

基于路径空间层次划分的区域匹配算法*

史扬^{1,2}, 凌云翔³, 金士尧³, 胡华平³

¹(海军潜艇学院 软件中心, 山东 青岛 266071);

²(哈尔滨工程大学 自动化学院, 黑龙江 哈尔滨 150001);

³(国防科学技术大学 计算机学院, 湖南 长沙 410073)

E-mail: shi yang@netease.com

摘要: 随着仿真应用在广度及深度上的发展, 基于因特网的分布交互仿真成为未来的发展方向. 其面临的主要挑战之一是系统可扩展性问题. 采用高效的数据过滤机制是解决问题的有效手段. 首先讨论了分布交互仿真存在的可扩展性问题, 介绍了HLA(high level architecture)框架下数据过滤机制的原理、实现及存在的问题. 针对大规模系统过滤匹配计算量大的问题, 提出了一种基于路径空间层次划分方法. 该方法利用空间位置关系知识进行启发式搜索, 从而排除大量不相关区域的匹配计算, 有效地降低了过滤机制的计算复杂度.

关键词: 分布交互仿真; 高级体系结构; 可扩展性; 数据过滤; 路径空间层次划分; 启发式剪枝搜索

中图分类号: TP393 文献标识码: A

分布交互仿真^[1]以网络技术为基础, 将地理位置分布的同构或异构的仿真器集成在一个时间和空间相一致的环境中, 并实现它们之间的互操作, 从而完成特定的任务. 分布交互仿真对于解决复杂、综合系统的研究、设计、验证具有重要的意义, 目前广泛应用于军事及国民经济的各个领域, 如军事演习、武器对抗模拟、复杂工程模型的协作设计与分布仿真、网上交互游戏等.

随着仿真应用在广度及深度上的发展, 基于因特网的分布交互仿真成为未来的发展趋势. 其面临的主要技术问题之一是系统规模的可扩展性问题. 因特网上影响系统可扩展性的主要因素是有限的网络带宽和仿真平台处理资源. 可扩展性是当前分布交互仿真技术面临的主要技术障碍, 本文从系统资源的角度出发, 分析了系统可扩展性问题, 以及新一代仿真技术框架HLA下的数据过滤机制原理及实现, 以降低过滤机制开销为目的提出了一种基于路径空间层次划分的启发式搜索匹配算法. 本文第1节介绍了分布交互仿真的特点, 分析了系统规模受限的原因, 介绍了HLA框架下数据过滤机制的实现途径. 第2节分析了过滤机制存在的区域匹配计算开销问题, 提出了路径空间层次划分方法, 在此基础上实现了基于启发式剪枝搜索的区域匹配算法. 第3节总结了全文.

1 分布交互仿真系统可扩展性问题及数据过滤技术

1.1 分布交互仿真系统特点及系统可扩展性问题

分布交互仿真系统中各结点机分别运行特定的仿真应用程序, 它由若干个仿真实体(被仿真的

* 收稿日期: 1998-11-30; 修改日期: 2000-02-25

基金项目: 国家自然科学基金资助项目(69525205); 国家“九五”国防科技预研基金资助项目(16.2.6.2)

作者简介: 史扬(1971-), 男, 河北广平人, 博士, 讲师, 主要研究领域为分布交互仿真, 地理信息系统, 虚拟现实技术; 凌云翔(1972-), 男, 四川成都人, 博士, 主要研究领域为分布交互仿真, 分布/并行处理技术; 金士尧(1937-), 男, 江苏扬州人, 教授, 博士生导师, 主要研究领域为计算机系统结构, 性能评价, 计算机仿真; 胡华平(1967-), 男, 湖南长沙人, 博士, 副研究员, 主要研究领域为性能评测, 复杂系统建模与仿真.

对象)模型构成.我们称结点内部的实体为本地实体,反之为远地实体.为保证系统整体状态的一致性,仿真实体间必须及时地通过网络交换有关信息,包括实体状态信息(如实体位置、速度)和交互信息(如碰撞、开火等).由于未来的分布交互仿真是建立在网络上的,并且要求能支持成千上万个仿真实体的运行^[2],因此,分布交互仿真面临的一个主要技术问题是系统规模的可扩展性问题.系统规模主要受系统资源(处理机资源及网络带宽资源)的限制,一方面,分布交互仿真系统规模(实体和仿真结点机数目)的增加,必然加大了系统内部信息的交换量,过多的网络数据量会导致网络拥塞,影响数据延迟特性;另一方面,对于现有的处理机平台(如工作站,微机),受操作系统等因素的影响,网络处理开销较大,对大量网络数据的处理会影响正常的仿真运行.而传统的 DIS 协议采用广播方式则加剧了这种趋势(广播方式引起大量网上数据传输及大量冗余网络数据的处理),因而严重影响了系统的可扩展性^[3].

