# 基于连续模型和动力学仿真模型的 高密度人群仿真算法

### 孙立博 孙晓峰 秦文虎

(东南大学仪器科学与工程学院 南京 210096)

**摘 要**为了真实模拟高密度下的人群运动,文中提出了一种将人群运动连续模型和人群动力学仿真模型相结合的高密度人群仿真算法.该方法首先将人群假设为连续的流体,并进行密度场的转化;然后通过计算最大速度场和单位消耗场,建立全局的动态势能场,并根据势能场的梯度得到期望的全局速度场;接着利用有限压缩性规则对人群的密度进行限制,通过高密度环境下产生的压力项校正速度场,从而插值更新人群位置,并对更新后的人群做最小距离限制,实现网格内部个体之间的避碰;最后引入 GPU 加速算法实现高密度人群快速仿真.仿真结果表明,基于连续模型和动力学仿真模型的高密度人群仿真算法能够逼真地模拟向同一目标或指定目标运动、躲避动静态障碍物、人群交汇等多种场景下的人群运动,这对现实生活中交通疏散方案的制定具有一定的指导意义.

关键词 虚拟人群仿真;连续模型;有限压缩性规则;GPU加速 中图法分类号 TP391 **DOI**号 10.11897/SP.J.1016.2016.01375

# Dense Crowd Simulation Based on Continuum Model and Aggregate Dynamics Model

SUN Li-Bo SUN Xiao-Feng QIN Wen-Hu

(School of Instrument Science and Engineering, Southeast University, Nanjing 210096)

**Abstract** We present a novel approach for simulating crowd motions in high density realistically based on continuum model as well as aggregate dynamics model. In our approach, we firstly consider the crowd as continuum flow and splat the crowd onto a density grid with density conversion function. In a second stage, the speed field and unit cost field are computed to construct the dynamic potential field so that we can compute the velocity field according to the gradient of the potential field. Next, unilateral incompressibility constraint is integrated to constrain crowd density and the velocity field is corrected through the nonnegative pressure imposed by dense crowd. We update each person's position by interpolating the velocity field and enforce a pair-wise minimum distance between the people to ensure no two people intersect in the same grid cell. Finally, GPU accelerating algorithm is applied to simulate dense crowd at interactive rates. Experimental results show that our approach can well simulate the crowd motions in various scenarios such as moving towards the same or assigned target, avoiding the collisions with dynamic or static obstacles, intersecting with each other, which has certain guiding significance for formulating traffic evacuation plans in real life.

**Keywords** crowd simulation; continuum model; unilateral incompressibility constraint; GPU acceleration

收稿日期:2015-01-29;在线出版日期:2015-10-26.本课题得到国家自然科学基金(61300101)、江苏省自然科学基金(BK20130638)资助. 孙立博,女,1983年生,博士,讲师,主要研究方向为虚拟现实、虚拟人群仿真技术.E-mail: sunirisgrace@gmail.com.孙晓峰,男,1986年 生,硕士,主要研究方向为虚拟现实、秦文虎(通信作者),男,1969年生,博士,教授,博士生导师,主要研究领域为虚拟现实、汽车安全.

# 1 引 言

近年来,随着虚拟现实技术的不断发展和完善、 图形硬件运算速度的快速提高,虚拟人群仿真技术 越来越被人们广泛关注,逐渐成为计算机图形学及 相关领域的研究热点.

虚拟人群仿真技术在城市规划、影视娱乐、军事 训练、疏散模拟、大型体育仿真、古遗产还原重建等 领域都有着广泛的应用前景:

(1) 在城市规划领域,随着汽车保有量和城市 人口规模的不断增加,上下班高峰期的道路拥堵问 题严重困扰着人们的日常生活.通过引入虚拟人群 仿真技术,根据人群的出发时间、目的地和道路的实 际状况,可以为人群和通行车辆规划出拥堵少的路 径,达到减少人们等待时间,保证道路基本畅通的 目的.

(2) 在影视娱乐方面,《权力的游戏》中恢宏的 军队对战场景印证了引入大规模群体动画的必要 性,其不仅可以为观众带来超级震撼的视觉体验,而 且可以极大地减少动画师的手工操作,并为导演进 行更完美的构思提供有效途径.

(3)在军事训练方面,很多特种兵的实战化训 练都是比较危险的,引入群体仿真技术,不仅可以在 一定程度上降低训练的危险系数、保证战士的安全, 还可以提高训练效率,为全面提升我国军事力量奠 定基础.

(4)在疏散模拟领域,如何在紧急情况发生时, 迅速将人员撤离到安全场所是人们关注的重要问题.通过引入虚拟人群仿真技术,不仅可以判断建筑 物的出口数量、出口位置等设计是否合理,还可以根 据人群疏散仿真的结果,为建立人群的应急疏散方 案提供帮助,并在此基础上进一步验证方案的有效 性和合理性.

(5)在大型体育仿真方面,直接根据导演的创 意进行真人彩排,不仅需要大量的人员配合,更需要 花费大量的时间去验证进而修改完善编排方案.而 引入虚拟人群仿真技术,就可以有效地解决上述问 题,使导演在短时间内完成精彩绝伦的设计成为可 能,北京奥运会惊艳全球的开幕式就是最好也是最 有力的证明.

(6) 在古遗产还原重建方面,虽然重现古建筑 原貌是人们关注的首要问题,但是将该时期的居民 也包括其中,不仅可以提高场景的生动性,而且可以 使人们对古建筑所反映的历史和文化有更加直观和 深入的认识.

学

报

机

笡

目前根据不同的仿真目的,我们可以将虚拟人 群仿真方法分成两类:第1类是宏观仿真方法,第2 类是微观仿真方法:宏观仿真方法主要侧重于整个 群体的真实性,也就是不再著重于个体,而是使人群 整体的运动看起来真实即可.比较有代表性的方法 包括 Treuille 等人<sup>[1]</sup>提出的连续人群运动模型和 Narain 等人<sup>[2]</sup>提出的将离散和连续表示相结合的 可扩展的动力学人群仿真方法. Treuille 等人将运 动看成是粒子能量最小,人群按照目标的不同进行 分组,并且结合全局的路径规划和局部的碰撞避免, 保证人群所走的路径是最优化的. 在连续人群运动 模型中,Treuille 等人还考虑了人群密度对速度的 影响.即当人群密度较低时,人群保持最大的速度前 进,而当人群密度较高时,速度会受到相应的影响, 密度越高,速度越小,这会给人比较真实的感觉.另 外,由于人群是按照目标是否相同进行划分的,所以 一个组内的人群只做一次路径规划,这样使用的计 算资源也就相应减少,从而保证人群运动是实时的. Narain 等人对密集人群所表现出的聚集行为进行 仿真,其使用混合方法,即将 agent 看成是离散个体 和连续整体的统一体.在进行连续性设置时,将单一 不可压缩性的动态限制条件引入到模型中,对大规 模的人群行为进行建模并且对密集度较高的场景中 人与人之间的碰撞避免进行加速,当人群密度较高 时,由于每个个体的自由受到了周围个体的限制,因 此 Narain 等人将人群建模成可以用密度和流速描 述的连续流体,并把局部避碰映射转换到流体域, 从而得到限制人群流的可变参量,这个限制量就 是单一不可压缩限制(Unilateral Incompressibility Constraint, UIC). 该方法的具体实现可描述如下, 其将局部规划模块和全局规划器相结合,首先通过 全局规划器获得每个 agent 的理想速度,其次将 agent 的离散集合用密度和速度转换成连续表示,再 次引入 UIC 来确定合理的速度域. 所谓 UIC, 是根 据人群的特性提出来的,当密集度较高时,人群虽然 很像流体系统,但是也不完全相同.这就意味着它既 不是完全不可压缩的,又不是完全可压缩的.也就是 说,可以使用 UIC 对人群的密度加以限制,从而使 其密度不可以无限制地增大,这也保证了人与人之 间的距离不会过小.最后,每个 agent 根据真实速度 进行运动,从而实现位置上的更新.

