

# 一个基于异构型局域网的雕塑实体布尔操作并行算法

唐敏 董金祥

(浙江大学 CAD/CG 国家重点实验室, 人工智能研究所, 杭州 310027)

**摘要** 为了在某些限时操作下得到实时交互的速度, 提出了一个分布式的雕塑实体模型布尔操作并行算法。该算法在各个阶段抽取的并行性和原串行算法的拓扑相关性之间采取了折中的策略, 并对于不同的并行任务, 采用了不同的任务均衡手段。该算法已经在由 10 台 P II /350 微机和由 5 台 SGI O2 工作站组成的异构型局域网中得到实现。实验结果显示, 该算法获得了理想的加速比, 对于复杂雕塑实体的造型, 操作平均可在 3s~4s 完成, 因此能满足实时交互的需要。

**关键词** 计算机并行处理(520·3020) 雕塑实体 布尔操作 限时操作 任务均衡

**中图分类号**: TP391.9 **文献标识码**: A **文章编号**: 1006-8961(2003)03-0292-07

## A Distribute Algorithm for Boolean Operations of Sculptured Solids Based on a Heterogeneous LAN

Tang Min, Dong Jin-xiang

(State Key Laboratory of CAD&CG, AI Institute, Zhejiang University, Hangzhou 310027)

**Abstract** To fulfill the demand of some time-critical operations, we presented a distribute algorithm for Boolean operations of sculptured solids. The algorithm exploits parallelism at all stages of the Boolean operation between sculptured solids, and made a good compromise with the relevancy of the topological entities involved in. And different load balancing strategy is used according to different kinds of the distribute tasks undertaken. The algorithm can distribute the notoriously computation cost of Boolean operations between complex sculptured solids on a heterogeneous LAN, and have been implemented in a heterogeneous LAN which is composed by 10 PentiumII/350 PCs and 5 SGI O2 workstations. A good degree of parallel computation can get from the experiment datum. The algorithm is used in a modeling kernel for a feature-based parametric modeling system named GS-CAD, and is able to interactively evaluate the boundaries of Boolean operations of complex sculptured solid in average 3-4 seconds. It can also be easily extended to collision detection algorithm between sculptured solids.

**Keywords** Sculptured solids, Boolean operations, Time-critical operations, Load balancing

### 0 引言

在一些虚拟制造系统中, 由于允许用户用布尔操作来对产品模型进行实时操作, 因而对系统的计算性能提出了更高的要求, 即在进行产品造型时, 实时交互式设计是用户的迫切要求, 但鲁棒精确的雕塑实体的布尔操作一直是造型系统的效率瓶颈<sup>[1]</sup>, 其原因一方面是由于单机的硬件性能难以满足需

要, 而另一方面则由于它常常存在于一个多机的局域网环境中, 而网络中的其他机器可能正处于闲置状态, 因此, 采用分布式并行算法是加快布尔操作的一个有效方法。

采用分布式系统有许多潜在的优点, 其主要是:

(1) 能提高系统的可用性, 如当系统中的某些机器出现故障时, 可以采取有可能降低性能的方法使系统不停止工作;

(2) 能提高系统的坚固性;

基金项目: 国家教育部博士点基金资助项目(2000033553); 浙江省自然科学基金资助项目(6001107)

收稿日期: 2002-02-28; 改回日期: 2002-10-08

(3) 具有高灵活性,便于扩充或修改;

(4) 具有高性能,这主要是通过并行处理来得到.由于一个分布式系统,不但有并行处理能力,而且可采用分布式控制,因此可以避免各类“瓶颈”与同步等待,所以有潜在的高性能;

(5) 有较佳价格性能比.由于超大规模电路技术的飞速发展,单个芯片的功能越来越强,而价格却越来越低,因此用这种芯片或微型机(或工作站)来组成分布式系统,其价格/性能比往往比单机系统低.