1.2 分布交互仿真数据过滤技术

根据分布交互仿真任务的特点,使用数据过滤技术^[4]可大大缓解对网络带宽以及处理资源的需求,为系统的可扩展性提供了可能.支持数据过滤的依据是仿真实体的局部特性,即任何时刻某一仿真实体只与系统中部分实体发生交互,我们称此子集为实体的“感兴趣集”.只有在感兴趣集内部的实体之间才需要交换数据.在现实世界中,这种现象是很常见的,如模拟战场中的交战区域、武器的探测和作用区域及协作工程中的工作组等,都可以构成感兴趣集.在大规模分布交互仿真中,一个实体的感兴趣集一般只占系统实体集的一小部分,而且随着规模的增大,感兴趣集的大小趋于稳定^[5].例如,在地面战场装甲车辆对抗演习时,战场区域的扩大意味着实体规模的增加,但受模型的限制(视野、火炮射程及物理空间),其所交互的实体数量随规模的增加趋于稳定,而非随规模成正比.数据过滤主要有两个目的:一个是尽可能减少不相关数据的产生,以减少网络带宽的占用;另一个是降低仿真结点接收冗余数据时引起的处理开销.这两个目的都有助于系统的可扩展性.

1.3 HLA 框架下基于路径空间的过滤机制实现及存在的问题

1.3.1 基于路径空间的数据过滤机制实现途径

高级体系结构 HLA 是美国国防部于 1996 年提出的新一代分布交互仿真技术框架.它针对现有仿真技术(如 DIS 协议,ALSP 协议)的不足,解决分布交互仿真互操作性和可重用性^[6],以满足未来仿真发展的需求.目前,IEEE 已通过了 HLA 的标准草案,HLA 将成为未来分布交互仿真的标准.过滤机制的实现需要全局过滤信息的交换,过滤信息包括实体对发送和接收数据的规则说明.在 HLA 框架下规定使用路径空间 RS(routing space)作为标准的过滤信息描述格式.RS 是一个抽象的多维坐标系,它有 3 个要素:(1)坐标系统的维数;(2)路径变量,对应于坐标系统的每一维,用于表达感兴趣区域的特征(如地理坐标、运动速度等);(3)路径变量在每维上的定义(如范围、单位刻度等).

过滤信息分为两部分:公布区域信息和订购区域信息.区域(region)是 RS 上的子集,实体通过公布区域(publish region)和订购区域(subscribe region)描述发送数据和接收数据的约束条件(在各维上的范围),实体准备发送的或接收的数据与区域建立起对应关系.根据区域匹配计算的结果建立数据源与一组目的地的映射关系,在此基础上为它们分配组播地址,利用网络的组播功能(multicast),发送方只需一次发送即可将数据发送给一组接收方.组播通信既减少了发送和接受双方处理机的通信开销,又减少了网络的流量,有利于改善系统的可扩展性.

1.3.2 区域匹配算法存在的问题

HLA 只规定了路径空间作为过滤信息的描述方式,而没有规定过滤机制的具体实现方法.在实现中,过滤机制的匹配算法直接影响到系统的性能.衡量区域匹配算法的两个指标是:(1) 匹配算法的精确度,对于一个给定的区域是否能够准确地找出与其相交的所有区域,且只找出与其相交的区域,这决定了数据过滤的效率;(2) 匹配算法的复杂度,决定着过滤机制的系统开销.

一种直观的过滤机制匹配算法是采用完全匹配方式,即将所有外部实体的区域扫描一遍,以找出其中所有的相关实体.这种方法虽然数据过滤效果最佳,但对于实体数目成千上万的大规模系统来说开销较大. Van Hook 等人提出使用区域聚合 (clustering)^[7]方法减少待匹配区域的数量,以降低计算复杂度.这种方法是牺牲匹配精度(原本不相关的区域判为相关)为代价,影响了数据过滤效率,而且由于区域变化会引起复杂的解除及重新聚合处理,因此开销较大.

2 基于层次空间划分的区域匹配算法

2.1 路径空间层次划分的基本原理

根据分布交互仿真的局部性,分布在 RS 上的实体区域大部分之间是没有关系的(即定购和公布区域不重叠),如果能在区域匹配前排除明显不相关的区域,则可以大大降低匹配计算复杂度.我们将路径空间层次划分的思想引入到区域匹配过程中,通过结构化的路径空间划分,开发并利用区域的空间位置关系的知识,排除了冗余的区域匹配,为匹配计算复杂度的降阶提供了可能.

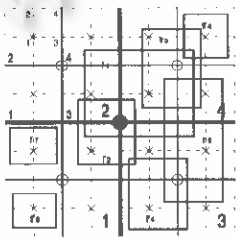


Fig. 1 Three-Level dividing of RS

图 1 路径空间三层划分示意图

如图 1 所示,为一个二维路径空间的三层划分示意图.空间被均匀划分成 4 个一级子空间(图中 1~4 为各层下的子空间标号),每个一级子空间又被划分成二级子空间.采用层次划分的优点是:(1) 有利于区域空间位置关系的表达、分类与查找;(2) 可适应不同区域尺寸,更加精确地开发区域空间位置关系的细节.

如 r_7, r_8 现在位于两个不同的二级子空间中,因此可以排除它们匹配计算的可能性.层次划分后的路径空间可用四叉树结构清晰地表达空间之间的包容关系.

