

(19) 中华人民共和国国家知识产权局



## (12) 发明专利申请

(10) 申请公布号 CN 102831241 A

(43) 申请公布日 2012. 12. 19

(21) 申请号 201210332537. 9

(22) 申请日 2012. 09. 11

(71) 申请人 山东理工大学

地址 255086 山东省淄博市高新技术产业开发区高创园 D 座 1012 室

(72) 发明人 孙殿柱 史阳 刘华东 李延瑞

(51) Int. Cl.

G06F 17/30 (2006. 01)

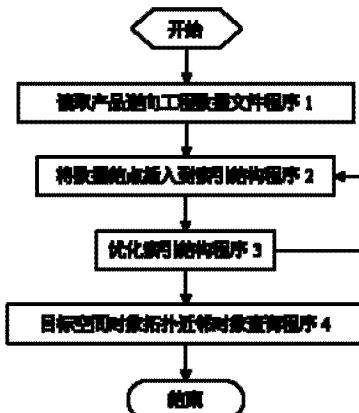
权利要求书 2 页 说明书 7 页 附图 5 页

(54) 发明名称

产品逆向工程数据动态索引多目标自适应构建方法

(57) 摘要

本发明提供一种产品逆向工程数据动态索引多目标自适应构建方法，其特征在于：首先读取产品逆向工程数据文件，建立各空间对象的轴向包围盒，依据轴向包围盒的中心及外接球半径建立各空间对象对应的数据结点，并存入数据结点序列，通过选择插入位置、强制重新插入、结点分裂、调整结点轴向包围盒等步骤将序列中各数据结点插入到索引结构中，将轴向包围盒体积较大的数据结点重新插入到索引结构中，进一步优化索引结构，实现产品逆向工程数据动态索引结构的建立。本发明可建立各种复杂产品逆向工程数据的空间索引结构，具有参数依赖性低、稳定性强、查询效率高的特点。



1. 一种产品逆向工程数据动态索引多目标自适应构建方法，其特征在于包含以下步骤：一、读取产品逆向工程数据，建立各散乱点云、多边形网格及分片连续曲面的轴向包围盒，依据轴向包围盒的中心及外接球半径建立各空间对象对应的数据结点，并存入数据结点序列，其中结点包括索引结点和数据结点，索引结点包含根索引结点、内部索引结点和叶索引结点，索引结构的最上层结点为根索引结点、最下层结点为叶索引结点、其余结点为内部索引结点，定义  $M$  为结点的最大子结点数、 $m$  为结点最小子结点数，其中  $M$  为大于 2 的整数， $m$  为小于或等于  $M/2$  的整数，除根索引结点外，每个索引结点的子结点数均小于等于  $M$  且大于等于  $m$ ；索引结构中每个结点的轴向包围盒恰好包围该结点的所有子结点；二、将数据结点插入到索引结构中，步骤具体是：1) 为结点选择插入位置，具体步骤为：(1) 令当前结点为 `current_node`，如果索引结构为空则返回空，否则令 `current_node` 为索引结构根索引结点；(2) 令结点将要插入的层数为 `level`，若结点为数据结点则 `level` 为索引结构的叶子层，其他类型结点的插入是由强制重新插入引起的，`level` 为其重新插入前所在层数；(3) 计算 `current_node` 的每个子结点与待插入结点的轴向包围盒外接球重叠度，选择重叠度最小的作为 `current_node`，其中计算两个结点的轴向包围盒外接球重叠度的方法为：令任意两结点  $c_i, c_j$  的轴向包围盒外接球半径分别为  $r_i, r_j$ ，轴向包围盒中心间的距离为  $d_{ij}$ ，采用公式  $\xi_{ij} = \frac{r_i + r_j}{d_{ij}}$  计算两结点轴向包围盒的外接球重叠度；(4) 重复步骤

(2) 直到索引结构的 `level` 层为止；2) 将结点插入到步骤 1) 中得到的插入位置；3) 令结点插入到结点 `node` 下，判断结点 `node` 的子结点数是否大于结点的最大子结点数  $M$ ，如果大于则对结点 `node` 进行溢出处理，若结点 `node` 为非根索引结点且在插入一个空间对象过程中该结点所在层第一次进行溢出处理，则计算溢出结点 `node` 的  $M+1$  个子结点的轴向包围盒的中心到结点 `node` 的轴向包围盒的中心的距离，以距离值为关键字，对结点 `node` 的子结点进行降序排序，选出前  $P$  个子结点将它们重新插入索引结构的该层中，否则将结点 `node` 的子结点划分为  $k$  簇  $\{C_i | i = 1, \dots, k\}$ ，将分簇  $C_1$  作为结点 `node` 的子结点，为分簇集合  $\{C_i | i = 2, \dots, k\}$  分别新建结点  $\{\text{new\_node}_i | i = 2, \dots, k\}$ ，计算新结点的轴向包围盒，并将新节点  $\{\text{new\_node}_i | i = 2, \dots, k\}$  作为结点 `node` 的父结点的子结点插入到索引结构中，实现结点的分裂；4) 调整各结点的轴向包围盒，具体过程为：(1) 设新插入到索引结构中的结点的父结点为 `src_node`；(2) 调整父结点 `src_node` 的轴向包围盒，使其恰好包含父结点 `src_node` 的所有子结点；(3) 若父结点 `src_node` 为根索引结点，程序返回，否则继续执行；(4) 令父结点 `src_node` 为步骤 (1) 中父结点 `src_node` 的父结点，返回步骤 (2)；三、将体积过大的轴向包围盒重新插入到索引结构中，实现索引结构的优化；四、基于产品逆向工程数据动态索引结构，实现散乱点云、多边形网格以及分片连续曲面的拓扑近邻查询，其中查询任一空间对象 `T` 的邻接对象的具体步骤如下：1) 令空间对象 `T` 的轴向包围盒外接球为 `S`；2) 令  $\text{Search}(T, n)$  表示在以结点 `n` 为根索引结点的索引结构中查询空间对象 `T` 的邻近对象集合，若结点 `n` 为数据结点且与外接球 `S` 相交，则返回其包含的空间对象集合，若结点 `n` 为内部结点，则  $\text{Search}(T, n) = \text{Search}(T, n_1) \cup \text{Search}(T, n_2) \dots \text{Search}(T, n_i)$ ，其中  $n_i$  表示结点 `N` 中与外

