

立体几何造型系统图形数据库

梁友栋 胡希明 毛根生
(浙江大学)

摘 要

一个实用的、完整的几何造型系统必须有一个精心设计的图形数据库作为图形信息的存贮和管理中心。本文以几何体的边界表示法为基础,提出了一个几何造型系统的图形数据库设计方法,并介绍了该方法在 UNIFY 关系数据库管理系统上的实现及 C 语言接口设计。

一、引 言

近些年来,由于 CAD 技术在许多工程系统中得到成功的应用,以及软件、硬件技术的发展,促使立体几何造型系统的研究发展极为迅速。目前已有比较成功的系统如美国的 Romulus, PADL, 日本的 TIPS, 西德的 Compac, 法国的 Euclid 及英国的 Build 系统等。据 1984 年 9 月在西德召开的国际计算机图形学及应用会议的一份报告^[1], 1984 年初在全世界约有 350 个具有立体造型功能的工作站, 到 1984 年底就猛增到 1200 个。据国外一些专家估计, 五年后立体几何造型将成为工程设计和加工的日常基本手段^[2]。当前立体造型系统的研究有如下几个特点:

1. 实用性。三、四年前许多系统, 包括 Build, Euclid, Romulus 等, 都还只是学术性研究系统, 目前均已逐步发展成为在 CAD 中显示强大生命力的实用性系统, 并且开始成为商品化的系统。美国的 Romulus 系统已出售给 Douglas、航天中心、波音公司等。

2. 几何造型与软件工程新技术相结合。与二维的绘图相比, 几何造型系统需要处理复杂得多的几何与拓扑的关系, 需要处理大量的数据, 软件技术的新发展, 特别是数据库技术的发展, 使得可以发展建立数据库系统, 有效地处理大量数据, 避免数据的重复, 而且易于存取^[3]。许多几何造型系统在原有基础上都已经或正在改在数据库上工作。例如西德的 Compac 系统是在层次数据库上工作, 如图 1 所示。日本东京大学 1984 年发表的 Ceomap III 系统是在关系数据库上工作。原来在 70 年代发表的 Geomap II 是在网状数据库支撑下工作的^[4]。

3. 系统以及系统的有效使用。Computer Graphics 1984 年 3 月的一期上刊登的一篇美国科学基金会的报告^[5]说, 几何造型研究的重点已开始从体的表示, 特定的算法等转移到系统以及系统的有效使用的研究。主要包括如下几个方面: (1) 多种表示的造型系统, 特别是几何体的造型与曲面造型相结合。例如美国犹他大学过去只研究曲面造型, 现