微观仿真方法侧重于个体行为的真实性和多样

性,也就是说其对每个 agent 都进行行为建模,包括 感知、规划、心理、性格、情感等.目的是使每个 agent 在对待同一事件时,能够表现出不同的反应,从而提 高仿真的真实性.根据研究方式的不同,微观仿真方 法又可细分为基于模型的方法和基于数据驱动的方 法.基于模型的方法以社会力模型[3]、元胞自动机模 型[4]和基于规则的模型[5-6]为主要代表.社会力模型 以牛顿力学为基础,假设个体受到社会力的作用,其 可认为是社会心理和生理的"力"的合力,从而来控 制人群运动,由于社会力模型为每个个体解牛顿力 学方程,将人与人之间、人与静态障碍物之间的斥 力,人与目标之间的引力以及摩擦力考虑进来,并且 对个体行为合理建模,因此能够真实地再现人群拥 挤时的挤压力,逼真地模拟紧急状态下人群的疏散 过程, 元胞自动机模型将人的运动限制于离散的方 格之中,方格的大小固定了人群的密度,使得人群流 量不真实.而且由于其在时间和空间上都是离散的, 所以用该方法仿真出来的人群运动不连续,因而缺 乏真实性.基于规则的模型在人群密度较低或者中 等的情况下,能够仿真出更加真实的人群运动,但是 不能处理人与人之间的接触,因此无法仿真出挤压 的效果. 鉴于人群行为本身的多样性和复杂性, 基于 模型的方法大多是针对人群行为模型的某一方面展 开研究的,因此近年来涌现出很多扩展模型[7-8]对其 进行丰富和完善,基于数据驱动的方法主要包括运 动捕获数据驱动的方法[9-10] 和真实录像分析的方 法[11-12]. 运动捕获技术可以很容易地获得很多真实 人体运动的数据,因此使用这一技术生成群体动画 在目前非常流行,也得到了广泛应用.但是运动捕获 技术也存在一些不足,它需要对数据进行运动编辑 (motion editing)、运动重定向(motion retargeting) 或者使用运动图(motion graph)等技术自动生成相 似动作间的转换从而增加了更多新的动作片段,以 实现对更多种身材比例的模型进行运动控制,达到 更真实模拟人群运动的目的.基于真实录像分析方 法的思想是通过对真实人群的例子或者对拍摄到的 真实视频中人群的运动轨迹进行观察分析,从而仿 真人群行为,以达到更加真实再现实际场景的目的. 该方法近来也成为研究学者关注的焦点.

最近,很多采用 GPU 仿真人群的研究也取得 了一定的成果<sup>[13]</sup>. Courty 等人<sup>[14]</sup>早期在快速人群 的研究中就实现了 GPU 人群仿真的效果,与 CPU 相比,它的加速性能得到了显著的提高. Erra 等 人<sup>[15]</sup>在 GeForceFX5800 的 GPU 上实现了基于离 散行为模型的大规模群体(boids)仿真. Richmond 和 Romano<sup>[16]</sup> 提出了一种利用 GPGPU 实现的基于 个体的行人仿真,取得了不错的效果.他们设计了一 个支持 GPU 脚本的界面,这样就可以定义更多复 杂的个体行为.为了提高仿真性能,行人的数据和仿 真都放在 GPU 中进行,达到了减少数据传输花费 的目的.同样地,他们通过 GPU 上的 LOD 系统提 升了渲染和仿真性能. D' Souza 和他的团队<sup>[17]</sup>在 GPU上模拟了大型人群,仿真的个体数量超过了 200万,并达到了每秒更新 50 次的效果. 他们用 GPU 纹理存储器来保存个体的属性,然后利用 GLSL 来编写着色语言. Passos 等人[18-19] 在最近的 一个研究中通过 NVIDIA 的 CUDA 技术证明了实 时的大规模人群仿真是具有可行性的.他们成功地 模拟了超过 100 万的 boids,并且将帧速保持在 30 帧左右,效果相当于利用多核 CPU 模拟的 15 000 个 boids. 实验结果表明基于通用并行计算的方法能 极大地增强实时人群仿真的效果. Karthikeyan<sup>[20]</sup> 也在他的硕士学位论文中利用 CUDA 技术对虚拟 人群进行渲染,并对比了在 GPU 和 CPU 上的动画 计算.结果表明,GPU能够比 CPU 节约 1000 个左 右虚拟角色的计算.

综上所述,当仿真中小规模人群时,采用微观方 法能够体现个体行为的真实性、完整性和差异性,但 是不管采用基于模型的仿真方法还是基于数据驱动 的仿真方法,都会面临对数据的需求量较大,处理起 来比较繁琐的问题或者对行为建模时考虑的因素过 多而使得算法的计算复杂性增高的问题;而当仿真 较大规模人群时,由于计算资源消耗增大,采用宏观 方法更为合适.本文旨在模拟高密度下的人群运动 过程,比较注重仿真的实时性而对个体行为的差异 性不做过多要求,因此我们将采用基于连续流体模 型[1]的宏观方法对人群进行仿真.考虑到人群在密 度较高时,既不是完全不可压缩的,也不是完全可压 缩的,为了突出人群的这一特性,本文在 Treuille 等 人[1]提出的连续人群运动模型的基础上引入了有限 压缩性规则<sup>[2]</sup>,对人群的密度加以合理地限制,并引 入 GPU 加速算法提高仿真速度,从而比较真实地 模拟了各种场景下的人群运动.

本文第2节阐述基于连续模型和动力学仿真模型的高密度人群仿真算法的总体框架、局部人群密度的定义以及各算法的具体步骤;第3节介绍人群 仿真算法在 GPU 上的扩展;第4节通过多组仿真 实验,对提出的算法进行验证,并对仿真结果进行详 细的分析;第5节比较并分析本文提出的算法和已 有的虚拟人群宏观仿真方法;第6节总结全文.

# 2 基于连续模型和动力学仿真模型的 高密度人群仿真算法

#### 2.1 高密度人群仿真算法框架

通过对体育馆离场时高密度人群的运动观察发现,人群在向各出口运动的过程中,个体的自由度会由于附近人群的限制而大大降低,这与流体运动非常相似.因此,本文将人群看作一组连续的个体,通过密度和流动速度来表示它们的特性;而随着人群密度的增加,会出现个体之间发生碰撞的现象,故本文将局部避碰方法引入连续场中,以更好地控制人群的行为.

本文提出的高密度人群仿真算法的总体框架如 图1所示:在每一个时间步长,首先将离散的人群转 化成基于密度场(density field)和速度场(speed field)的连续表示方法,然后在局部避碰方法中利 用有限压缩性规则(Unilateral Incompressibility Constraint,UIC)对连续人群中的个体速度进行校 正,并插值得到个体的实际运动速度;最后将两者结 合起来,更新人群的位置,从而实现高密度人群仿真.



图 1 高密度人群仿真算法框架

#### 2.2 基于连续流体的全局规划算法

本文采用 Treuille 等人<sup>[1]</sup>的方法将人群看作一 组连续的个体.为了体现人群的流动特性,我们首先 做4个重要的假设.