对于布尔操作算法,国外主要用于大型 CSG 模型到 B-rep 模型的边界求取<sup>[1~4]</sup>.目前对于雕塑实体布尔操作的研究主要集中于计算 Bspline 曲面或 Bezier 曲面对应的求交问题<sup>[5~8]</sup>.文献[9]中提到了用 HutDesign 系统进行交互式实体造型的问题,但并行的布尔操作算法在文献中却很少提及.最新的工作是 Krishnan 在一台 4 个 CPU 共享主存的 SGI 工作站上,实现了交互式的 CSG 模型到 B-rep 模型转换的系统<sup>[10]</sup>.该算法是针对多 CPU 共享主存的硬件体系结构而设计的,虽然受硬件价格条件的限制,一般计算机的 CPU 个数很难达到很多,但算法对于实体模型的拓扑相关性也未能充分利用,并且在鲁棒性方面仍存在问题<sup>[11~12]</sup>.

基于异构多机局域网分布式雕塑实体布尔操作的并行算法,是在原串行算法<sup>[11~12]</sup>的基础上,通过提取各阶段的并行性和针对原算法的拓扑相关性,同时采用折中的策略,并对于不同的并行任务,采用不同的负载均衡手段来将任务分布到不同的处理器上,以获得实时计算效果.另外,对于常见的网络出错情况,算法也进行了相应的处理,因而能确保任务的最终完成.

### 1 模型表示

将实体模型的拓扑结构表达为半边数据结构,它是雕塑实体几何信息放置的框架.图 1 中给出了雕塑实体模型,大家知道,每个实体在拓扑结构上,由面、边、顶点组成,由图 1 可见,其中面包含一个或多个环,而环由半边构成,每条边分为一对方向相反的半边.其几何信息由分别附加在面、边、顶点上的裁剪曲面、空间曲线、空间点信息组成.

体模型中,面的几何信息为一张参数裁剪曲面,而边的几何信息为三维空间  $R^3$  上的一条空间样条,这是为了避免两个邻接面分别定义在曲面参数域上

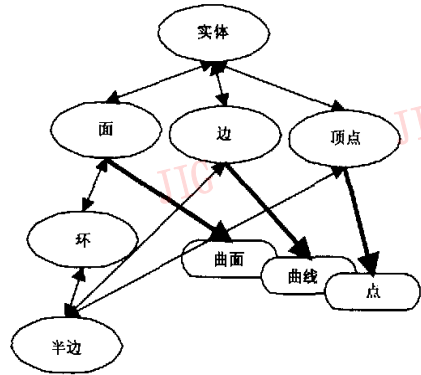


图 1 雕塑实体模型

的边界曲线,由于误差而产生裂纹的情况发生,其顶点的几何信息为  $R^3$  上的一个位置矢量.

为了精确表达雕塑曲面,每个实体面的几何信息可通过一组参数裁剪曲面片来定义,用张量积有理 Bezier 曲面  $F(s, t)$  来表示每个曲面片:

$$F(s, t) = \frac{\sum_{i=0}^m \sum_{j=0}^n \omega_{i,j} V_{i,j} B_i^m(s) B_j^n(t)}{\sum_{i=0}^m \sum_{j=0}^n \omega_{i,j} B_i^m(s) B_j^n(t)}$$

$(0 \leq s \leq 1, 0 \leq t \leq 1)$

其中,  $V_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$  为曲面片的控制顶点,

$B_i^m(s) = \binom{m}{i} s^i (1-s)^{m-i}$  为 Bernstein 多项式,而

$F(s, t)$  可看成是从参数空间  $(s, t)$  平面到三维空间  $R^3$  的一个映像.其裁剪曲面是通过原参数空间和其上的裁剪曲线来定义,由于裁剪曲线一般是高阶曲线,不能精确参数化,因此本文用参数域上的分段线性链来近似表示,其得到的裁剪区域为曲面参数域上的多边形,为了表示紧凑,在一定误差范围内,可用样条来近似高阶交线,这样裁剪区域就可通过曲面参数域上的一组样条来定义.

### 2 算法概述

算法采用了 Supervisor-Worker-Collector 模型(如图 2 所示),其中 Supervisor 负责任务划分分配和处理机状态控制,Worker 完成子任务的计算工作,计算结果由 Collector 统一收集.

