

二维概念格图形的三维重构研究

郑珂, 马骏, 陈明

(河南大学 计算机与信息工程学院, 河南 开封 475004)

摘要:介绍了二维概念格图形向三维空间转化和延伸的必要性和现状。通过对传统概念格图形分层定位布局方法的研究与分析,提出并实现了一种新的以具有大量的平行四边形和有向线段为基本特征的概念格在三维空间的自动布局算法,描述了一种基于该算法的二维概念格图形的三维重构机制,有效地解决了节点横向过度扩张的问题并减少了线段交叉,较好地实现了复杂概念格图形的三维可视化,为知识发现和知识处理提供了良好的基础。

关键词:概念格;子格;概念格可视化;分层布局;三维重构

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2008)05-0025-03

3D Lattice Reconstruction from 2D Graphics

ZHENG Ke, MA Jun, CHEN Ming

(College of Computer and Information Engineering, Henan University, Kaifeng 475004, China)

Abstract: Introduced the necessity and actuality for the transformation from 2D concept lattice to 3D lattice. According to researching and analyzing layers layout of lattice a new algorithm of nodes layout which is based on lots of parallelograms and dilines in 3D space is proposed and implemented. A solution of transformation from 2D concept lattice to 3D lattice is described. Thereby, this policy solves the problem of overage outspread nodes on landscape orientation and reduces the amount of crossing lines. It implements visualization of some complicated concept lattice in 3D space better, and offers a good foundation for knowledge finding and management.

Key words: concept; lattice; sub-lattice; concept lattice visualization; layout in layer; three-dimensional reconstruction

0 引言

随着计算机3D技术的发展,三维图形在网络多媒体、文档表达、虚拟现实和仿真模拟等许多方面的应用越来越广泛。对于概念格而言,数字化三维图形可以增强概念格中对对象和属性的表达能力,打破了二维平面布局的局限性,使得呈现出的格结构更加清晰易懂,更便于数据表示^[1,2]。目前,概念格的三维可视化还是一个比较新的领域,特别是在概念格的数字化三维展现方面,国内还没有这方面的实质性研究成果。针对以上需求,笔者提出了一种新的概念格节点的三维布局方法,结合二维概念格的三维重构机制,可将二维概念格的节点数量和前驱后继关系等信息作为形式概念的输入信息,通过对二维概念格模型的组织与分析,将其转换为三维概念格以增强信息的表达能力。

概念格是以形式背景为基础,以形式概念为基本

元素的一种格结构。概念格的每个节点称为一个形式概念,由对象和属性两部分组成。对象也叫概念的外延,是概念所覆盖的实例;属性也叫概念的内涵,是概念的描述,即该概念所包含的共同特征。概念格可以用图形表示为有标号的线图,图中的每个节点表示一个形式概念,节点间的连线表示节点间存在的偏序关系。目前常用分层图^[3,4]对概念格进行布局,其优点是节点间的偏序关系清晰,但缺点是线段的交叉会由于节点间偏序关系的增多而增多,不利于概念格的三维展示。因此提出了一种新的三维布局方法。在这种方法中,概念格以平行四边形和柱体为特征,绘制出的三维概念格美观清晰,有效地减少了线段的交叉。同时为了使节点间的偏序关系更清晰,采用有向图的形式,节点间的线段是有方向的。这样不但保留了分层图中偏序关系清晰的优点,又克服了分层图中线段交叉较多的不足。

1 概念格的平行边有向图构造算法

在二维概念格图形中,线图(Line Diagram)是最基本的表示方法。在线图中,每个节点表示一个形式概

收稿日期:2007-08-19

基金项目:河南省高校杰出科研人才创新工程项目(2007KYCX018)

作者简介:郑珂(1981-),男,硕士研究生,主要研究方向为知识发现及网络应用;马骏,副教授,硕士生导师,主要研究方向为分布式处理,知识发现及网络应用。

念,如果节点 P_1 在节点 P_2 的上方,并且节点 P_1 和节点 P_2 存在偏序关系,则节点 P_1 是节点 P_2 的父节点或前驱节点,节点 P_2 是节点 P_1 的子节点或后继节点,并将这两个节点用连线连接起来。连接后的线图使概念之间的关系变得清晰和易于理解,为实现概念格的三维转化奠定了基础。在传统的概念格分层布局中,若节点到外延最大节点的最大链的长度是 n ,则该节点是第 n 层的格节点^[5]。这种按照偏序关系定位节点的方法满足了概念格布局的基本要求,格节点之间的偏序关系清晰易懂。但由于父节点和子节点分别在不同的层,也带来了格结构表达不清晰、边的交叉数较多等缺点,特别是在某些复杂的二维概念格中,节点横向过度扩张和线段交叉仍是一个无法有效解决的问题^[6]。

