经过式(9)修正的标架前后对比图。

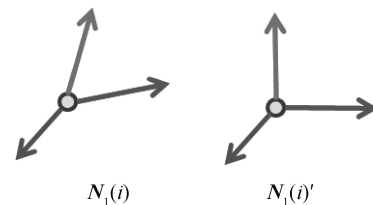


图7 修正前后的标架对比示例

Fig. 7 Before and after correction of the frame

3.4.2 第 II 类虚拟控制点的约束规则

对于第 II 类虚拟控制点,假设其标架为 $N_2(i) = (n_x, n_y, n_z)$, 在初始化时把 X 轴方向即 n_x 设置为

所在骨骼的延伸方向。

通过移动控制骨骼的端点进行形状编辑时,如果该控制骨骼上有第Ⅱ类虚拟控制点,也需要对其进行相应的变换。对第Ⅱ类虚拟控制点的变换进行相关的约束:

1) 骨骼上的第Ⅱ类虚拟控制点始终保持在原来相对骨骼端点的等分点上;

2) 标架变换遵循最小变形原则,在变换的过程中, $\mathbf{n}_x$ 方向始终沿着骨骼延伸方向,并保持 $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$ 符合右手定则正交坐标系。所谓的“最小变形”,其具体做法是:令第Ⅱ类虚拟控制点的初始标架 $\mathbf{n}_x$ 经过变换后 $\mathbf{n}'_x$,则 $\mathbf{n}_x$ 和 $\mathbf{n}'_x$ 所在平面的法向量是 $\mathbf{n} = \mathbf{n}_x \times \mathbf{n}'_x$, $\mathbf{n}_x$ 和 $\mathbf{n}'_x$ 的夹角为 $\alpha = \frac{\mathbf{n}_x \cdot \mathbf{n}'_x}{|\mathbf{n}_x| \cdot |\mathbf{n}'_x|}$ 。因此标架变换可以看做原始标架绕着法向量 $\mathbf{n}$ 旋转 α 角度,令旋转矩阵为 $\mathbf{T}$,则可得到变换后的标架为 $\mathbf{N}_2(i)' = \mathbf{T} \times \mathbf{N}_2(i)$, $\mathbf{n}_x = \mathbf{n}'_x$ 。

图8展示了控制骨骼从上向下旋转时骨骼上第Ⅱ类虚拟控制点的位置和标架变形情况。在此过程中始终约束第Ⅱ类虚拟控制点遵循上述两条约束。

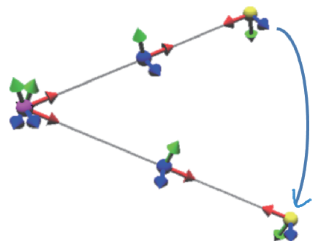


图8 第Ⅱ类虚拟控制点标架的变形示例

Fig.8 Frame transformation of the second type of virtual control points

3.5 形状编辑与并行加速

在完成所有虚拟控制点的生成,并且所有控制点都保证满足式(3)支持度约束后,就可以通过变换输入的控制点和控制骨骼来编辑形状。在用户操作完控制点或控制骨骼后,则通过式(1)来更新形状的所有顶点。

对于控制点 $\mathbf{H}(i)$,假设其标架位置为 $\mathbf{O}(i) = (o_x, o_y, o_z)$,局部标架为 $\mathbf{N}(i) = (\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z)$,则对于输入的封闭形状上的任意一点 $\mathbf{p}(j) = (x_j, y_j, z_j)$,可以计算其相对控制点 $\mathbf{H}(i)$ 的坐标 $\Delta p_{ij} = (\Delta_x, \Delta_y, \Delta_z)$,即

$$\Delta p_{ij} = [\mathbf{p}(j) - \mathbf{O}(i)] \cdot \mathbf{N}(i) \quad (10)$$

并通过3.4节中的 C^2 权值计算方法计算出控制点 $\mathbf{H}(i)$ 对点 $\mathbf{p}(j)$ 的权重 $\omega(\mathbf{p}(j))$ 。