接球 S 相交的子结点 ;3) 将当前结点 N 初始化为索引结构的根索引结点, 则空间对象 T 的邻近对象集合为  $\text{Search}(T, N)$ 。

2. 如权利要求 1 所述的产品逆向工程数据动态索引多目标自适应建立方法, 其特征在于 : 在步骤二的步骤 3) 中, 结点分裂的步骤具体是 :1) 对结点 node 的子结点进行二进制编码, 0 表示非聚类中心, 1 表示聚类中心, 并构造及初始化指定规模的种群  $P(t)$ ,  $t=1$ , 计算其目标函数和适应值 ;2) 依据个体的目标函数选出非支配解集  $E(t)$  ;3) 对种群  $P(t)$  进行选择、交叉、变异操作, 得到下一代种群  $P(t+1)$ , 令  $t=t+1$ ;4) 计算种群  $P(t)$  的目标函数值与适应值 ;5) 计算种群  $P(t)$  的非支配解集, 然后更新非支配解集  $E(t)$  ;6) 若达到截止的进化代数则跳转到步骤 7), 否则跳转到步骤 3);7) 对非支配解集  $E(t)$  进行解码, 然后从中选取轴向包围盒重叠度与轴向包围盒体积之和最小的分裂方案作为结点 node 的最优分裂方案 ;8) 令结点 node 的最优分裂方案为  $\{C_i | i = 1, \dots, k\}$ , 将分簇  $C_1$  作为结点 node 的子结点, 为分簇集合  $\{C_i | i = 2, \dots, k\}$  分别新建结点  $\{\text{new\_node}_i | i = 2, \dots, k\}$ , 计算新结点的轴向包围盒, 并将新节点  $\{\text{new\_node}_i | i = 2, \dots, k\}$  作为结点 node 的父结点的子结点插入到索引结构中。

3. 如权利要求 1 所述的产品逆向工程数据动态索引多目标自适应建立方法, 其特征在于 : 在步骤三中对索引结构进行优化, 步骤具体是 :1) 遍历索引结构, 计算叶索引结点层轴向包围盒的平均体积  $volume\_avg$  ;2) 遍历索引结构各叶索引结点, 若该叶索引结点轴向包围盒的体积大于  $\mu \times volume\_avg$  ( $\mu$  为用户设定的阈值, 通常取  $3^{\sim}5$ ), 则将其包含的数据结点添加到临时序列 L 中, 并将其包含的数据结点从索引结构中删除 ;3) 将序列 L 中的数据结点重新插入到索引结构中, 实现索引结构的全局优化。

## 产品逆向工程数据动态索引多目标自适应构建方法

### 技术领域

[0001] 本发明提供一种产品逆向工程数据动态索引多目标自适应构建方法，属于产品逆向工程技术领域。

### 背景技术

[0002] 在产品逆向工程技术领域，所处理的原始数据通常是来自实物表面采样而获得的散乱点云、多边形网格模型等数据格式，基于该类原始数据进行曲面重建生成分片连续曲面是产品逆向工程的核心技术。由于散乱点云、多边形网格以及分片连续曲面这些数据格式均表现为大规模甚至海量空间几何对象的复合结构，为这些数据类型构建一种通用且高效的索引技术，对于提高产品逆向工程数据处理效率具有重要意义。

[0003] 对现有文献检索发现，现有产品逆向工程数据的索引技术通常仅适用于某种特定的数据类型。在散乱点云处理中，空间八叉树与 K-D 树分别是应用最为广泛的静态索引与动态索引。周海在其博士学位论文“细分曲面造型技术研究”（南京航空航天大学，2005）中采用空间八叉树作为三角网格模型的空间索引结构，依据三角面片包围盒中心的位置将三角面片插入到空间八叉树中，建立三角网格模型索引结构，组织三角面片间的近邻关系，该方法以三角面片包围盒中心表示三角面片，不能准确反映三角面片所在位置及所占空间区域大小，准确性差，降低了索引结构的质量及基于该结构的空间查询效率。王占礼在其博士学位论文“面向虚拟制造的数控加工仿真技术研究”（吉林大学，2007）中采用一个大包围盒包围三角网格模型，将该包围盒作为根索引结点，然后将其中的三角面片分割成两部分，每一部分用一个包围盒包围，再对每一个包围盒递推进行分割，直到一个包围盒只包含一个三角面片，建立三角网格模型的非平衡二叉树索引结构，该结构提高了三角网格模型的空间查询效率，但由于该结构为非平衡二叉树，故只适用于分布较为均匀的三角网格模型，当三角网格模型分布疏密不均时，容易出现树的某一分支层数过多现象，导致数据结构急剧恶化，严重降低索引查询效率。孙殿柱等人在其学术论文“基于四维聚类的 R\*- 树结点分裂算法”（机械工程学报，2009, 45 (10) :180–184）中对 R\*- 树进行了改进，使之可统一索引散乱点云、多边形网格等数据类型，继而在其学术论文“三角 Bézier 曲面快速求交算法”（机械工程学报，2011, 47 (3) :89–94）中将改进的 R\*- 树作为分片连续曲面的索引结构以提高相交三角 Bézier 曲面片查询效率，但是由于改进的 R- 树在索引结点分裂过程中采用了 k- 均值聚类算法，需要用户交互设定聚类簇数，聚类簇数的不同会导致差别很大的索引结点分裂结果，导致索引结构与性能不稳定，此外 k- 均值聚类算法是一种局部搜索算法，对初始值过于敏感，采用爬山法迭代搜索最优的索引结点分裂结果，容易陷入局部极值，难以获得全局最优的索引结点分裂结果，导致未能充分发挥 R\*- 树的优势。

[0004] 综上所述，目前的产品逆向工程数据的动态索引结构已经具备了一定的通用性，可基于统一的索引机制处理各种类型的空间几何对象的复合结构，但是依然存在数据适应性较差、索引性能较低并且系统资源消耗较高等问题，为产品逆向工程数据构建稳定、高效的索引机制已成为本领域技术人员亟待解决的技术问题。

