

一种基于关联挖掘的自适应构件检索方法

薛云皎¹, 钱乐秋¹, 花 鸣², 张天戈¹, 杨 珉¹

(1. 复旦大学计算机科学与工程系, 上海 200433; 2. 复旦大学计算机与信息技术系, 上海 200433)

摘 要: 构件复用过程中, 用户常因对构件描述机制认识有限而难以提出准确的检索需求, 从而影响查准率. 针对基于刻画描述的软件构件, 借鉴数据挖掘中关联规则挖掘的有关理论, 提出了带有用户反馈的自适应构件检索模型以及基于关联挖掘的自适应学习算法, 从用户检索的历史记录中挖掘用户的显式检索条件与隐性检索需求之间的内在联系, 从而完整化和精确化用户的检索条件, 提高构件检索的查准率. 同时, 用实验结果证明了该方法的有效性和可行性.

关键词: 构件库; 构件检索; 刻画描述; 关联挖掘; 自适应

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2004) 12A-203-04

A Self-Adaptive Approach of Component Retrieval Based on Association Mining

XUE Yun-jiao¹, QIAN Le-qiu¹, HUA Ming², ZHANG Tian-ge¹, YANG Min¹

(1. Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China;

2. Department of Computing and Information Technology, Fudan University, Shanghai 200433, China)

Abstract: During the process of component reusing, users often have no all-around understanding of the component description mechanism. So it's difficult for users to propose exact retrieval requirements, which leads to lower precision. We refer to association mining theory in data mining field, and present a self-adaptive component retrieval model with user-feedback and the association mining-based self-adaptive learning algorithm. It can obtain the internal relationships between users' explicit retrieval conditions and their implicit requirements through mining in the retrieval history, thus making the retrieval conditions more complete and precise, which increase the precision of component retrieval. The experimental results show that this solution is feasible and effective.

Key words: component repository; component retrieval; faceted description; association mining; self-adaptive

1 引言

在基于构件复用的软件开发过程中, 对大量可复用软件构件的管理非常重要^[1]. 构件库作为一种支持软件复用的基础设施, 提供了对于可复用构件的描述、分类、存储和检索等功能. 其中, 构件检索是构件库研究中的重要问题, 有效的构件检索机制能够降低构件复用成本. 一个构件库检索系统的检索质量通常通过查全率、查准率和检索效率进行衡量^[2]. 检索系统的查全率指在一次检索过程中, 检索结果中实际符合用户检索需求的构件个数与构件库中满足用户检索需求的构件个数的比值; 查准率指的是检索结果中实际符合用户检索需求的构件个数与整个检索结果中构件个数的比值; 效率指在构件库中进行一次检索所需的时间. 在实际的应用过程中, 构件的复用者并不是构件的生产者或构件库的管理员, 故在对构件进行检索时, 常常对构件库中描述构件的刻画和术语理解不充分, 因而难以给出完整和精确的检索需求, 从而对检索的查准率造成影响. 但实际上可以根据用户最终选取的结果反映出其真实需求. 如果能够从用户的检索行为及用户对

检索结果的反馈中, 推断出用户的检索特点即反映用户给出的非精确检索条件与用户实际需要的精确检索条件之间内在联系的模式, 则利用这些模式可以对新的不完整的检索需求进行补充, 从而提高系统的查准率.

本文在研究构件检索特点的基础上, 针对基于刻画描述的构件, 借鉴数据挖掘中关联规则挖掘的理论, 将数据挖掘方法引入构件检索, 从用户检索行为及反馈中, 挖掘出非精确检索条件与精确检索结果之间的关联规则, 从而调整检索机制, 提高构件检索的查准率.

本文第二部分介绍构件检索方面的工作; 第三部分介绍数据挖掘中关联规则挖掘的相关理论; 第四部分提出基于关联挖掘的自适应构件检索过程模型, 以及模型的核心部分——基于关联挖掘的自适应学习算法. 最后, 通过实例和实验结果展示了本方法对提高查准率的作用.