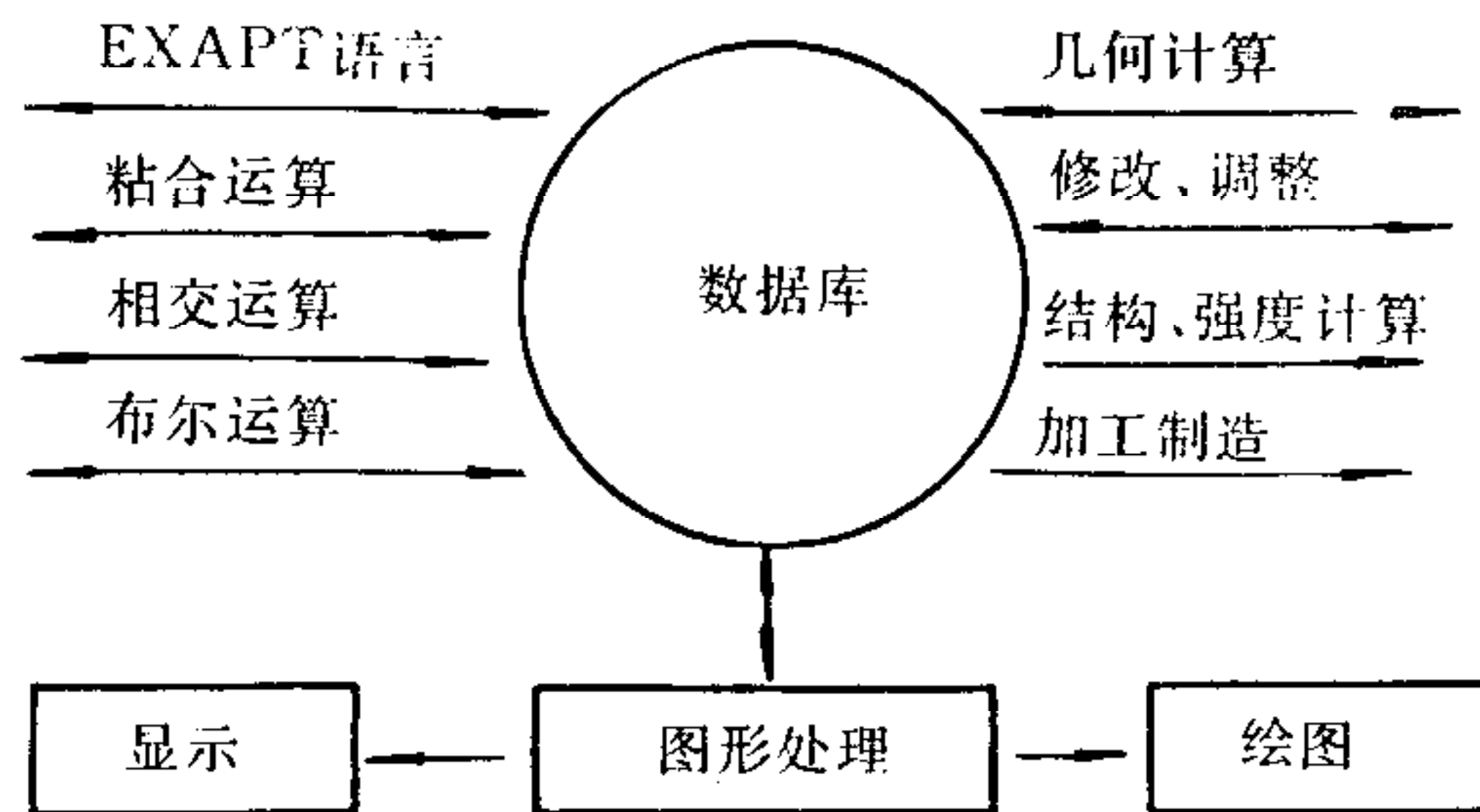


图 1

在正在研制与三维几何造型相结合的 Alpha 系统。(2) 干涉的检查亦即判断交集是否为空的算法,这与求交算法不同,它不是关心交集是什么,而是判断是否相交。它在加工问题和决定机器人运动路径等问题中是很重要的。(3) 具有真实感的彩色图象。新近发展起来的光线投射与光线跟踪算法由于真实感强,正越来越吸引人们普遍的兴趣。(4) 图形交互技术,特别是方便的人机接口,菜单功能以及输入输出等,这对几何造型系统的实用性是极为重要的。

4. 造型算法和数据库的硬件化。集合的布尔运算和数据库的操作相当费时,这已成为当前 CAD 技术进一步应用的重要障碍。所以算法和数据库系统的硬件化正成为当前图形领域发展的一个新动向。

二、体的数学模型与信息模型

人们已做了许多努力来解决体的定义问题,日本 TIPS 系统采用半空间的并、交、差布尔运算定义几何体。美国 Rochester 大学 Requicha 定义体为 E^3 中的有界、正规和半解析的闭集^[6],同时定义正规运算使得体的全体在正规运算下是封闭的。可是体的这些定义存在如下几个问题:(1) 不完整性。用集合的布尔运算定义体掩盖了最本质的体、面、边、点之间的关系。由此定义较难推出这些元素间的几何与拓扑关系。(2) 不便于计算机表示和运算。由于用内点的闭包来定义体和体的运算,需要作极限运算,带来极为复杂的计算,不适合于计算机的处理。

