

油水两相流系统可视化应用研究

靖 固, 李 丹

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘 要: 油水两相流在密封管道中的流动现象是不可见的, 将其真实流动现象可视化将为科学发现和控制决策提供非常有价值的信息. 采用 RealFlow 模拟两相流液体流动, 应用并行双曲面函数 Multiquadric 插值方法对检测数据图像重构后的稀疏矩阵插值以加强对图像细节的显示, 并结合 C4D 软件中的灯光和材质对两相流系统液态流动效果进行制作. 结果表明应用改进的径向基插值算法能解决检测数据稀疏的困扰; 将 RealFlow 和 C4D 平台交互应用能真实有效的模拟两相流四种流型的连续动态过程, 其结果真实自然.

关键词: 油水两相流; 可视化; RealFlow 技术; 并行插值算法

DOI: 10.15938/j.jhust.2015.02.013

中图分类号: TP391.41 **文献标志码:** A **文章编号:** 1007-2683(2015)02-0069-06

The Application and Research of Visualization for Oil-water Two-phase System

JING Gu, LI Dan

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: The phenomenon of oil-water two-phase flow in a sealed pipeline is not visible. The visualization of the real flow phenomenon will provide very valuable information for the scientific discovery and control decisions. In this paper, simulated two-phase fluid flow using RealFlow, applied of parallel hyperboloid function Multiquadric interpolation method for the test data image after reconstruction of sparse matrix interpolation to enhance image detail display, and created the liquid flow effect of two-phase system, combined with lighting and materials of C4D software. The results show that the application of improved radial basis function interpolation algorithm can solve the test data sparse problems. RealFlow and C4D platform interactive applications can effectively simulate the real two-phase flow pattern of four kinds of continuous dynamic process, and the result is real and nature.

Keywords: oil-water two-phase; visualization; RealFlow technology; parallel interpolation algorithm

0 引 言

在石油开采过程中由于地层中含有一定量的水并挟带天然气,使得石油管道形成油气水三相流,但

随着油井开采深度的增加,天然气含量逐渐减少,而水相的含量逐渐增加,更多以油水两相流为主^[1]. 事实上,国内的许多油田已进入开采中后期,2011年全国油田平均含水量超过85%,部分油田水相含量甚至高达98%.这种流动现象在密封管道中是不

收稿日期: 2014-04-26

基金项目: 黑龙江省自然科学基金(F222805); 黑龙江省教育厅科学技术研究项目(12531107).

作者简介: 靖 固(1961—),女,博士,教授,E-mail: jinggu@hrbust.edu.cn;

李 丹(1987—),女,硕士研究生.

可见的,将其真实流动现象可视化将为科学发现和控制决策提供非常有价值的信息^[2-3]. 本文应用 RealFlow 技术模拟两相流中四种典型流型的流动过程,利用脚本程序控制粒子模拟液体的流动;同时结合径向基插值算法,实现并行双曲面函数 Multiquadric 插值设计,加强图像细节显示,计算出正确的运动轨迹;再以质点与质点间产生平滑的多边形网格,保存序列文件,然后在 Cinema 4D (简称 C4D) 中引入网格,赋予材质,渲染输出,实现两相流系统可视化仿真.

1 RealFlow 两相流管道流动系统设计

流体包括液体和气体,其流动过程存在着复杂的相互作用关系,比如流体的混合交互、对流与扩散、紊流、表面张力等,以往的可视化技术很难表现出流体的真实流动效果^[4]. RealFlow 是由西班牙 Next Limit 公司出品的流体动力学模拟软件,它是一款独立的模拟软件,可以计算真实世界中液体的运动,是目前唯一具有对物体实施作用的力学模拟系统^[5]. 它的粒子具有动能,能够推动其它物体,使它们产生运动或改变运动方式. 在相同硬件条件下,RealFlow 能处理的粒子数是 3ds max 或 Maya 等 3D 平台所能处理的粒子数的几十倍乃至更多. 利用脚本程序还可以精确控制粒子、物体的运动,它已成为使用者创造流体真实效果的强大工具. 但 RealFlow 也存在缺陷,它没有完整的建模功能且没有提供任何照明和渲染功能,模拟时需结合外部建模软件. 本文将 RealFlow 结合具有高效建模和快速渲染功能的 C4D 软件完成两相流可视化仿真.