2 相关工作

现有构件的分类方法^[4] 主要包括人工智能方法^[9,10,11]、超文本方法^[12] 和信息科学方法^[13,14]. 其中, 信息科学方法是

收稿日期: 2004-09-01; 修回日期: 2004-11-14

基金项目: The Development Foundation for Key Disciplines of Shanghai Education Commission of China under Grant (No. B990105); 上海市科技攻关项目“上海构件库及其应用研究”(No. 025115014); 国家 863 计划项目“上海构件库及其应用”(No. 2002AA114010)

实际应用中较为普遍的途径,该方法中又以关键字分类和刻面分类两种应用最多。

构件的刻面描述是一种逐渐受到重视的描述方法,对于它的检索技术的研究主要是基于 XML 的文本检索结合术语辞典的联想检索技术。文献[5]介绍了青鸟系统中支持刻面、属性、关键字等多种查询方法组合的构件查询工具,它还支持以构件间关系为超链的超文本游历和以术语间层次关系为线索的构件浏览。文献[6]结合模式分析中的树匹配思想,根据刻面描述特点,提出了一个包含 3 个层次、5 种匹配类型的刻面匹配模型,并给出了该模型的泛型算法以及具体情况下的算法实现与时间复杂度讨论。

在构件检索机制的自动调整方面,文献[7]提出了一个包括结构匹配和功能匹配的精确的构件描述匹配方法,其中功能匹配保证检索出符合需求的构件,而结构匹配用来检查可适应性。文章提出用属性标识的转换系统来定义结构匹配和功能匹配,并且提出了构件自适应的方法。文献[8]提出了构件的精确检索和近似检索,当用户精确检索失败时,系统自动进行近似检索,并且保证近似检索的结果集是构件库中最接近用户需求的,需要的调整代价最小。

3 关联规则挖掘

在数据挖掘理论中,对大量数据中数据项间的相关联系有比较深入的研究。Jiawei Han 等在[3]中介绍了在大型数据库中寻找频繁项集的 Apriori 算法,以及由频繁项集产生强关联规则的过程。

3.1 基本概念

事务数据库中的原子数据称为项,项的集合称为项集,包含 k 个项的项集称为 k -项集。设 $I = \{i_1, i_2, \dots, i_m\}$ 是所有项的集合,在事务数据库 D 中,每个事务 T 可以表示其涉及到的项的集合,有 $T \subseteq I$ 。设 A 是一个项集,事务 T 包含 A 当且仅当 $A \subseteq T$ 。一个项集的出现频率指数据库中包含该项集的事务的数量。最小支持度阈值 ($min. sup$) 是一个由用户指定的数量值,用以界定项集的频繁出现程度。如果一个项集的出现频率大于或等于 $min. sup$ 与 D 中事务总数的乘积,就称该项集满足最小支持度 $min. sup$,并称之为频繁项集。

关联规则是形如 $A \Rightarrow B$ 的蕴涵式,其中 $A \subset I, B \subset I$, 并且 $A \cap B = \phi$ 。关联规则 $A \Rightarrow B$ 表示 A 和 B 之间具有联系(联系的紧密程度由下文的强规则定义)。规则 $A \Rightarrow B$ 在事务数据库 D 中的支持度 s 定义为: D 中包含 $A \cup B$ (即 A 和 B 中的所有项)的事务占所有事务的百分比,即概率 $P(A \cup B)$ 。规则 $A \Rightarrow B$ 在事务数据库 D 中的置信度 c 定义为: D 中同时包含 A 和 B 的事务占包含 A 的事务的百分比,即条件概率 $P(B|A)$ 。

同时满足最小支持度阈值 ($min. sup$) 和最小置信度阈值 ($min. conf$) 要求的规则称为强规则,其中最小支持度阈值和最小置信度阈值由用户给定。强规则揭示的是两个项集之间的内在联系,找出强规则,就能找到某条件的出现(即强规则左部)频繁地导致另一条件(即强规则右部)同时出现的模式。

3.2 挖掘频繁项集的 Apriori 算法

文献[3]中提出的 Apriori 算法是一种最有影响的挖掘布

尔关联规则频繁项集的算法。算法使用逐层搜索的迭代方法, k -项集用于探索 $(k+1)$ -项集。并且,算法基于一种称作 Apriori 性质的重要性质:频繁项集的所有非空子集都必须也是频繁的。