我们力图用代数拓扑与微分几何的方法来解决这些问题。在 [7] 中,已经证明了具有定向的曲面片簇可唯一确定一簇闭曲面,由此可给出体的定义如下:称具有定向的曲面片簇所唯一确定的一簇闭曲面所包围的区域为体。若此曲面族只包含一个闭曲面,那么称此体为单体。[7] 中还给出按拓扑分类的判别条件和算法以及检查作为体和单体的充要条件。根据体的上述定义,可以建立体的如下信息模型,见图 2。其中 V 代表体, F 代表面, E 代表边, P 代表点。上图不是树结构而是一般联系表示。暂时撇开每个元素的几何含义,用整数连续编号代表各层上各别的元素,图中每一条边表示高一维的元素(即上一层的元素)与其所支撑(拓扑意义下)的低一维边界上的元素之间关系。这些关系满足定义中的条件(而且只有这些条件)。因此体的信息模型实质上就是关系的关系,并且图 2 所表示体的信息模型是体的最简化的信息模型。每一个元素及其关系在“关系的关系”

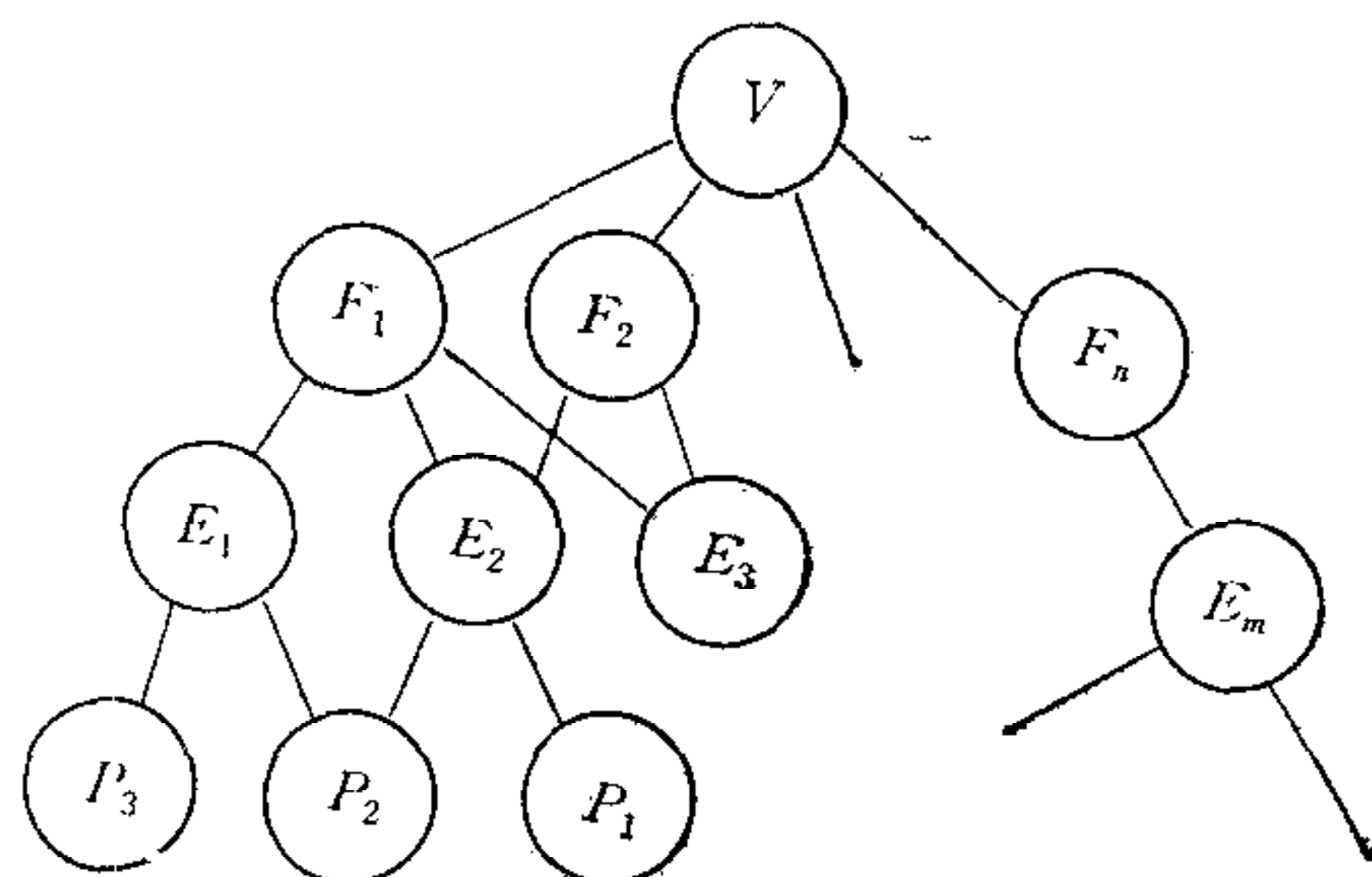


图 2

下都是独立的。用四层元素及它们之间的三个联系,可用七个表来代表几何体,每个表都是独立的,设 $X_i(i=1, 2, \dots, 7)$ 表示七个表,那么体就可以表示为 (X_1, X_2, \dots, X_7) , 即七维空间的一个点。反之,满足给定关系的 7 个表亦代表某一物理几何体。这样,这些表所代表的元素全体就构成七维空间的一个集合。

三、图形数据库

在大型的复杂的图形处理系统中,人们逐步地发现用数据文件的方法管理和存贮这些数据带来处理多个文件的复杂性,从而使得系统设计的统一性、规模和功能都受到限制。由于附属于程序的数据文件的数据结构只对某一特定程序有意义,而其他程序则无法利用这些数据。各自独立的多个程序数据文件使得管理和查找数据花费大量的时空代价,并且大大增加了程序的复杂性。