Supervisor-Worker-Collector 模型是广泛使用的 Client-Server 模型的一种改进模型,与后者相比,该模型具有以下优点:

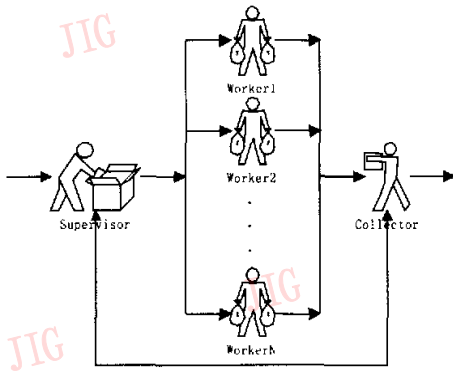


图 2 分布式系统结构模型

(1) 具有更大的灵活性;

(2) 除了支持任务级并行系统的实现外,还支持算法级并行系统的实现;

(3) 坚固性好,并且容易使负载平衡.

整个算法大体分为如下 3 个阶段<sup>[10]</sup>:

(1) 求交检测 首先对两个输入实体的几何元素顶点、边、面,采用包围盒测试和线性规划法来去掉不相交的情况,并记录可能相交的元素对,同时计算每个曲面片的包围盒,因为每个曲面片的包围盒可独立计算,所以这一步可并行执行,若某对包围盒不相交,则相对应的曲面片必不相交(有理 Bezier 曲面片的凸包性<sup>[13]</sup>),此时冗余面对可通过包围盒的简单排序而去掉;然后,采用如下线性规划方法寻找相交曲面片:如果两个曲面片之间存在一个分离面,则曲面片不相交,由此,即可去除包围盒之间存在分离面的面对;同时使用 Seidel 的线性规划算法<sup>[14]</sup>来寻找相交曲面片.这样对实体实施了上两步操作后,只余下了少数有较大相交可能的面对.

(2) 曲面片求交和分割 对于求交检测后所记录的可能相交的元素对,按照顶点、边、面的层次关系进行几何求交,即按 Vertex-Vertex, Vertex-Edge, Vertex-Face, Edge-Edge, Edge-Face, Face-Face 求交的求交次序来确保拓扑分割的一致性.另外,为了提高求交速度和稳定性,还根据面面相交情况的分析结果,沿已求得的交线走向进行跟踪求交.在求交过程中,需适当插入新顶点、边,并通过分割曲面片来产生新曲面片,其得到的交线,按照次序记录在交线链中.由于每个面对的求交运算彼此独立,因此这步操作可并行执行.对于参数曲面求交,系统采用了解析法和步进法相结合的方法将求交问题代数化,即通过把交线的投影作为一条代数平面曲线来求出其

上点切向量的性质,算法是利用矩阵计算来跟踪交线,其跟踪过程是几何操作,即通过计算平面曲线来降低几何复杂度,其所采用跟踪算法可以在跟踪阶段检测出奇异点的存在,从而可精确高效地得到交线.曲面片的交线可以看成是参数域上的开环和闭环(如图 3 中的  $l'$  和  $l$ ).其中,对于开环,它必然与曲面边界有一个交点,这样再结合层次求交技术,这个交点可通过 Edge-Face 的求交,即曲线和曲面的求交得到,本文用它作为跟踪的初始迭代点;而对于闭环,则用 Hohmeyer 提出的基于 Gauss 图<sup>[6]</sup>进行判别,然后通过将有环曲面分割为多个子曲面,将其转化到开环的情况.