## 发明内容

[0005] 为克服现有产品逆向工程数据的索引机制的不足,本发明目的在于提供一种产品逆向工程数据动态索引多目标自适应构建方法,使之能索引各种类型的逆向工程数据,具有稳定性强、数据查询效率高的特点,技术方案如下:

一种产品逆向工程数据动态索引多目标自适应构建方法,其特征在于包含以下步骤:一、读取产品逆向工程数据,建立各空间对象的轴向包围盒,依据轴向包围盒的中心以及外接球半径建立其对应的数据结点,并存入数据结点序列;二、将数据结点插入到索引结构中,结点插入到索引结构的具体步骤是:1)为结点选择插入位置;2)将结点插入到步骤1)中得到的位置;3)令结点插入到结点 node 下,判断结点 node 的子结点数是否大于结点的最大子结点数,如果大于则对结点 node 进行溢出处理,若结点 node 为非根索引结点且在插入一个空间对象过程中该结点所在层第一次进行溢出处理,则在结点 node 中有选择地取出一部分结点,将它们重新插入索引结构的该层中,否则进行结点分裂;4)调整各结点的轴向包围盒;三、将体积过大的轴向包围盒重新插入到索引结构中,实现索引结构的优化;四、基于产品逆向工程数据动态索引结构,实现散乱点云、多边形网格以及分片连续曲面的拓扑近邻查询。

[0006] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,在步骤一中,读取逆向工程数据文件,若空间对象为散乱点则以该点为中心建立边长为单位1且各条棱均平行于坐标轴的轴向包围盒,若空间对象为多边形网格则建立恰好包围网格顶点的轴向包围盒,若空间对象为分片曲面片则建立恰好包围其控制顶点的轴向包围盒,建立各空间对象对应的数据结点,并将其存入数据结点序列,数据结点包含空间对象的信息以及对应的轴向包围盒信息。

[0007] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,在步骤二中,将数据结点插入到索引结构,方法是:结点包括索引结点和数据结点,索引结点包含根索引结点、内部索引结点和叶索引结点,索引结构的最上层结点为根索引结点、最下层结点为叶索引结点、其余结点为内部索引结点,定义  $M$  为结点的最大子结点数 ( $M$  为大于 2 的整数)、 $m$  为结点最小子结点数 ( $m$  为小于或等于  $M/2$  的整数),除根索引结点外,每个索引结点的子结点数均小于等于  $M$  且大于等于  $m$ ;索引结构中每个结点的轴向包围盒恰好包围该结点的所有子结点。

[0008] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,在步骤二中为结点选择插入位置的步骤具体是:1)令当前结点为 current\_node,如果索引结构为空则返回空,否则令 current\_node 为索引结构根索引结点;2)令结点将要插入的层数为 level,若结点为数据结点则 level 为索引结构的叶子层,其他类型结点的插入是由强制重新插入引起的, level 为其重新插入前所在层数;3)计算 current\_node 的每个子结点与待插入结点的轴向包围盒外接球重叠度,选择重叠度最小的作为 current\_node;4)重复步骤2)直到索引结构的 level 层为止。

[0009] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,在步骤二中,将结点插入到索引结构,令任意两结点  $c_i$ 、 $c_j$  轴向包围盒的外接球半径分别为  $r_i$ 、 $r_j$ ,则结点  $c_i$  与结点  $c_j$  的轴向包围盒外接球重叠度为:

、 $r_j$ ，轴向包围盒中心间的距离为  $d_{ij}$ ，两结点  $c_i$ 、 $c_j$  轴向包围盒的外接球重叠度的计算公式为  $\xi_{ij} = \frac{r_i + r_j}{d_{ij}}$ ，以结点轴向包围盒外接球重叠度衡量两结点间的相似性大小，重叠度越大则结点间的相似性越大，否则越小。

[0010] 为实现发明目的，所述的产品逆向工程数据动态索引多目标自适应构建方法，在步骤二的步骤 3) 中，选择重新插入结点的步骤具体是：1) 对溢出结点 node 的  $M+1$  个子结点，计算它们的轴向包围盒的中心到结点 node 的轴向包围盒的中心的距离；2) 以步骤 1) 中计算的距离值为关键字，对结点 node 的子结点进行降序排序，选出前  $p$  个子结点。

[0011] 为实现发明目的，所述的产品逆向工程数据动态索引多目标自适应构建方法，在步骤二的步骤 3) 中，结点分裂的步骤具体是：1) 对结点 node 的子结点进行二进制编码，0 表示非聚类中心，1 表示聚类中心，并构造及初始化指定规模的种群  $P(t)$ ， $t=1$ ，计算其目标函数和适应值；2) 依据个体的目标函数选出非支配解集  $E(t)$ ；3) 对种群  $P(t)$  进行选择、交叉、变异操作，得到下一代种群  $P(t+1)$ ，令  $t=t+1$ ；4) 计算种群  $P(t)$  的目标函数值与适应值；5) 计算种群  $P(t)$  的非支配解集，然后更新非支配解集  $E(t)$ ；6) 若达到截止的进化代数则跳转到步骤 7)，否则跳转到步骤 3)；7) 对非支配解集  $E(t)$  进行解码，然后从中选取轴向包围盒重叠度与轴向包围盒体积之和最小的分裂方案作为结点 node 的最优分裂方案；8) 令结点 node 的最优分裂方案为  $(C_i | i = 1, \dots, k)$ ，将分簇  $C_1$  作为结点 node 的子结点，为分簇集合  $(C_i | i = 2, \dots, k)$  分别新建结点  $(new\_node_i | i = 2, \dots, k)$ ，计算新结点的轴向包围盒，并将新节点  $(new\_node_i | i = 2, \dots, k)$  作为结点 node 的父结点的子结点插入到索引结构中。