借助于不可约减的对象或属性,格图形以具有大量的平行边为特征,将大大改善它的可读性。由于三维概念格以平行四边形和柱体结构展示可以有效地减少边的交叉,因此提出了基于三维空间的概念格平行边有向图构造算法,使绘制出的三维概念格以具有大量的平行四边形或柱体结构为基本特征。这种方法和传统的分层布局方法相比,层与层之间有效地减少了线段交叉,使格结构既达到了清晰美观的要求,又方便地实现了人机交互。

在引入复杂概念格的重构方法之前,首先考虑一种任意节点的直接后继节点均不超过 6 个的情况。假设在二维概念格中,第一层的节点表示为 $P_{1,1}$,第二层的节点依次表示为 $P_{2,a}$ ($1 \leq a \leq 6$),节点 $P_{2,1}$ 的后继节点依次表示为 $P_{3,1}$ 至 $P_{3,b}$ ($b \geq 1$), $P_{2,2}$ 的后继节点依次表示为 $P_{3,b+1}$ 至 $P_{3,c}$ ($c \geq b+1$), $P_{2,3}$ 的后继节点依次表示为 $P_{3,c+1}$ 至 $P_{3,d}$ ($d \geq c+1$),依次类推。如果节点有多个前驱节点,节点的命名就会产生冲突,此时以该节点的第一个前驱节点对该节点的命名为准。对节点命名后,首先建立三维坐标系,并把外延最大节点 $P_{1,1}$ 定位于坐标原点,然后分别定位 $P_{1,1}$ 的后继节点于 X 轴、 Y 轴和 Z 轴的正向与负向上。为了使绘制出的概念格尽量以平行四边形的形式展示,具有相同后继节点的同层节点在定位时不能位于同一坐标轴上。

在后继表中遍历节点 $P_{2,i}$ 的后继节点 $P_{3,j}$,查找 $P_{3,j}$ 的前驱表,若 $P_{3,j}$ 和第二层的其他节点无关,即不存在偏序关系,则 $P_{3,j}$ 定位在有向线段 $P_{1,1}P_{2,i}$ 的延长线上,并可经过几何计算使其满足 $P_{1,1}P_{2,i}$ 和 $P_{2,i}P_{3,j}$ 长度相等;若 $P_{3,j}$ 和除 $P_{2,i}$ 的第二层的其他节点 $P_{2,k}$ 有关,即 $P_{2,k}$ 是节点 $P_{3,j}$ 的一个前驱,那么节点 $P_{3,j}$ 的定位满足有向线段 $P_{2,i}P_{3,j}$ 和有向线段 $P_{1,1}P_{2,k}$ 平行且相等即可。

若第 n 层的第 m 个节点的第 j 个后继节点 $P_{n+1,j}$

和第 n 层的其他节点没有偏序关系,则节点 $P_{n+1,j}$ 定位在 $P_{n,m}$ 的父节点 $P_{n-1,i}$ 与节点 $P_{n,m}$ 连接成的有向线段 $P_{n-1,i}P_{n,m}$ 的延长线上,并可经过几何计算使其满足 $P_{n-1,i}P_{n,m}$ 和 $P_{n,m}P_{n+1,j}$ 相等;若 $P_{n+1,j}$ 和除 $P_{n,m}$ 的第 n 层的其他节点 $P_{n,k}$ 有关,即 $P_{n,k}$ 是节点 $P_{n+1,j}$ 的一个前驱,则节点 $P_{n+1,j}$ 的定位满足有向线段 $P_{n,m}P_{n+1,j}$ 和有向线段 $P_{n-1,i}P_{n,k}$ 平行且相等即可。