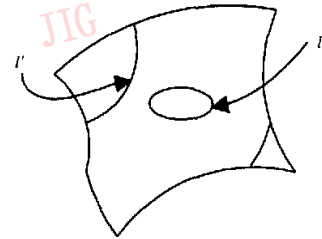


图 3 交线环分析,  $l'$  和  $l$  分别对应于参数域上的开环和闭环

(3) 区域分类和生成实体 由于参加布尔操作的两个实体封闭紧凑,因此上一步得到的交线链必然是实体边界上的一系列闭环,且它们把实体划分为独立的边界部件.生成实体时,先进行区域分类,判别每个边界部件相对于另一个实体边界的 in-out 关系,然后根据布尔操作类型来选择适当的边界部件,以便沿着交线环进行边界粘合,最后生成结果实体.由于构成每个组分的所有曲面片相对另一个实体都具有相同的内/外特性,因此对于每个组分,仅需计算其中任意一点的内/外特性即可;若一个实体是闭合,且不自交的,则可根据经过该组分中任一点的一射线与实体的交点个数来判断其内外特性,若交点个数为奇数,则点在体内,否则点在体外.为了得到交点个数,必须对另一个实体的每个曲面片都进行线面求交计算.为避免对另一个实体的每个曲面片的重复耗时计算,本文使用了各组分之间的相连信息(在上一步中求得),以使每个实体只需操作一次,而且其基于光线投射的曲面片的组分分类可并行执行.另外,类似求相交面对,本文使用包围盒计算和线性规划算法来加速运算.

图 4 以两个轴向垂直的圆柱的并操作为例,演示了在 3 个 Worker 上运行的并行算法的流程.