**假设1.** 人群中的每个人都试图到达一个目标 G,这个目标点或目标区域对于每个人来说都是 唯一的.

假设 2. 人群会尽可能地以最大速度运动,也 就是说存在一个最大速度场 f 使得在位置 x 沿 $\theta$  方 向运动的行人的速度可用式(1)来表示:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{\theta}) \boldsymbol{n}_{\boldsymbol{\theta}} \tag{1}$$

其中, $n_{\theta} = [\cos\theta, \sin\theta]^{T}$ 表示指向方向 $\theta$ 的单位向量.

当人们能够畅通无阻地移动时,他们通常会选择一些特定的路线,比如行人过马路时会走斑马线; 人们会选择一条比较熟悉的路线行走,尽管该路线 可能要远一些.为了体现人群的上述特性,引入不适 度函数这一概念,动态的不适度可用来仿真人群之 间以及人群与障碍物之间的避碰行为.

假设 3. 存在一个不适度函数 g,在其他条件相同的情况下,如果  $g(x_0) > g(x)$ ,人们则会选择向 x 点移动而不是向  $x_0$ 点移动.

一般来说,人们会选择最短路径到达目标位置. 但实际上,人们会更趋向于避免拥挤堵塞和环境不 好的路径(即不适度较高的区域),也就是说,人们会 在选择路径时综合考虑如下 3 个方面的因素,力图 使三者的线性组合最小化:

(1) 路径长度;

(2)到达目标位置所需的时间;

(3)沿着路径方向单位时间内的不适度.

**假设 4**. 若 *Π* 是所有可行路径的合集,当速度 场 *f*,不适度 *g* 和目标位置*G* 都确定时,行人所选择 路径 *P*∈*Π* 的代价可以用式(2)表示:

$$\alpha \int_{P} 1 ds + \beta \int_{P} 1 dt + \gamma \int_{P} g dt$$
 (2)

其中:第1项为路径长度的耗费,第2项为移动时间 的耗费,第3项为不适度引起的耗费.α,β和γ分别 为各自的权重;ds表示路径长度的积分,而dt表示 时间的积分,这两个变量的关系如下:

$$\mathrm{d}s = f\mathrm{d}t \tag{3}$$

通过式(3),化简可以得到

$$\int_{P} C \, \mathrm{d}s \tag{4}$$

其中, $C \equiv \frac{\alpha f + \beta + \gamma g}{f}$ 表示单位消耗场.  $\alpha, \beta$ 和  $\gamma$  值可以由用户根据场景进行设置.

接下来进行密度场的转化,并在每一时间步长, 为每一组人群计算最大速度场和单位消耗场.其中, 为了保证每一组人群选择的路径都是最优路径,定 义势函数 \$,在目标点时 \$=0;而在其他点时 \$ 满足 eikonal 方程:

$$\nabla \phi(\mathbf{x}) = C \tag{5}$$

其中,单位消耗场 C 是沿着势函数梯度方向的. 根据 式(5),可得到每个人在点 x(二维坐标)处的速度:

$$\dot{\mathbf{x}} = -f(\mathbf{x}, \theta) \frac{\nabla \phi(\mathbf{x})}{\|\nabla \phi(\mathbf{x})\|}$$
(6)

其中: $\dot{x}$  表示速度; $f(x,\theta)$ 为运动方向上最大允许的速度值.

最后构建全局动态势能场,并根据势能场的梯 度得到期望的全局速度场,从而驱动人群运动.鉴于 建立动态势能场是整个算法中最复杂最耗时的步 骤,本文将对其详细介绍.

由于隐式的 eikonal 方程不能直接计算出来, 因此需要采用有效的方法来解决.目前,有效的解决 方法主要采用 Rouy 和 Tourin<sup>[21]</sup>提出的迭代算法 和 Sethian<sup>[22]</sup>提出的快速行进法(Fast Marching Method).迭代算法对每一个网格点都进行迭代更 新直至其收敛为止,该方法虽然易于实现,但是由于 其对每一网格点都进行访问并评估直至所有网格点 收敛,因此该方法的效率不高.而快速行进法(FMM) 则采用波阵面窄带的思想,通过维护堆结构选择 性地更新网格点,在大小为 M×N 的网格上,FMM 算法的时间复杂度仅为 O(MNlog(MN)).鉴于 FMM 在速度上的优势以及其在各个领域的广泛应 用,我们选择使用快速行进法来求解 eikonal 方程, 以达到保证全局规划算法效率的目的.

快速行进法的基本思想如图 2 所示,首先将场 景中已划分好的网格存入 3 种列表:已知列表、未知 列表和候选列表,设目标处的势能 \$ 为 0,并将相应 的网格存入已知列表,其余的网格存入未知列表,势 能 \$ 设为∞;然后将与已知列表中网格相邻的未知 网格存入候选列表,并通过方程的有限差分法估计 候选列表中各网格的势能 \$;最后将势能最小的候 选网格存入已知列表,再根据更新后的已知列表确 定相应的候选列表,并计算候选列表的势能;重复这 些过程,直到将所有的网格都存入已知列表,候选列 表为空为止.这样场景的势能场 \$ 就确定完毕了. 快速行进法的实现可描述如算法 1.



图 2 快速行进法

算法1. 快速行进法.

输入: grids U, 网格数量  $num_grids$ , 目标 G 势能  $\phi_G$ ,

已知列表 KList,未知列表 UList,候选列表 CList

```
输出:场景的势能场 Φ
```

- 1.  $KList = \{U_G\}$ ,  $UList = U U_G$ , CList = null;
- 2. while  $length(KList) \leq num_grids$
- 3. for  $U_i$  in U do
- 4. if  $(U_i \in KList)$
- 5. 计算 *U<sup>i</sup>*所有的邻居节点 *N<sup>i</sup>*
- 6. for  $n_{ij}$  in  $N_i$  do
- 7. if  $(n_{ij} \in UList)$
- 8. 将 *n<sub>ij</sub>*添加到 *CList* 中
- 9. end if
- 10. end for
- 11. end if
- 12. end for

13. for  $C_i$  in *CList* do

- 14. 计算势能值  $\phi(C_i) = CompPotential[x[i], y[i]];$
- 15. end for
  - 16. 找出 CList 中势能最小网格  $C_k = \arg\min(\phi(C_i))$
  - 17. 将 C<sub>k</sub>添加到 KList
- 18. end while

有限差分是微分方程的一种近似数值解法,该 方法的基本原理是利用网格剖分将连续的定解区域 用离散化网格来代替;将待解方程中的微商用各离 散点上的函数的差商来近似,积分采用积分和来逼 近.这样,待求的微分方程定解问题即可转化为相应 的有限差分方程组问题,通过求解有限差分方程组 即可获得原问题在离散点上的近似解,最后,再应用 插值方法便可从离散解求得定解问题在整个场域上 的近似解.通常利用有限差分法计算所采用的步骤 可概括如下: (1)采用一定的网格剖分方式离散化场域;

(2)进行差分离散化处理,用离散的、含有限个 未知数的差分方程组来近似代替场内具有连续变量 的偏微分方程及其边界条件;

(3)结合选定的代数方程组的解法,编写程序 获得差分方程组的数值解.

