



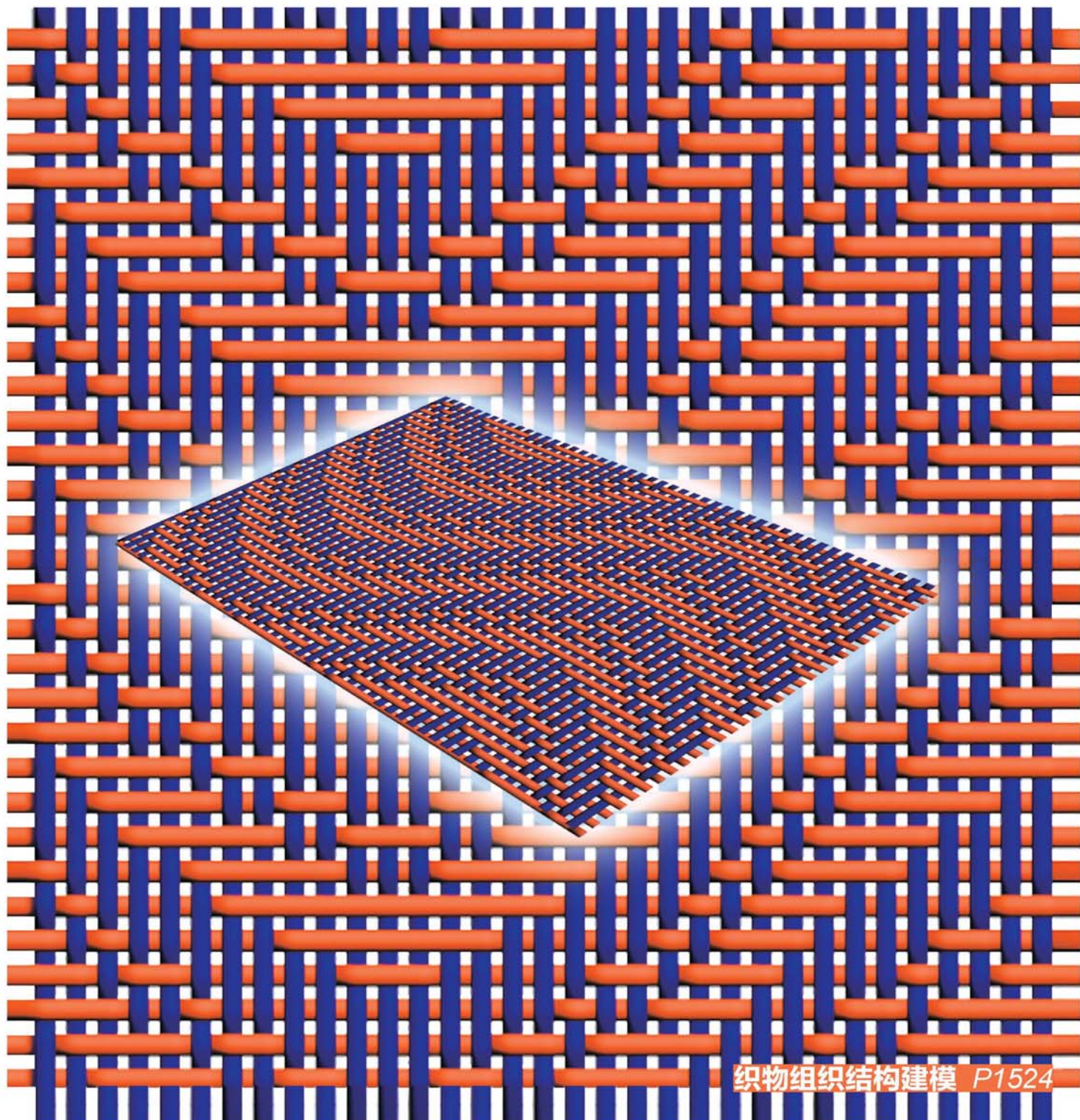
JOURNAL OF IMAGE AND GRAPHICS

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

中国图象图形学报

2014
10
VOL.19

ISSN1006-8961
CN11-3758/TB

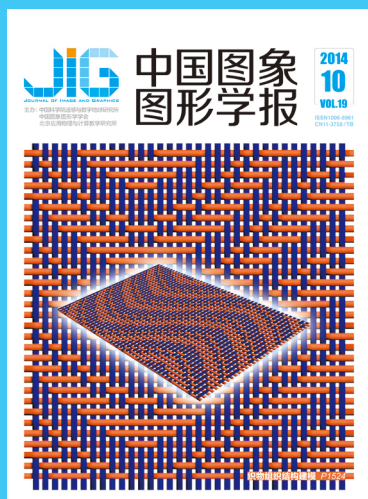


织物组织结构建模 P1524

中国图象图形学报

刊名题字: 宋健

月刊 (1996年创刊)



第19卷第10期 (总第222期)

2014年10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

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 李小文

编辑出版 《中国图象图形学报》编辑出版委员会

邮政信箱 北京9718信箱

邮 编 100101

电子信箱 jig@irsa.ac.cn

电 话 010-64807995

网 址 www.cjig.cn

广告经营许可证 京朝工商广字第0361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 LI Xiaowen