[0012] 为实现发明目的，所述的产品逆向工程数据动态索引多目标自适应构建方法，在步骤二的步骤 3) 的结点分裂过程中，目标函数包括分簇的簇数  $k$  与分簇的类内聚  $G_c$ ，分簇的簇数  $k$  为个体编码中 1 的个数，计算类内聚  $G_c$  需先对个体解码，即从个体编码中提取出聚类中心  $N(N_1, N_2, \dots, N_k)$ ，计算其他结点的轴向包围盒中心到聚类中心的距离，将其归到距其最近的聚类中心所代表的分簇，形成分簇集合  $(C_i | i = 1, \dots, k)$  实现个体的解码，则类内聚的定义为  $G_c = \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - z_i\|$ ，其中  $k$  表示簇数， $x_j$  为结点， $z_i$  为分簇  $C_i$  的聚类中心；若

不存在任何  $x \in F$  使得公式  $f_i(x) \geq f_i(x^*)$ ,  $i \in [1, q]$ ,  $x \in F$ ,  $x^* \in F$  成立，且至少一个解是严格不等的，则认为解  $x^*$  为非支配解，其中  $F$  表示所有的可行解， $f_i$  表示各个目标函数，即簇数

$k$  与类内聚  $G_c$ ，由所有非支配解构成非支配解集  $E(t)$ ；如果可行解中的两个解  $x, y$  满足公式  $\forall i \in \{1, 2, \dots, q\}: f_i(x) \geq f_i(y)$  和  $\exists j \in \{1, 2, \dots, q\}: f_j(x) > f_j(y)$  则认为  $x$  支配  $y$ ；适应值定义

为  $r(x) = \frac{1}{r(x)}$ ，其中  $r(x)$  表示个体  $x$  的排序值，如果个体  $x$  为非支配解则其排序值为 1，否则其排序值为支配该个体的的个体数目加一。

[0013] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,将结点插入到索引结构,调整结点轴向包围盒的步骤具体是:1)设新插入到索引结构中的结点的父结点为src\_node;2)调整父结点src\_node的轴向包围盒,使其恰好包含父结点src\_node的所有子结点;3)若父结点src\_node为根索引结点,程序返回,否则继续执行;4)令父结点src\_node为步骤1)中父结点src\_node的父结点,返回步骤2)。

[0014] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,在步骤三中对索引结构进行优化,步骤具体是:1)遍历索引结构,计算叶索引结点层轴向包围盒的平均体积 $volume\_avg$ ;2)遍历索引结构各叶索引结点,若该叶索引结点轴向包围盒的体积大于 $\mu \times volume\_avg$ , $\mu$ 为用户设定的阈值,通常取3~5,则将其包含的数据结点添加到临时序列L中,并将其包含的数据结点从索引结构中删除;3)将序列L中的数据结点重新插入到索引结构中,实现索引结构的全局优化。

[0015] 为实现发明目的,所述的产品逆向工程数据动态索引多目标自适应构建方法,在步骤四中,查询任一空间对象的邻接对象的具体步骤如下:1)令空间对象T的轴向包围盒外接球为S;2)令 $Serach(T, N)$ 表示在以结点N为根索引结点的索引结构中查询空间对象T的邻近对象集合,若结点N为数据结点且与外接球S相交,则返回其包含的空间对象集合,若结点N不是数据结点,则 $Serach(T, N) = Serach(T, n_1) \cup Serach(T, n_2) \dots Serach(T, n_i)$ ,其中 $n_i$ 表示结点N中与外接球S相交的子结点;3)将当前结点N初始化为索引结构的根索引结点,则空间对象T的邻近对象集合为 $Serach(T, N)$ 。

[0016] 本发明与现有技术相比,具有以下四个特点:

1)依据结点轴向包围盒外接球重叠度衡量空间对象的相似性,既反映了空间对象所在位置又反映了其所占空间区域大小,提高了结点在空间范围内的聚合性,分簇结果更合理;

2)结合遗传多目标优化算法进行结点分裂求解结点分裂的最优解,降低了结点分裂过程的参数依赖性,提高了空间对象的空间索引结构的建立效率;

3)将体积过大的叶索引结点包含的数据结点重新插入到索引结构中,避免了轴向包围盒奇异结点的产生,提高了索引结构的质量;

4)采用深度优先遍历方法快速准确的获取三角面片的拓扑近邻三角面片,可有效缩小查询范围,提高目标三角面片拓扑近邻面片的查询效率。

## 附图说明

[0017] 图1是本发明产品逆向工程数据动态索引结构建立程序实现流程图;

图2是多边形网格及其轴向包围盒示意图;

图3是本发明索引结构的平面结构示意图;

图4是本发明索引结构的树状结构示意图;

图5是本发明产品逆向工程数据动态索引结构新结点插入流程图;

图6是本发明产品逆向工程数据动态索引结构结点分裂流程图;

图7是本发明人脸多边形网格模型;

图 8- 图 11 是本发明对人脸多边形网格模型建立的空间索引结构各层结点轴向包围盒。

## 具体实施方式

[0018] 下面结合附图对本发明作进一步说明：

图 1 是本发明产品逆向工程数据动态索引结构建立程序实现流程图，产品逆向工程数据动态索引结构建立程序包含读取产品逆向工程数据文件程序 1，将数据结点插入到索引结构程序 2，优化索引结构程序 3 及目标空间对象近邻对象查询程序 4，其中，读取产品逆向工程数据文件程序 1 读取产品逆向工程数据文件，建立各空间对象的轴向包围盒，依据轴向包围盒的中心以及外接球半径建立其对应的数据结点，并存入数据结点序列；将数据结点插入到索引结构程序 2 读取数据结点序列，选择结点插入位置，判断结点的子结点数是否大于结点的最大子结点数，若大于则进行结点溢出处理，若结点为非根结点且在插入一个空间对象过程中该结点所在层第一次进行溢出处理，则在结点中有选择地取出一部分结点，将它们重新插入索引结构的该层中，否则进行结点分裂；自轴向包围盒发生变化的结点，沿着索引结构的一条分枝，自底向上调整各结点的轴向包围盒；优化索引结构程序 3 删除轴向包围盒体积过大或过于狭长的叶索引结点，将其包含的子结点重新插入到索引结构中，实现索引结构的优化；目标空间对象拓扑近邻对象查询程序 4 深度优先遍历产品逆向工程数据索引结构，获取与目标空间对象轴向包围盒外接球相交的数据结点，从其包含的空间对象中选择目标空间对象的拓扑近邻对象。