对于式(1)中的 $T_i \mathbf{p}$,可以认为是用户对控制点进行了一个变换矩阵为 T_i 的变换,如果把编辑后的变换尽可能保持局部相对坐标不变,则式(1)中的 $T_i \mathbf{p}$ 可以写为

$$T_i \mathbf{p} = \Delta_i + O(i) \quad (11)$$

也就是更新后的顶点

$$\mathbf{p}' = \sum_{i=1}^m \omega_i(\mathbf{p}) \cdot (\Delta_i + O(i)) \quad (12)$$

式中, Δ_i 表示 $\mathbf{p}$ 点相对于控制点 $\mathbf{H}(i)$ 的相对位移。在实际编辑中,采用式(12)来更新形状上顶点坐标,从而实现对形状的编辑。当所有控制点变换的位置和变换后的标架确定后,待编辑的形状上顶点的位置也可以唯一确定下来,而且形状上各个顶点之间互不影响,可以采用并行的算法进行编辑加速,从而提高编辑的效率。

4 实验结果与讨论

4.1 GPU 并行计算效率

实验环境配置为 Windows7 操作系统、Intel i5-3470 3.20 GHz CPU、8.00 GB 内存和 Nvidia GeForce GTX 650TI 显卡。

表1是使用GPU并行计算更新顶点坐标的效率以及和串行效率的对比结果。其中 t_p 表示使用GPU并行计算耗时, t_c 表示串行计算耗时。 $|v|$ 表示网格模型顶点数目, R 表示并行比串行计算效率提升的百分比。

表1 并行与串行效率对比

Table 1 Efficiency comparison between parallel and serial

数据	$ v /10^3$	t_p/ms	t_c/ms	$R/\%$
Armadillo	15.0	15.1	47.3	68.0
Dragon	25.0	30.0	110.4	72.8
Human	34.0	9.1	29.6	69.3
Lion	91.9	25.8	56.4	54.3
Walk Man	34.0	8.9	14.2	37.3

4.2 算法编辑效果

根据图1所示的流程处理用户输入的网格模型、控制点以及控制骨骼后,通过操控预输入的控制

点编辑网格模型。

允许用户对控制点进行3种操作:1)平移控制点;2)旋转控制点;3)操控控制骨骼。提供两种策略:一是严格保持控制骨骼的长度,对定长控制骨骼的操作转化为绕关节旋转;二是允许控制骨骼伸缩,其长度不受约束,则可以任意拖拽骨骼一端到任意位置。

图9是4组实验结果,每两行表示一组实验,其中第1行显示了初始的控制点(含算法生成的虚拟

控制点)和控制骨骼位置和标架,第2行展示了编辑前后相应的渲染效果。图9(a)只输入控制点来进行编辑。图9(b)只使用控制骨骼对恐龙模型进行编辑。用户通过对手部控制点和身躯控制骨骼的编辑使得模型呈现出行走姿态。图9(c)(d)则是约束控制骨骼是否变动长度的编辑效果。从图9看到,本文方法可以使用不同的控制单元对模型进行有效编辑。