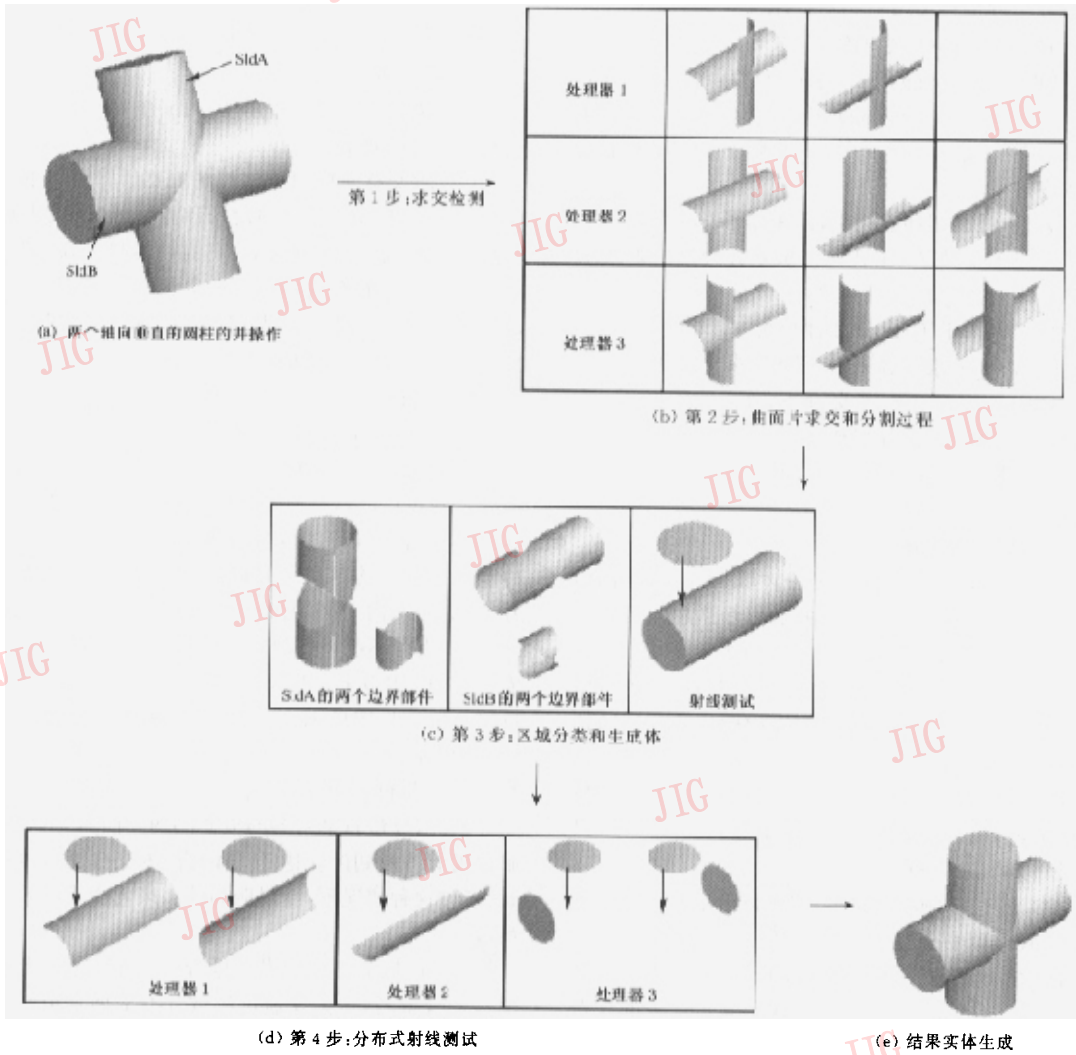


图 4 分布式 Boolean 操作实例

### 3 并行处理技术

#### 3.1 拓扑相关性

在串行算法<sup>[11,12]</sup>中,是利用拓扑层次结构来进行层次求交.由于在实体中,高维元素的边界是通过低维元素来定义的,因此在求交过程中,应先完成低维元素间的求交分割,然后再完成高维元素间的求交,这个过程称为层次求交.例如一个面与另一个面求交的结果需依赖于它与定义该面的边、顶点相交的结果,另外,低维的每条边也会参与其两个高维邻面与其他表面的求交,如果重复计算它与其他面的

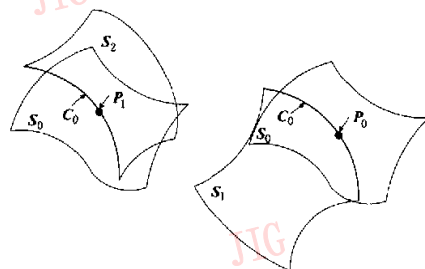


图 5  $S_1, S_2$  分别与  $S_0$  求交,因误差引起交点  $P_0 \neq P_1$ , 曲线  $C_0$  被  $S_0$  分割不一致

交点,则不仅费时,还可能引起分割的不一致性(参见图 5);由图 5 可见,曲面  $S_1, S_2$  分别与  $S_0$  求交,由

于边界空间曲线  $C_0$  两次与  $S_0$  求交, 又因误差引起交点  $P_0 \neq P_1$ , 同时引起曲线  $C_0$  被  $S_0$  分割不一致, 因此, 采用层次求交的算法, 将求交过程按照 Vertex-Vertex 求交、Vertex-Edge 求交、Edge-Edge 求交和 Vertex-Face 求交、Edge-Face 求交、Face-Face 求交的次序进行。

另外, 根据拓扑连续性, 还采用了交线跟踪分割技术来求出交线链, 而且在区域划分阶段的射线求交算法中, 也利用了拓扑连续性。

由于利用了拓扑层次结构, 在并行算法中, 也按 Vertex-Vertex 求交、Vertex-Edge 求交、Edge-Edge 求交和 Vertex-Face 求交、Edge-Face 求交、Face-Face 求交的次序来分配任务。同样, 对于拓扑连续性, 则针对交线跟踪的情况来动态生成求交子任务, 并分配到各处理器。

### 3.2 任务均衡策略

当一个算法需要在多个处理机上并行运行时, 便产生了均衡负载的问题, 由于并行算法的运行时间由各台处理机工作时间的最大值决定, 因此, 最高效的算法中, 任务应平均分配至各个处理机。由于没有各种并行算法的通用解法, 故均衡负载的问题长期以来一直受到关注。由于不同解法的效率因应用特性的变化而不同, 因此, 需要一种新的基于问题特性的分析方法, 用于选择最有效的均衡负载技术。以下是可采用的 3 种方法<sup>[10]</sup>。

(1) 静态均衡负载法 即先将给定的  $n$  个任务分为  $p$  (处理器个数) 组, 然后对每个处理器分配一组任务的方法, 其每个问题组的规模可预先计算并且在运行过程中是不变的。这样当满足任务的耗费时间预先可确定时, 且在执行中, 任务数不变的条件, 此算法最优。