每个节点可由一个三元组 (S, F, A) 表示,其中 S 表示节点本身, F 表示该节点的前驱节点, A 表示节点的后继节点。显然由于外延最大节点没有前驱节点,内涵最大节点没有后继节点,所以可通过查找算法找到外延最大节点和内涵最大节点。算法描述如下:

输入: (S, F, A)

输出: 节点的坐标,即节点的定位布局

查找外延最大节点 $P_{1,1}$,并将其定位于坐标原点
遍历后继表定位 $P_{1,1}$ 的后继节点于坐标轴

FOR($n=2; n \leq \text{层数}-1; x++$)

//控制除外延最大节点和内涵最大节点的层的循环

FOR($y=1; y \leq n+1$ 层的节点数量; $y++$)

//控制除外延最大节点和内涵最大节点的其余各层内的节点循环

IF $P_{n+1,j}$ 和第 n 层的其他节点没有偏序关系

THEN 节点 $P_{n+1,j}$ 定位在有向线段 $P_{n-1,i}P_{n,m}$ 的延长线上

ELSE

IF 节点 $P_{n+1,j}$ 的一个前驱是 $P_{n,k}$

THEN 定位 $P_{n+1,j}$ 使 $P_{n,m}P_{n+1,j}$ 和 $P_{n-1,i}P_{n,k}$ 平行且相等

END_IF

END_IF

END_FOR

END_FOR

按照这种节点定位和布局算法构造出的三维概念格图形中,边的交叉和传统的布局方法相比明显减少,绘制出的格结构规则且美观。但是该算法只适用于简单格结构的情况。对于后继节点多于 6 个的情况,算法将无法处理;还有一种情况是当某节点的多个后继节点和该节点同层的其他节点都没有偏序关系时,该算法也无法处理。为了解决这些问题,需要首先对输入的概念格进行预处理,把概念格拆分为若干简单子格,然后对子格执行以上的算法,最后合并三维子格,得到所希望的结果。

2 重构机制的设计和实现

2.1 重构机制架构设计

文中描述的重构机制由二维概念格的输入和预处理、三维概念格的可视化生成引擎两个模块组成。二

维概念格的输入和预处理主要是输入概念格形式背景、格节点数量和节点间的偏序关系等信息,建立节点的前驱表和后继表并存入相应的数据库表中。三维概念格的可视化生成引擎最核心的功能是根据格节点之间的前驱和后继关系,对格节点的定位进行判断,以便确定概念格的空间结构。详细处理流程如图1所示。

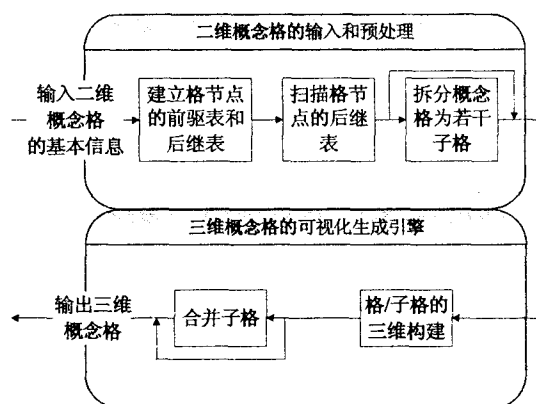


图1 三维概念格的转化重构流程

2.2 二维概念格的输入和预处理

为了进行三维概念格的可视化生成,首先需要对输入的二维概念格的基本信息进行预处理。预处理分两步,第一步需要建立概念格节点的前驱表和后继表,并存入数据库;第二步扫描每个格节点的后继表,如果节点的后继节点超过6个或某层的多个节点都只有一个前驱节点且其前驱节点相同时,则需对概念格的形式背景进行拆分,再根据子背景构造出子格。经过预处理后,即可在可视化生成引擎模块中对子格进行三维构建。

对于一个概念格 V 而言,满足重叠区间的容差关系的块总是能够形成满足重叠区间映射的子块,且它的和是 $V^{[7]}$ 。这就为概念格的拆分与合并提供了理论依据。在系统实现时,首先提取形式背景的子背景,通过子背景构造子格,并选择合适的重叠区域,再依次对子格进行叠加,直到获得完整的三维格。