算法首先通过遍历事务数据库 D 找出出现次数超过 $min. sup$ 的 1 项集,称为频繁 1 项集。然后将 1 项集两两连接形成 2 项集,由于频繁项集的子项集必定频繁,所以经过连接而成的 2 项集包含了所有的频繁 2 项集,故称为频繁 2 项集的候选集合。以此类推,通过频繁 $k-1$ 项集产生频繁 k 项集,直到再也找不到频繁的项集。

Algorithm: Apriori

Input: Transaction Database D ; minimum support threshold $min. sup$

Output: frequent itemsets L

Method:

- (1) Scan D to find frequent 1-itemset L_1
- (2) for ($k=2; L_{k-1} \neq \phi; k++$) {
- (3) join L_{k-1} with itself to generate the candidate sets C_k of frequent k -itemset
- (4) for each transaction t in D {
- (5) find all itemsets in t which belongs to C_k to form C_t
- (6) for each item c in C_t
- (7) c.count ++
- (8) }
- (9) $L_k = \{c \in C_k | c.count \geq min. sup\}$
- (10) }
- (11) return L_k

图 1 挖掘频繁项集的 Apriori 算法

4 基于关联规则的自适应构件检索方法

在基于刻面描述的构件的检索中,用户结合复用需求和构件库提供的刻面以及术语集提出检索关键词,提交检索系统。用户对检索需求描述的完整性和精确性取决于用户对构件库所采用的刻面及术语的熟悉程度。由于用户并非构件生产者或构件库管理员,难以完全理解刻面和术语的含义,故可能在提出检索需求时遗漏部分刻面或术语,这称为用户的隐含需求。在构件库检索系统返回检索结果后,用户可以根据实际需要在结果集中进行进一步筛选。我们可以根据用户在筛选过程中表现出的行为特点,挖掘出用户的隐含需求,从而在检索中自动加入隐含需求的约束,在一定程度上提高构件库检索系统的查准率。

4.1 带有用户反馈的自适应检索模型

本文在构件检索过程模型的基础上提出了基于关联挖掘的自适应构件检索过程模型:

在图 2 所示的检索过程模型中,用户的检索需求被同时送到检索系统和学习系统,检索系统执行检索后,将结果返回给用户,用户在结果集中进行筛选,得出反馈的检索结果送到学习系统。学习系统对一定数量的检索需求和反馈的检索结果进行分析后,得出形如 $A \Rightarrow B$ 的关联规则,其中 A 是出现在用户检索需求中的刻面或术语集合, B 是出现在反馈的检索

结果中的构件的描述剖面或术语集合 A 和 B 共同出现在一次查询中的概率大于给定的阈值 $min. sup$, 并且包含 A 的查询同时也包含 B 的概率大于 $min. conf$, B 就是用户在检索中忽略的隐性需求。

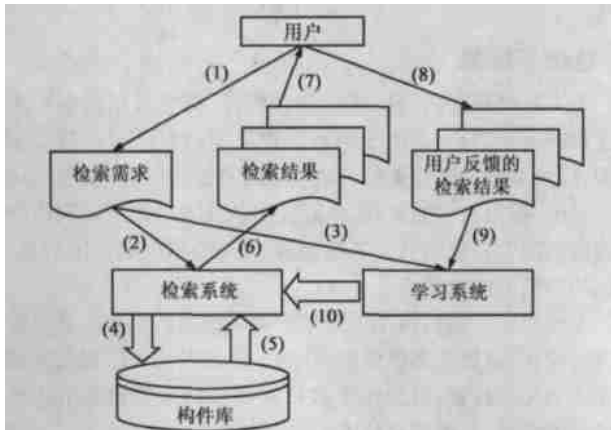


图 2 基于关联挖掘的自适应构件检索过程模型

- (1) 用户提出检索需求
- (2) 将检索需求发送到检索系统
- (3) 将检索需求发送到学习系统
- (4) 检索系统发送查询到构件库
- (5) 构件库返回查询结果
- (6) 检索系统将返回的信息组织成检索结果
- (7) 检索结果送达用户
- (8) 用户对检索结果筛选得反馈的检索结果
- (9) 将用户反馈的检索结果送到学习系统
- (10) 学习系统将调整方案送到检索系统