因此,在造型系统设计时必须考虑统一的数据结构,以便能为系统中各种功能的图形处理程序提供必要的统一的信息。为了支持这样复杂的数据结构,需要有保存和管理这些信息的中心,最好的方法是建立图形数据库。图形数据库至少必须具备下列功能:

- 能够有效地存贮系统所需要的所有有关的数值和非数值的几何及拓扑图形数据。
- 各种应用程序(例如布尔运算等)在运行期间能够查找、增加、删除、和修改数据库内的数据。
- 针对图形处理的特点,能一次性地给出或存入一个体或一幅画面的具有某一特定条件的整体数据。

建立图形数据库的首要问题是进行数据模型的设计。非图形信息(如为了帮助图形显示的说明、文字、标注、表格等)可采用通常事务处理数据库系统的方法进行设计。而图形信息如前所述,最终可归结为边界定义和表示,即点、线、面、体之间的相互联系的表达及相应坐标值的确定。全面考虑这些信息的存贮以及系统图形处理的要求(如消隐、集合运算等)进行合理的数据模型设计,以得到概念模型(Conceptual Model)。对概念模型的计算机语言描述就成为总模式(Conceptual Schema)。

自 Codd, E. F.^[8] 提出关系数据库的概念,和一些有效的关系数据库管理系统(DBMS)^[9] 出现后,目前许多图形数据库设计都用关系模型。关系结构的数据库系统在模型描述上较为简单,易被一般用户理解。关系的方法也很适合于如前提出的“七维空

间”的概念。数据的操作过程对用户来说只要提出查找的需求,而不必描述查找的过程,查找过程由系统自行优化实现。另外对数据库模型的修改较少或不影响原来的数据和程序,这使得系统设计有更大的发展余地。最突出的是,对关系数据库的查找和存贮可以以整个数据文件或大批数据为整体一次完成,这也最适宜于图形处理的数据要求。