[0019] 图 2 是若干多边形网格集合。定义索引结构中结点的最小子结点数  $m=3$ 、最大子结点数  $M=8$ ，图 3 是对图 2 所示多边形网格建立的空间索引结构中结点轴向包围盒示意图，图 4 为索引结构的树状结构示意图，结点 A 是根索引结点，B、C、D 为叶索引结点，E、F、G、H、I、J、K、L、M、N、O、P、Q 为数据结点，每个数据结点包含一个多边形网格。

[0020] 图 5 所示为将结点插入到索引结构程序 2 实现流程图，调用选择结点插入位置程序 1) 选择结点应插入的位置，将结点插入到程序 1) 得到的位置，判断结点的父结点的子结点数是否大于结点的最大子结点数，若大于则进行结点溢出处理，若结点为非根索引结点且在插入一个空间对象过程中该结点所在层第一次进行溢出处理，则调用选择重新插入结点程序 2) 有选择的从结点中选择若干个结点重新插入到索引结构中，否则调用结点分裂程序 3) 对新插入结点的父结点进行结点分裂，调用调整结点轴向包围盒程序 4) 对自新插入结点，沿着索引结构的一条分枝，调整各结点的轴向包围盒，使各结点的轴向包围盒恰好包围其包含的子结点。

[0021] 将结点插入到索引结构程序 2 中，选择结点插入位置程序 1) 的步骤具体是：1) 若当前索引结构为空，则返回空；否则设当前结点  $current\_node$  为索引结构的根索引结点；2) 计算当前结点  $current\_node$  的每个子结点  $c_i$  与待插入结点  $c_j$  的的轴向包围盒中心间的

距离为  $d_{i,j}$ ，采用公式  $\xi_{i,j} = \frac{r_i + r_j}{d_{i,j}}$  计算两结点的外接球重叠度，其中  $r_i$ 、 $r_j$  分别为结点  $c_i$  与  $c_j$  的轴向包围盒外接球半径，选择重叠度最小的作为当前结点  $current\_node$ ；3) 重复步骤 2) 直到当前结点  $current\_node$  所在层为叶索引结点层。

[0022] 将结点插入到索引结构程序 2 中,选择重新插入结点程序 2) 的步骤具体是: 1) 对溢出结点 node 的  $M+1$  个子结点,计算它们的轴向包围盒的中心到结点 node 的轴向包围盒的中心的距离;2) 以步骤 1) 中计算的距离值为关键字,对结点 node 的子结点进行降序排序,选出前  $P$  个子结点。

[0023] 将结点插入到索引结构程序 2 中,若结点 node 的子结点数  $n$  大于最大子结点数  $M$ ,则调用结点分裂程序 3) 对结点 node 进行结点分裂,其具体过程如图 6 所示,过程(1)中个体采用长度为  $M+1$  的二进制串表示,每一位对应于一个结点,1 表示聚类中心,0 表示非聚类中心,随机地将个体中每一位置为 0 或 1,实现个体初始化,然后分配并初始 population\_size 个个体,完成种群的初始化;过程(2)中目标函数包括簇数  $k$  以及类内距  $G_c$ ,统计个体中 1 的个数即为簇数  $k$ ,计算类内距需先对个体解码,提取出个体中 1 所对应的结点  $N(N_1, N_2, \dots, N_k)$ ,即分簇中心,然后计算其他结点到各分簇中心的距离,将其归为距离最近的分簇中心所表示的分簇,形成本簇集合  $\{C_i | i = 1, \dots, k\}$ ,计算类内距

$$G_c = \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - z_i\|, \text{ 其中 } k \text{ 表示簇数, } x_j \text{ 为结点, } z_i \text{ 为分簇 } C_i \text{ 的聚类中心; 过程(3)计算种}$$

群中个体的适应值,计算公式为  $f(x) = \frac{1}{r(x)}$ ,其中  $r(x)$  表示个体  $x$  的排序值,若个体  $x$  与  $y$ ,满足  $k(x) \leq k(y)$  和  $G_c(x) \leq G_c(y)$  且其中有一个公式是严格不等的,则个体  $x$  支配个体  $y$ ,若没有任何个体能支配个体  $x$ ,则个体  $x$  为非支配解,对种群中的非支配个体分配排序值 1,其他个体的排序值等于支配该个体的个体数目加一;过程(4)构造 Pareto 解集合  $E(t)$ ,遍历每个个体,对于每个个体寻找是否有能支配该个体的个体存在,若不存在则将该个体加入集合  $E(t)$ ;过程(5)对 Pareto 解集合  $E(t)$  的个体解码,从中选择轴向包围盒重叠度与轴向包围盒体积之和最小的分裂方案作为结点的最优分裂方案  $\{C_i | i = 1, \dots, k\}$ ,将分簇  $C_1$  作为结点的子结点,为分簇集合  $\{C_i | i = 2, \dots, k\}$  分别新建结点  $(new\_node_i | i = 2, \dots, k)$ ,计算新结点的轴向包围盒,并将新节点  $(new\_node_i | i = 2, \dots, k)$  作为结点的父结点的子结点插入到索引结构中。

[0024] 优化索引结构程序 3 的步骤具体是:1) 遍历索引结构,计算叶索引结点层轴向包围盒的平均体积  $volume\_avg$ ;2) 遍历索引结构各叶索引结点,若该叶索引结点轴向包围盒的体积大于  $\mu \times volume\_avg$  ( $\mu$  为用户设定的阈值,通常取 3-5),则将其从索引结构中删除,并将其包含的空间对象添加到临时序列 L 中;3) 将序列 L 中的空间对象重新插入到索引结构中,实现索引结构的全局优化。

[0025] 图 7 是人脸多边形网格模型,该模型型面特征较为复杂,由 47765 个多边形网格构成,采用本发明建立其空间索引结构,各层结点轴向包围盒效果如图 8- 图 11 所示。图 8 为根索引结点的轴向包围盒,图 9 为内部索引结点的轴向包围盒,图 10 为叶索引结点的轴向包围盒,图 11 为数据结点的轴向包围盒,除根索引结点外,每个索引结点的子结点数均在 8~20 范围内,每个数据结点包含一个多边形网格。