2.2 空间编码及存储组织

空间层次划分将空间各部分的包容关系开发出来,有利于 RS 上的区域分类.层次划分的一个主要问题是如何有效地存储和表达空间的层次关系,以及如何描述 RS 上所定义的区域,使它能与层次空间建立映射关系.这涉及到对空间的层次编码的问题.本文采用文献[8]提出的基于四叉树的空间编码方案,它根据编码规则赋予每个层次上的空间一个编码,称为空间的关键值(key value).关键值采用 5 进制表示,以区别某一层的父结点与 4 个字节节点空间.如图 2 所示为三层空间划分下的空间编码.假设空间划分成 m 级,设 k 为某个 $l(i>0)$ 级子空间的关键值,其父结点的关键值为 k' ,则有如下关系:

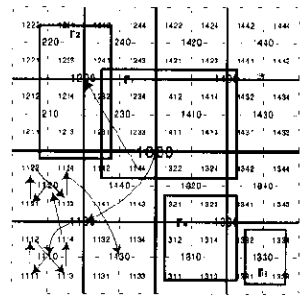


Fig. 2 Three-Level dividing and encoding

图 2 三层空间划分及编码示意图

$$k = k' + s * 5^{m-l}, \quad s \in (1, 2, 3, 4). \tag{1}$$

其中: $s=1$: if k 为 k' 的左下子结点;
 $s=2$: if k 为 k' 的左上子结点;
 $s=3$: if k 为 k' 的右下子结点;
 $s=4$: if k 为 k' 的右上子结点.

其中 0 级空间(即整个空间)的关键值定义为 5^m (例如,图 2 中 0 级空间编码为 1000_5). 这样,任意一个 l 级结点的关键值可递归定义为

$$k_l = \sum_{i=0}^l s_i 5^{m-i}, \quad s \in (1, 2, 3, 4). \quad (2)$$

由于每个空间的关键值惟一且按照一定的顺序编码(图 2 中的箭头标明空间编码顺序),空间用升序的线性链表结构存储,可表示为

1000, 1100, 1110, 1111, ..., 1114, 1200, ..., 1300, ...,
 1400, ..., 1440, 1441, 1442, 1443, 1444.

2.3 区域的编码及空间位置细节知识的进一步开发

区域编码的目的是将空间上定义的区域映射到某个子空间上,利用空间覆盖关系排除冗余匹配计算. 区域编码采用最小覆盖空间编码方法,即将完全覆盖区域的最低一级空间的关键值赋予此区域. 假设 RS 划分为 m 层, r 为路径空间 RS 上的一个区域,不失一般性,设 RS 的各维坐标经过归一化(normalization)处理后的各坐标轴范围为 $[0, 1]$. 设 $(x_1, y_1), (x_2, y_2)$ 分别为 r 的左下点和右上点坐标,各维坐标则可以 2^{-m} 精度的编码表示如下:

$$\begin{cases} x_1 = \sum_{i=0}^m a_i 2^{-i} \\ x_2 = \sum_{i=0}^m b_i 2^{-i} \\ y_1 = \sum_{i=0}^m c_i 2^{-i} \\ y_2 = \sum_{i=0}^m d_i 2^{-i} \end{cases}, \quad a_i, b_i, c_i, d_i \in (0, 1).$$

如果 r 的 4 个点落在 j 级空间上,则必有如下等式成立:

$$\begin{cases} a_i = b_i \\ c_i = d_i \end{cases}, \quad i=0, \dots, j.$$

这样,覆盖 r 的最小空间的关键值可由符合 $a_i \neq b_i$ 或 $c_i \neq d_i$ 条件的最小 j 推导得出:

$$k(r) = \sum_{i=0}^{j-1} (2a_i + c_i + 1) 5^{m-i}, \quad 0 \leq j \leq m+1. \quad (3)$$

例如,图 2 中 r_1, r_2, r_3, r_4 相对应的 j 分别为 1, 1, 3, 2, 根据式(3),它们的关键值分别为

$$k(r_1) = 1 \times 5^3 = 1000_5; \quad k(r_2) = 1 \times 5^3 = 1000_5;$$

$$k(r_3) = 1 \times 5^3 + 3 \times 5^2 + 3 \times 5^1 = 1330_5; \quad k(r_4) = 1 \times 5^3 + 3 \times 5^2 = 1300_5.$$

在图 2 中, r_1, r_2, r_3, r_4 的编码分别为 $1000_5, 1000_5, 1330_5, 1300_5$, 这表明 r_1 和 r_2 与其他所有区域都可能相交. 但实际上, r_2 仅位于空间左半部,并不与 r_3, r_4 相交, r_1 不覆盖空间 1400_5 的中心点,因此也不可能与 r_3 相交. 这主要是因为最小覆盖空间编码方法隐藏了空间关系的细节知识,如 r_2 不可能与空间 1000_5 的 3, 4 子空间(及其下级子空间)上的区域相交. 为此,我们在最小覆盖空间编

码的基础上提出了基于标定点的区域匹配算法.其基本思想是:利用某级空间的4个子空间的中心点作为标定点,提高区域在空间内部定位的精度,并利用这种知识避免不必要的匹配计算.对于任意区域 r ,我们增加 a_i, b_i, c_i, d_i 的第 $j+1$ 位,并利用第 j 位和第 $j+1$ 位来纪录区域在其最小覆盖空间的大体位置.