四、图形关系数据库的实现

在当前各档计算机中,配有或正在配备通用数据库软件系统。该软件本身已提供描述模式的语言(DDL)及操作管理数据的语言(DML)。因此可以作如下的考虑:(1)建立合理的能应付各种图形处理要求的数据模型,通过DDL描述成数据库模式,从而建立图形数据库。(2)根据不同的图形处理的需要提供数据库内的数据或把图形处理后的结果存入数据库,为此必须用DML设计相应的一组数据操作程序。(3)最重要的一点,必须设计图形处理程序所依赖的计算机语言与数据库操作语言的接口。对以上的考虑,笔者在UV/68超级微型计算机上作了图形数据库的实现。在UV/68系统上,配有关系数据库系统软件UNIFY^[10]。UNIFY具有强有力的用户使用接口,对数据库的操作用户可使用SQL(Structure Query Language)语言进行查询,亦可用C语言编写应用程序通过接口操作数据。此外,它还向用户提供对数据的完整、安全性和使用环境等管理的手段。

在用UNIFY设计图形数据库模型时,考虑如下的因素:在数据库内既可以存贮基本的体和简单的基础图形,又可以存贮经过图形处理的复杂图形的数据。在输入手段上,可以对简单图形直接的数据输入,又可以用图形处理手段,如Sweeping方法,图形的集合运算方法或光标(光笔)的方法输入图形,变换成数据后存入数据库。存贮的数据及提供数据的手段能适应各种图形处理的需要。对用户可按名存取几何体而不涉及繁琐的具体数据。根据以上的考虑,运用前面提出的几何体的边界定义和表示的理论,建立如下的关系数据库模型。在各模型中,各几何元素的编码均以单个体为标准,不采用连续编码。

Object (Objkey*, Objname)

Object 为屏幕图形关系,一个屏幕可以有多个独立的几何体图形。

Volume (Volkey*, Volname, Volwgx, Volwgy, Volwgz, Voltyp, ...)

Volume 为几何体表达,每个几何体有不同的体号。Volwgx, Volwgy, Volwgz 为体的重心坐标,可以缺省,或由程序调用装入函数来装入。其用处是可为画面体之间的覆盖。Voltyp 为几何体的类型,可以为多面体、旋转体、二次曲面体等。程序可据此识别而采用不同的处理算法。

OVrela (Objkey, Volkey, ...)

OVrela 关系表达一幅屏幕由那些独立几何体构成。以上的Volume及以下六个关系,构成七维空间的体表达:

Face (Facekey*, Facetyp, ...); VFrela (Volkey, Facekey, ...)

Edges (Edgekey*, ...); FErela (Facekey, Edgekey, ...)

Points (Pointkey*, ...); EPrela (Edgekey, Pointkey, ...)

对这样的模式,用 SQL 语言可以获取各种图形数据,如找出某体名所有边的有关信息,

```

Select *
From Edges
Where Edgekey = Select Edgekey
                From FRela
                Where Facekey = Select Facekey
                                From VRela
                                Where Volkey =
Select Volkey
From Volume
Where Volname = "give-name"