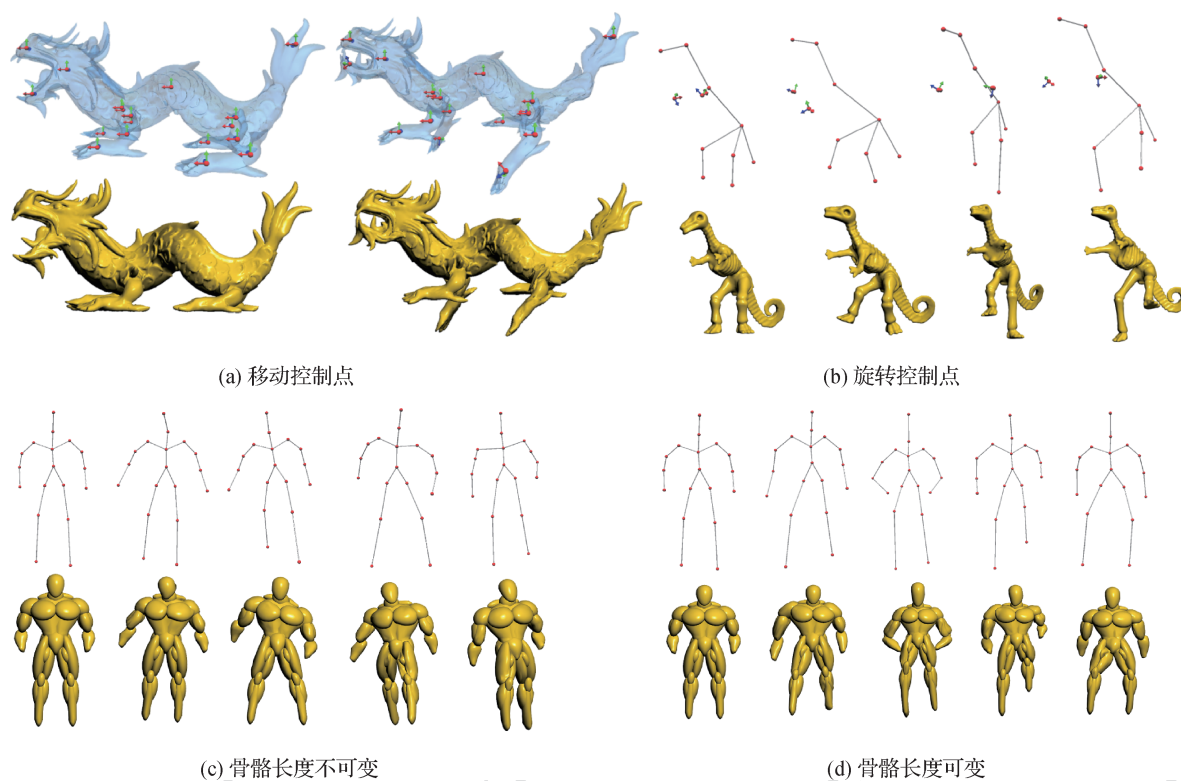


图9 控制点和控制骨骼的编辑效果

Fig. 9 Editing results of control points and skeletons((a) moving control points;
(b) rotating control points;(c) invariable length control skeleton;(d) variable length control skeleton)

4.3 实验对比

Sorkine 和 Alexa(2007)提出 ARAP 编辑算法,用户选取锚点,然后通过固定某些锚点并拖动另一些锚点实现变形。为了达到尽可能刚性的编辑效果,算法首次引入一个描述两形状间差异的能量,通过最小化这组能量获得较好的编辑效果。Chao 等人(2010)在 Sorkine 和 Alexa(2007)算法的基础上,引入弹性能量方程,模拟物理上的弹性概念,获得了更加具有弹性和自然平滑过渡的编辑效果。Levi 和 Gotsman(2015)在 Sorkine 和 Alexa(2007)算法的基础上引入一个能离散化三角网格表面的能量,使用这个能量可以获得比只使用 ARAP 能量更高质量的

编辑效果。上述3种方法的编辑效果较好,编辑过程都需要迭代计算和更新坐标位置,求解一组带约束的线性方程,相对比较耗时。

而本文方法基于控制单元的形状编辑,与上述3种方法相同,通过操纵控制单元(或锚点)实现对3维形状的编辑,但本文方法能够实现控制骨骼区域的刚性变换和自然平滑的编辑效果,而且不需要迭代计算,编辑效果和效率都有所提高。因此选择了与改进后的 ARAP 编辑算法即 Chao 等人(2010)提出的算法(SR-ARAP)以及 Levi 和 Gotsman(2015)提出的算法(Enhanced-ARAP)进行对比。实验将从以下4个方面比较:

1) 编辑细节对比。图 10 展示采用本文方法和 SR-ARAP 算法的编辑结果对比实验。实验中使用少量的控制点操作 Armadillo 模型。图 10(a) 从左至右分别为原始网格模型姿态和控制点位置、SR-ARAP 算法以及本文方法编辑后模型姿态和控制点位置。图 10(b) 展示其对应的渲染效果。从该

例子可以看出, 采用 SR-ARAP 算法对 Armadillo 的手部进行编辑时会出现非常明显的凹陷, 编辑效果很不理想。与之相反, 采用本文方法则产生非常平滑的变形结果, 这表明, 本文方法在编辑大幅度旋转时, 明显能取得比 SR-ARAP 算法更好的效果。

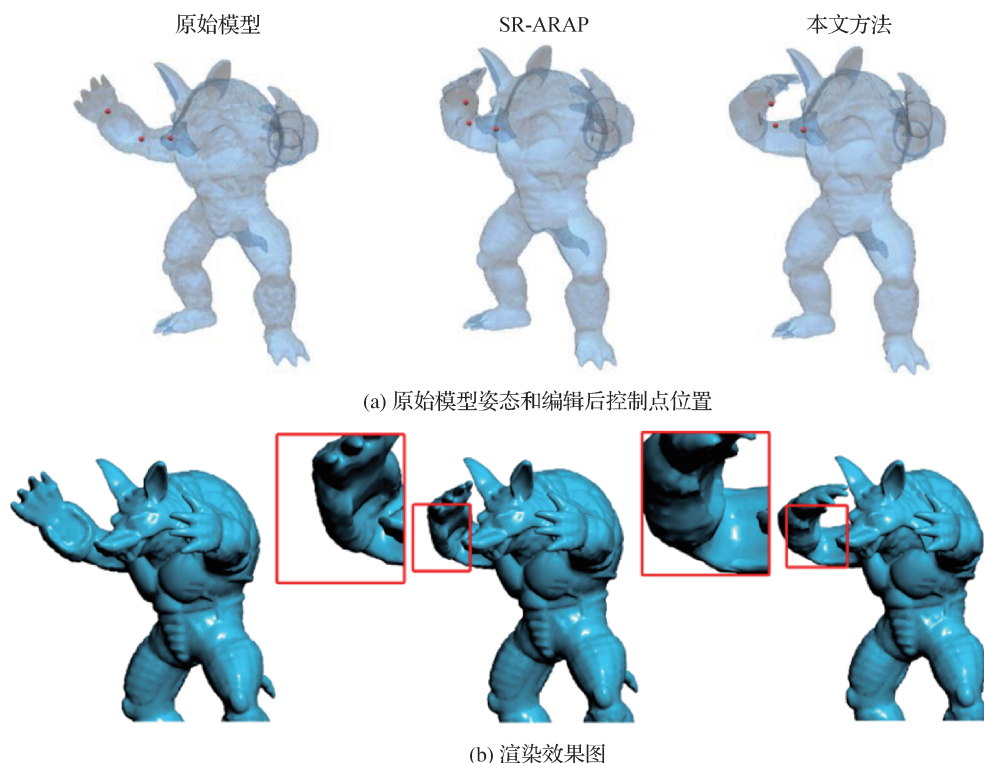


图 10 编辑结果局部细节对比图

Fig. 10 Editing results local details comparison between SR-ARAP algorithm and ours
((a) original model pose and position of control points; (b) rendering diagrams)

2) 区域连通性。SR-ARAP 和 Enhanced-ARAP 算法编辑的对象是封闭的三角形网格。对于非严格封闭域, 采用这两个方法会产生难以预测的编辑结果。针对这种情况, 使用由多个封闭域叠加而成的模型。图 11 是实验结果。红色标注点为控制点。从图 11(b) 效果图对比发现, SR-ARAP 算法在本组实验中编辑效果显然不合理, 它只在模型封闭域之间产生了位移而没有遵循控制点变换, 这表明 SR-ARAP 算法不适用多个封闭域叠加的情况, 而本文方法则依然适用且获得了良好的编辑效果。