学习系统将计算出的关联规则送到检索系统,在以后的检索中,若用户检索需求中出现剖面或术语集合 A ,则检索系统在检索中自动加上隐性需求 B ,由 A 和 B 共同为约束条件进行检索,则检索结果在大部分情况下比仅用 A 作为条件进行的检索更能满足用户的需求。

4.2 基于关联挖掘的自适应学习算法

学习系统是检索过程模型的核心部分,它采用关联挖掘手段学习用户的检索模式,本节介绍学习系统的核心算法——基于关联挖掘的自适应学习算法。

4.2.1 基本概念 设构件库中术语集为 $T = \{t_1, t_2, \dots, t_n\}$, 用户的检索需求表示成 $Q (Q \subset T)$, 而用户反馈的检索结果集包含 m 个构件,每个构件的描述术语集为 $D_i (D_i \subset T)$. 学习系统维护一个用户检索行为历史记录数据库 H , 每执行一次检索,将用户的检索需求 Q 和反馈的检索结果集中每一个构件的描述术语集 D_i 分别存入数据库 H .

用户的检索需求与隐性需求间的关联规则即形如 $A \Rightarrow B$ 的蕴涵式,其中 $A \subseteq T, B \subseteq T$, 并且 $A \cap B = \phi$. 规则 $A \Rightarrow B$ 的含义为:用户提出的检索需求仅包含 A 时,他对检索的实际隐含需求为 B . 规则在数据库 H 中成立,具有支持度 s , 其中 s 是 H 中记录包含 $A \cup B$ 的百分比,即概率 $P(A \cup B)$. 规则 $A \Rightarrow B$ 在数据库 H 中具有置信度 c , 其中 c 是 H 中包含 A 的事务同时也包含 B 的百分比,即条件概率 $P(B|A)$. 同时满足最小支持度阈值 ($min. sup$) 和最小置信度阈值 ($min. conf$) 的规则称为强规则。

4.2.2 算法描述 基于关联挖掘的自适应学习算法采用 Apriori 算法寻找用户检索中频繁出现的术语集,以及在这一检索条件下,用户经常选择的构件的信息。

用户的检索条件 (Q) 和反馈的检索结果集中每一个构件的描述术语集 (D) 保存在检索历史数据库 H 中. 算法首先用 Apriori 算法找出用户检索术语集 Q 中的频繁检索术语集 QL (出现次数超过给定的最小支持度阈值). 对于 QL 中每一个术语集 ql , 在它对应的用户反馈检索结果集中寻找频繁描述术语 DL (出现次数超过反馈检索结果集记录总数 \times 置信度阈值). 选择 DL 中出现次数最多的频繁术语集,与 ql 一起组成强规则。

Algorithm: Self-adaptive Learning Algorithm Based On Association Mining
 Input: retrieval history database H , minimum support threshold $min. sup$,
 minimum confidence threshold $min. conf$

Output: Strong rule sets R

Method:

- (1) Find frequent retrieval term sets QL in Q using Apriori
- (2) for ($i = 1; QL \neq \phi; i++$) {
- (3) fetch one frequent term set ql in QL whose corresponding user feedback retrieval result set is D
- (4) find frequent term set DL whose support counting is larger than $count(D) \times min. conf$ in D using Apriori
- (5) select the frequent term set $dl (dl \in ql)$ which has the largest support counting in DL
- (6) add the rule $ql \Rightarrow dl$ into R
- (7) remove ql from QL
- (8) }

图 3 基于关联挖掘的自适应学习算法

学习系统将强规则集合 R 送到检索系统,检索系统在检索时,首先找出检索中包含的最长频繁检索术语 ql , 然后将 ql 对应的强规则右边的频繁描述术语 dl 合并到 ql 进行检索. 由于合并后的描述信息比 ql 中丰富,故能提高检索的查准率。