```

表 1 是由计算机查找打印的图 3 几何体的有关面、边、点的几何数据及拓扑数据,它

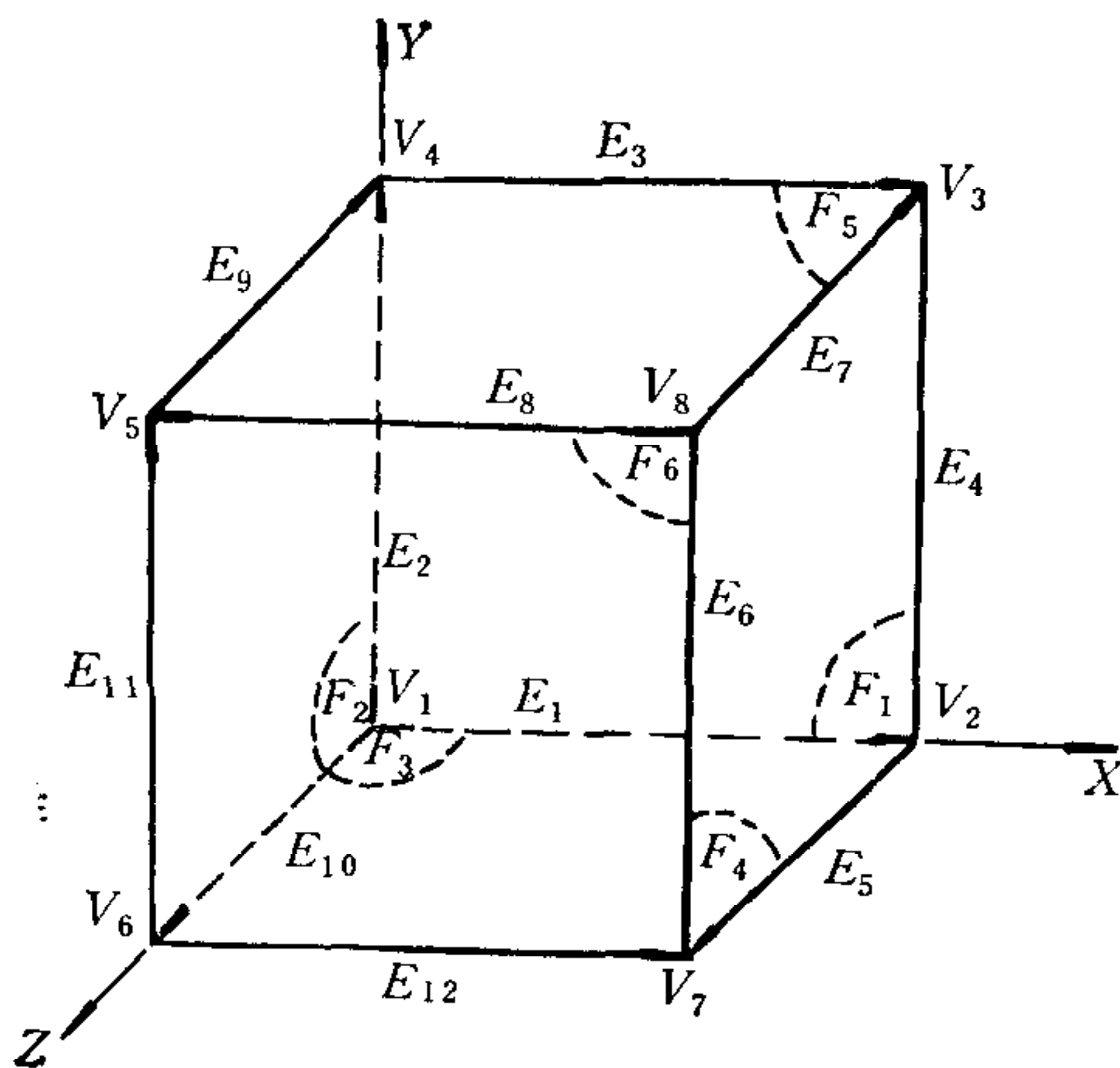


图 3

可以清楚查找出一个面由那些边及点构造,及边与点在该面的方向,以便确定该面的法线方向。

用 SQL 语言编写的查询块可以完成选择、投影、连接、合并、求差等基本关系运算,多个查询块的嵌套可以查遍整个数据库。然而查找的结果是在 UNIFY 管辖下的缓冲区内,其他计算机语言的程序无法使用这些结果。虽然 UNIFY 提供有 100 多个 C 函数可供 C 语言调用,由于图形处理往往需要的是一个体或一幅画面的具有某一特定条件的整体数据,于是笔者研制了另外形式的 C 语言或 FORTRAN 语言的接口。其基本思想