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

P.O.Box 9718, Beijing, P.R.China

Zip code 100101

E-mail jig@irsa.ac.cn

Telephone 010-64807995

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

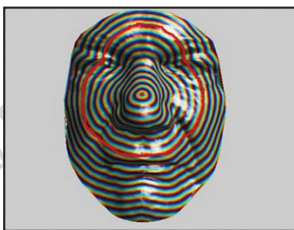
国外发行代号 M1406

国内邮发代号 82-831

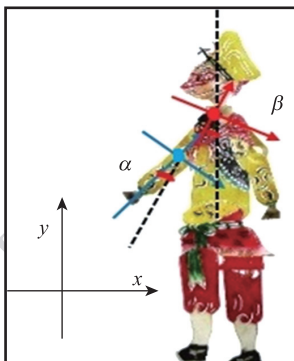
国内定价 50.00元



自适应梯度重建分水岭分割算法(第1430页)



基于特征点针对表情变化的3维人脸识别(第1459页)



数字皮影的交互仿真和动画绘制(第1490页)

综述

2维至3维图像/视频转换的深度图提取方法综述

李可宏, 姜灵敏, 龚永义 1393

图像处理和编码

优化参数化二元模映射信息隐藏

廖琪男, 柯琦, 赖振丹 1407

基于分数阶微积分的噪声检测和图像去噪

杨柱中, 周激流, 郎方年 1418

图像分析和识别

自适应梯度重建分水岭分割算法

张志禹, 孟令辉, 雷涛 1430

鲁棒空间约束的模糊聚类图像分割

刘金尧, 纪则轩 1438

灰度级信息的目标边界精确周长估算

吴秦, 周琪, 梁久祯 1449

基于特征点表情变化的3维人脸识别

李燕春, 达飞鹏 1459

图像理解和计算机视觉

曲率无穷范数约束的曲线3维重建

贺磊盈, 武传宇, 杜小强 1468

光照变化条件下的光流估计

刘骏, 祖静, 张瑜, 张红艳 1475

GPU近实时线性双目立体代价聚合

陈彬, 陈和平, 李晓卉 1481

计算机图形学

数字皮影的交互仿真和动画绘制

陈璇, 张明敏, 潘志夷 1490

遥感图像处理

由航空影像自动重建大范围3维地形

杨阿华, 李学军, 谢剑薇, 李东岳 1500

结合结构相似度的自适应多尺度SAR图像变化检测

崔莹, 熊博莅, 蒋咏梅, 匡纲要 1507

第11届全国智能CAD与数字娱乐学术会议专栏

特征驱动先验的归一化卷积超分辨率重建

徐枫, 严锡君, 黄陈蓉, 郑胜男, 黄凤辰, 徐立中 1514

机织物组织图的数学描述与3维几何建模

杨允出, 李基拓, 陆国栋 1524

虚拟手术流血模拟的GPU加速实现

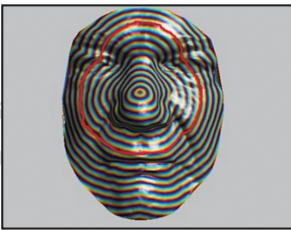
赖颢升, 向辉 1532

CONTENTS

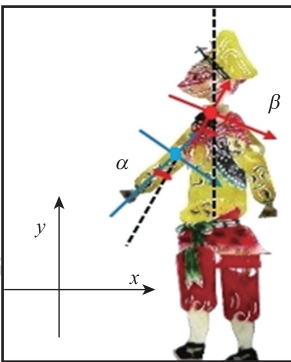
JOURNAL OF IMAGE AND GRAPHICS



Adaptive gradient reconstruction for watershed based image segmentation (P1430)



Expression-insensitive 3D face recognition method based on facial fiducial points(P1349)



Interaction and animation simulation of digital shadow play(P1490)

Review

- Depth map extraction methods in 2D-3D image/video conversion
Li Kehong, Jiang Lingmin, Gong Yongyi 1393

Image Processing and Coding

- Optimal parameterized binary modulo mapping steganography
Liao Qinan, Ke Qi, Lai Zhendan 1407
- Noise detection and image de-noising based on fractional calculus
Yang Zhuzhong, Zhou Jiliu, Lang Fangnian 1418

Image Analysis and Recognition

- Adaptive gradient reconstruction for watershed based image segmentation
Zhang Zhiyu, Meng Linghui, Lei Tao 1430
- Robust spatial factor based fuzzy clustering for image segmentation
Liu Jinyao, Ji Zexuan 1438
- Accurate perimeter estimation of target boundary based on gray-level information
Wu Qin, Zhou Qi, Liang Jiuzhen 1449
- Expression-insensitive 3D face recognition method based on facial fiducial points
Li Yanchun, Da Feipeng 1459

Image Understanding and Computer Vision

- Three dimensional curve reconstruction under the L-infinity norm of curvatures
He Leiying, Wu Chuanyu, Du Xiaoqiang 1468
- Optical flow estimation method under the condition of illumination change
Liu Jun, Zu Jing, Zhang Yu, Zhang Hongyan 1475
- Near real time linear stereo cost aggregation on GPU
Chen Bin, Chen Heping, Li Xiaohui 1481

Computer Graphics

- Interaction and animation simulation of digital shadow play
Chen Xuan, Zhang Mingmin, Pan Zhigeng 1490

Remote Sensing Image Processing

- Automatic reconstruction of large area 3D terrain from aerial imagery
Yang Ahua, Li Xuejun, Xie Jianwei, Li Dongyue 1500
- Multi-scale approach based on structure similarity for change detection in SAR images
Cui Ying, Xiong Boli, Jiang Yongmei, Kuang Gangyao 1507