应用 RealFlow 建立两相流管道流动系统的制作思路是将 C4D 软件中制作的石油管道模型导入 RealFlow 来设置流体模型,模拟和调节完成后,再将粒子或网格物体从 RealFlow 导回 C4D 软件中进行照明和渲染. RealFlow 模拟两相流实现过程如图 1 所示:

1.1 石油管道三维模型的导入

在 C4D 中制作场景,可以保存两种 RealFlow 接受的格式,不动的模型可以保存为 Obj 格式,动画的模型可以保存为 SD 格式,视需要而定. 本文利用 C4D 软件制作出石油管道模型,由于 RealFlow 只能识别三角形的网格物体,将其简化的模型转换为可编辑面片,变为三角网格物体,最终导出为 Obj 格式文件. 图 2 为从 C4D 导入 RealFlow 中管道的三维模型.

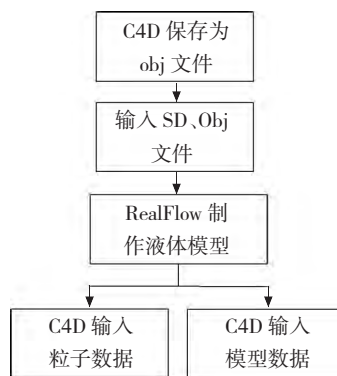


图1 两相流管道流动系统的制作流程图

1.2 建立 RealFlow 两相流管道流动模型

1.2.1 脚本程序设计

Python 脚本是 RealFlow 的重要组成部分,是扩展 RealFlow 功能和开发自定义解决方案的一种很好的工具. 在 Python 脚本中有 RealFlow 的响应模块和功能扩展包,可以实现 RealFlow 的操纵发射器、创建自定义力场,修改刚体力学参数、使用 RealWave 曲面并控制软体动力学功能等指令集操作.

RealFlow 可以根据图片控制发射器和动力学参数等,本系统利用 Python 脚本程序将插值后图像的像素信息传递给发射器,通过 Python 可以直接访问每一个单独的粒子并设置属性、设定运动方式、受力情况及动画等的设置,完成四种流型的模拟. 以点滴流型为例,在定义图片路径后根据矩阵中有效部分的点阵控制发射器的粒子,定义两个粒子发射器. 其中一个作为水相的发射器,发射器的初始状态设置为全水,然后初始化再重置,同时把溢流出来的粒子灭杀、体积灭杀 (age),提高流动速度 (speed),将 speed 设置为 2,让流体做矢量方向的运动;另一个发射器根据矩阵中像素“1”的位置设置油滴发射器并定义为间隔发射. 下面的程序是将水相发射器转移粒子到油滴发射器的操作程序,每个粒子有特定的 ID,当遍历到这个粒子时,ID 根据像素信息区分是水粒子 (water) 还是油滴粒子 (oil),确定两个发射器属性后根据管道中油水两相物理属性设置内外压力、密度、粘滞性等动力学参数及对粒子运动属性、发射时间等并进行调整,最终形成点滴流流动模型.

系统部分脚本程序设计如下:

```

...
water = scence.getEmitter( "Circle01" )
oil = scence.getEmitter( "Circle02" )
particle = water.getFirstParticle( )
while( particle ):

```

```

vel = particle.getVelocity()
pos = particle.getPosition()
id = particle.getId()
if( pixe_val = 1)
oil. addparticle( pos,vel)
water. removeParticle. getNextPartele()
...