[0026] 查询任一空间对象 T 的邻接对象的步骤具体是 :1) 求解空间对象 T 的轴向包围盒外接球 S ;2) 定义函数  $Serach(T, N)$  表示在以结点 N 为根索引结点的索引结构中查询空间对象 T 的邻近对象集合, 若结点 N 为数据结点且与外接球 S 相交, 则返回其包含的空间对象集合, 若结点 N 为内部结点, 则  $Serach(T, N) = Serach(T, n_1) \cup Serach(T, n_2) \dots Serach(T, n_i)$  , 其中  $n_i$  表示结点 N 中与外接球 S 相交的子结点 ;3) 将当前结点 N 初始化为索引结构的根索引结点, 则空间对象 T 的邻近对象集合为  $Serach(T, N)$  。

[0027] 其它产品逆向工程数据的动态索引结构构建方法同上。

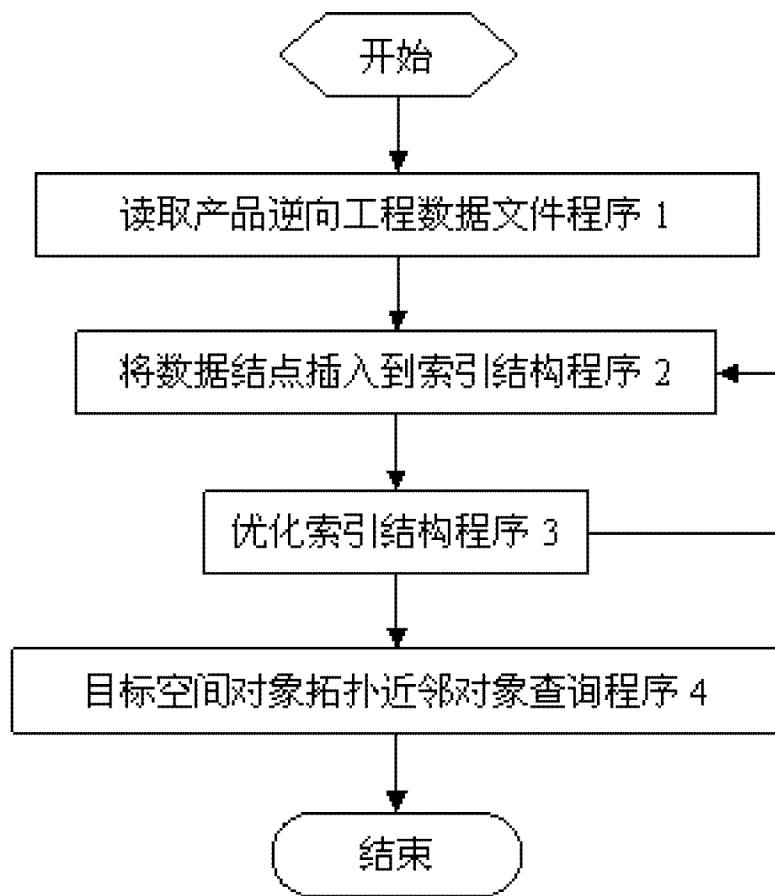


图 1

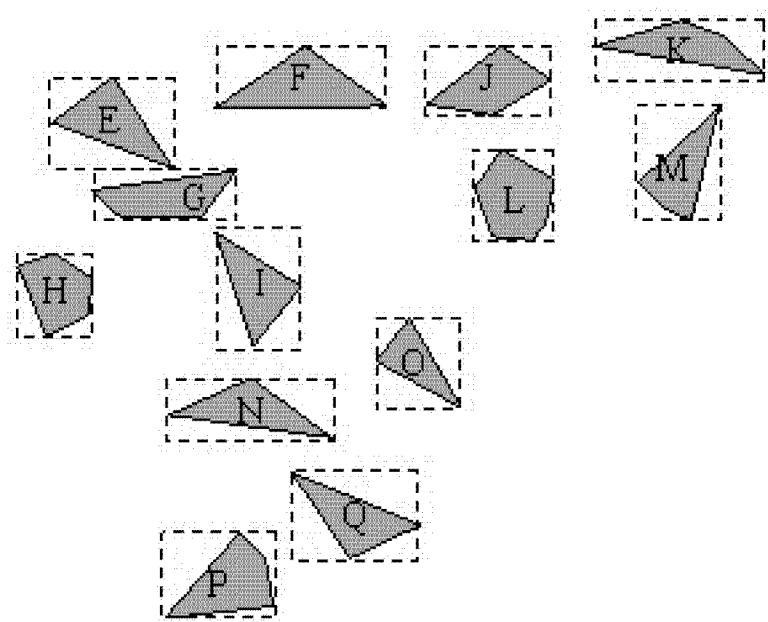


图 2

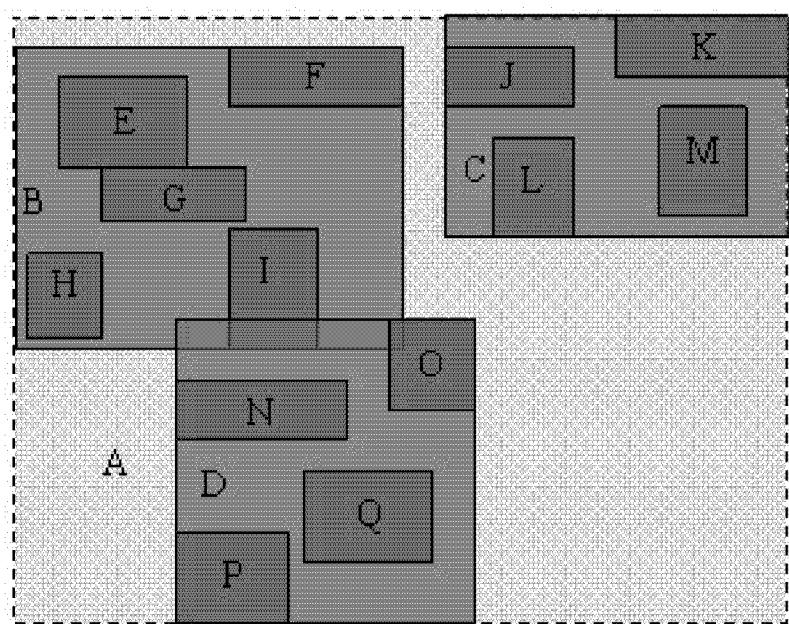


图 3

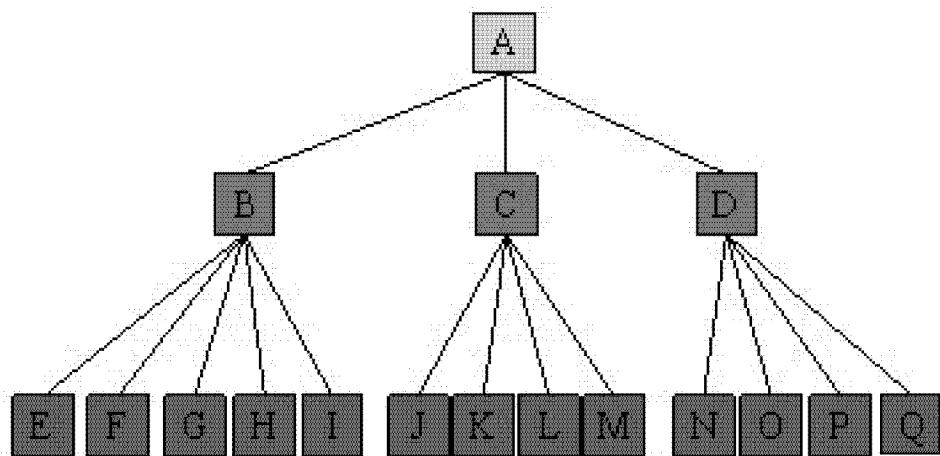


图 4

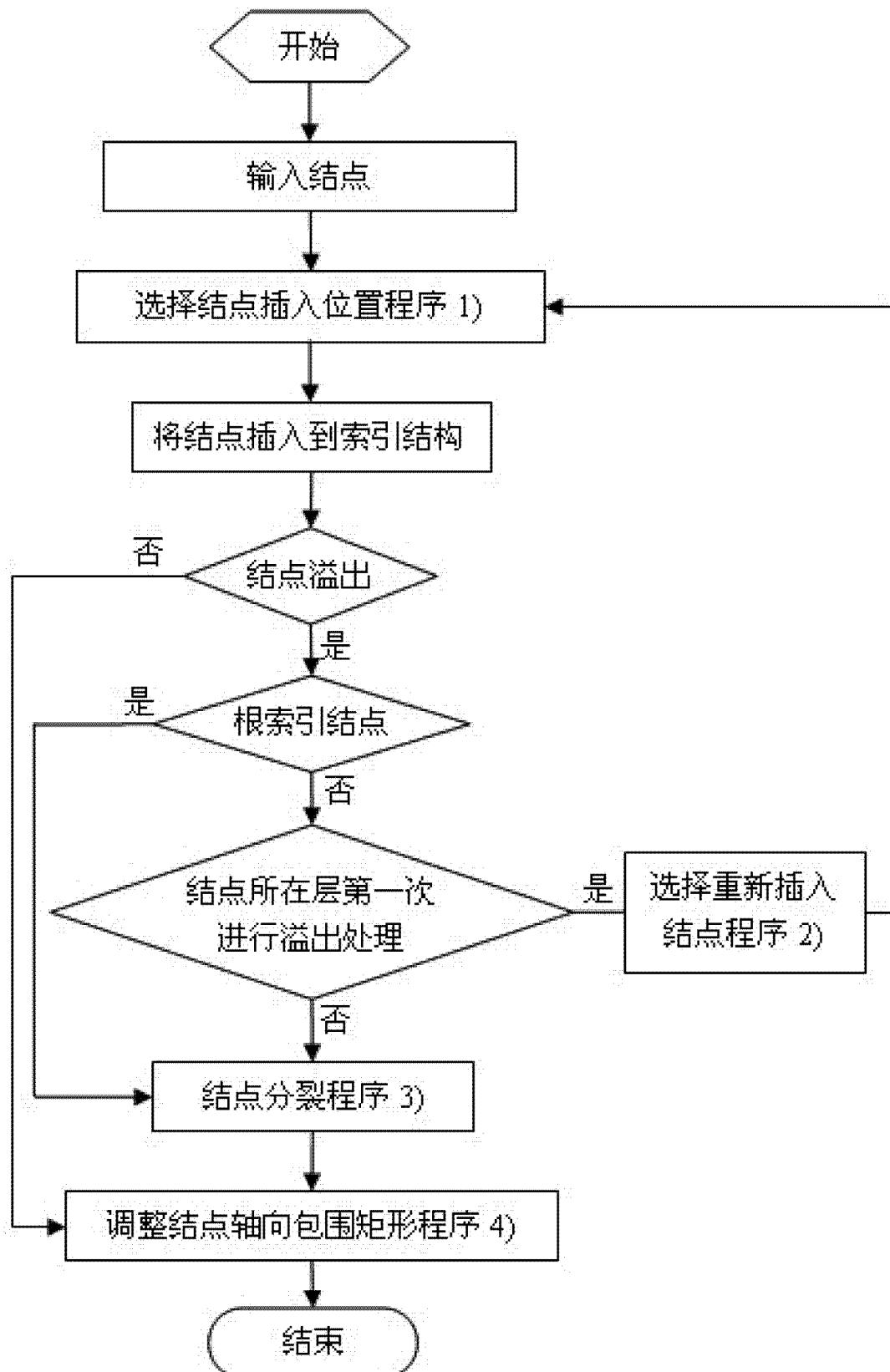