Column of CIDE'2014

- Normalized convolution based super-resolution reconstruction using feature-driven prior
Xu Feng, Yan Xijun, Huang Chenrong, Zheng Shengnan, Huang Fengchen, Xu Lizhong 1514
- Mathematical expression and 3D geometrical modeling of weave diagram
Yang Yunchu, Li Jituo, Lu Guodong 1524
- Bleeding simulation of virtual surgery implemented on GPU
Lai Haosheng, Xiang Hui 1532

中图法分类号: TP391.9 文献标识码: A 文章编号: 1006-8961(2014)10-1532-07

论文引用格式: Lai H S, Xiang H. Bleeding simulation of virtual surgery implemented on gpu[J]. Journal of Image and Graphics, 2014, 19(10): 1532-1538. [赖颢升, 向辉. 虚拟手术流血模拟的 GPU 加速实现[J]. 中国图象图形学报, 2014, 19(10): 1532-1538.] [DOI: 10. 11834/jig. 20141016]

虚拟手术流血模拟的 GPU 加速实现

赖颢升, 向辉

山东大学计算机科学与技术学院, 济南 250100

摘要: 目的 流血效果是虚拟手术模拟器视觉效果的重要组成部分, 血流与固体交互的庞大计算量使得实时的流血模拟效果具有很大的挑战性。提出一种基于图形处理单元(GPU)加速的虚拟手术流血效果模拟方法。**方法** 该方法以 Müller 等人提出的光滑粒子动力学(SPH)作为基础, 采用温度项使粒子具有不同速度模拟血流形成的血槽, 同时基于构建均匀空间网格的思想, 利用通用并行计算架构(CUDA)多线程并行加速技术完成粒子控制方程的求解和血流与固体交互的计算, 从而取得实时的效果。**结果** 实验结果表明, 本文方法能够满足虚拟手术中切割表面流血和血液在器官中流动的模拟需求, 在粒子个数为 9 000 时仅需 20 ms, 相比于纯 CPU 的实现取得 20.15 倍的加速比, 实现了大量粒子下的实时流血模拟。**结论** 本文方法具有较好的灵活性和实时性的特点, 可以应用于虚拟手术仿真系统之中。

关键词: 虚拟手术; 流血模拟; 光滑粒子动力学(SPH); CUDA 加速; 温度项

Bleeding simulation of virtual surgery implemented on GPU

Lai Haosheng, Xiang Hui

College of Computer Science and Technology, Shandong University, Jinan 250100, China

Abstract: **Objective** Bleeding simulation is an important visual effect. Real-time bleeding simulation is challenging because of the extensive computations required for blood-solid interactions. This study proposes a simulation method for surgical bleeding based on graphics-processing units (GPUs). **Method** The proposed method is derived from the smooth particle hydrodynamics (SPH) proposed by Müller et al. SPH uses the temperature item to generate particles with different speeds and simulate bloodstream. The multi-thread parallel technology compute unified device architecture (CUDA), which is implemented on GPUs, is used to rapidly solve the control equation of the particle and the blood-solid interactions. Thus, this method allows real-time bleeding based on a uniformly spaced grid. **Result** The proposed method can simulate bleeding after a cut. It can also simulate blood flowing over complex obstacles in a surgical simulator. The method only takes 20 ms when the particle number is 9 000 and has 20.15 times faster computation than the central processing unit implementation. Thus, real-time bleeding simulation can be obtained with the method using a large particle number. **Conclusion** The proposed method is flexible and capable of simulating real-time surgical bleeding.

Key words: virtual surgery, bleeding simulation, smooth particle hydrodynamics, compute unified device architecture acceleration, temperature item

收稿日期: 2014-04-01; 修回日期: 2014-05-27

基金项目: 山东省优秀中青年科学家科研奖励基金项目(BS2009DX017); 山东大学自主创新基金项目(2011JC020)

第一作者简介: 赖颢升(1988—), 男, 山东大学数字媒体技术和艺术专业硕士研究生, 主要研究领域为人机交互与虚拟现实、计算机图形学。E-mail: laihaosheng13194@gmail.com

通信作者: 向辉, E-mail: hxiang@sdu.edu.cn

0 引言

虚拟手术仿真系统通过力反馈设备和虚拟技术实现了手术过程的虚拟化,为医疗从业者提供了沉浸式的训练平台,被应用于医学教育、手术计划制定和手术辅助支持中。国内外学者提出了 Spring, GiSpi, SOFA 等虚拟手术框架,通过这些框架开发人员可以迅速搭建针对特定手术过程的仿真系统。而血液和烟雾等视觉特效是虚拟手术仿真系统的重要组成部分,它能增加场景的真实感和提高系统的训练作用。

在过去的10年中,学者们针对血液模拟的研究可以分为两大类:一类是视觉模拟,主要应用于游戏或仿真系统;另一类是科学计算,主要应用于医学针对血液和血管之间的交互力分析或者进行病理推测。在虚拟手术仿真系统中常用的方法可分为以下4类:

1) 基于网格的方法。1999年 Stam^[1]提出了一个基于统一网格的无条件稳定流体模拟器。该方法将模拟空间进行离散化形成网格,在每一网格顶点求解 Navier-Stokes(以下简称 NS)方程得到下一时刻流体经过该点的状态。该方法适用于模拟空间固定且边界满足对称性的实现如血管内血液的流动,因为每一个固定点都需要经过多次迭代才能得到下一刻该点的状态,因此空间离散化的分辨率越高,需要的计算量就越大。Zátonyi 等人^[2]基于 Stam 的想法在宫腔镜模拟器中实现了2维和3维的血液融于透明流体的场景,为了达到实时的效果采用基于CPU的多线程技术进行并行加速。Rianto 等人^[3]采用曲线插值的方式来求解 NS 方程,该方法能够应用于粗糙的网格上,结合 GPU 加速在3维心脏手术模拟器中取得了实时的视觉效果。