对于 a_i, b_i 有以下情况:

j 位	$a_i=b_i=0$			$a_i=b_i=1$			$a_i=0, b_i=1$				
区域位置	空间左半部			空间右半部			整个空间				
$j+1$ 位	$a_i=0$	$a_i=1$	$a_i=0$	$a_i=0$	$a_i=1$	$a_i=0$	$a_i=0$	$a_i=0$	$a_i=1$	$a_i=1$	$a_i=1$
覆盖标定点	\emptyset	\emptyset	$\{1,2\}$	\emptyset	\emptyset	$\{3,4\}$	$\{1,2\}$	$\{1,2,3,4\}$	\emptyset	$\{3,4\}$	$\{3,4\}$

对于 c_i, d_i 有以下情况:

j 位	$c_i=d_i=0$			$c_i=d_i=1$			$c_i=0, d_i=1$				
区域位置	空间左半部			空间右半部			整个空间				
$j+1$ 位	$c_i=0$	$c_i=1$	$c_i=0$	$c_i=0$	$c_i=1$	$c_i=0$	$c_i=0$	$c_i=0$	$c_i=1$	$c_i=1$	$c_i=1$
覆盖标定点	\emptyset	\emptyset	$\{1,3\}$	\emptyset	\emptyset	$\{2,4\}$	$\{1,3\}$	$\{1,2,3,4\}$	\emptyset	$\{2,4\}$	$\{2,4\}$

根据区域编码规则(当 $i=j$ 时,必有 $a_j \neq b_j$ 或 $c_j \neq d_j$), j 位的可能组合只有 $\left[\begin{matrix} a_j=b_j=0 \\ c_j=0, d_j=1 \end{matrix} \right]$,
 $\left[\begin{matrix} a_j=b_j=1 \\ c_j=0, d_j=1 \end{matrix} \right]$, $\left[\begin{matrix} a_j=0, b_j=1 \\ c_j=d_j=0 \end{matrix} \right]$, $\left[\begin{matrix} a_j=0, b_j=1 \\ c_j=d_j=1 \end{matrix} \right]$ 这几种情况.因此,根据 a_i, b_i, c_i, d_i 的
 j 和 $j+1$ 位组合可以进一步判定区域 R 在空间中更为精确的位置.例如,在图2中,区域 r_1 的 a_i, b_i
 位有 $\left[\begin{matrix} a_j=0, b_j=1 \\ a_{j+1}=1, d_{j+1}=1 \end{matrix} \right]$,它可能覆盖整个空间,且只可能覆盖标定点 $\{3,4\}$;对 c_i, d_i 位有
 $\left[\begin{matrix} c_j=0, b_j=1 \\ c_{j+1}=1, d_{j+1}=1 \end{matrix} \right]$,它可能覆盖整个空间,且只可能覆盖标定点 $\{2,4\}$.从而判断 r_1 覆盖整个空间的
 4个部分,但只覆盖标定点4,这意味着 r_1 不可能与位于空间 $1110_5, 1130_5, 1120_5, 1210_5, 1220_5,$
 $1310_5, 1330_5$ (包括它们的低级空间)上的区域相交.因此,利用 a_i, b_i, c_i, d_i 的两位编码可以进一步排除
 不相关区域的匹配计算.

2.4 基于启发式搜索的区域快速匹配算法

在前面工作的基础上,我们提出了区域匹配的新算法,该算法基于区域空间位置的细节知识(由编码信息表示)进行启发式的剪枝搜索(称之为HPS算法).在描述算法之前,首先定义区域信息存储的链表组织结构,如图3所示.该链表以空间编码的升序存储.

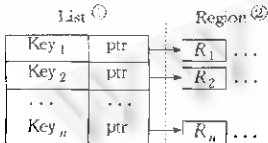


Fig.3 List structure of region
图3 区域的链表存储结构

图4为区域内部的结构信息, l 为区域所在空间的级数,用 $l(k)$ 表示关键值为 k 的空间的级数.根据编码规则,两个空间的关系可采用下面的公式来判定:

R_{id}	l	a_j	b_j	c_j	d_j	a_{j+1}	b_{j+1}	c_{j+1}	d_{j+1}	Description of a region ^①	ptr
----------	-----	-------	-------	-------	-------	-----------	-----------	-----------	-----------	--------------------------------------	-----

①区域描述信息.

Fig.4 Inner information of region
图4 区域内部结构信息

覆盖关系:设 k 和 k' 为两个空间的关键值,且有 $k < k'$,则空间 k 覆盖空间 k' 当且仅当

$$k' - k < 5^{m-l(k)}. \quad (4)$$

兄弟关系:设 k 为某一空间的关键值,则根据空间编码规则有

$$k' = k + 5^{m-l(k)}. \quad (5)$$

· 若 k 不为其父结点空间的 4 号子空间,则 k' 为 k 的下一个兄弟空间的关键值.

· 若 k 为其父结点空间的 4 号子空间,且其父结点不是上一级空间的 4 号子空间,则 k' 为 k 的父结点的下一个兄弟空间.

· 若 k 为其父结点空间的 4 号子空间,且其父结点也为上一级空间的 4 号子空间,则返回其祖父结点的下一个兄弟空间,依此类推.