是把查询结果整体数据读入到 C 程序的结构数组中。下面给出了部分接口函数的名称和功能:

```
Getobject (object-name);
```

根据 object-name 可以找出该 object 的信息。

```
Getvolume (volume-name);
```

```
Getfaces (volume-name);
```

```
Getedges (volume-name);
```

```
Getpoints (volume-name);
```

```
Getvolall (volume-name);
```

根据 volume-name 找出该体的有关面、边、点的所有信息,与此对应的装入函数为 Putobject, Putvolume 等。这些查询函数由 SQL 与 C 语言结合编写, C 程序可直接调用这些函数,通过它可自动查询数据库或把数据装入数据库中,其框图如图 4 所示。

表 1

面号	边号	方向	边类型	点号	x 坐标	y 坐标	z 坐标	边中的起终点
10001	10001	0	1	10001	.00000	.00000	.00000	0
10001	10001	0	1	10002	1.00000	.00000	.00000	1
10001	10002	1	1	10001	.00000	.00000	.00000	0
10001	10002	1	1	10004	.00000	1.00000	.00000	1
10001	10003	1	1	10004	.00000	1.00000	.00000	0
10001	10003	1	1	10003	1.00000	1.00000	.00000	1
10001	10004	1	1	10003	1.00000	1.00000	.00000	0
10001	10004	1	1	10002	1.00000	.00000	.00000	1
10002	10002	0	1	10001	.00000	.00000	.00000	0
10002	10002	0	1	10004	.00000	1.00000	.00000	1
10002	10010	1	1	10001	.00000	.00000	.00000	0
10002	10010	1	1	10006	.00000	.00000	1.00000	1
10002	10011	1	1	10006	.00000	.00000	1.00000	0
10002	10011	1	1	10005	.00000	1.00000	1.00000	1
10002	10009	1	1	10005	.00000	1.00000	1.00000	0
10002	10009	1	1	10004	.00000	1.00000	.00000	1