```

1.2.2 Mesh 网格模型的生成

在调整粒子达到最终效果后,创建网格模型对粒子进行包裹.在模拟管道与液体接触的面与两相液体表面之间的角度尽可能与真实的场景相似,就需要对 Mesh 网格进行调整.通过对网格各项参数进行调试,最终达到真实的液体流动的效果.生成的网格序列帧在项目文件夹的 Meshs 目录下,文件格式为. bin. 然后将 bin 文件通过接口插件导入到 C4D 中完成灯光材质的制作和最终的渲染输出,完成系统仿真.本系统点滴流型将 Mesh 网格模型导入 C4D 中如图 3 所示.

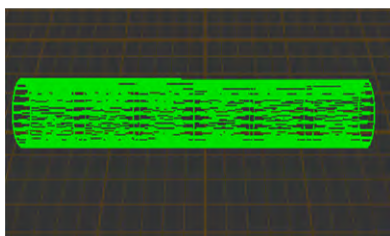


图2 三维模型

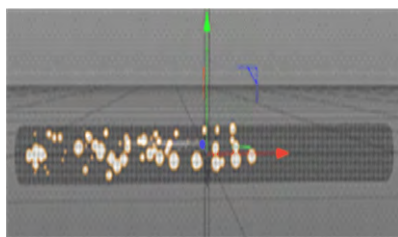


图3 网格模型

1.2.3 两相流可视化数据

系统可视化数据来源于图像重构后的检测现场电容传感器的检测数据,经迭代的代数重建法 ART 最后得到两相流 4 种典型像素点的灰度值矩阵^[8],包括点滴流、环流、中心流、层流.在图 4 表示的 4 种典型流型的矩阵中横断面用 100 个像素点来表示,其中“0”表示水,“1”表示油,两相流管道流动系统设计就是基于这四种流型的二维图像矩阵.根据给定矩阵中有效部分的点阵就可以确定流体的具体形状,用以控制脚本程序中粒子发射器的设置.

0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0

(a)点滴流

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	0
0	1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	0	0	1	1
1	1	1	1	0	0	0	1	1	1
0	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(b)环流

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(c)中心流

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

(d)层流

图4 检测数据重构后的四种流型的二维图像矩阵

2 并行双曲面 Multiquadric 插值算法

由于重构的图像数据稀疏,不能真实反映两相流系统连续的物理流动过程,必须对其进行插值运算增加数据密度以提高三维图像的细节显示.插值方法及插值基函数的选择是可视化技术的一个关键问题,因此选择适合本系统的插值方法就显得尤为重要.

径向基函数(radial basis functions, RBF)是以每个插值中心为径向对称基函数线性组合的插值函数, RBF 被认为是散乱数据插值中最精确和最稳定的方法之一^[9],而散乱数据特点正符合两相流系统.常用的 RBF 有 Gaussians, Thinplate splines, Multiquadric 及 Inverse Multiquadric 等^[8],其中 Multiquadric 最初由 Hardy 在 1971 年提出的,1990 年 MADYCH 等.描述了一类通用的 Multiquadric 函数,使用双曲面函数作为径向基函数的最大特点就是计算结果与原始形式相近^[8],这正符合本系统数据稀疏,需提高三维图像的细节显示的要求.

2.1 插值原理

二维双曲面 Multiquadric 函数公式为:

$$H(x_j, y_j) = \sum_{i=1}^n a_i Q_i(x_j, y_j). \quad (1)$$

式中: $H(x_j, y_j)$ 为空间变量场; $Q_i(x_j, y_j)$ 为径向基函数; a_i 为权重因子,由检测值决定或者以某种方式设定.

$$Q_i(x_j, y_j) = -\sqrt{\frac{|x_j - x_i|^2 + |y_j - y_i|^2}{c^2} + 1.0}. \quad (2)$$

式(1)也可以用矩阵形式表示:

$$H_j = Q_{ij} a_i, \quad (3)$$

对式(3)进行矩阵变换得到:

$$a_i = Q_{ij}^{-1} H_j, \quad (4)$$

从而求得系数 a_i . 那么任意一点 (x_g, y_g) 可由下式插值得到:

$$H_g = Q_{gi} Q_{ij}^{-1}, \quad (5)$$

该插值函数的准确率完全由原始检测点分布密度决定,即由系数 a_i 来决定.