如图 3 所示,假设需要为网格 M 求解 eikonal 方程,首先要找到 x 轴和 y 轴两个方向上具有较低 消耗 的相邻网格单元:

$$m_{x} = \operatorname*{argmin}_{i \in \{W, E\}} \{ \phi_{i} + C_{M \rightarrow i} \},$$
  
$$m_{y} = \operatorname*{argmin}_{i \in \{N, S\}} \{ \phi_{i} + C_{M \rightarrow i} \}$$
(7)



图 3 场景的离散网格结构<sup>[1]</sup>

然后通过向后差分求解方程的有限差分近似 值,其中,∳<sub>M</sub>满足如下关系:

$$\left(\frac{\phi_{M}-\phi_{m_{x}}}{C_{M\rightarrow m_{x}}}\right)^{2}+\left(\frac{\phi_{M}-\phi_{m_{y}}}{C_{M\rightarrow m_{y}}}\right)^{2}=1$$
(8)

如果  $m_x$ 或者  $m_y$ 由于邻近网格的势能都是无穷而没 有定义,上述方程中就应该去掉该轴项.在计算得到  $\phi_M 后,即可通过其与相邻网格的势差求得梯度 <math>\nabla \phi$ , 并可进一步地根据梯度求得该点的速度场 v,而在 使用局部避碰算法对速度 v进行校正之后,就可以 通过对速度场的插值来更新每一个人的位置.

鉴于计算动态势能场需要遍历场景的每一个网格,当场景面积比较大、划分的网格数比较多时,运 算量就会变得非常庞大,直接影响全局规划算法的 运行效率,而考虑到人群有可能只集中到某一区域, 在其他区域分布较少,本文首先采用 AABB 轴向包 围盒法(如图 4 所示)来确定人群的分布范围,然后 只对位于包围盒内部的网格执行密度场和速度场的 更新,而对于没有人群分布的网格(区域 A、区域 B 和区域 C)不再进行密度场和速度场的计算,这样就 可以在一定程度上降低全局规划算法的计算复杂 性,以保证算法的实时性.



图 4 采用 AABB 包围盒划分后的场景

#### 2.3 基于 UIC 的局部避碰算法

在进行 2.2 节的处理后,我们可以将人群看作 带有密度和流动速度的连续流体,理论上在该模型 中不会出现两个人相交的现象;当两个人互相靠近 时,密度会变得非常高,速度方程完全由平均速度项 决定,这样他们之间的相对速度就会降为零.但实际 上,该方法只能解决网格之间的人群相交问题,对于 在同一网格内的两个人可能发生碰撞的问题并不能 很好地解决,因此该方法只适合仿真中密度人群.当 人群密度非常高时,场景所划分的一个网格单元中 就包含了很多人,这时需要将局部避碰算法引人到 连续场中,从而获得一些变量的约束条件.本文结合 Narain 等人<sup>[2]</sup>的方法,将一种可以与基于流体的全 局规划算法相结合的基于 UIC 的局部避碰算法引 人进来,从而实现大规模人群的仿真.

有限压缩性规则(UIC)在宏观上能够保证个体 之间不发生碰撞.首先,我们对全局场进行设置,用 连续密度场和速度场来表示人群的状态和运动.这 是一个类似于流体的系统,但是,其与物理流体又不 完全相同.它既不是完全可压缩的,因为人群不能被 无限度地压缩;也不是完全不可压缩的,因为人群可 以轻易地聚集或分散从而改变密度.因此,可以将两 者结合起来,把人群看做一种单向不可压缩的流体, 也就是说可直接对人群的密度设置约束条件,防止 其超过最大密度值 *P*<sub>M</sub>.

假设个体之间距离小到一定值后就不会再靠 近,该距离即为人与人之间的最小距离,用 d<sub>min</sub>表 示,据此可以给出人群密度 ρ<sub>M</sub>的上限,我们将其称 为 UIC 限制规则:

$$\rho \le \rho_M = 2\alpha / (\sqrt{3}d_{\min}^2)$$
 (9)

其中: $\rho_{M}$ 表示允许的最大密度;参数  $\alpha \leq 1$ 表示个体之间相互接近的程度.

根据人群模型的连续性,密度 ρ 和速度 ν 满足 连续方程:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{10}$$

为了确保满足 UIC 限制,我们需对期望速度 ν 进行校正,使得密度 ρ 保持在合适的范围内,最终的 校正速度 ν 在某种程度上接近期望速度 ν 并能维持 UIC 限制条件,具体方法见文献[2].

假设人群都尽量以最大的速度向他们期望的方向运动,同时还要避免碰撞,那么校正速度 v 的选择 旨在使积分∫ρv · v 最大化,当然还要满足最大运动 速度 || v || ≤ v<sub>max</sub>的限制.

根据上述条件,可以得到校正速度 v 的最佳 形式:

$$\mathbf{v} = v_{\max} \frac{\widetilde{\mathbf{v}} - \nabla p}{\|\widetilde{\mathbf{v}} - \nabla p\|} \tag{11}$$

其中,"压力"p满足如下条件:

$$\rho < \rho_{\rm max} \Rightarrow p = 0 \tag{12}$$

$$p > 0 \Rightarrow \nabla \cdot \mathbf{v} = 0 \tag{13}$$

在式(12)、(13)中,p表示人群流中存在一个非负压 力项,其用来防止人群聚集导致密度高于最大值  $\rho_{M}$ .在压力p的影响下,个体会加强对速度的校正, 以获得最大允许速度 $v_{max}$ .式(12)也可以如式(14) 表示:

$$p > 0 \Rightarrow \rho = \rho_{\max} \tag{14}$$

直观地说,压力只有在密度取最大值的时候才出现;当密度小于最大值时,个体不会很密集,相互之间所造成的影响比较小,人群可以比较自由地流动.

使用 UIC 规则只能确保个体之间在整体上保 持运动速度与方向的一致性,为了达到更好的效果, 我们需要在每对独立个体之间执行最小距离规则, 即查找出每对距离小于一定值的个体,将他们均匀 地分开.执行最小距离规则的算法过程如图 5 所示.

#### 2.4 算法流程

如图 6 所示,将全局规划与局部避碰相结合,总 结人群仿真算法的实现步骤如下:

(1) 在每个时间步长的开始,已知每个个体的 位置 x<sub>i</sub>,根据人群分布计算 AABB 包围盒从而确定 网格更新范围;

(2)执行全局规划算法,将离散的人群转化为 密度场表示;

(3) 对每组人群,构建相应的单位消耗场,并计



图 5 执行最小距离规则的算法流程

算动态势能场及其梯度,从而计算速度场;

(4)利用基于 UIC 的局部避碰算法,计算校正 速度场;

(5)获得每个个体的实际速度,并进行下一时 间步长位置的更新;

(6)最后,对更新的位置执行两两最小距离限制,从而避免个体之间的碰撞.

## 3 人群仿真算法在 GPU 上的扩展

本文采用的全局规划算法是基于连续流体的方法,这种基于空间网格计算的方法特别适合模拟大量个体.整个环境由一个消耗函数来表示,该函数通过 eikonal 方程来计算从任何位置到目标位置的剩余势能.沿着这个势能场的梯度前进,就能确保人群沿着最短路径向目的地移动.

虽然该方法为每一组人群只规划一次路线,能 够节约计算资源,但是其在每一步都进行密度场和 速度场的计算,运算量还是很大的,当人群数目不断 增多时,还是会影响算法的实时性.因此,本节将高



图 6 基于连续流体模型和动力学仿真模型的 高密度人群仿真算法流程

密度人群仿真算法在 GPU<sup>[23]</sup>上进行扩展,从而进 一步提高算法的效率.