3) 非三角网格模型的编辑。使用一个封闭的人体四边形网格模型, 如图 12 所示。红色标注点为控制点。从图 12 可以看出, 对于非三角形网格, SR-ARAP 算法会产生奇异的结果。而本文方法编辑结

果则自然平滑过渡, 并保持了模型良好的外观特征, 满足对 3 维动画形状编辑的要求。

4) 编辑的效率对比。实验在确保 3 种算法对同一个模型的编辑结果尽可能相近的情况下进行, 统计 3 种算法编辑时间, 如表 2 所示。对于 Enhanced-ARAP 和 SR-ARAP 算法的收敛误差均设置为 0.000 1, 最大迭代次数设置为 999, 算法收敛或者达到最大迭代次数则停止迭代。实验中, Enhanced-ARAP 算法未收敛, 平均迭代时间是 0.079 9 s。而 SR-ARAP 算法则需要迭代 139 次才收敛, 其编辑的总时间需要 14.738 s, 平均迭代时间是 0.105 8 s。对于同一模型, 本文方法编辑总时间是 0.013 9 s。对比另外两种算法, 本文方法比其一次迭代的时间还短。可见, 采用本文方法编辑效率非常高。

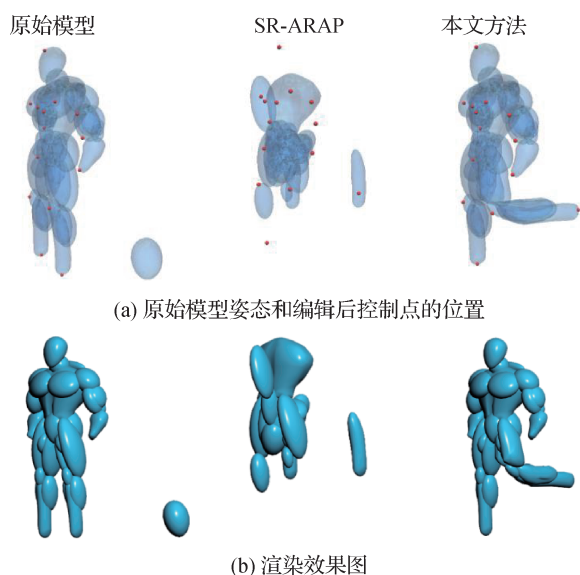


图 11 区域连通性实验对比

Fig. 11 Regional connectivity experiments comparison between SR-ARAP algorithm and ours ((a) original model pose and position of control points; (b) rendering diagrams)

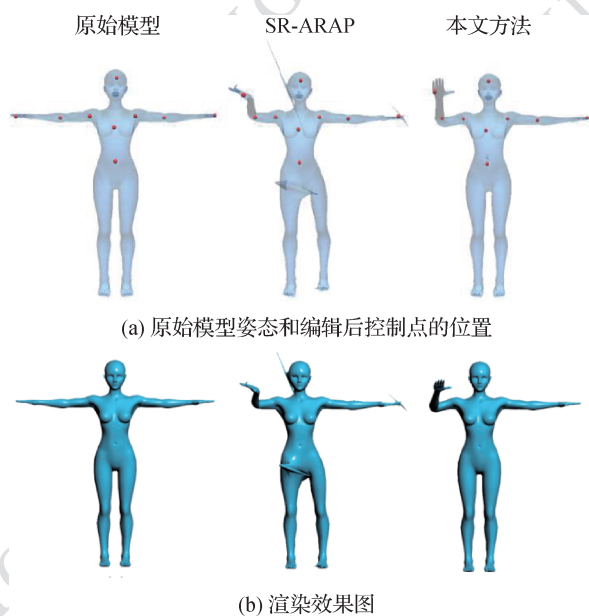


图 12 针对非三角网格的实验对比图

Fig. 12 Experiment for non-triangular meshes between SR-ARAP algorithm and ours ((a) original model pose and position of control points; (b) rendering diagrams)