2.3 三维概念格的可视化生成引擎

在该模块中,对于简单的二维概念格,可直接利用平行边有向图造格算法对其进行三维构造。如果是复杂的概念格,由于在预处理模块中已经对二维概念格进行了拆分,所以在此只需对每个子格进行三维构造,然后通过节点匹配和三维坐标变换合并三维子格即可。但是要注意,对每个子格进行三维构建时,子格中边的长度要符合一致性要求,这样不但便于子格的合并,而且合并后的三维概念格也更美观。

对于每个子格,首先在数据库表中扫描每个节点的信息,然后对任意两个子格的节点进行匹配,如果某

两个子格的某些节点能够匹配,则说明这两个子格具有重叠区域需要合并。在合并的过程中,可通过矩阵变换实现对每个子格的坐标变换。

设 H 是三维坐标系变换到其他坐标系的变换矩阵的集合,函数 $f: \text{relationships} \rightarrow H$ 将 relationships 中每个序偶元素与一个变换矩阵相关联,对于任意一个子格,根据关系 relationships 中的相应元素可以将该子格与一个属于 H 的变换矩阵相关联。经过相应的变换矩阵变换后,所有子格都拥有了相同的坐标系,然后根据匹配的节点将各个子格组合在一起。

笔者使用C#语言,通过组件和Direct3D技术设计并实现了3DLRS(3D Lattice Reconstruction System)系统,对文中描述的关于节点定位的平行边有向图造格算法和二维概念格的三维重构机制进行了验证。对于图2所示的二维概念格,转换后的三维效果如图3所示。

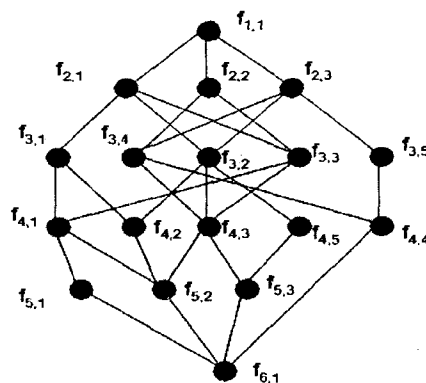


图2 输入系统的二维概念格结构

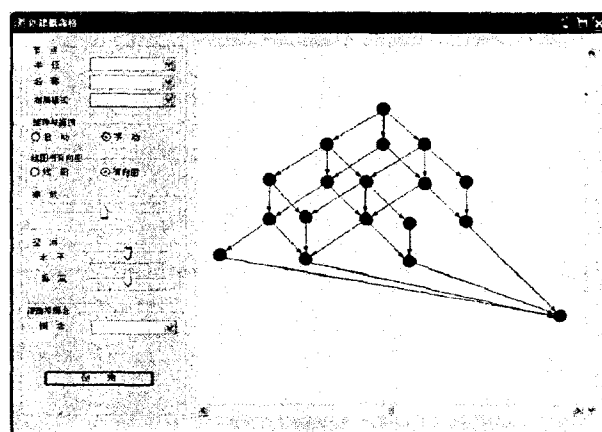


图3 转化后的三维概念格结构

从图2和图3的对比中不难发现,三维概念格的空间结构不但比二维概念格更加美观清晰,边的交叉也明显减少。随着概念格越来越复杂,采用平行边有向图造格算法和二维概念格的三维重构机制进行节点定位布局的优越性将越来越明显。

(下转第30页)