#### 3.1 基于 GPU 的快速迭代法

在 2.2 节提到的所谓 eikonal 方程,是一个非线性的偏微分方程,可以表示如下:

 $H(\mathbf{x}, \nabla \mathbf{x}) = |\nabla \phi(\mathbf{x})|^2 - \frac{1}{f^2(\mathbf{x})} = 0, \quad \forall \mathbf{x} \in \Omega$ (15)

其中:Ω表示所在的n 维空间; ¢(x)表示到目标点所 需要经历的时间或者距离,而 f(x)表示速度函数. 在 2.2 节中,我们所采用的解 eikonal 方程的方法是 快速行进法,但是该方法并不适合于并行处理,这就 会影响算法的运行效率.本节为了提高人群仿真算 法的实时性,采用快速迭代法<sup>[24-25]</sup>在 GPU 架构中 求解 eikonal 方程.这里首先介绍 GPU 架构,以便 更好地阐述快速迭代法的特点.CUDA 是由著名显 卡厂商 NVIDIA 公司提出的一种基于 GPU 通用并 行计算的架构<sup>①</sup>,它利用 GPU 的处理能力,通过并 行计算来大幅提升计算性能.在 CUDA 推出之前, GPU 的并行计算一直受限于图形 API 的着色模型,只有符合顶点着色器和片元着色器属性的问题 才能利用 GPU 去并行处理.为了让 GPU 并行计算 变得更加简单实用,NVIDIA 公司以 C 语言为基 础,对其架构进行了一些扩展和两项改进,让开发者 能够根据 CPU 和 GPU 的架构分配代码和数据.

在支持 CUDA 的 GPU 架构下,一个程序可以 分成两个部分:主机端(host)和设备端(device).在 CPU上运行的部分即为主机端,而在 GPU上运行 的部分即为设备端,设备端又可被称为"Kernel(内 核函数)".通常显卡内存中的数据由主机端程序复 制提供,经由 GPU 执行 Kernel 对数据处理完成后, 再由主机端程序将其取回.

以并行计算为核心思想的 CUDA 技术将 GPU 视为一个协处理器,通过对线程(Thread)这一基本 单元的分配和管理来实现并行计算,其中,线程内部 的代码按顺序执行,而各线程之间的代码按并行执 行.共享同一存储空间协助完成计算的多个线程组 成一个线程块(Thread Block),大小相同且相互独 立的一组线程块可构成一个 Kernel,其中,各线程 块也按并行执行.CUDA 执行时,将 Kernel 函数设 计成可并行处理的数据,并由主机端发送到 GPU 上的 Grid(网格)上执行,其中,每个 Grid 由若干个 Block 组成,而每个 Block 又可以执行多个线程. CUDA 架构的基本原理如图 7 所示.

区别于传统的 GPU 顺序处理流程"顶点处理 器操作顶点属性变量——片元处理器处理片元及相 关数据——像素操作输出结果",支持 CUDA 的 GPU引入了流式多处理器(SM),大大提高了 CUDA 的并行计算能力,例如,GTX295 中包含了 60 个 SM,每个 SM 中又包含 8 个流处理器(SP),每个 SP 处理执行一个线程,一个 Block 单元必须分配到同 一个 SM 中,它是一种单指令多数据(SIMD)的执行 方式.这种执行方式是 CUDA 高速运算的基础,同 时由于 SM 还拥有片内共享存储器,因此可降低片 外访问延迟.

快速迭代法是 Jeong 等人<sup>[24]</sup> 通过对 Rouy 等人<sup>[21]</sup>提出的迭代算法和 Sethian<sup>[22]</sup>提出的快速行进

① NVIDIA Corporation. NVIDIA CUDA Compute Unified Device Architecture Programming Guide, http://www.cs. ucla.edu/~palsberg/course/cs239/papers/CudaReference-Manual\_2.0.pdf, 2008



图 7 CUDA 架构基本原理图

法(FMM)进行分析而开发出来的求解 eikonal 方程 的方法.快速迭代法整体运算性能好,适合并行扩展,主要具有如下特点:

(1)算法不设置特定的更新顺序;

(2)算法不需要另外使用一个数据结构来进行 分类排序;

(3)算法能够同时更新多个网格点.

与快速行进法不同,快速迭代法(Fast Iterative Method)不需要建立一个复杂的数据结构来保存所有网格的信息,而是通过建立一个激活列表来保存需要被更新的网格点.每次迭代,都对激活列表进行扩展,并同时更新所有受影响的网格点.本文通过逆向差分方法求解当前网格势能的局部近似值,当势能收敛时就将该网格点从激活列表中移除.势能收敛的判断方法为计算当前势能值与之前势能估计值的差,判断其是否在固定的阀值 ε 内.与快速行进法类似,采用 Godunov 逆向差分方法,通过式(16)来估算该网格的势能值:

 $[(U_{i,j} - U_{i,j}^{x\min})^+]^2 + [(U_{i,j} - U_{i,j}^{y\min})^+]^2 = \frac{1}{f_{i,j}^2} (16)$   $\ddagger \psi : U_{i,j}^{x\min} = \min(U_{i-1,j}, U_{i+1,j}), U_{i,j}^{y\min} = \min(U_{i,j-1}, U_{i,j})$ 

 $U_{i,j+1}$ ),(x)<sup>+</sup>=max(x,0), $U_{ij}$ 表示网格(*i*,*j*)处势 能的离散估计值.

#### 3.2 快速迭代法的流程与实现

快速迭代法包括以下两个部分:

(1)初始化.设置网格的边界条件和初始值,将目标网格的每一相邻网格加入激活列表 L,而将其他网格点的值设为∞;

(2)更新. 对激活列表 L中的每个网格  $x \in L$ , 根据式(16)计算新的势能值  $U_x$ ,并与原来的值进行 比较以判断其是否收敛;若收敛,则将该网格所有不 在激活列表 L中并且未收敛的相邻网格加入到激 活列表 L中,然后将该网格从激活列表 L中移除; 为了保证新加入激活列表 L的网格能够在下一次 迭代中得到更新,本文使用双向链表来表示激活列 表 L.重复更新步骤,直到激活列表 L为空.快速迭 代法中的更新算法可用伪代码形式描述,具体步骤 请参见算法 2.

**算法 2.** 更新算法 Update(X).

- 输入:网格集合 X,激活列表 L
- 输出:各网格收敛的势能值
- 1. Initialization (X: grid points, L: active list):
- 2. for each x in X do

3.	if $x$ is source
4.	$U_x = 0$
5.	else $U_x = \infty$
6.	end
7.	for each $x$ in $X$ do
8.	if any neighbor of $x$ is source
9.	add $x$ to $L$ ;
10.	end
11.	Update grid points in L:
12.	while $L$ is not empty
13.	for each $x$ in $L$ do
14.	$p = U_x$
15.	$q\!=\!\mathrm{new}\;U_x$
16.	$U_x = q$
17.	if $ p-q  < \epsilon$
18.	for each neighbor $x_n$ of $x$
19.	if $x_n$ is not in L
20.	$p = U_{x_n}$
21.	$q\!=\!\mathrm{new}\;U_{x_n}$
22.	if $(p > q)$
23.	$U_{x_n} = q$
24.	add $x_{r}$ to L:

25.	end
26.	end
27.	end
28.	remove $x$ from $L$
29.	end
30.	end
31.	end

图 8 显示了快速迭代法的二维迭代过程. 左下 角的网格为起始点,黑色网格表示已计算过势能的 网格,灰色网格表示处于激活列表中的网格,黑色箭 头表示窄带的推进方向.图 8(a)是初始状态,图 8 (b)是第 1 次更新后的状态,图 8(c)是第 2 次更新 后的状态.灰色网格的势能值只能由相邻的黑色网 格决定,因此激活列表中所有的灰色网格都能同时 进行势能值的更新.快速迭代法虽然为迭代算法,即 会一直更新激活列表中的网格直到其收敛为止.但 在大多数情况下,网格只需要迭代一次就能获得收 敛的势能值.为了更清晰地说明该现象,本文进行如 下分析:首先,定义特征路径为从某一网格到另一网 格的最优轨迹;然后根据特征路径与窄带推进的方 向夹角,从两方面进行讨论.