图 5

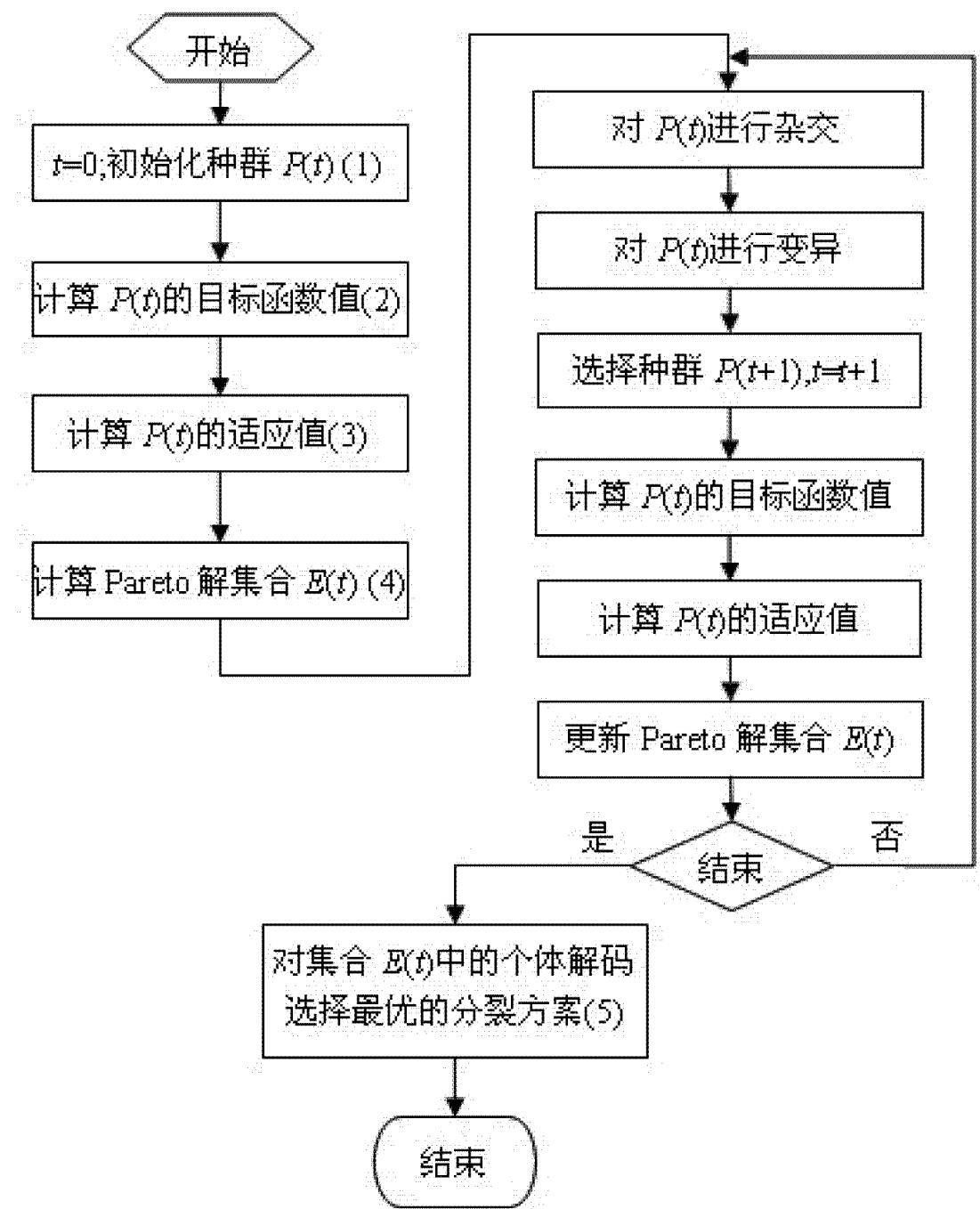


图 6

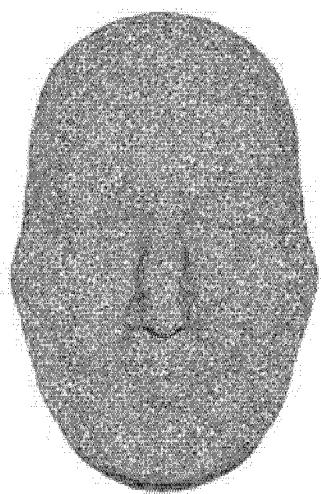


图 7

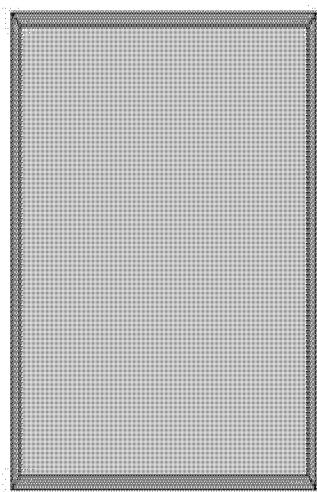


图 8

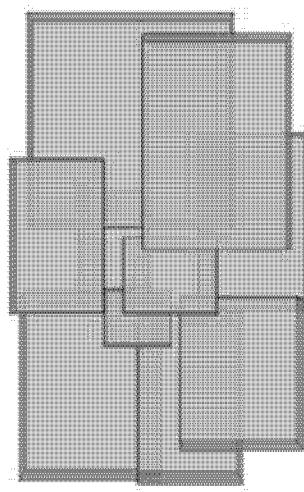


图 9

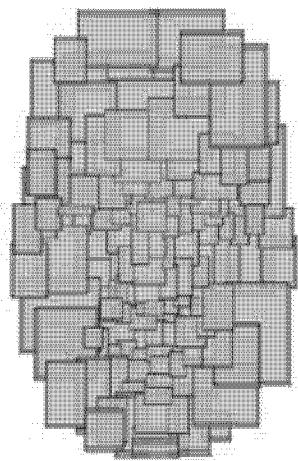


图 10

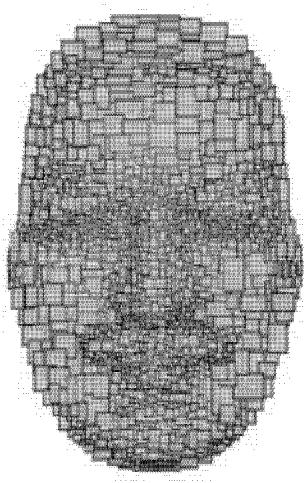


图 11