2) 基于粒子的方法将流体离散成独立的带属性粒子。通过核近似等方法求解控制方程,得到粒子下一时刻的状态。Müller 等人^[4]采用光滑粒子动力学(SPH)方法实现了一个可交互的血流模拟器,为了让表面看起来光滑,其在 NS 方程上引入了表面张力项。Qin 等人^[5]在 SPH 方法的基础上通过使用物理处理单元(PPU)加速技术取得实时的模拟效果,并通过调节粒子的黏性系数来模拟血液的非牛顿流体的特性。Ueda 等人^[6]通过引入范德华力项来模拟血液在表面的流动。基于粒子的方法实现简

单,且适合于细节的模拟,但是随着粒子数量的增加计算量也随之增大,取得实时的效果比较困难,同时如何提取光滑的流体表面也是该方法的一个难点。

3) 基于高度值(流体体积)的方法。该方法以 Kass 等人^[7]提出的近似浅水方程作为控制方程,相比于 NS 方程,求解该方程所需的计算量大大减少。Borgeat 等人^[8]结合基于高度值的表面模型和物体的分层深度表示方法模拟了血液在表面的流动。基于高度值的方法虽然具有计算速度快的特性,并且适用于动态的边界,但是其固有的2维特性大大降低了应用价值。

4) 虽然上述方法是当前的主要模拟方法,但是仍有学者提出了一些另辟蹊径的方法来取得流血视觉效果。Oppenheimer 等人^[9]采用基于图像的方法模拟流血效果,首先在真实手术中录制视频然后从视频中提取流血的图像并进行分割处理,然后将图像绘制到流血的面。徐凯等人^[10]采用简单的元球模型模拟了少量流血在皮肤表面流动后形成的血槽。任巨等人^[11]结合粒子系统和弹性腔血液流动物理模型实现了血液在血管内的流动,相比于 NS 方程和浅水方程,该模型计算量更少。

血液模拟在虚拟手术中使用的范围比较广,因此不仅面临着模拟需求多样性的困难,而且计算方面由于涉及本身控制方程的求解和与可变形固体的交互,其庞大的计算量使其难以取得实时模拟效果。而 GPU 中的计算单元远多于控制单元,并且流体模拟中的计算过程是一个单指令多数据处理,因此学者们开始尝试使用 GPU 进行计算。Amada 等人^[12]采用 CPU 和 GPU 相结合的方式实现 SPH,获得两倍加速比;Harada 等人^[13]采用 OpenGL 和 Cg 技术将整个 SPH 在 GPU 上实现,相比于 CUDA 该实现需要比较高的技术;Goswami 等人^[14]采用 CUDA 技术模拟流固交互现象并取得了实时的效果。

采用 SPH 的方法对 NS 方程进行离散求解。通过温度来影响粒子的速度与黏度系数,模拟血液流过皮肤表面后形成的血槽,使其能同时满足切割流血和血液在器官中流动的模拟需求。同时利用 CUDA 并行加速和与 OpenGL 可交互技术提高计算与绘制速度以取得实时的效果。在本文中,重点介绍血流的建模和使用 CUDA 技术完成表面流血模拟的实现。

1 血流建模

1.1 NS 方程离散化

SPH 在计算流体动力学(CFD)中也被称为基于拉格朗日的方法,其把流体离散成粒子的集合,每个粒子都携带着浓度、压强、速度和加速度等属性,每个粒子下一时刻的状态通过对周围一定空间的粒子使用核函数加权求和得到,任意位置粒子的属性 A_i 的梯度和拉普拉斯算子的计算方程为

$$A_i(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \quad (1)$$

$$\nabla A_i(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (2)$$

$$\nabla^2 A_i(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (3)$$

式中, m_j 是粒子 j 的质量, ρ_j 是密度, $\mathbf{r}$ 、 $\mathbf{r}_j$ 是位置, $W(\mathbf{r}, h)$ 是半径为 h 的光滑核函数。

采用 Müller 的方法对 NS 中的质量守恒和动量守恒方程进行离散化,实现对血流的运动进行控制,其中包括压力、重力和黏性力。

$$a_i = \frac{F_i}{\rho_i} = \frac{f_i^{\text{pressure}} + f_i^{\text{gravity}} + f_i^{\text{viscosity}}}{\rho_i} \quad (4)$$

$$f_i^{\text{pressure}}(\mathbf{r}) = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (5)$$

$$f_i^{\text{gravity}}(\mathbf{r}) = \rho_i g \quad (6)$$

$$f_i^{\text{viscosity}}(\mathbf{r}) = \mu \sum_j m_j \frac{v_j - v_i}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (7)$$

