

# 一个面向纺织的图案数据库的应用设计

## ——基于实例的图案生成模型\*

申立志<sup>1</sup>, 徐白雁<sup>2</sup>, 赵磊<sup>1</sup>

(1. 浙江大学 计算机学院, 浙江 杭州 310027; 2. 浙江教育报社, 浙江 杭州 310012)

**摘要:** 提出了一个基于 Case 推理的图案生成模型。采用结构与数据相结合的实例描述方法, 将图案设计实例分解为基于框架描述的实例结构和基于多类型图元的实例数据, 并介绍了基于实例的检索方法和回溯求解策略。

**关键词:** 图案设计; Case 推理; 回溯求解

中图法分类号: TP311.132

文献标识码: A

文章编号: 1001-3695(2004)06-0125-03

## Application and Design of Image Database Based on Textile

SHEN Li-zhi<sup>1</sup>, XU Bai-yan<sup>2</sup>, ZHAO Lei<sup>1</sup>

(1. College of Computer, Zhejiang University, Hangzhou Zhejiang 310027, China; 2. Education Newspaper Office of Zhejiang, Hangzhou Zhejiang 310012, China)

**Abstract:** This paper presents a model based on Case which can produce pattern. The method of case description which combine data and structure disassembles pattern design case into case structure based on frame description and case data based on multi-type pattern element. This paper also introduces retrieve method based on case and retrospect solution strategy.

**Key words:** Pattern Design; Case Reasoning; Retrospect Solution

图案设计作为计算机辅助设计领域的一个分支, 其应用日益广泛, 已成为产品创新设计的重要内容。目前, 国内外已有众多的软件系统支持图案的辅助设计, 但它们中的大部分还仅仅局限于提供交互构图的功能, 即图案设计的主体还是设计人员。这类软件系统能够做到对设计人员创作思想的充分体现, 当然这是以复杂烦琐的交互过程为代价的。

Case 推理方法应用于图案 CAD 领域是一个新的尝试。本文提出了一个基于 Case 推理的图案生成模型, 采用结构与数据相结合的实例描述方法, 将图案设计实例分解为基于框架描述的实例结构和基于多类型图元的实例数据, 而把整个图案生成过程视为特定的实例结构。针对不同实例数据(图元)的例化过程, 对实例库的组织与检索、实例的改编与再设计等 Case 推理中的关键技术, 提出了新的求解策略。同时, 利用知识推理方法, 在图案生成过程中引入了基于设计上下文的引导式交互技术以及局部操作到整体布局的自动提升等智能化交互技术, 简化了图案创作过程。实际结果表明, Case 推理方法对于系列化、规格化的图案设计是一种有效的求解方法。

### 1 实例的分解表示

图元是图案构成的基本单元, 我们可以对一幅图案进行图元一级的分解。基于图元的图案分解如图 1 所示。

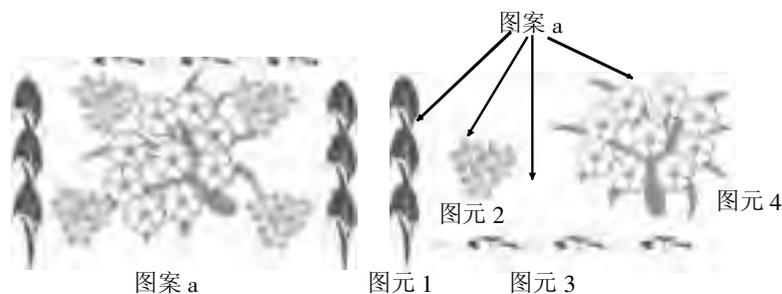


图 1 基于图元的图案分解

根据实例(Case) = 实例结构(Case Structure) + 实例数据(Case Data)的实例描述方法, 我们可以将图元从设计实例中抽取出来, 成为实例数据, 而脱离了具体数据的设计实例的抽象结构框架就成了实例结构。实际应用中, 针对实例结构和实例数据不同的存储要求, 对两者分别建库并实现各自不同的存储管理功能。这样, 一个实例就对应于实例库中的一条结构框架记录和素材库中的多个图元。实例的查询、检索、改编、维护主要针对实例结构进行并被转换成相应的数据库操作, 简化了实例推理的难度。而在实际设计过程中, 通过实例结构对实例数据的例化, 又使两者结合起来, 直观地再现于用户。

### 2 实例结构的描述方法

实例结构是图案总体布局的大致描述, 是对图案设计知识的抽象。本文采用框架(Frame)知识描述方法来定义实例结构, 其一般形式如下:

< 框架 >

Frame < Frame Name > in < case base >

Isa : < frame name > // 框架继承关系槽, 其值是上层框架名

收稿日期: 2003-04-18; 修返日期: 2003-10-28

基金项目: 国家“863”计划自动化领域 CIMS 主题资助项目(863-511-820-021); 国家发展计划委员会重大资助项目(98-15-5)

```

Keyword : < keyword value > //对该实例的关键字描述
Mark : < mark value > /* 用户对特定实例框架所作的标志, 以上三个槽值构成实例检索的基础 * /
With : //指明该实例框架在各个不同侧面的参数和属性
EleNum : < number of elements > //该实例所用图元个数
Cascade-prior : < prior list >
//图元素材在实例显示时的显示优先数序列
Cascade-mode : < mode list >
/* 各图元按优先数序列显示时所用的层叠模式, 具体值有覆盖、透明、半透明(参数可调)等 * /
Center : X : < center value1 >
          : Y : < center value2 >
/* 表示该实例显示时的参考位置, 该实例的所有图元在实际显示时都相对于 Center * /
ElementSet : < element1, element2, ..., elementN >
//所用的图元序列
Always : < constrain values >
//实例改编时必须满足的约束
If-modified : < constrain rules >
//为维护 Always 而执行的过程操作
EndFrame
    
```

其中, Always 槽值表明了实例所必须满足的约束, 它或是一些构图知识约束, 如某些图案所要求的中心辐射约束, 或是一些领域知识约束, 如纺织工业对图案的四方连续性约束等; If-modified 槽值则是对 Always 槽值约束性保护, 一般是一组产生式规则, 指明在不同的前提条件下为保持 Always 约束所做的一系列操作。Always 槽值约束和 If-modified 槽值的自动维护可以弥补实例分解而带来的散乱性缺陷。

### 2.1 实例数据的描述方法

作为实例数据的图元, 主要有三种类型: 点阵图元、矢量图元和文字图元, 它们的显示技术和处理算法都存在很大差异。采用面向对象的数据描述方法, 则可以封装这些差异性, 为上层应用提供一个图元操作的通用接口。以下是基本图元类 CElement 的定义:

```

class CElement
{
public:
    UINT ElementID; //对应于该图元在素材库中的 ID 号
    UINT ElementType; /* 图元类型, 可以是 RASTER(点阵图元), VERTEX(矢量图元) 或 WORD(文字图元) * /
    CPoint Position; //显示时图元中心点相对于 Center 的偏移量
    CRect Rect; /* 图元的最小外接矩形, 显示时应根据该变量调整缩放比 * /
public:
    Virtual BOOL Display( UINT CascadeMode, Crect rect) = 0;
    Virtual BOOL Rotate( int Angle, Crect rect) = 0;
    ...
    /* 不同类型图元的通用操作, 如显示、旋转等, 这里仅仅以纯虚函数的形式给出了通用的调用接口 * /
};
    
```

CElement 类定义了图元的共同属性和通用调用接口, 但并没有具体实现。以 CElement 类作为超类, 定义了三个子类 CrasterElement, CvertexElement, CwordElement 来对三种类型的图元作个性定义, 并完成通用接口依类型不同的具体实现。如:

```

class CrasterElement : public CElement
{
private:
    CLongBinary RasterInfo; //图元的点阵信息
    CPalette Palette; //图元显示时的调色板信息
public: //通用调用接口针对点阵图元的具体实现
    Virtual BOOL Display( UINT CascadeMode, Crect rect) ;
    Virtual BOOL Rotate( int Angle, Crect rect) ;
    ...
public: //点阵图元的个性化操作
    BOOL SetTransparentColor( CPalette Palette, UINT index) ;
    //设置透明色
    ...
    
```

};

利用面向对象的多态性特征, 在实际应用中, 我们只需声明一个公共超类 CElement 的指针, 通过它指向不同类型的图元对象。CElement 类中由虚函数定义的通用接口将会依据指针指向图元类型的不同, 自动执行不同的实现。

面向对象的一些优点, 如封装性、继承性、多态性, 将不同图元在处理上差异封装在对象内部, 从而可以在更高层次上对不同类型的图元作统一的透明处理, 使它们作为一个独立个体参与图案布局, 简化了推理难度。

### 2.2 实例结构与实例数据的存储(图 2)

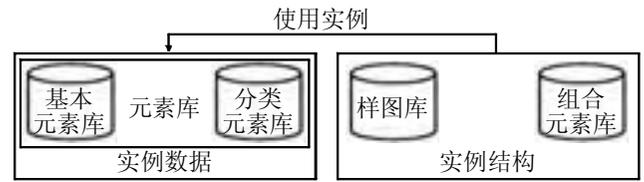


图 2 图案分解后实例结构与实例数据的存储(图案数据库的应用)