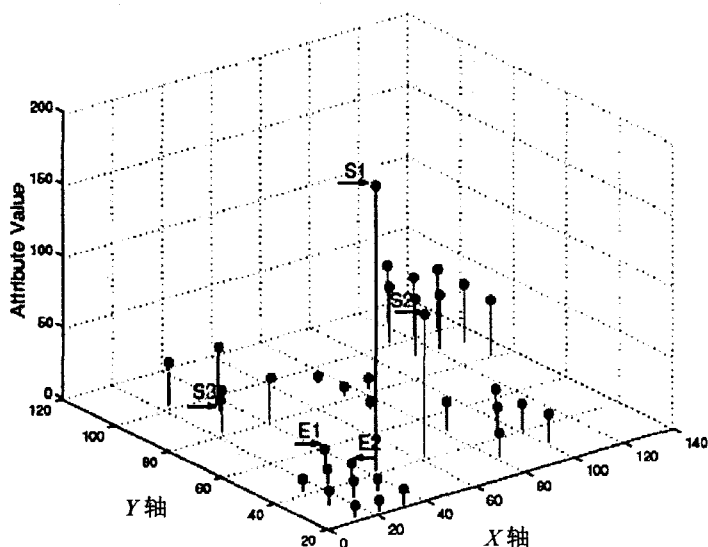


图 1 空间数据集

图中,空间对象定位在 $X-Y$ 平面,竖轴的高度表示每个对象的属性值。

表 1 给出了实验的比较结果,可以看出,文中的算法准确地检测到三个真正的空间离群点 S1、S2、S3,而 Scatterplot 算法和 Moran Scatterplot 算法则均错误地把 E1 和 E2 当成了空间离群点。实验的结果表明文中提出的算法可以成功地挖掘出空间离群点而同时避

表 1 实验结果

离群点	算法		
	Scatterplot	Moran Scatterplot	文中算法
1	E1	S1	S1
2	E2	E1	S2
3	S2	E2	S3

(上接第 27 页)

3 结束语

基于三维空间的平行边有向图构造算法和概念格的三维重构机制有效地解决了概念格在三维空间中的节点定位布局和线段交叉问题,使格结构更加清晰易懂。而且采用这种重构方法构造出的概念格具有特别良好的结构,增强了概念格的表达能力,为知识处理提供了良好的数据基础。

但是,采用这种方法进行构造,由于系统对于复杂的概念格需要进行形式背景的拆分并构造子格,系统的时间复杂度和空间复杂度将会成倍增加,因此寻找一种更优化的算法将是今后进一步的研究目标;同时由于子格的合并是一个非常复杂的过程,如何简化合并过程并得到精确的结果也将是今后研究的重点。

参考文献:

[1] Wille R. Lattices in data analysis: How to draw them with a

避免了将非空间离群点错误地当成了空间离群点。

4 结束语

文中提出了一种基于迭代的空间离群点检测算法,该算法通过多次迭代检测离群点,并在迭代过程中对离群点的属性值进行修正,可以提高检测结果的正确性,并能检测局部离群点。实验验证了该算法的有效性。

参考文献:

- [1] Tan Pang-Ning, Steinbach M, Kumar V. Introduction to Data Mining[M]. 北京:人民邮电出版社,2006.
- [2] Shekhar S, Lu C T, Zhang P. A Unified Approach to Spatial Outlier Detection Geoinformatica[J]. International Journal on Advances of Computer Science for Geographic Information System, 2003, 7(2): 139-166.
- [3] 王占全. 基于 GIS 空间数据挖掘若干关键技术的研究[D]. 杭州:浙江大学,2005.
- [4] Lu Chang-Tien, Chen Dechang, Kou Yufeng. Detecting Spatial Outliers with Multiple Attributes[C]// Proceedings of the 15th International Conference on Tools with Artificial Intelligence. [s.l.]:[s.n.], 2003.
- [5] Haining R. Spatial data Analysis in the Social and Environmental Sciences[M]. Cambridge: Cambridge University Press, 1993.
- [6] Luc A. Local Indicators of Spatial Association: LISA[J]. Geographical Analysis, 1995, 27(2): 93-115.
- [7] computer[M]. Berlin: Berlinpringer, 1993: 47-158.
- [2] Krohn U, Davies N J, Weeks R. Concept lattices for knowledge management[J]. BT Technol J, 1999, 17(4): 106-114.
- [3] Cole R. Automated Layout of Concept Lattice Using Layer Diagrams and Additive Diagrams[C]// Concept Lattices: Second International Conference on Formal Concept Analysis, ICFA 2004. Sydney: Griffith University, 2004: 31-42.
- [4] Sugiyama K, Tagawa S, Toda M. Methods for visual understanding of hierarchical system structures[J]. IEEE Transactions on Systems, Man and Cybernetics, 1981, 11(2): 109-125.
- [5] 谢润,李海霞,马骏,等. 概念格的分层及逐层建格法[J]. 西南交通大学学报, 2005, 40(6): 837-841.
- [6] 马骏,沈夏炯,刘宗田. 基于三维空间的概念格自动布局[J]. 计算机科学, 2006, 33(5): 244-246.
- [7] Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations[M]. [s.l.]: Springer Verlag, 1999.