所使用的核函数为

$$W_{\text{poly6}}(r, h) = \frac{315}{64\pi h^9} (h^2 - r^2)^3 \quad (8)$$

$$\nabla W_{\text{spiky}}(r, h) = - \frac{45}{\pi h^6} (h - r)^2 \frac{\mathbf{r}}{r} \quad (9)$$

$$\nabla^2 W_{\text{viscosity}}(r, h) = \frac{45}{\pi h^6} (h - r) \quad (10)$$

式中, μ 是黏性系数, p_i 是压强, v_i 为粒子的速度, $r = |\mathbf{r}|$, 密度 ρ_i 和压强 p_i 通过下面的方程进行计算,即

$$\rho_i(\mathbf{r}) = \sum_j m_j W(\mathbf{r} - \mathbf{r}_j, h) \quad (11)$$

$$p_i = K(\rho_i - \rho_0) \quad (12)$$

式中, K 是气体常量, ρ_0 为初始密度, 模拟中血液初始密度为 1.056 kg/m^3 。

1.2 温度项

当血流在表面进行流动时,需要与空气和表面

接触,在血小板的作用下凝结成血块形成血槽。Ueda 采用分层的思想,根据粒子到表面的距离为粒子设定不同的黏性系数,使得越靠近表面的粒子凝结得越快。但是这种方法需要计算粒子到最近表面的距离,并且血液是一种非牛顿流体,其黏度应是时间独立的与温度相关。因此本文采用将温度项加入粒子的属性中,考虑血液内部粒子之间和外部粒子与表面的温度传递。

内部粒子间温度的传递及采用 SPH 粒子近似方法进行离散化表示为

$$\frac{dT}{dt} = k^{\text{heat}} \nabla^2 T$$

$$\nabla^2 T(\mathbf{r}) = \sum_j m_j \frac{T_j - T_i}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (13)$$

式中, k^{heat} 是温度传递系数, T_i 、 T_j 是粒子的温度。

根据粒子的温度改变黏性系数以模拟血液非牛顿流体的特性。当粒子的温度低于 0°C 时,让其停止形成血块。

$$\mu = b \exp(-aT_i), a > 0, b > 0 \quad (14)$$

式中, a 、 b 是系数,在模拟中分别取值为 200 和 0.5 以满足随着温度降低黏性系数变大的要求。

1.3 吸附力

为了使血液沿着表面流动,需要构造一个表面对粒子的吸附力。Ueda 提出了基于物理的范德华力,该力与粒子到表面的距离的 7 次方成反比,当粒子距离表面越近时力越大,导致粒子容易停留在一个地方。采用 Clavet 等人^[15]所使用的计算方法,根据粒子到表面的距离构造一个垂直表面的力,当距离小于阈值 h 时该力表现为排斥力,当距离大于阈值时表现为吸引力。

$$\nabla V = -\Delta t * k^{\text{stick}} * d * (d - h) * \mathbf{n} \quad (15)$$

式中, k^{stick} 是表面的黏性系数, Δt 为模拟时间片, d 是粒子到表面的距离, $\mathbf{n}$ 是表面的法向。

随着吸附力的加入,粒子容易渗入固体之中,采用回溯将粒子投影到面片表面的方法对粒子的位置进行修正。为了满足均匀分布的要求,若粒子温度大于 0°C 时,让粒子以垂直三角面片法向的速度继续流动,若粒子温度小于或等于 0,则将粒子停止在该处。

2 CUDA 加速实现

CUDA 是基于其显卡的通用并行计算架构,该架构让开发者能够利用 GPU 资源解决科学研究等

的复杂计算问题。为了在大量粒子的条件下取得实时的流血模拟效果, 将整个 SPH 计算过程移植到 GPU 中, 采用 CUDA 多线程技术进行并行加速, 整个模拟流程如图 1 所示。

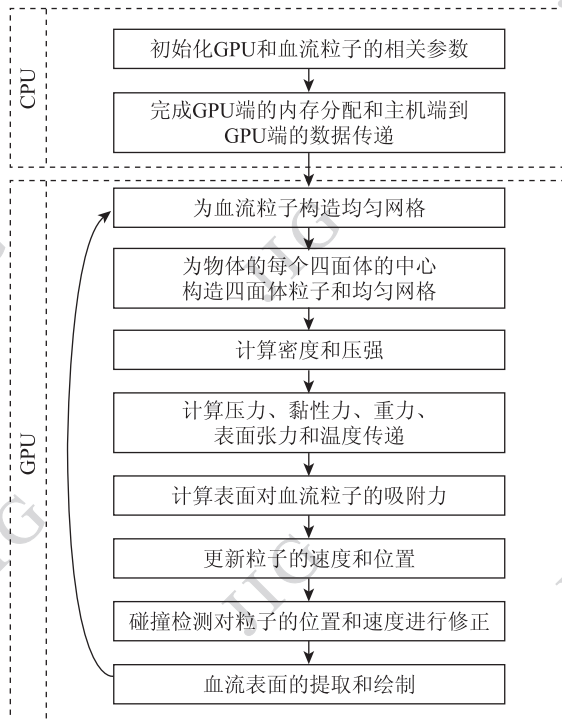


图 1 流血效果模拟流程图

Fig. 1 Flow char of bleeding simulation

2.1 邻居粒子查找

SPH 方法中主要的计算量分布在邻居粒子的查找和粒子间相互作用力的计算上。若采用暴力遍历的方法, 则整个邻居粒子查找方法的复杂度为 $O(n^2)$ 。根据核函数可知, 每个粒子仅需用到距离小于或等于 h 的邻居粒子来计算粒子相互之间的作用力。因此, 采用构建均匀空间网格的方法来加速粒子的查找, 需要在 GPU 上分配网格开始粒子位置缓存、网格结束粒子位置缓存、粒子编号缓存和粒子网格索引缓存。该计算方法实现步骤如下:

1) 按照边长 h 将整个模拟空间划分为均匀网格, 每一个网格根据其顶点的位置进行编号, 每个网格对应着一个开始粒子位置缓存和一个结束粒子位置缓存。

2) 每个粒子根据位置信息计算出所属网格的编号并存储在粒子的网格索引缓存中, 同时每个粒子有一个编号缓存。

3) 采用 CUDA 中 Thrust 库所实现的堆排序, 根

据粒子的网格索引对粒子的编号缓存进行排序, 使得在同一网格的粒子编号聚集在一起。

4) 将属于相应网格的第 1 个粒子的编号缓存位置存储于网格的开始粒子位置缓存, 最后一个粒子的编号缓存位置存储于网格的结束粒子位置缓存, 如图 2 所示。

5) 当需要查找周围粒子时, 只需查找粒子所属网格周围的 27 网格所包含的粒子, 然后计算粒子间的距离进行判断便可得到周围的粒子。

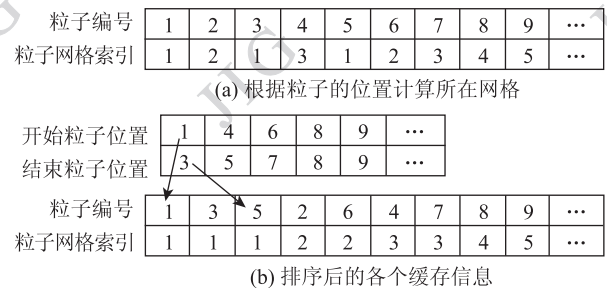


图 2 排序后缓存信息

Fig. 2 Message of buffer after sorting

2.2 计算 NS 方程和温度传递

NS 控制方程由重力、压力和黏性力 3 部分组成, 每一部分的计算都需要搜索邻居粒子。为了减少搜索次数并结合数据的相关性, 可将该计算分成两大步, 每一步在 CUDA 中采用一个 Kernel 函数进行实现。首先, 根据式 (11) (12) 计算粒子的密度和压强。其次, 根据式 (5) — (7) 和式 (13) 计算粒子的压力、重力、黏性力、表面张力^[4]和粒子间温度的传递。在计算过程中, 为每个粒子分配独立的线程, 以 128 个线程作为一组, 将相邻的粒子信息复制到共享寄存器中, 提高访问内存的速度。

2.3 计算吸附力和更新粒子的位置

计算表面对粒子的吸附力, 需要得到粒子到表面的距离。一般采用的方法是对每个表面三角面片查找周围的粒子, 根据粒子到表面的距离计算吸附力。但是在并行计算中该方法会导致多个线程同时要求对数据进行修改, 于是并行将退化串行, 降低了加速的效果。本文中物体采用四面体的方式进行表示, 为加速邻居四面体的查找, 基于构造均匀空间网格的思想将四面体信息存储为四面体粒子, 四面体粒子包含四面体的编号信息和四面体中心的位置信息, 依据邻居粒子查找的方法让血液粒子能够快速查找到周围的四面体, 根据粒子到表面的距离计算吸附力, 得到粒子的加速度后采用 Leap-Frog 模

式^[16]更新粒子的速度和位置信息。

2.4 碰撞检测

随着吸附力的加入,粒子容易在模拟时间片内渗入固体的内部,因此需要判断粒子是否在四面体内。而通过构造四面体粒子可以为血液粒子快速检索到周围的四面体,若一个粒子在四面体内部,则它处于四面体所有面的同一侧,通过该原则可以判断一个粒子是否在四面体内。而判定粒子是否在三角面片的正面则可通过计算粒子和三角面片的顶点的连线与三角面片法向的点积得到。若粒子在四面体内部则表明粒子和三角面片发生了碰撞,采用按照粒子的速度进行反向运动的回溯方法最终将其投影到四面体的外表面上,则该点就是发生碰撞的点,回溯计算的次数一般为1~2次。由于仅需与附近的四面体进行碰撞检测,大大减少了计算量。

2.5 血液的绘制

基于粒子的流体表面绘制方法有很多,本文将 Lorensen 等人^[17]提出的 Marching Cube 方法采用 CUDA 实现,以提取血液的等值面。根据周围粒子的密度,对每个包含血液粒子的网格顶点进行插值会得到该点的值,其余顶点赋值为0,然后根据等值面与网格相交得到血液表面的三角面片。为了减少主机端和 GPU 端之间的数据传递,利用 CUDA 和 OpenGL 的可交互性技术,在计算前先在显存中分配一块顶点缓存,将计算得到的血液表面存储在该顶点缓存中,然后结合着色器技术直接将血液表面在 GPU 上进行绘制,大大提高了绘制的速度,以得到实时的模拟效果。

3 实验结果

程序是在 fedora 系统下结合切割仿真系统进行开发与测试,CUDA 的版本为 4.0,PC 的配置如下: Intel 奔腾双核 E6500 2.93 GHz CPU、8 GB 内存、NVIDIA GeForce GTS450 显卡。

切割模型是由 7 986 个四面体和 5 184 个顶点组成,切割仿真系统采用 CPU 进行实现,模型信息从主机端到 GPU 端的数据传递平均需要 6 ms, Marching Cube 网格的分辨率为 40×40×40。

采用构造均匀空间网格的方法,加速邻居粒子和邻居四面体的查找,将查找量从 $O(n)$ 降低为 $27k$ (k 为每个网格平均包含的粒子数)。相比于 Wen 等人^[18]采用的体纹理碰撞检测方法,构造四面体粒