## 3 基于实例的检索方法

实例结构描述的紧凑性给实例检索的有效性提供了可能。在实例库中, 根据实例框架的 Isa 槽值, 所有实例被组织成一个标准树型的层次结构, 从根节点到任一实例的遍历就决定了实例结构大的类别特征, 如印染图案设计实例、墙纸图案设计实例等, 同时这也提供了一种最简单的实例检索方式。但这种检索方式仅仅依靠 Isa 槽值, 信息量不足, 在检索过程中又使用了严格的布尔操作, 从而导致了查全率和查准率的下降; 另外它对检索出来的元素不区分其重要性, 不能进行相关排序, 这也不能符合设计人员的检索要求。

实例框架中的 Keyword 槽值是实例结构的索引, 决定了实例框架的索引属性及其上的隶属度。如一实例框架关于工艺适用性属性上的值是: { 毛毯/0.75, 服装面料/0.10, 针织物/0.23, 装饰品/0.15 }, 则表明该素材元素有 75% 的应用场合适用于毛毯图案, 10% 适用于服饰图案, 等等。基于 Keyword 槽值, 可以实现实例结构的索引属性的模糊查询。

基于隶属度的模糊检索函数的构造, 不仅应考虑属性的隶属度, 还应考虑该属性对整个检索结果的权重以及用户对检索结果的精确度要求。为此我们引入了 P 次开方模型, 例如, 图案设计师对素材元素的下列属性提出了检索要求: 含有红花 (A<sub>1</sub>)、绿叶 (A<sub>2</sub>) 的适用于纺织 (A<sub>3</sub>) 毛毯 (A<sub>4</sub>) 的素材元素。其中, 属性 A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub>, A<sub>4</sub> 对检索结果的影响程度分别是 d<sub>1</sub>, d<sub>2</sub>, d<sub>3</sub>, d<sub>4</sub> ( $\sum_{i=1}^4 d_i = 1$ ), 任一素材元素 Element<sub>i</sub> 在这些属性上的隶属度是 [S<sub>i1</sub>, S<sub>i2</sub>, S<sub>i3</sub>, S<sub>i4</sub>], 则检索函数为

$$SeekFuc(Element_i) = 1 - \sqrt[p]{\frac{d_1^p(1-S_{i1})^p + d_2^p(1-S_{i2})^p + d_3^p(1-S_{i3})^p + d_4^p(1-S_{i4})^p}{d_1^p + d_2^p + d_3^p + d_4^p}}$$

系统按检索函数 SeekFuc 作用于每一实例结构上的值来对检索结果排序, 并提交给用户。其中, p 表示对检索结果的精确度要求, p 值越大, 符合条件的检索结果越少, 也越精确, 当然我们应尽量避免检索条件的过约束情况。

## 4 基于实例的回溯求解策略

回溯是实例再设计的基本控制策略, 它支持设计过程中的

试探性行为 and 实例设计过程的再现。当设计者认为某一时刻上的设计状态是有保留价值的,他可以在此设置一个保存点。实际上,保存点就是系统状态在该时刻的快照(Snapshot),它通过保存系统在该点的设计状态来实现回退机制。

设计状态是一个四元组 (PatternFrame, ElementSet, ActiveElement, Tools), 其中, PatternFrame 是当前图案布局的框架表示; ElementSet 是图元集合, 包括图元的当前位置、操作状态等; ActiveElement 是 ElementSet 的一个成员, 表示当前编辑图元; Tools 则是所调用的编辑工具。

将各保存点的设计状态组织起来就形成了一个设计过程序列 (Design Process Series), 如图 3 所示。设计者可以通过 Undo 和 Redo 操作在设计过程序列上进行线性搜索以便寻求一个有用的设计点, 同时对设计过程的标签记忆可以实现跨越多个设计点的状态恢复。这种描述的紧凑性, 使得对整个设计过程的记忆成为可能, 将设计过程序列用于实例记忆, 即将实例的整个设计过程序列存储于实例库中, 则可实现实例的整个设计过程的再现。所需做的工作仅仅是依次取出系统的每一保存点上的设计状态, 通过实例结构框架针对具体图元的例化过程, 将实例的设计过程直观地再现于用户。其间, 通过响应用户输入, 使实例的解答改编细化到交互点上进行。

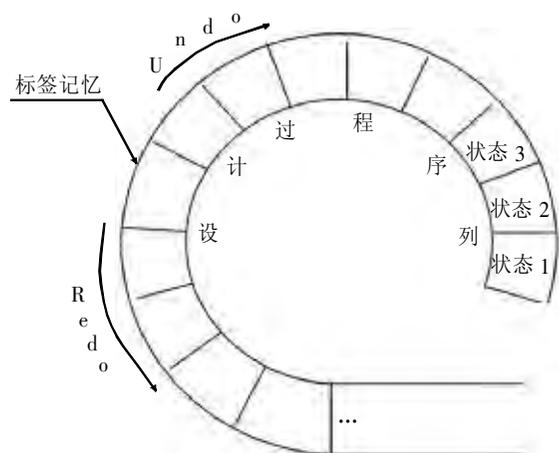


图 3 设计过程序列与回退机制

作执行顺序的约束, 它有顺序型、并发型等几种形式, 它用一组规则来确定动作的执行顺序。如动作推理过程可表述如下:

(1) 确认该交互点上的系统当前状态; (2) 根据系统当前状态, 匹配设计上下文树, 并用广度优先算法搜索出系统可能

的状态迁移; (3) 针对不同的状态迁移, 产生出相应的动作, 放入动作队列, 并依据领域知识和对用户行为特征的数据统计排序; (4) 弹出动作列表, 等待用户确认。

## 5 总结

图案素材库系统以行业为导向, 以应用为驱动, 旨在为了触发图案设计师的灵感, 拓展设计师的思维空间, 以支持面向创新的图案设计。它与其他素材库相比, 具有高效存储、方便查询且直接针对图案 CAD 的特点, 目前已应用于国家“863”项目“新一代轻纺图案 CAD/CAM 系统”以及国家计委重点项目“面向区域经济的高技术产品开发系统”中, 并产生了良好的效果。目前, 图案素材库系统较多地支持了提供基于描述、关键字一类的检索和查询, 在基于内容的检索方面, 只提供了在分色基础上的色彩检索方法。展望未来, 随着计算机视觉与图像理解研究的进一步深入, 面向知识的、主动的图案数据库将是发展方向。

## 参考文献:

- [1] on Androutsopoulos, et al. Natural Language Interface to Database Introduction [J]. Journal of Natural Language Engineering, 1995, 1(1): 29-81.
- [2] nil K Jain, Aditya Vailaya. Image Retrieval Using Color and Shape [J]. Pattern Recognition, 1996, 29(8): 1233-1244.
- [3] amesh Jain, S N Jayaram Murthy, et al. Similarity Measures for Image Databases [C]. SPIE 2420, 1995. 58-65.
- [4] ean Michel Marie-Julie, Hassane Essafi. Image Database Indexing and Retrieval Using the Fractal Transform [C]. Serge Fdida, Michele Morganti. LNCS 1242: Multimedia Applications, Services and Techniques, ECMAST '97, 1997. 169-182.
- [5] Y Lee, M K Shan, W P Yang. Similarity Retrieval of Iconic Image Database [J]. Pattern Recognition, 1989, 22(6): 675-682.

## 作者简介:

申立志 (1973-), 男, 硕士研究生, 主要研究领域为数据库; 徐白雁 (1964-), 女, 工程师; 赵磊 (1976-), 男, 硕士研究生, 主要研究方向为计算机图形图像、CSCW、图形处理。

(上接第 113 页)



图 1 Lena 原始图像



图 2 图 1 置乱 1 次 (t=1)

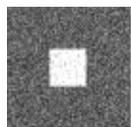


图 3 中心剪切及噪声攻击 (t=5)



图 4 图 3 的解密图像



图 5 二值化后四角污染 (t=5)



图 6 图 5 的解密图像

## 4 结论

基于配对和移位变换, 本文提出了一种新的置乱算法, 置乱参数由 Chebyshev 混沌系统的密钥流产生。该算法是在位置空间进行的, 具有较强的灵活性, 运算简单, 没有引起数据量的膨胀, 可适用于各类图像加密。分析表明算法有较大的密钥

空间和置乱方案, 试验结果显示了对噪声、破损和污染攻击的抵抗能力, 说明该置乱算法具有较好的安全性和稳健性。

## 参考文献:

- [1] 丁玮, 齐东旭. 数字图像变换及信息隐藏与伪装技术 [J]. 计算机学报, 1998, 21(9): 838-843.
- [2] i Dongxu, Zou Jincheng, Han Xiaoyou. A New Class of Scrambling Transformation and Its Application in the Image Information Covering [J]. Sciences in China, Ser. E, 2000, 43(3): 304-312.
- [3] ing Wei, Yan Weiqi, Qi Dongxu. Digital Image Watermarking Based on Discrete Wavelet Transform [J]. Computer Science and Technology, 2002, 17(2): 129-136.
- [4] 张景中, 熊金城. 函数迭代与一维动力系统 [M]. 成都: 四川教育出版社, 1992.
- [5] 屈婉玲. 代数结构与组合数学 [M]. 北京: 北京大学出版社, 1998. 257-329.

## 作者简介:

赵学峰 (1965-), 男, 副教授, 硕士, 研究方向为信息隐藏、算法分析。