由式(2)知 Multiquadric 函数建立的插值方程组的系数矩阵为非奇异的,解此插值方程组可采用 LDL^T 分解,也称为改进 Cholesky 分解法. 此方法将系数矩阵分解为单位下三角矩阵 L 、对角矩阵 D 和矩阵 L^T 的乘积再求解方程组,这样可减少存储量和计算量,提高算法效率.

2.2 并行算法设计

2.2.1 待插值点及插值参考点的选择

对于径向基插值算法,找到所有待插值点求解插值方程组是最直接的方法,理论上这种方法是最精确的,但在实际应用中,当散乱点数量增多时,求解系数的线性方程组不断扩大,插值几乎无法实现. 事实上,径向基插值方法具有某种屏蔽性质,即距离较远的数据点的影响极小^[11]. 本文采用与双三次线性插值类似的方法选择插值参考点,即选取待插值点周围 $4 \times 4 = 16$ 个整数点作为插值参考点,待插值点的选择如图 5(a) 所示. 其中 $\circ$ 表示插值参考点, $\bullet$ 表示待插值点.

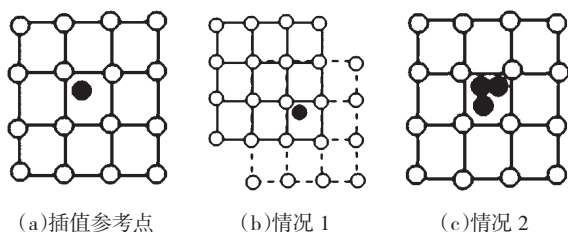


图5 待插值点及插值参考点的选择

需要注意两种情况,若待插值点到矩阵某个或某两个边缘太近而无法取到 4×4 个点,则调整 4×4 个点的插值参考点不得超出矩阵范围,图 5(b) 为待插值点在矩阵右下角很近的情况,虚线部分位于矩阵边界以外. 此时需向上、向左各移动 1 格,取实线构成 4×4 个方格的顶点为插值参考点;插值程序按照由左到右、由上到下的顺序依次进行计算,相邻的待插值像素多数情况下为图 5(c) 所示 3 个待插

值点的方程组系数矩阵相同,只需做一次 LDL^T 分解,这样可以节省大量时间.

2.2.2 并行插值算法设计

由于当前微机系统的主流配置都配备了双核、四核甚至多核 CPU,在体系结构方面已具备了实现多核多线程并行计算的硬件条件,且这种内核模式具有共享存储空间的结构特点^[11]. 如果将所求应用问题采取一定的分配策略将计算任务分配给多核 CPU,多线程同时并行求解,可大大提高计算效率. 本算法应用计算机并行自动选择待插值和插值参考点并求解插值方程组. 插值算法的实现步骤如下:

- 1) 读入原图像矩阵,得到图像的高度和宽度 (10×10);
- 2) 根据插值算法得到新图像的高度和宽度 ($h \times w$),并创建空白新图像;
- 3) 插值得到像素点 (x, y) 的值并写回新图像,求取待插值像素点的过程如图 6 所示;
- 4) 得到插值后图像矩阵.

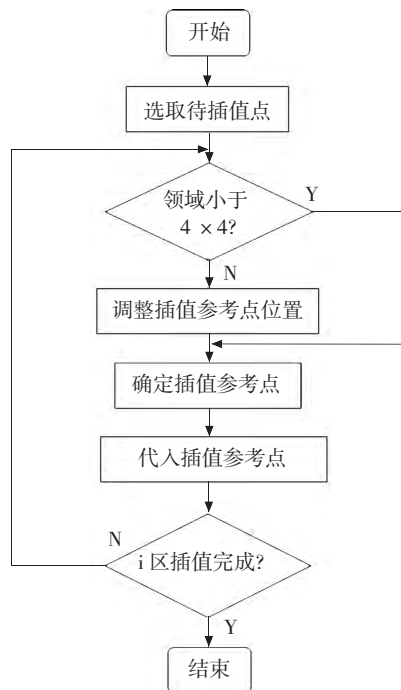


图6 求取待插值点过程

其中第 3 步是设计该算法最为关键的环节,即径向基插值部分,运算量大,其快慢直接关系到插值的速度. 插值时待插值点的值只由输入像素的邻域像素确定,后一像素输出结果不依赖于前面像素的输出结果,而且原图像与插值后的图像分别利用两个数组进行存储,因此插值过程以及所利用的像素