do



(1)当特征路径的方向与窄带推进的方向之间 的夹角小于 45°时,与快速扫描法<sup>[26]</sup>的原理类似,只 进行一次更新,激活列表中网格的势能值就不再改 变,也就是说,更新一次后,激活列表中网格的势能 值已收敛,窄带可向其一阶邻居网格推进;

(2)当特征路径的方向与窄带推进的方向之间 的夹角大于 45°时,对于特征路径的方向发生改变 的网格,其势能值起初会由于使用错误的迎风相邻 网格计算而导致需要在后续的迭代中对其进行修 正,因此,该网格不会从激活列表中移除直到其更新 至正确的势能值.也就是说,即使特征路径的方向改 变使得窄带范围增大,波前扩展将势能值本已收敛 的网格重新放入激活列表中,也会由于新信息的扩展而计算得出该网格正确的势能值.因此,即便在最坏的情况下,快速迭代法的性能也是次优的.Jeong 通过实验表明:快速迭代法的运行时间一般仅为快速行进法的一半.

### 4 仿真结果

本文的硬件实验平台为一台 PC 电脑, CPU 为 Core 双核 E8400, 主频为 3.0 GHz, 内存为 4 GB, 显 卡为 NVIDIA Geforce 9800GT. 本文设计的全部实 验场景主要采用 Unity3D 引擎搭建, 算法由 C # 语 言加以实现. 在本文实现的全局规划算法中,为了计 算单位消耗场,我们设定路径长度、时间和不适度的 权值分别为  $\alpha = 1$ 、 $\beta = 1$ 、 $\gamma = 1$ .

本文构建的高密度虚拟人群仿真界面如图 9 所 示,用户在进入每个场景之前,可以对人群参数进行 设置,主要包括虚拟人群的数量,个体的半径,个体之间的最小距离,个体的最大运动速度;同时也可以对场景网格进行设置,包括网格的单位长度和 X 轴、Y 轴方向的网格数.设置完成之后,用户可以根据自己的意愿,模拟不同场景下、不同数量的虚拟人群运动.

运动,在运动的过程中个体之间没有发生碰撞;

图 10(c)显示了人群即将到达目标点的情形,这里

有必要强调的是为了防止人群最后全部集中在目标

点上,本文允许先到达目标点的个体先行离开;与

图 10(b)相比,在图 10(d)中,由于未使用最小距离

规则限制,可以看出在同一网格的内部,个体相互重

叠在一起,发生了碰撞,这与现实生活中的人群运动

场景设置	人群设置		网格设置
M. dri a mil fall fat	数量:	10	单位长度: 1.0
甲组人群模拟	半径:	0.15	X轴网格数: 14
两组人群交汇	最小距离:	0.4	Y轴网格数: 10
四组人群交汇	最大速度:	1.2	
退出			完成



严重不符.

图 10 显示了单组人群向同一目标点行进的过程,其中,用户设置的人群数量为 12 人,个体半径为 0.15 m,最小距离为 0.4 m,最大速度为 1.2 m/s;根 据场景的大小和人群的数量,用户将场景划分为 14×10 的网格.图 10(a)显示了人群的初始状态,即 将人群的初始位置和目标点(显示为黑色立方体)设 置完成,并在初始位置绘制人群,显示为白色圆柱 体;在图 10(b)中,人群聚集到一起并向同一目标点





(b)使用最小距离规则





(c)到达目标状态

(d) 未使用最小距离规则

图 10 单组人群向同一目标运动(右侧黑色立方体为目标位置)

图 11 和图 12 为某一时刻检测到的人群密度场 和速度场的分布,其中,为了更清晰地体现密度场和 速度场的特性,用户同样将场景划分为 14×10 的网 格.从图中可以看出,每个人只对其相邻网格的密度 产生影响(可参见图 4,相邻网格在二维平面中即指 其上、下、左、右 4 个网格),而对其他网格密度的影 响可忽略不计.进一步地,速度场是在计算动态势能 场及其梯度后,根据式(7)计算得到的,速度场受到 人群密度场的影响,即在低密度区域,速度在平面上 保持常量;而在高密度区域,速度则取决于附近人群 的流动,即逆流时速度会受到抑制,而顺流时则无 影响.

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.53	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.92	1.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.59	1.68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.61	2.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.29	2.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.62	2.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.86	0.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

图 11 某一时刻的人群密度场

1.31	1.29	1.28	1.26	1.23	1.21	1.17	1.13	1.07	0.99	0.88	0.74	0.52	0.00
1.35	1.33	1.32	1.30	1.28	1.26	1.24	1.20	1.15	1.08	0.99	0.85	0.62	0.00
1.40	1.39	1.36	1.34	1.33	1.32	1.30	1.27	1.24	1.19	1.11	0.99	0.76	0.00
1.40	1.40	1.39	1.38	1.37	1.36	1.36	1.34	1.33	1.30	1.26	1.17	0.99	0.00
1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	0.00
1.40	1.40	1.38	1.38	1.37	1.36	1.36	1.34	1.33	1.30	1.26	1.17	0.99	0.00
1.40	1.40	1.36	1.34	1.33	1.32	1.30	1.27	1.24	1.19	1.11	0.99	0.76	0.00
1.40	1.40	1.34	1.30	1.28	1.26	1.24	1.20	1.15	1.08	0.99	0.85	0.62	0.00
1.40	1.38	1.32	1.26	1.23	1.21	1.17	1.13	1.07	0.99	0.88	0.74	0.52	0.00
1.36	1.32	1.25	1.21	1.18	1.15	1.11	1.06	0.99	0.91	0.80	0.65	0.45	0.00

图 12 某一时刻 x 轴方向的速度场

图 13 显示了两组人群在无障碍物的场景中交 汇的情形,其中,为了便于区分,将一组人群显示为 白色,而将另一组人群显示为黑色,每组人群的数目 均为 120 人.在图 13(a)中,两组人群由初始位置分 别向对方所在的位置,也就是其目标点运动;在 图 13(b)中,每组人群以尽可能大的速度运动;当发 现自己即将与对面的人群发生碰撞导致不适度升高 时,两组人群会重新计算密度场和最大速度场,从而 改变运动方向来避免碰撞的发生,如图 13(c)、(d)所 示;在成功避免碰撞后,两组人群重新以最大速度向 目标点前进,如图 13(e)所示;最终,两组人群成功 到达目标点,如图 13(f)所示.

图 14 显示了虚拟人群及其所在场景的三维效 果图,其中,静态障碍物用正方体表示,4 组人群用 圆柱体表示.每组人群数量均为 25 人,拥有相同的 目标.图 15 显示了 4 组人群在有障碍物的场景中交 汇的二维效果图,其中,4 组人群用不同灰度区别. 图 15(a)显示了 4 组人群的初始状态,即每组人群 都在初始位置准备向各自的目标点——对角线位置 运动;在图 15(b)中,4 组人群分别以最大速度向目 标点运动;在图 15(c)、(d)中,当靠近静态障碍物时,由于不适度升高,4 组人群在重算速度场后,采用改变运动方向的方式成功躲避静态障碍物;图 15(e)和(f)显示了 4 组人群交汇的过程,为了保证人群之间不发生碰撞,本文在每一时间步都计算密度场,构建单位消耗场,从而计算速度场,并采用 UIC 规则对速度场进行校正;图 15(g)显示了 4 组人群成功避碰后,继续向目标点运动的过程;图 15(h)显示了 4 组人群最终顺利到达目标点的情形.