5 结 论

针对 3 维动画形状,提出了一种高效的线性混合形状编辑算法。本文方法允许用户通过操控预输

表 2 3 种编辑算法效率对比

Table 2 Efficiency comparison of three editing algorithms

方法	收敛误差	编辑 总时间/s	平均迭代 时间/s	迭代次数
Enhanced-ARAP	0.000 1	79.854	0.079 9	999
SR-ARAP	0.000 1	14.738	0.105 8	139
本文	—	0.013 9	—	—

注:本文方法无需迭代,在表中收敛误差、平均迭代时间及迭代次数用“—”标识。

入的控制点和控制骨骼使 3 维动画形状进行相应形变,从而达到特定的形状编辑的目标。

本文方法原理简单易懂,实现容易,用户操作友好且编辑效率高,基本可以实时交互。主要通过计算 C^2 连续的控制点影响权重和插入虚拟控制点的关键步骤使得编辑结果过渡平滑,自然流畅,同时有效保持模型表面的细节特征,尤其是保持控制骨骼的控制区域周围的形状。除此之外,GPU 并行计算大大提高了算法的编辑效率,与对比算法的实验结果表明,本文方法编辑效果满足用户需求,用户反馈实时,用户体验更佳。

本文方法也存在一些局限。在计算内部距离时,为了提高计算效率,采用了离散化空间的方法,但这样做会带来两个问题:1) 如果被编辑的形状本身不是封闭的,则无法区分内部和外部,即无法定义“内部距离”;2) 计算内部距离的精度与离散化的精度有关,如果需要得到更为精确的距离,则需要选取更小的体素尺寸,这将会增加计算时间和内存开销。第 1 个问题的解决方案是先对输入模型进行“补洞”修复,使其成为一个封闭的模型再运用本文算法。解决第 2 个问题则需要寻求另外的不依赖体素化的方法来进行距离计算。如何克服以上两个缺点,将关系到本文方法的计算效率和适应性。将会在未来继续探讨相关的解决方法。

参考文献 (References)

- Alexa M. 2002. Linear combination of transformations//Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. San Antonio: ACM: 380-387 [DOI: 10.1145/566654.566592]
- Casti S, Livesu M, Mellado N, Rumman N A, Scateni R, Barthe L and Puppo E. 2019. Skeleton based cage generation guided by harmonic