资源之间不存在关联性,不同的输出值可以独立进行计算互不干涉,具有良好的并行性. 本算法插值任务被划分成多个子任务,线程组中每个线程分别执行对应的子任务,各待插点同时插入对应的数据区域,并行区结束后派生的线程结束,主线程继续执行,并行算法流程图如图7所示.

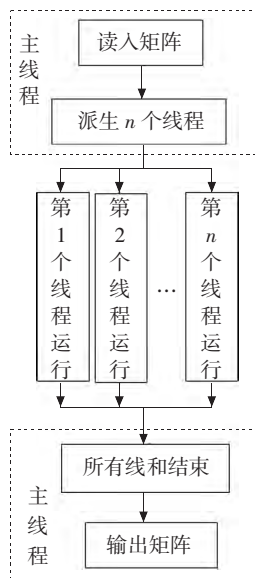
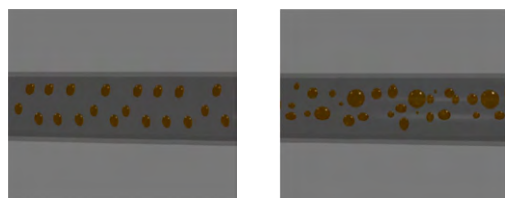


图7 并行算法流程图

2.3 算法应用实例

根据4种流型的灰度值矩阵设图像每一像素的宽度为1,则 $f(x, y)$ 表示在以左上为原点的右手二维坐标系上像素 (x, y) 的值,其中 x, y 为正整数或0,则得到若干个 (x_i, y_j) ,点滴流型经本文方法得到插值后效果如图8所示. 插值后油水两相符合实际工业现场离散相实体的几何形状、相对位置和空间分布等三维信息的状态,明显改善了插值前成像效果.



(a)点滴流插值前效果 (b)点滴流插值后效果

图8 插值前后对比

为了验证本文算法的有效性,将不同像素矩阵分别应用普通 Multiquadric (简称 MQ) 插值算法和本文并行插值算法,以加速比作为衡量并行计算性能的指标,实验数据如表1所示.

表1 普通和并行插值算法实验数据

图像矩阵/pixel	普通 MQ/ms	本文算法/ms	加速比
10 × 10	2.02	1.98	1.02
20 × 20	2.94	2.65	1.10
40 × 40	3.71	3.09	1.21

从表1可以看出,通过并行化处理能缩短插值计算时间,提高算法的有效性,解决径向基函数求取速度较慢的问题. 并且可以看出随着图像矩阵的增大,加速效果明显,并行计算的优势可更加明显地表现出来.

3 可视化结果分析

图9为 RealFlow 结合 C4D 得到的可视化仿真结果截图;图10为文中用 3ds max 结构建模得到的可视化仿真结果截图.