从以上的仿真过程可以看出,本文采用的基于 连续流体模型和动力学仿真模型的高密度人群仿真 算法能够很好地完成对同一目标、指定目标、躲避障 碍物、多组人群交汇等不同情况的模拟,适用于仿真 各种人群的运动场景;同时由于引入 UIC 限制条件 和使用最小距离规则,算法能够很好地仿真个体之 间的避碰行为;进一步地,本文提出的高密度人群仿 真算法还允许用户对人群参数和场景参数进行设 置,也就是说,虚拟人群既可以自主运动,也可以根 据用户的意愿进行运动.因此,本文提出的虚拟人群 仿真算法适合于模拟高密度环境下的人群运动.



(f)成功运动到目标点



图 14 4 组人群交汇的三维场景

如图 16 所示, Unitv3D 引擎能够实时地检测场 景的一些状态,包括帧数、占用的线程数、内存以及 纹理等信息.实验中的所有个体都由 Unity 内部提 供的三维圆柱体表示,势必会占用一定的资源,而目 算法整体上计算量非常庞大,从表1即可看出,当人 数增加至 200 人时,算法的运行效率急剧下降,帧速 仅有 1.8 fps, 难以达到实时的效果. 引入 GPU 加速 算法后,由于 GPU 采用高度并行化的 SIMD 多处 理器架构,同一指令可由多个流处理器的执行单元 同时执行,因此,与在 CPU 上执行的快速行进法 (FMM)相比,在 GPU 上执行的快速迭代法(FIM) 速度提高了很多,从表2即可看出,网格数为32×32, 原本需要 70ms 的计算缩短到 13.6ms 就能解决,而 网格数为256×256,原本需要140ms的计算缩短到 18.3ms就能解决,加速效果非常明显,因此,本文 对高密度人群仿真算法在 GPU 上所进行的扩展, 确实为提高人群仿真的实时性做出了贡献.

表 1	人群数量与算法仿真速率之间的关系
-----	------------------

人群数量	仿真速率
10	30.0
20	30.0
50	6.1
100	3.2
200	1.8

表 2 快速行进法与快速迭代法在不同场景下的运行时间

场景网格数	FMM(CPU)/ms	FIM(GPU)/ms
$16 \times 16$	68.5	11.4
$32 \times 32$	71.0	13.6
$64 \times 64$	83.8	15.2
$256\!\times\!256$	140.0	18.3

# 5 比较与讨论

目前,已有很多工作对虚拟人群进行仿真,包括 宏观仿真方法和微观仿真方法.而宏观仿真方法尤 以 Treuille 等人提出的连续流体模型和 Narain 等 人提出的动力学仿真模型最具代表性.本文提出的 高密度人群仿真算法属宏观仿真方法的范畴,该方 法结合了连续流体模型和动力学仿真模型的优点, 即在保证人群运动真实、合理、自然的同时,还考虑 了人群在高密度情况下体现的既不是完全可压缩 的、也不是完全不可压缩的特征.

下面通过表 3 对本文提出的高密度人群仿真算 法与连续流体模型及动力学仿真模型进行比较和分 析.在避免路径拥堵及陷入死锁方面,全局规划算法 具有绝对的优势,动力学仿真模型由于主要侧重局 部避碰,对此考虑较少,只使用简单的路线图方法作 为全局规划器,虽然也能完成规划任务,但是在预测 路径拥堵、防止路径陷入局部最优或者死锁方面尚 有不足;而对于人群数目不断增多,人群密度却不能 无限制地升高的问题,连续流体模型并没有考虑,本 文提出的方法结合了全局规划算法和基于 UIC 的 局部避碰方法的优势,虽然每一时间步都进行密度 场和速度场的更新,使得计算量较大,运行效率下 降,实时性难以保证,但是在引入局部人群密度概念 和 GPU 加速以后,算法的实时性得到提高,从表 3 即可看出,同样是对 2000 人进行仿真,连续流体模 型在引入单独的并行绘制线程后,帧速为 24 fps;动 力学仿真模型由于仅关注个体之间的局部避碰,因



图 15 4 组人群交汇的仿真过程

Statis	FULS
Graphics:	9.8 FPS (102.2ms)
Main Thread: 102.2ms F Draw Calls: 1 batched: Tris: 200 Verts: 121 Used Textures: 0 - 0 B Render Textures: 0 - 0 B Screen: 739x598 - 5.1 M VRAM usage: 5.1 MB to 9 VBO Total: 25 - 101.9 KB Shadow Casters: 0 Visible Skinned Meshes:	Renderer: 0.0ms 0 8 switches: 0 1B 5.2 MB (of 0.96 GB) B 0 Animations: 0
Network: (no players c	onnected)

图 16 场景实时状态

	全局规划	预测拥堵	基于 UIC 的局部避碰	用户控制	并行绘制线程 或 GPU 加速	实时性 (2000人仿真)			
Treuille 等人提出的 连续流体模型	有	能	无	有	采用单独的 并行绘制线程	使用并行绘制后, 帧速为 24 fps			
Narain 等人提出的 动力学仿真模型	非常简单的 路线图(roadmap)	不能	有,使用 UIC 规则	无	无	帧速为 62 fps			
我们的高密度 人群仿真算法	有	能	有,使用 UIC 规则	有	GPU 加速	进行加速后, 帧速为 28 fps			

表 3 虚拟人群宏观仿真方法对比

此运行效率较高,帧速为 62 fps;而本文提出的算法 帧速为 28 fps,也就是说,在引入 AABB 轴向包围盒 确定人群的分布范围后,就不需要对没有人群分布 的网格进行密度场和速度场的计算,这在一定程度 上节省了计算资源;而且,将全局规划算法在 GPU 架构上进行扩展后,也使得实时性不再成为制约本 文提出的高密度人群仿真算法的瓶颈.另外,在用户 交互控制方面,连续流体模型允许用户创建移动的 障碍物来达到调整不适度从而验证仿真人群对动态 环境反应性的目的;动力学仿真模型对此方面没有 考虑;而本文算法允许用户设置人群的数量,个体的 半径,个体之间的最小距离,个体的最大运动速度以 及场景的网格数目,即在交互控制方面给予用户足 够的自由度,以达到仿真各种场景下不同规模人群 运动的目的.

### 6 结 论

本文提出了一种基于连续流体模型和动力学仿 真模型的高密度人群仿真算法.首先,本文实现了基 于连续流体模型的全局规划算法,通过4个重要的 假设,根据人群初始位置和目的地求出人群的密 度场以及最大速度场,从而构建单位消耗场,再根 据动态势能场及其梯度计算速度场.其次,引入基 于UIC限制规则的局部避碰算法,通过对密度的 控制来校正速度值.然后,将全局规划和局部避碰 相结合,即利用校正得到的速度值更新人群的位 置,最后再利用最小距离限制保证网格内部的个 体之间不发生碰撞,从而实现适用于高密度情况的 人群仿真算法.通过多组实验对本文提出的高密度 人群仿真算法进行验证,结果表明该算法能够模拟 各种场景下的人群运动,仿真效果良好.考虑到在人 数较多的情况下,整体计算量比较庞大,算法的运行 效率下降,难以达到实时的效果,本文又进一步对高 密度人群仿真中的全局规划算法在 GPU 上进行了 扩展,达到了加速的目的,有效地提高了算法的实 时性.