子的方法更容易处理一个网格内出现多个三角面片的情况,其使用 GPU 进行加速在粒子个数为 6 000 时每帧需要 51.45 ms,而本文仅需 17 ms。Qin 等人^[19]使用 CPU 模拟血液在血管中流动,当粒子数为 5 000 时,取得了 18.2 帧/s,而本文加速后可取得 66.7 帧/s。表 1 展示了结合切割后模型伤口大小分别进行粒子数为 300,2 000,4 700,9 000 的流血模拟采用 CPU 和 GPU 所需要的时间及加速比信息。实验结果表明,虽然从 CPU 端到 GPU 端的数据复制需要占用很大的时间,但是采用 CUDA 加速技术后计算速度得到明显的提高,取得良好的加速比。

表 1 采用 CPU 和 GPU 的模拟时间
Table 1 Simulation time using CPU and GPU

粒子个数	CPU/ms	GPU/ms	加速比
300	34	10	3.40
2 000	108	11	9.82
4 700	220	15	14.67
9 000	403	20	20.15

相比于 Basdogan 等人^[20]提出的使用两种不同的算法来实现血液在器官表面和血管内的流动,本文方法具有良好的灵活性,在引入温度项后只需通过简单设置外部温度便可满足 2 维和 3 维下不同的模拟需求如图 3 和图 4 所示。图 3 切割流血模拟中

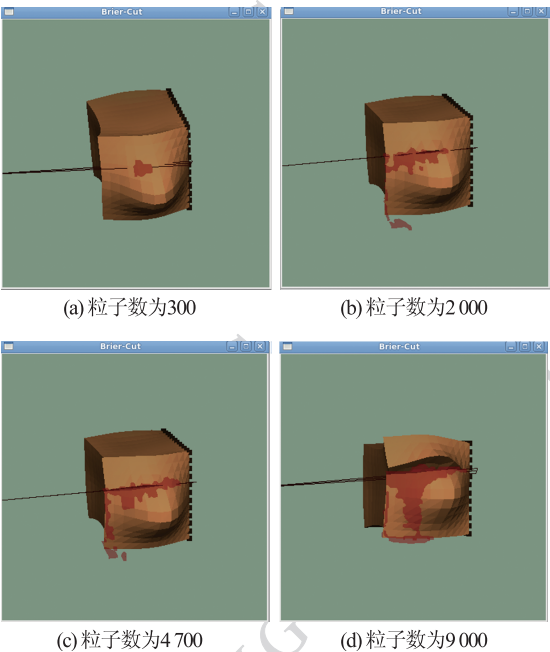


图 3 结合伤口大小不同粒子下的模拟效果

Fig. 3 Visual effect of bleeding simulation with different particles

将外部温度设为 0, 在温度项的影响下可以看到血液在表面形成的血槽。图 4 血液在器官中流动模拟则将温度设为 100, 血液只有流动没有血槽的效果。图 5 模拟了血管爆裂后的流血效果。由于本文仅采用了 OpenGL 纹理映射技术进行顶点着色, 因此模拟结果真实感一般, 对比其他模拟结果没有明显提高。

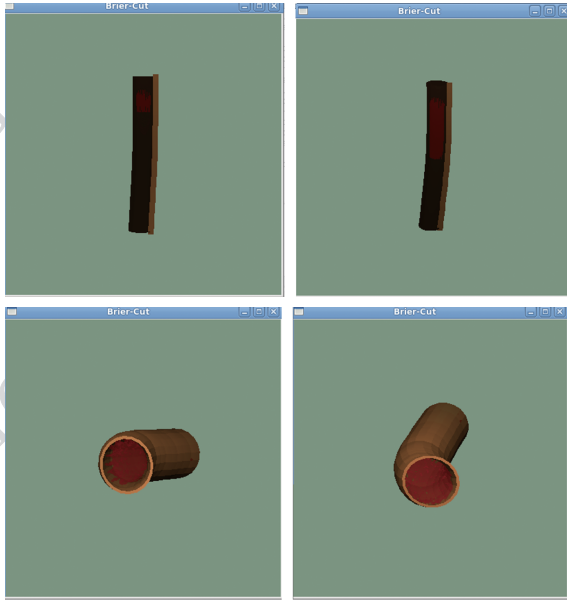


图 4 血液在血管中流动

Fig. 4 Blood flowing over vessel



图 5 血管爆裂后的流血效果

Fig. 5 Bleeding simulation of cracking blood vessel

4 结 论

采用 SPH 方法对血液进行建模并采用 CUDA 加速实现, 展示了与切割仿真系统结合的表面流血和血液在血管内流动的效果。实验结果表明, 引入温度项后可以真实的模拟血液在皮肤表面形成的血槽, 并具有良好的灵活性满足不同的模拟需求, 而基于 GPU 的加速技术相比于 CPU 的实现大大提高了计算速度, 满足了虚拟手术中实时性的要求, 可应用于虚拟手术仿真系统中。未来的工作是: 1) 采用异步方式减少从主机端到 GPU 端数据传递的等待时间; 2) 实现基于 GPU 的具有真实感的渲染算法, 增强模拟真实感; 3) 考虑血液与血管的交互作用。

参考文献 (References)