(2) 全局队列法 许多算法中, 由于要预测每个任务执行时间是不可能的, 例如, 求两个面的交线的计算将随着曲线组成数, 长度 (由需分析的点数决定) 的变化而变化, 因此在这种方案中, 当一个处理器从队列取任务时, 需采取将队列锁住的方法, 以保证互斥访问。尽管附加的置锁操作削弱了平衡负载带来的优势, 但此类方法仍是最优的。

(3) 动态均衡负载法 此方法中, 每个处理器都有一个局部队列。其任务的最初分配虽类似静态均衡负载, 然而由于任务分配的局部最优性, 使某些处理器将优先完成任务。此时, 由于空处理器将分担忙处理器的任务, 于是平衡负载被动态地实现。如果

能保证在任何时间, 忙处理器最多只被一个空处理器访问, 则不须置锁。同时要求每个任务只被处理一次, 且没有任务遗漏。

而针对算法如下不同阶段的工作, 采用了如下不同的负载均衡算法:

(1) 求交检测阶段 该算法主要计算量集中在几何元素顶点、边、面包围盒的计算与测试, 因为每个曲面片的包围盒可独立计算, 且计算量大致相等, 所以这一步可静态地实现均衡负载, 即可简单地将总任务大致平均地分配到各处理器, 同样, 包围盒的测试也可静态地均衡负载并行执行。

(2) 曲面片求交和分割阶段 这是算法计算量最大的部分, 对于通过了上面检测的几何元素对, 则先按照层次关系进行几何求交, 并在求交过程中, 适当插入新顶点、边, 并通过分割曲面片来产生新曲面片; 然后将得到的交线, 按照次序记录在交线链中。由于每个面对的求交运算彼此独立, 故这步操作可并行执行, 但由于必须按照拓扑层次关系次序来进行求交, 因此不能采用静态均衡负载方法, 同时由于需要对曲面片进行分割, 即需修改原有的曲面片, 这可能会改变后续的求交过程, 所以采用对曲面进行加锁的方法来保证基于互斥操作的全局队列均衡负载算法的顺利执行。具体实例为, 在一对曲面进行求交分割时, 算法自动对参与分割的两个曲面置锁, 以确保没有涉及到的其他任一曲面的求交分割可以同时进行。这种情况下, 全局队列的负载均衡策略效率最高。

(3) 区域分类和生成实体阶段 在经过了上面的求交分割后, 还需要通过区域分类来确定由曲面片组成的边界部件的 in-out 关系。这个阶段计算量用于进行射线与实体的求交, 同样射线求交时, 也采用了类似求交检测阶段的包围盒计算和测试方法, 这部分可以采用静态均衡负载法来并行执行, 而射线和曲面的求交工作, 由于互相独立, 并且不会修改曲面片, 因此可以简单地采用静态均衡负载方法来并行执行。

### 3.3 算法容错性

执行算法时, 有 Client-Server 方式, 或 Multi-Agent 方式两种选择。它们各有特色:

(1) Client-Server 模式 该模式下所有的 Client 均以一个高性能的 Server 为核心进行处理, 在 Client 中, 只进行简单的处理工作, 而大量的数据、复杂的计算都在 Server 上进行。

(2) Multi-Agent 模式 该模式下所有的 Agent 均以平等的地位出现,即各 Agent 互相联系,使得任务平衡分配,以共同完成计算任务。

总体上说,在系统性能上,虽 Multi-Agent 模式要优于 Client-Server 模式,但比较而言,Client-Server 模式较之更易实现,而且在一个处理机性能相差较大的环境中,特别是远程用户通过 WWW 服务来进行零件设计时,Client-Server 模式也存在着优越性,而对于局域网中的并行设计来说,Multi-Agent 显然是最佳选择。

在设计算法时,由于考虑到了分布式系统的特征,因而使得算法在部分物理处理机出现失败时,仍能运行,即具有容错性。

在算法运行的局域网中,部分物理处理机出现失败的主要原因有<sup>[15]</sup>:

- (1) 其他用户所在的处理机重新启动;
- (2) 其他用户处在 kill 外部进程;
- (3) 其他用户关机。

为了确保当某一 Worker 失败时,系统能再恢复其计算任务,以保证计算结果的完整性和正确性,现采用了以下机制:

(1) Supervisor 对发送给 Worker 的每个子任务赋予一个唯一标识号。

(2) Supervisor 维持一子任务 Cache,用来保存当前 Worker 正在处理的子任务。

(3) 定时对子任务 Cache 进行扫描,如果某个子任务在 Cache 中的时间超过一给定值,那么就可以认定该子任务的处理机已失败,此时,需把该子任务再加入到全局任务中。

(4) 当 Worker 回送子任务的结果时,必须附上相应子任务的唯一标识号,以便由 Super-visor 把已完成的子任务从子任务 Cache 中删除。

#### 4 系统实现

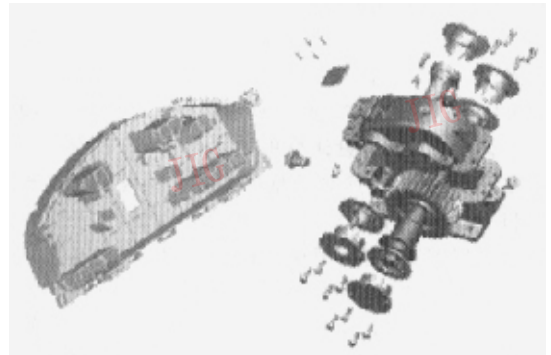
该算法已经在由 10 台 P II 350 微机和 5 台 SGI O2 工作站组成的异构型局域网上了实现了。该算法是对 GS-CAD98 系统中的造型核心——雕塑实体布尔操作的改进。GS-CAD98 是浙江大学人工智能所开发的一个基于 NT 平台的参数化特征造型系统,其底层网络通讯部分采用 Socket 编程,代码用 C/C++ 实现,而微机的操作系统可采用 WindowsNT/Windows95 或 Linux,工作站的操作

系统为 Unix。

### 5 实验结果与结论

#### 5.1 实验分析

图 6 为两个布尔操作造型实例,其中图 6(a)为一个由 143 次布尔操作生成的仪表盘。图 6(b)为一个由 42 个较简单的子部件组成的变速箱,其总共经过 212 次布尔操作生成。



(a) 仪表盘(零件 A) (b) 变速箱(零件 B)

图 6 两个布尔操作造型实例

算法在不同个数处理机下的实施效率如表 1 所示。

表 1 算法在不同个数处理器下的实验数据

任务	处理机个数	运行时间 (s)	加速比	效率 (%)
零件 A	1	34.66	1	100
	2	17.38	1.99	99.17
	3	11.66	2.97	99.09
	4	8.76	3.96	98.92
	5	7.06	4.91	98.19
	6	5.90	5.87	97.91
	7	5.10	6.80	97.09
	8	4.46	7.76	97.02
	9	3.98	8.71	96.76
	10	3.61	9.60	96.01
零件 B	1	32.92	1	100
	2	16.52	1.99	99.64
	3	11.05	2.98	99.31
	4	8.32	3.96	98.92
	5	6.70	4.91	98.27
	6	5.61	5.87	97.80
	7	4.85	6.79	96.97
	8	4.25	7.75	96.82
	9	3.80	8.66	96.26
	10	3.42	9.63	95.97

其中加速比和效率定义如下:

$$R = t_1/t_2$$

$$P = R/n$$

其中,  $R$  为加速比,  $t_1$  为主从模式下实际执行时间,  $t_2$  为  $n$  个主体协作的实际执行时间,  $P$  为算法实施效率。