在高密度虚拟人群仿真中,如何在保证速度的 同时,突显人群中每个个体行为的社会性和智能性, 一直都是研究的热点问题,希望在下一步工作中能 够对该方面进行更深入的研究.一种值得尝试的 方案是与微观仿真方法相结合,通过模拟个体的 心理、性格以及情感因素,更逼真地反映人群的各 种特性.

- Treuille A, Cooper S, Popovic Z. Continuum crowds. ACM Transactions on Graph, 2006, 25(3); 1160-1168
- [2] Narain R, Golas A, Curtis S, Lin M C. Aggregate dynamics for dense crowd simulation//Proceedings of the ACM SIGGRAPH Asia. Yokohama, Japan, 2009: 1-8
- [3] Helbing D, Farkas I, Vicsek T. Simulating dynamical features of escape panic. Letters to Nature, 2000, 407: 487-490
- [4] Chenney S. Flow tiles//Proceedings of the ACM SIGGRAPH/ Eurographics Symposium on Computer Animation. Grenoble, France, 2004: 233-242
- [5] Reynolds C. Flocks, herds and schools: A distributed behavioral model. ACM Computer Graphics, 1987, 21(4): 25-34
- [6] Thalmann D, Musse S R. A hierarchical model for real time simulation of virtual human crowds. IEEE Transactions on Visualization and Computer Graphics, 2001, 7(2): 152-164
- [7] Kapadia M, Singh S, Reinman G, Faloutsos P. A behavior authoring framework for multi-actor simulations. IEEE Computer Graphics and Applications, 2011, 31(6): 45-55
- [8] van Toll W, Cook IV A, Geraerts R. Real-time densitybased crowd simulation. Computer Animation and Virtual Worlds, 2012, 23(1): 59-69
- [9] Yahia-Cherif L, Gilles B, Molet T, Magnenat-Thalmann N. Motion capture and visualization of the hip joint with dynamic MRI and optical systems. Computer Animation and Virtual Worlds, 2004, 15(3): 377-385
- [10] OBrien J F, Bodenheimer B E, Brostow G J, Hodgins J K. Automatic joint parameter estimation from magnetic motion capture data//Proceedings of the Graphics Interface 2000. Montreal, Canada, 2000: 53-60
- [11] Lerner A, Chrysanthou Y, Lischinski D. Crowds by example. Computer Graphics Forum, 2007, 26(3): 655-664
- [12] Lerner A, Fitusi E, Chrysanthou Y, Cohen-Or D. Fitting behaviors to pedestrian simulations//Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New Orleans, USA, 2009: 199-208
- [13] Jeremy S, Christopher O, Joshua B. GPU crowd simulation// Proceedings of the ACM SIGGRAPH ASIA. Singapore, 2008: 1
- [14] Courty N, Musse S R. Fastcrowd: Real-time simulation and interaction with large crowds based on graphics hardware.



SUN Li-Bo, born in 1983, Ph. D., lecturer. Her research interests include computer animation, virtual reality and crowd simulation. INRIA Grenoble, France: Technical Report, 2004

- [15] Erra U, De Chiara R, Scarano V, Tatafiore M. Massive simulation using GPU of a distributed behavioral mode of a flock with obstacle avoidance//Proceedings of the Vision, Modeling and Visualization. Standford, USA, 2004: 233-240
- [16] Richmond P, Romano D. A high performance framework for agent based pedestrian dynamics on GPU hardware//Proceedings of the EUROSIS ESM 2008. Le Havre, France, 2008: 1-8
- [17] D' Souza R M, Lysenko M, Rahmani K. SugarScape on steroids: Simulating over a million agents at interactive rates//Proceedings of the Agent 2007 Conference. Chicago, USA, 2007: 1-7
- [18] Passos E, Joselli M, Zamith M, et al. Supermassive crowd simulation on GPU based on emergent behavior//Proceedings of the 7th Brazilian Symposium on Computer Games and Digital Entertainment. Belo Horizonte, MG, Brazil, 2008: 81-86
- [19] Joselli M, Zamith M, Passos E, et al. A neighborhood grid data structure for massive 3D crowd simulation on GPU// Proceedings of the VIII Brazilian Symposium on Games and Digital Entertainment. Rio de Janeiro, 2009, 121-131
- [20] Karthikeyan M. Real Time Crowd Visualization Using the GPU [M. S. dissertation]. Virginia Polytechnic Institute and State University, USA, 2008
- [21] Rouy E, Tourin A. A viscosity solutions approach to shapefrom-shading. SIAM Journal on Numerical Analysis, 1992, 29(3): 867-884
- [22] Sethian J. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences, 1996, 93(4): 1591-1595
- [23] Zhang Shu, Chu Yan-Li. High Performance GPU Computing with CUDA. Beijing: China Water Power Press, 2009(in Chinese)
  (张舒,褚艳利. GPU高性能运算之 CUDA. 北京:中国水 利水电出版社, 2009)
- [24] Jeong Won-Ki, Whitaker R. A fast iterative method for eikonal equations. SIAM Journal on Scientific Computing, 2008, 30(5): 2512-2534
- [25] Jeong Won-Ki, Whitaker R. A fast eikonal equation slover for parallel systems//Proceedings of the SIAM Conference on Computational Science and Engineering. Costa Mesa, USA, 2007: 1-4
- [26] Zhao H. A fast sweeping method for eikonal equations. Mathematics of Computation, 2004, 74: 603-627

SUN Xiao-Feng, born in 1986, M.S. His research interest is virtual reality.

**QIN Wen-Hu**, born in 1969, Ph. D., professor, Ph. D. supervisor. His research interests include virtual reality and vehicle safety.

#### Background

This article focuses on high density crowd simulation. To produce behaviorally interesting agents, crowd simulation takes one of two approaches: microscopic approach or macroscopic approach. Microscopic approaches focus on the realism of individual behavior by simulating the perception, memory, planning and emotion of every agent. As a result, the computational cost increases rapidly due to the complexity of behaviors and variety of agents composing the crowd. Macroscopic approach aims at achieving real-time simulation for very large crowds, thus the behavior of each individual is not as important as long as the overall crowd movement produces realistic emergent behavior. The focus is on locomotion and collision avoidance while maintaining appropriate velocities, motions and directions. Classically this was done with continuum crowds based on dynamic potential field integrating global navigation with moving obstacles and a real-time and hybrid approach with a dual representation for simulating agents as both a discrete and single continuous system. The former method is more concerned with global planning while the latter method is more concerned with local collision avoidance. However, local collision avoidance alone cannot properly model real crowds with specific goal due to the possibility of getting stuck while it is time consuming to consider all static and dynamic obstacles/agents for global planning. Therefore, we present a novel approach for simulating crowd motions in high density realistically based on continuum model as well as aggregate dynamics model. We firstly consider the crowd as continuum flow and construct the dynamic potential filed to compute the velocity field. Then we integrate unilateral incompressibility constraint to constrain crowd density and correct the velocity field through the nonnegative pressure imposed by dense crowd. Finally, we adopt GPU accelerating algorithm to simulate dense crowd at interactive rates. Experimental results show that our approach can well simulate the crowd motions in various scenarios such as moving towards the same or assigned target, avoiding the collisions with dynamic or static obstacles and intersecting with each other.

This research is supported by a grant from the National Natural Science Foundation of China (No. 61300101) and a grant from the Natural Science Foundation of Jiangsu Province (No. BK20130638). These projects aim to simulate realistic crowd behaviors in real-time using advanced techniques in research areas of computer graphics and animation, in which simulating crowd motions in high density is one of the key problems. The team has been working on the research of crowd simulation for about ten years and has published some papers in respectable international journals and conferences.