算法中用到的几个函数定义如下:

(1) NEXT(k):返回关键值为 k 的下一个区域.若此时已没有这样的区域,则返回链表中下一个关键值大于 k 的空间上的区域.若到达链表尾部,则返回 NULL 值.

(2) OVERLAP(k, k'):若编码为 k 和 k' 的空间有覆盖关系,则返回 TRUE,否则返回 FALSE.使用式(4)判定空间的覆盖关系.

(3) MATCH(r, r'):根据区域 r 和 r' 的详细描述信息进行匹配计算,判断区域是否相交,若相交,则返回 TRUE.

(4) INCR(k):返回链表中空间 k 的下一个兄弟空间,使用式(5).

区域匹配问题定义:设 r^* 为路径空间 RS 上的某个区域, R 为 RS 上一组区域的集合 ($R \subset RS$),求 $R' = \{r; \forall r (r \in R), r \text{ 与 } r^* \text{ 相交}\}$.算法描述如下:

算法. 基于启发式剪枝搜索的匹配算法 HPS

Step 1. /* 初始化 */

计算区域 r^* 及 $r (r \in R)$ 的关键值即编码,并将 r 按关键值升序序列插入到链表中;

$R' = \emptyset; k = 0$. /* 初始化 R', k */

Step 2. $k = \text{NEXT}(k)$; /* 当 $k = 0$ 时, NEXT(k) 返回 0 */

若 $k = \text{NULL}$, 则转 Step 8. /* 搜索已到达链表尾部,结束 */

Step 3. 若 OVERLAP(k^*, k) = TRUE, 则转 Step 4; /* 空间 k^* 和 k 存在覆盖关系 */

否则, $k = \text{INCR}(k)$, 并转 Step 3. /* 寻找下一个与空间 k^* 有覆盖关系的空间 */

Step 4. 若 $l(k^*) = l(k)$, 则转 Step 5; /* 同级 */

若 $l(k^*) > l(k)$, 则转 Step 6; /* k^* 的级别要低于当前 k 的级别 */

若 $l(k^*) < l(k)$, 则转 Step 7. /* k^* 的级别要高于当前 k 的级别 */

Step 5. /* 排除 r^* 和 r 分别在不相交空间半部的匹配 */

若 r^* 的 $a_j = b_j = 0$ 且 r 的 $a_j = b_j = 1$, 则转 Step 2;

若 r^* 的 $a_j = b_j = 1$ 且 r 的 $a_j = b_j = 0$, 则转 Step 2;

若 r^* 的 $c_j = d_j = 0$ 且 r 的 $c_j = d_j = 1$, 则转 Step 2;

若 r^* 的 $c_j = d_j = 1$ 且 r 的 $c_j = d_j = 0$, 则转 Step 2;

若 MATCH(r^*, r), 则将 r 加入 R' , 并转 Step 2.

Step 6. /* k^* 的级别要低于当前 k 的级别,不存在剪枝的可能性 */

根据 k^* 的第 $l(k) + 1$ 位判断 k^* 落在 k 的哪个子空间上:

(1) /* 空间 k^* 被空间 k 的 1 号子空间覆盖 */

若 r 的 $a_j = b_j = 1$ 或 $c_j = d_j = 1$, 则转 Step 2; /* r 不覆盖其所在空间的 1 号子空间,跳过匹配计算 */

否则,若 $l(k^*) \geq l(k) + 2$, /* 判断 r 在空间 k 中的位置 */

则根据 k^* 的第 $l(k) + 2$ 位判断 k^* 落在 k 的哪个孙子空间上:

① 若 r 的 $a_{j+1} = 1$ 或 $(a_{j+1} = 0, c_{j+1} = 1)$, 则转 Step 2; /* 子空间 1 的 1 号子空间 */

② 若 r 的 $a_{j+1}=1$, 则转 Step 2; /* 子空间 1 的 2 号子空间 */

③ 若 r 的 $c_{j+1}=1$, 则转 Step 2; /* 子空间 1 的 3 号子空间 */

(2) 与(1)类似; /* 空间 k^* 被空间 k 的 2 号子空间覆盖 */

(3) 与(1)类似; /* 空间 k^* 被空间 k 的 3 号子空间覆盖 */

(4) 与(1)类似; /* 空间 k^* 被空间 k 的 4 号子空间覆盖 */

若 $\text{MATCH}(r^*, r) = \text{TRUE}$, 则将 r 加入 R' , 并转 Step 2. /* 区域匹配成功 */

Step 7. /* k^* 的级别要高于当前 k 的级别, 存在剪枝的可能性 */

根据 k 的第 $l(k^*)+1$ 位判断 k 落在 k^* 的哪个子空间上;

/* $k^* - 5^{m-l(k^*)-1}$ 为 k^* 的 1 号子空间, 做剪枝操作, 以跳到 k^* 的 2 号子空间 */

(1) 若 r^* 的 $a_j=b_j=1$ 或 $c_j=d_j=1$, 则 $k = \text{INCR}(k^* + 5^{m-l(k^*)-1})$ 并转 Step 3;

否则, 若 $l(k^*) \geq l(k) + 2$, /* 判断 r 在空间 k 中的位置 */

则根据 k 的第 $l(k^*)+2$ 位判断 k 在 k^* 的哪个孙子空间上:

/* $k^* + 5^{m-l(k^*)-2} + 5^{m-l(k^*)-2}$ 为 k^* 的子空间 1 的 1 号子空间, 剪枝, 跳到下一子空间 */

① 若 r^* 的 $a_{j+1}=1$ 或 $(a_{j+1}=0, c_{j+1}=1)$, 则 $k = \text{INCR}(k^* + 5^{m-l(k^*)-1} + 5^{m-l(k^*)-2})$, 并转 Step 3;

② 若 r^* 的 $a_{j-1}=1$, 则 $k = \text{INCR}(k^* + 5^{m-l(k^*)-1} + 2 * 5^{m-l(k^*)-2})$, 并转 Step 3;

③ 若 r^* 的 $c_{j+1}=1$, 则 $k = \text{INCR}(k^* + 5^{m-l(k^*)-1} + 3 * 5^{m-l(k^*)-2})$, 并转 Step 3;

(2) 与(1)类似; /* 空间 k^* 被空间 k 的 2 号子空间覆盖 */

(3) 与(1)类似; /* 空间 k^* 被空间 k 的 3 号子空间覆盖 */

(4) 与(1)类似; /* 空间 k^* 被空间 k 的 4 号子空间覆盖 */

若 $\text{MATCH}(r^*, r) = \text{TRUE}$, 则将 r 加入 R' , 并转 Step 2. /* 区域匹配成功 */

Step 8. 结束.

算法是一个自顶向下的二叉树搜索过程, 在路径空间层次划分和编码的基础上, 进一步开发空间内部区域位置关系的细节, 实现了启发式的剪枝搜索. 由算法可知, 对于高级别区域匹配, 低级别区域能利用其区域的位置关系知识进行剪枝操作, 即使是匹配比其级别高的区域, 也可以利用这种位置关系排除明显不相关的区域进行匹配计算.

2.5 对高维路径空间的处理

路径空间层次划分可扩展到三维路径空间上(有类似的编码和存储结构), 而对于大于四维的高维空间, 则可通过将高维空间分解为多个基本空间(2 维或 3 维)的处理方法, 对每个基本空间进行候选区域的筛选, 并取它们的交集做进一步的匹配计算. 本文的算法只要稍加修改即可使用, 限于篇幅, 本文不再讨论.

3 模拟试验结果

3.1 空间层次划分下区域的分布情况

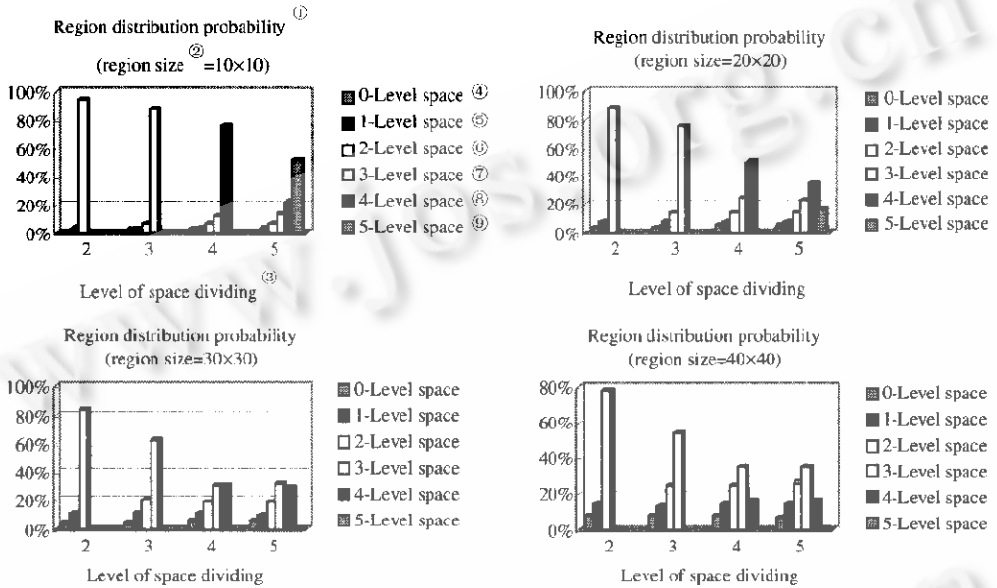
区域在各层空间的分布情况对匹配算法的影响较大. 区域应尽可能位于级别较低的空间上(0 级为最高级), 以便于较为精确地确定区域在空间中的位置, 从而减少区域在空间链表中搜索及匹配的次数. 为此, 我们模拟了区域在各个层次空间上的分布情况. 假定一个二维路径空间, 其尺寸为 1024×1024 , 对其进行 l 层划分. 在该空间上随机产生 n 个尺寸为 $s \times s$ 的区域, 如图 5 所示为区域在各层空间上的分布情况.

图 5 的模拟结果表明, 当区域相对于路径空间较小时, 绝大部分区域落在级别较低的空间(这里是指区域的最小覆盖空间)上.