- fields. *Computers and Graphics*, 81: 140-151 [DOI: 10.1016/j.cag.2019.04.004]
- Chao I, Pinkall U, Sanan P and Schröder P. 2010. A simple geometric model for elastic deformations. *ACM Transactions on Graphics (TOG)*, 29(4): #38 [DOI: 10.1145/1778765.1778775]
- Eisemann E and Thiery J M. 2017. Linear blend skinning weight optimization utilizing skeletal pose sampling. U. S., No. US2016/043604
- Floater M S, Kós G and Reimers M. 2005. Mean value coordinates in 3D. *Computer Aided Geometric Design*, 22(7): 623-631 [DOI: 10.1016/j.cagd.2005.06.004]
- Igarashi T, Moscovich T and Hughes J F. 2005. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics (TOG)*, 24(3): 1134-1141 [DOI: 10.1145/1073204.1073323]
- Jacobson A, Baran I, Popović J and Sorkine O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics (TOG)*, 30(4): #78 [DOI: 10.1145/2010324.1964973]
- Jacobson A, Weinkauff T and Sorkine O. 2012. Smooth shape-aware functions with controlled extrema. *Computer Graphics Forum*, 31(5): 1577-1586 [DOI: 10.1111/j.1467-8659.2012.03163.x]
- Joshi P, Meyer M, DeRose T, Green B and Sanocki T. 2007. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3): #71 [DOI: 10.1145/1276377.1276466]
- Kavan L, Collins S, Žára J and O'Sullivan C. 2007. Skinning with dual quaternions//2007 Symposium on Interactive 3D Graphics and Games. Seattle: ACM: 39-46 [DOI: 10.1145/1230100.1230107]
- Le B H and Deng Z G. 2017. Interactive cage generation for mesh deformation//Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. San Francisco: ACM: #3 [DOI: 10.1145/3023368.3023369]
- Le B H and Hodgins J K. 2016. Real-time skeletal skinning with optimized centers of rotation. *ACM Transactions on Graphics (TOG)*, 35(4): #37 [DOI: 10.1145/2897824.2925959]
- Levi Z and Gotsman C. 2015. Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE Transactions on Visualization and Computer Graphics*, 21(2): 264-277 [DOI: 10.1109/tvcg.2014.2359463]
- Levi Z and Levin D. 2014. Shape deformation via interior RBF. *IEEE Transactions on Visualization and Computer Graphics*, 20(7): 1062-1075 [DOI: 10.1109/tvcg.2013.255]
- Lewis J P, Cordner M and Fong N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation//Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press/Addison-Wesley Publishing Co.: 165-172 [DOI: 10.1145/344779.344862]
- Lipman Y, Levin D and Cohen-Or D. 2008. Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3): #78 [DOI: 10.1145/1360612.1360677]
- Lipman Y, Sorkine O, Levin D, Cohen-Or D. 2005. Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics*, 24(3): 479-487 [DOI: 10.1145/1073204.1073217]
- Mukai T and Kuriyama S. 2016. Efficient dynamic skinning with low-rank helper bone controllers. *ACM Transactions on Graphics (TOG)*, 35(4): #36 [DOI: 10.1145/2897824.2925905]
- Schäfer H, Keinert B, Nießner M and Stamminger M. 2015. Local painting and deformation of meshes on the GPU. *Computer Graphics Forum*, 34(1): 26-35 [DOI: 10.1111/cgf.12456]
- Sederberg T W and Parry S R. 1986. Free-form deformation of solid geometric models. *ACM SIGGRAPH Computer Graphics*, 20(4): 151-160 [DOI: 10.1145/15922.15903]
- Sorkine O and Alexa M. 2007. As-rigid-as-possible surface modeling//Proceedings of the 5th Eurographics Symposium on Geometry Processing. Barcelona Spain: ACM: 109-116
- Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rössl C and Seidel H P. 2004. Laplacian surface editing//Proceedings of 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. Nice: ACM: 175-184 [DOI: 10.1145/1057432.1057456]
- Sumner R W, Schmid J and Pauly M. 2007. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3): #80 [DOI: 10.1145/1276377.1276478]
- Wang Y, Jacobson A, Barbič J and Kavan L. 2015. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG)*, 34(4): #57 [DOI: 10.1145/2766952]
- Xian C H, Jin S and Wang C C L. 2018. Efficient C2-weighting for IM-AGE warping. *IEEE Computer Graphics and Applications*, 38(1): 59-76 [DOI: 10.1109/mcg.2017.2801409]
- Yu Y Z, Zhou K, Xu D, Shi X H, Bao H J, Guo B N and Shum H Y. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics (TOG)*, 23(3): 644-651 [DOI: 10.1145/1015706.1015774]

作者简介



冼楚华, 1982年生, 男, 副教授, 主要研究方向为几何建模与处理、计算机图形学、计算机视觉、智能图形处理。

E-mail: chhxian@scut.edu.cn

杨煜, 女, 硕士研究生, 主要研究方向为计算机图形学。

E-mail: y18811321352@163.com

黄俊贤, 男, 硕士研究生, 主要研究方向为计算机图形学、并行计算。E-mail: 175269715@qq.com

李桂清, 男, 教授, 主要研究方向为计算机图形学、计算机动画、虚拟现实、CAGD。E-mail: ligq@scut.edu.cn