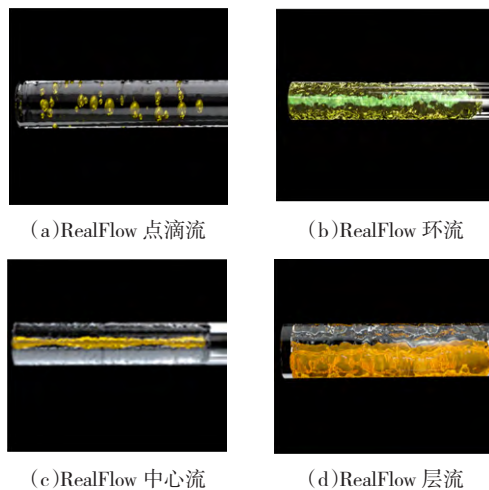


图9 两相流系统仿真结果截图

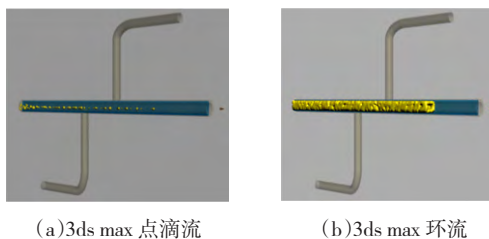


图10 3ds max 两相流系统仿真结果截图

从上面的仿真结果对比之下,可以看出使用 3ds max 中 MAXScript 脚本语言进行结构建模只表现了油相运动,没有油相与水相的交互运动,且粒子的特效仍与真实流体差距较大给人感觉生硬. RealFlow 结合 C4D 作为实验平台,RealFlow 流体力学模拟技术较好的解决了粒子碰撞计算与逼真度平衡的

问题,仿真结果突出两相液体交互运动更真实生动,流体感更强.

4 结 论

本文研究的两相流系统可视化是以重构的二维灰度值矩阵为基础,通过三维图像重建,将检测到的现场数据进行可视化处理,再现两相流流型变化的连续流动过程.与以往采用 3ds max 制作相比,将 RealFlow 与 C4D 平台结合应用在流体可视化中具有真实生动更流体化的优点,这为可视化系统开发中流体运动的模拟提供了一种较好的解决方法;通过采用并行双曲面函数 Multiquadric 插值算法,明显改善了两相流可视化系统的成像质量,加强了流型的细节显示,有效地解决了稀疏数据可视化问题,并利用计算机多核优势,实现了算法的加速处理.但研究成果真正用到检测勘探应用系统,还需进一步提速,以满足应用系统的实时性要求.

参 考 文 献:

- [1] SELAMET O F, PASAOGULLARI U, SPERNJAK D, et al. Two-phase Flow In a Proton Exchange Membrane Electrolyzer Visualized in Situ by Simultaneous Neutron Radiography and Optical Imaging [J]. International Journal of Hydrogen Energy, 2013, 38 (14): 5823 - 5835.
- [2] 杜萌,金宁德,高忠科,等.油水两相流水包油流型多尺度排列熵分析 [J].物理学报,2012,23:113-121.
- [3] 龙勇,袁静,康凤举,等.可视化仿真中三维建模策略研究 [J].系统仿真学报,2011,12:2682-2687.
- [4] 柳有权,王章野,朱鉴,等.基于物理的流体动画加速技术的研究进展 [J].计算机辅助设计与图形学学报,2013,3:312-321.
- [5] 王树杰,徐超,袁鹏,等.流体真实性仿真的国内外研究进展 [J].系统仿真学报,2010,2:280-286.
- [6] 靖固,宋扬,于晓洋.基于 3DMAX 结构建模的两相流系统可视化应用研究 [J].哈尔滨理工大学学报,2011,4:67-72.
- [7] CHEN T L, GEMAN S. Image Warping Using Radial Basis Functions [J]. Journal of Applied Statistics, 2014, 41 (2): 242 - 258.
- [8] 邹友龙,胡法龙,周灿灿,等.径向基函数插值方法分析 [J]. Applied Geophysics, 2013, 4: 397 - 410, 511.
- [9] ZOU Y L, HU F L, ZHOU C C, et al. Analysis of Radial Basis function Interpolation Approach [J]. Applied Geophysics, 2013, 10 (4): 397 - 410.
- [10] 方林聪,汪国昭.基于径向基函数的曲面重建算法 [J].浙江大学学报:工学版,2010,4:728-731.
- [11] 陆俊华,刘慧娜,王建鑫,等.多核多线程技术综述 [J].计算机应用,2013 (S1): 239 - 242.

(编辑:关毅)