3.2 剪枝操作的有效性验证

当待匹配区域位于高层空间上时, 将导致该区域与所有位于该层空间及其下级空间上的候选

区域进行匹配,如图5中区域 r_2 与0级空间的两条中轴相交(因此位于0级空间上).这时,可利用空间位置的细节知识进行剪枝操作,以减少匹配量.为了验证空间层次划分算法对剪枝操作的有效性,我们进行了模拟实验.以前面的试验为基础,人为控制待匹配区域所在的空间层次(如控制待匹配区域与0级空间的中轴线相交),以比较剪枝与非剪枝情况下的区域匹配次数.表中路径空间划分层次为4,其中 l 为待匹配区域所在的空间层次, N 为候选区域的数量,候选区域在空间的位置由随机数产生,区域尺寸为 20×20 .表1为模拟结果(取100次试验的平均数).



①区域分布概率,②区域尺寸,③空间划分层次,④0级空间,⑤1级空间,⑥2级空间,⑦3级空间,⑧4级空间,⑨5级空间.

Fig. 5 Statistic of region distribution on each level RS

图5 区域在各层空间上的分布情况

Table 1 Simulation result

表1 模拟结果

N \ l	1 000		2 000		3 000		4 000	
	Non-Prune ^①	Prune ^②	Non-Prune	Prune	Non-Prune	Prune	Non-Prune	Prune
0	1 000	350	2 000	661	3 000	993	4 000	1326
1	270	104	549	223	848	341	1125	486
2	100	68	200	155	369	235	439	307

①非剪枝,②剪枝.

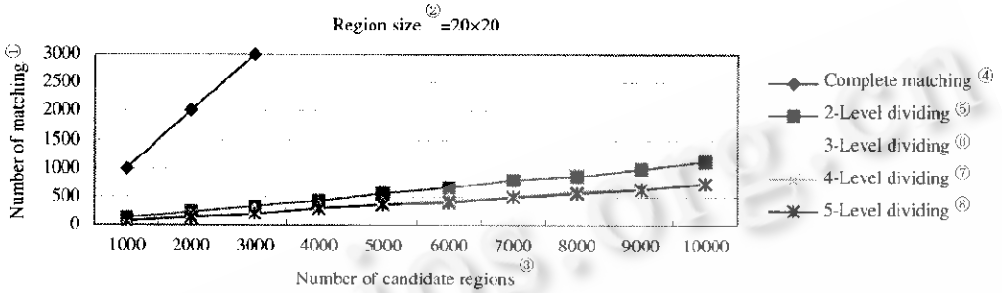
由表中的模拟结果可知,当待匹配区域位于较高层次时,它在空间中的位置靠近根结点,因此剪枝的效果更加明显.由此可见,剪枝操作进一步降低了区域匹配计算量,解决了待匹配区域位于高层空间时层次空间划分匹配算法效果急剧下降的问题.

3.3 层次划分区域匹配算法的有效性验证

为了验证基于路径空间层次划分的区域匹配算法的有效性,我们进行了如下的模拟试验:在上述二维路径空间上随机产生一定数量的候选区域,并产生一个待匹配区域进行算法的有效性验证.每种情况分别模拟了100次,并取平均数作为实验结果.

图6比较了完全匹配(区域-比较)方法与基于路径空间层次划分的区域匹配算法所需的匹

配次数. 由图 6 可见, 相对于完全匹配, 基于路径空间层次划分的匹配方法大大降低了匹配次数. 模拟结果还显示: 在空间划分大于 3 层时, 匹配次数没有较大的变化, 这说明对于特定的区域尺寸, 当层次划分到一定程度时就能充分开发出区域在空间的位置关系, 此时如果进一步增加划分层次, 并不能带来更大的收益.

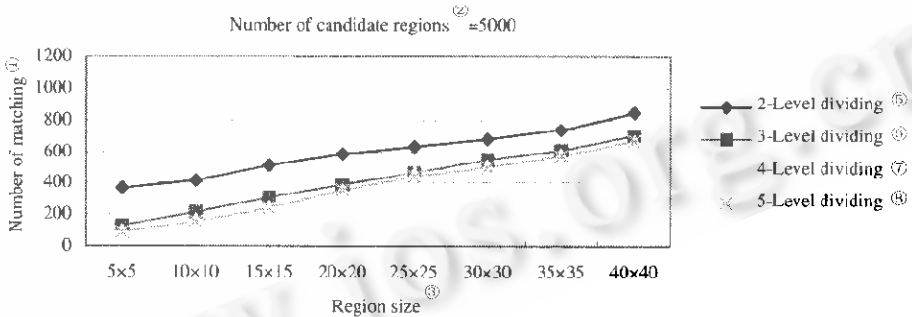


①匹配次数, ②区域尺寸, ③候选区域数量, ④完全匹配, ⑤2层划分, ⑥3层划分, ⑦4层划分, ⑧5层划分.

Fig. 6 Comparison of region matching times

图 6 区域匹配次数比较

图 7 研究了区域尺寸对匹配次数的影响(候选区域数量固定为 5 000). 模拟结果表明, 随着区域尺寸的增加(路径空间大小固定), 区域匹配次数也在增加, 这是因为较大的区域所在的空间级别也相应较高, 从而使空间搜索范围加大. 在大规模分布战场仿真环境中, 战场空间相对于实体定义的区域较大(如 STOW 项目^[9]中的战场环境为 500×750KM²), 因此区域层次划分能够满足要求. 这与我们研究大规模仿真环境下的系统可扩展的初衷是一致的.