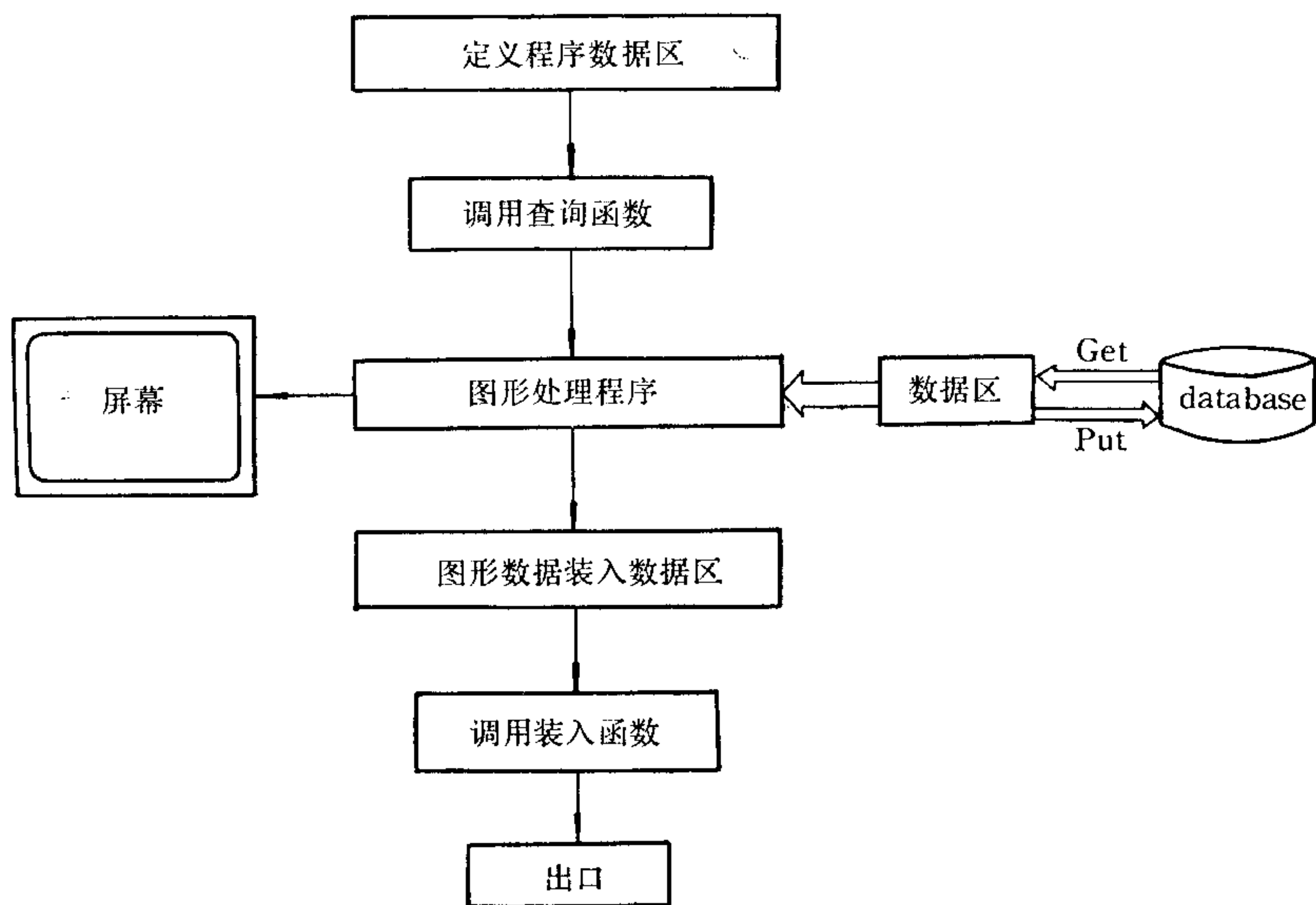


图 4

图形数据库为立体几何造型的各种图形处理程序提供和管理大量数据, 对各个应用程序的设计者来说, 仅要关心自己的数据结构, 而不必考虑数据的组织及存贮、删增和修改等管理。同时数据库内存贮的数据除了面向图形处理这一方面的用户外, 还可以为 CAD 的其它系统提供有效的数据, 这为实现大型的、实用的 CAD 系统打下稳固的基础, 下一步的工作是把多种形式的图形输入, 图形运算等应用程序与图形数据库连接起来, 以便形成较完整的几何造型系统。

参 考 文 献

- [1] Proceedings of Computer Graphics, CAMP 84, Berlin.
- [2] Robert Johnson, Geometric Modelling in CAD/CAM from 2D-Drawings to 3D-Solids, CAMP 84, Berlin.
- [3] 萨师焯, 王珊, 数据库概论, 高教出版社 (1984 年).
- [4] Geomap III, University of Tokyo, 1984 年.
- [5] Jerry Schneider, Selective Update, *Computer Graphics and its Application*, March 1984.
- [6] Requicha, A., Representation for Rigid Solids: Theory, Methods and System, *Computing Surveys*, 12(4), Dec. 1980.
- [7] 梁友栋, 余奕岳, 三维几何造型的数学原理, 高校应用数学学报, 1986, (1).
- [8] Codd, E. F., A Relational Model of Data for Large Shared Data Bank, *Communication ACM* Vol. 3, No. 6, June 1970.
- [9] Date, C. J., An Introduction to Database System, 3rd Edition, Addison-Wesley, 1981.
- [10] 毛根生, 胡希明, Unify 关系数据库系统, 微小型计算机开发与应用, 1985 年第 3 期.

A GRAPHIC DATABASE FOR 3-D GEOMETRIC MOULDING SYSTEM

LIANG YUODONG HU XIMING MAO GENSHENG

(Zhejiang University)

ABSTRACT

A practical stand-alone geometric moulding system necessitates a graphic database as its pictorial information storing and management kernel. The design method based on the boundary representation of geometric objects is proposed for a geometric moulding system graphic database. Its implementation on the UNIFY relational database management system and the design of C language interface are illustrated.