由表 1 可见, 当增加处理机个数时, 算法加速比提高较大, 且随着处理机个数的不断加大, 效率始终保持在 95% 以上, 可获得近似线性的加速比。

### 5.2 与同类算法的比较

与文献[10]的工作比较, 由于文献[10]算法是针对多 CPU 共享主存的硬件体系结构而设计的, 其可以共享内存资源, 因此算法实现紧凑。虽然受硬件条件的限制, 一般计算机的 CPU 个数很难达到很多, 但算法对于实体模型的拓扑相关性也未能充分利用, 另外, 在鲁棒性方面仍存在问题, 即经常算法会出现破面、裂纹等错误情况<sup>[11~12]</sup>, 由于本文算法可以利用异构型局域网内的任何空闲计算机 CPU 资源来很容易地将工作量分布到多台计算机上, 以获得理想的加速比, 并且由于利用了实体模型的拓扑相关性, 因此算法的鲁棒性大大提高, 其对于复杂雕塑实体的造型操作平均可在 3s~4s 完成。

### 5.3 结论

本文提出并实现了一个用于分布式雕塑实体模型布尔操作的并行算法, 该算法在保证稳定性和精确性的同时, 还充分发挥了并行性。实验结果显示, 该算法获得了理想的加速比, 由于其对于复杂雕塑实体的造型操作平均可在 3s~4s 完成, 因此能满足实时交互设计的需要。该算法思想也可以推广应用到雕塑实体模型的其他操作, 如局部操作、抽壳、物性分析、离散分割等, 这将是下一步研究的目标。

### 参 考 文 献

- Hoffman C H. Geometric and solid modeling[M]. San Mateo, California, USA: Morgan Kaufman Publishers, 1989.
- Mantyla M. An introduction to solid modeling[M]. Rockville, Maryland, USA: Computer Science Press, 1988.
- Requicha A A G, Rossignac J R. Solid modeling and beyond[J]. IEEE Computer Graphics and Applications, 1992, 12(5): 31~44.
- Requicha A A G, Voelcker H B. Boolean operations in solid modeling: Boundary evaluation and merging algorithms [J]. Proceeding of the IEEE, 1985, 73(1): 30~44.

- Hu Chun-Yi, Maekawa T, Patrikalakis N M *et al.* Robust interval algorithm for surface intersections[J]. Computer Aided Design, 1997, 29(9): 617~627.
- Hohmeyer M E. Robust and efficient intersection for solid modeling [D]. Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA, 1992.
- Krishnan S, Manocha D. An efficient surface intersection algorithm based on the lower dimensional formulation [R]. Technical Report TR94-064, Department of Computer Science, University of North Carolina, USA, 1994.
- Sederberg T W, Nishita T. Geometric hermite approximation of surface patch intersection curves[J]. Computer Aided Geometric Design, 1991, 8: 97~114.
- Mantyla M, Ranta M. Interactive solid modeling in HutDesign [A]. In: Proceedings of Computer Graphics[C], Tokyo, Japan, 1986.
- Krishnan S, Gopi M, Manocha D *et al.* Interactive boundary computation of boolean combinations of sculptured solids [J]. Computer Graphics Forum, 1997, 16(3): 67~78.
- 唐敏, 董金祥. 雕塑实体的精确求交算法[J]. 计算机学报, 2000, 23(4): 434~439.
- 唐敏, 董金祥, 李海龙等. 非正则精确模型的布尔操作[J]. 软件学报, 1999, 10(12): 1291~1297.
- Farin G. Curves and surfaces for computer aided geometric design: A practical guide[M]. New York, USA: Academic Press, 1993.
- Seidel R. Linear programming and convex hulls made easy[A]. In: Proc. 6th Ann. ACM Conf. on Computational Geometry [C], Berkeley, California, USA, 1990: 211~215.
- 潘志庚. 分布式并行图形处理技术及其应用[M]. 北京: 人民邮电出版社, 1997.



**唐敏** 1974年生, 1999年于浙江大学计算机系获博士学位, 现为浙江大学计算机系副教授。主要从事 CAD、几何造型等领域的研究工作。



**董金祥** 1945年生, 浙江大学计算机系教授, 博士生导师。主要从事 CIMS、用户界面、图形学等领域的研究工作。