4.2.3 算法实例

设构件库中构件的描述剖面为操作系统、应用领域和使

表 1 构件库中的所有记录

| 构件编号 | 操作系统 | 应用领域 | 使用的数据库管理系统 |
|------|---------|------|------------|
| 1 | Windows | 税务 | SQL Server |
| 2 | Windows | 金融 | Oracle |
| 3 | Linux | 税务 | Oracle |
| 4 | Unix | 保险 | Oracle |
| 5 | Windows | 金融 | Mysql |
| 6 | Unix | 税务 | SQL Server |
| 7 | Windows | 金融 | SQL Server |
| 8 | Windows | 税务 | Mysql |
| 9 | Linux | 保险 | Mysql |
| 10 | Linux | 税务 | Mysql |

用的数据库管理系统剖面. 操作系统剖面下有“windows”、“Linux”、“Unix”三个术语,应用领域剖面下有“税务领域”、“保险领域”和“金融领域”三个术语,使用的数据库管理系统剖面

下“Oracle”、“SQL Server”和“mysql”三个术语. 构件库信息与用户检索记录如下:

表2 用户检索及反馈情况

| 检索编号 | 检索术语(Q) | 检索返回构件集 | | 用户选择的构件集(D) | |
|------|----------------|---------|-----------------------|-------------|---------------------------|
| | | 编号 | 描述术语 | 编号 | 描述术语 |
| 1 | Windows、 税务 | 1 | Windows、税务、SQL Server | 1 | Windows、税务、 SQL Server |
| | | 8 | Windows、税务、Mysql | | |
| 2 | Windows、 金融 | 2 | Windows、金融、Oracle | 7 | Windows、金融、 SQL Server |
| | | 5 | Windows、金融、Mysql | | |
| | | 7 | Windows、金融、SQL Server | | |
| 3 | Linux、税务 | 3 | Linux、税务、Oracle | 10 | Linux、税务、 Mysql |
| | | 10 | Linux、税务、Mysql | | |
| 4 | Linux、保险 | 9 | Linux、保险、Mysql | 9 | Linux、保险、Mysql |
| 5 | Windows、 金融 | 2 | Windows、金融、Oracle | 7 | Windows、金融、 SQL Server |
| | | 5 | Windows、金融、Mysql | | |
| | | 7 | Windows、金融、SQL Server | | |

若给定最小支持度阈值为 40%, 最小置信度阈值为 90%. 则从用户检索术语记录 Q 中可以看出出现次数超过最小支持度阈值 (40% × 5 次检索 = 2 次) 的频繁检索术语集为: {windows}、{Linux}、{税务}、{金融}、{windows, 金融}. 然后寻找出现这些频繁检索术语集的检索中, 对应的用户选择的构件集中出现次数超过频繁检索术语集与最小置信度阈值乘积的频繁术语集. 例如“windows”术语出现在编号为 1、2、5 的检索中, 对应的用户选择的构件集为 1、7、7 号构件, 其中出现次数超过 2.7 次 (3 次 × 90%) 的频繁术语集为 {windows}、{SQL Server}.

最后, 能够从用户检索记录中产生的强规则为 (除去由自身推出自身的平凡规则):

{windows} ⇒ {windows, SQL Server}

{Linux} ⇒ {Linux, Mysql}

{金融} ⇒ {金融, SQL Server}

{windows, 金融} ⇒ {windows, 金融, SQL Server}

应注意的是, 为加强可读性, 上述规则中右部保留了左部的项, 以强调检索条件的完整化和精确化. 用户在以后的查询中, 若给定的关键字中包含以上规则中左边的术语集, 则检索系统自动以规则右边的术语集代替来进行检索.

5 实验

我们实现了一个构件库检索的原型系统, 采用带有用户反馈的自适应构件检索过程模型, 并实现了基于关联挖掘的自适应学习算法. 实验设计为: 采用 4 个刻面对构件进行分类和描述, 在构件库中存储了 500 个构件的描述信息, 然后模拟用户对构件进行检索. 首先, 在没有使用自适应学习算法的情况下检索, 共检索 100 次, 每 10 次检索统计一次查准率, 得知查准率均在 0.17 左右. 第二次实验应用了自适

应的学习算法, 每 10 次检索进行一次关联挖掘. 实验证明, 随着挖掘次数增加, 发现的强规则数量增多, 查准率逐渐提高. 针对每 10 次检索的查准率情况如图 4 所示, 其中 R 是没有应用自适应学习算法的检索查准率曲线, FR 是应用了自适应学习算法的检索查准率曲线.