①匹配次数, ②候选区域数量, ③区域尺寸, ④2层划分, ⑤3层划分, ⑥4层划分, ⑦5层划分.

Fig. 7 Effects of region size on matching times

图 7 区域大小对匹配次数的影响

4 总 结

网络与分布处理技术的发展使仿真由单一集中式发展到分布交互式, 基于因特网的分布交互仿真是仿真发展的方向. 高技术条件下新武器的研制、武器对抗、军事训练等对分布交互仿真提出了更高的要求. 未来的仿真要求系统在规模上是可扩展的, 能支持成千上万个仿真实体的运行, 以满足复杂系统以及地理分布环境下的仿真. 受系统资源限制, 网络环境下的分布交互仿真面临的一个主要技术问题是系统的可扩展性. 采用数据过滤机制可以大大缓解系统网络带宽及结点处理机资源增长的需求, 能有效地抑制网络资源耗费随系统规模的增加而急剧增长的势态, 提高系统可扩展

性。过滤机制的开销也随系统规模的增加而增加,因而影响过滤机制的高效性。为了降低过滤机制的计算复杂度,本文在HLA的数据过滤机制中引入了路径空间层次划分的概念,在此基础上进一步开发了区域在空间上的细节知识,并利用它排除不必要的匹配计算,从而降低了过滤机制的计算复杂度,其本质上是一种基于启发式的剪枝搜索算法。模拟结果表明,该算法能大大降低系统开销。

References:

- [1] Fullford, D. Distributed interactive simulation: it's past, present and future. In: Brummer, D., ed. Proceedings of the Winter Simulation Conference. San Diego, CA: ACM Press, 1996. 179~186.
- [2] Swaine, S., Stapf, M. Large DIS exercises-100 entities out of 100000. In: Medin, L., ed. Proceedings of the 15th Inter-service/Industry Training Systems and Education Conference. Orlando, FL: IEEE Computer Society Press, 1994. 4~13.
- [3] Stone, S., Zyda, M., Brutzman, D., et al. Mobile agents and smart networks for distributed simulation. In: Miller, D. C., ed. Proceedings of the 14th Distributed Interactive Simulation Workshop. Orlando, FL: IEEE Computer Society Press, 1996. 11~15.
- [4] Van Hook, D. J., Rak, S. J., Calvin, J. O. Approach to relevance filtering. In: Miller, D. C., ed. Proceedings of the 11th Distributed Interactive Simulation Workshop. Orlando, FL: IEEE Computer Society Press, 1994. 367~372.
- [5] Macedonia, M. Exploiting reality with multicast groups: a network architecture for large scale virtual environments. IEEE Computer Graphics and Application, 1995, 15(5):38~45.
- [6] Defense Modeling & Simulation Office (DMSO). High Level Architecture Interface Specification, Rules, Object Model Template: Version 1.0. 1996. <http://www.dmsu.mil>.
- [7] Van Hook, D. J., Rak, S. J., Calvin, J. O. Approaches to RTI implementation of HLA data distribution management service. In: Miller, D. C., ed. Proceedings of the 15th Distributed Interactive Simulation Workshop. Orlando, FL: IEEE Computer Society Press, 1995. 535~544.
- [8] Abel, D. J., Smith, J. L. A data structure and algorithm based on a linear key for a rectangle retrieval problem. Computer Vision, Graphics, and Image Processing, 1983, 24(1):1~13.
- [9] Wang, Cheng-wei. Virtual reality and distributed interactive simulation. Computer Research and Development, 1998, 35(12):1058~1063 (in Chinese).

附中文参考文献:

- [9] 汪成为. 虚拟现实和分布交互仿真. 计算机研究与发展, 1998, 35(12):1058~1063.

Region Matching Algorithm Based on Hierarchical Dividing of Routing Space*

SHI Yang^{1,2}, LING Yun-xiang³, JIN Shi-yao³, HU Hua-ping³

¹(Software Center, Navy Submarine Academy, Qingdao 266671, China);

²(Automation School, Harbin Engineering University, Harbin 150001, China);

³(School of Computer, National University of Defence Technology, Changsha 410673, China)

E-mail: shi_yang@netease.com

Abstract: With the simulation applications developing in extent and profundity, Inrerner-based distributed simulation becomes the trend in the future. It faces the challenge of system scalability. Data filtering is an efficient method to improve system scalability. In this paper, after discussing the problem of system scalability, the principle, the implementation, the problems of data filtering under HLA (high level architecture) are introduced. In order to solve the problem of too much matching computation, a novel method is proposed to reduce region match computation in data filtering based on hierarchical dividing of routing space. This method can reduce matching computation by using region location knowledge to execute efficient prune searching.

Key words: distributed interactive simulation; high level architecture; scalability; data filtering; hierarchical dividing of routing space; heuristic prune searching

* Received November 30, 1998; accepted February 25, 2000

Supported by the National Natural Science Foundation of China under Grant No. 69525265; the Defence Pre-Research Project of the 'Ninth Five-Year-Plan' of China under Grant No. 16. 2. 6. 2