- [1] Stam J. Stable fluids[C]//Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. New York, USA: Addison-Wesley Publishing Co., 1999: 121-128. [DOI:10.1145/311535.311548]
- [2] Zátanyi J, Paget R, Székely G, et al. Real-time synthesis of bleeding for virtual hysteroscopy[J]. Medical Image Analysis, 2005, 9(3): 255-266. [DOI:10.1016/j.media.2004.11.008]
- [3] Rianto S, Li L. Fluid dynamic visualisations of cuttings-bleeding for virtual reality heart beating surgery simulation[C]//Proceedings of the 33rd Australasian Conference on Computer Science. Darlinghurst, Australia: Australian Computer Society, 2010, 102: 53-60.
- [4] Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications[C]//Proceedings of the 2003 ACM SIGGRAPH, Switzerland Switzerland: Eurographics Association Aire-la-Ville, 2003: 154-159.
- [5] Qin J, Pang W M, Chui Y P, et al. Hardware-accelerated bleeding simulation for virtual surgery[C]//Proceedings of Computational Biomechanics for Medicine II Workshop. Brisbane, Australia: MICCAT, 2007, 133-140.
- [6] Ueda K, Fujishiro I. Visual simulation of bleeding on skin surface[C]//Proceedings of ACM SIGGRAPH New York, USA: ACM, 2011: 9. [DOI:10.1145/2037715.2037725]
- [7] Kass M, Miller G. Rapid, stable fluid dynamics for computer graphics[J]. ACM SIGGRAPH Computer Graphics, 1990, 24(4): 49-57. [DOI:10.1145/97880.97884]
- [8] Borgeat L, Massicotte P, Poirier G, et al. Layered surface fluid simulation for surgical training[M]//Medical Image Computing and Computer-Assisted Intervention. Berlin Heidelberg: Springer, 2011: 323-330. [DOI:10.1007/978-3-642-23623-5_41]
- [9] Oppenheimer P, Gupta A, Weghorst S, et al. The representation

- of blood flow in endourologic surgical simulations[J]. *Medicine Meets Virtual Reality 2001: Outer Space, Inner Space, Virtual Space*, 2001, 81: 365-371. [DOI:10.3233/978-1-60750-925-7-365]
- [10] Xu K, Xiong Y S, Tan K, et al. A model of bended bloodstream with small amount bleeding[J]. *Journal of National University of Defense Technology*, 2004, 26(5): 70-73. [徐凯, 熊岳山, 谭珂, 等. 一种小量流血形成的弯曲血槽模型[J]. *国防科技大学学报*, 2004, 26(5): 70-73.] [DOI: 10.3969/j.issn.1001-2486.2004.05.015]
- [11] Ren J, Xiong Y S, Tan K, et al. Simulation of blood flow in virtual system of cardiac intervention[J]. *Journal of National University of Defense Technology*, 2006, 28(2): 24-26. [任巨, 熊岳山, 谭珂, 等. 虚拟心脏介入手术系统的血流模拟[J]. *国防科技大学学报*, 2006, 28(2): 24-26.] [DOI: 10.3969/j.issn.1001-2486.2006.02.014]
- [12] Amada T, Imura M, Yasumuro Y, et al. Particle-based fluid simulation on gpu[C]//*Proceedings of ACM Workshop on General-Purpose Computing on Graphics Processors and SIGGRAPH*. New York, USA: ACM, 2004.
- [13] Harada T, Koshizuka S, Kawaguchi Y. Smoothed particle hydrodynamics on GPUs[C]//*Proceedings of Computer Graphics International. Petropolis: SBC*, 2007: 63-70.
- [14] Goswami P, Schlegel P, Solenthaler B, et al. Interactive SPH simulation and rendering on the GPU[C]//*Proceedings of the 2010 ACM SIGGRAPH*. Switzerland, Switzerland: Eurographics Association Aire-la-Ville, 2010: 55-64. [DOI:10.2312/SCA/SCA10/055-064]
- [15] Clavet S, Beaudoin P, Poulin P. Particle-based viscoelastic fluid simulation[C]//*Proceedings of the 2005 ACM SIGGRAPH*. New York, USA: ACM, 2005: 219-228. [DOI:10.1145/1073368.1073400]
- [16] Pozrikidis C. *Numerical Computation in Science and Engineering* [M]. New York: Oxford University Press, 1998.
- [17] Lorensen W E, Cline H E. Marching cubes: a high resolution 3D surface construction algorithm[C]//*Proceedings of ACM SIGGRAPH Computer Graphics*. New York, USA: ACM, 1987, 21(4): 163-169. [DOI:10.1145/37402.37422]
- [18] Wen C J, Ou J W, Jia J Y, et al. GPGPU-based smoothed particle hydrodynamic fluid simulation[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2010, 22(3): 406-411. [温婵娟, 欧嘉蔚, 贾金原, 等. GPU 通用计算平台上的 SPH 流体模拟[J]. *计算机辅助设计与图形学学报*, 2010, 22(3): 406-411.]
- [19] Qin J, Pang W M, Nguyen B P, et al. Particle-based simulation of blood flow and vessel wall interactions in virtual surgery[C]//*Symposium on Information and Communication Technology*. New York, USA: ACM, 2010: 128-133. [DOI:10.1145/1852611.1852636]
- [20] Basdogan C, Ho C H, Srinivasan M A. Simulation of tissue cutting and bleeding for laparoscopic surgery using auxiliary surfaces[J]. *Studies in Health Technology and Informatics*, 1999: 38-44. [DOI:10.3233/978-1-60750-906-6-38]