6 总结与展望

本文针对基于刻画描述的软件构件, 借鉴数据挖掘中关联挖掘的有关理论, 提出了带有用户反馈的自适应构件检索模型, 以及基于关联挖掘的自适应学习算法, 从用户检索的历史记录中, 挖掘隐含的条件, 从而找出从显式检索条件到用户对检索的隐含需求的强关联规则, 以增加检索中的有用信息, 提高构件检索的查准率.

本研究今后的目标在于对基于关联挖掘的自适应学习算法进行优化, 以提高算法效率, 并减少因偶发的关联规则实效而导致的无效检索; 以及将更多针对用户行为的研究引入到构件检索领域, 从而提高检索性能.

参考文献:

- [1] Hafedh Mili, Ali Mili, Sherif Yacoub, Edward Addy. Reuse-Based Software Engineering: Techniques, Organization, and Controls [M]. John Wiley & Sons, Inc, 2002.
- [2] Forbes Gbb, Colm McCartan, Ruairi O'Donnell, Niall Sweeney, Ruben Leon. The integration of information retrieval techniques within a software reuse environment[J]. Journal of Information Science, 2000, 26(4): 211 - 226.
- [3] Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques [M]. Morgan Kaufmann Publishers, 2000.
- [4] CHANG J C, Guo L F, Ma L. Representation and retrieval of reusable software components[J]. Computer Science, 1999, 26(5): 41 - 48.
- [5] CHANG J C, Li K Q, Guo L F, Mei H, Yang F Q. Representing and retrieving reusable software components in JB (Jadebird) system[J]. Electronica Journal, 2000, 28(8): 20 - 24. (in Chinese with English Abstract).

作者简介:



薛云皎 男, 1978 年生于云南建水, 博士生, 讲师, 主要研究领域为软件复用, 构件库系统, 面向方面的编程技术. E-mail: yjxue@fudan.edu.cn.



钱乐秋 1942 年出生, 男, 上海人, 教授, 博士生导师, 主要研究领域为软件工程, 软件复用, 软件过程.

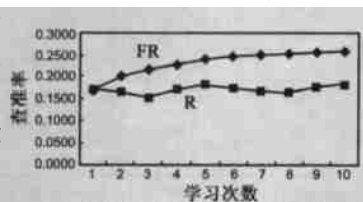


图4 使用自适应学习算法前后的查准率比较图

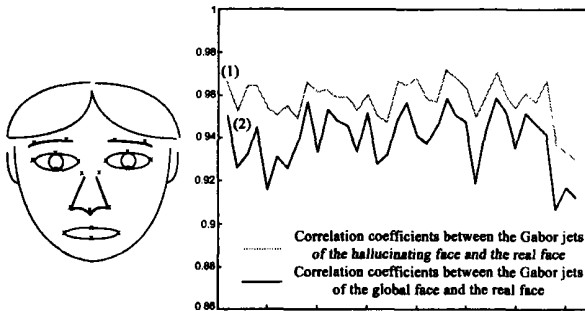


图 5 21 个人脸特征点 图 6 相关系数比较. 横坐标为测试图像编号, 纵坐标为相关系数. 曲线 (1) 描述了最终结果的 Gabor 特征与真实图像的 Gabor 特征之间的相关系数; 曲线 (2) 描述了整体图像的 Gabor 特征与真实图像的 Gabor 特征之间的相关系数

参考文献:

- [1] C Liu, H Y Shum, C-S Zhang. A two-step approach to hallucinating faces: Global parametric model and local nonparametric model [A]. In Proc of IEEE Conf on Computer Vision and Pattern Recognition [C]. Hawaii, USA, Dec 2001.
- [2] S Baker, T Kanade. Hallucinating faces [A]. In Proc of the 4th International Conf on Automatic Face and Gesture Recognition [C]. Grenoble, France, Mar. 2000.
- [3] R Schultz, R Stevenson. Extraction of high-resolution frames from video sequences [J]. IEEE Transactions on Image Processing, 1996, 5 (6): 996 - 1011.
- [4] R Hardie, K Barnard, E Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images [J]. IEEE Transactions on Image Processing, 1997, 6 (12): 1621 - 1633.
- [5] W Freeman, E Pasztor. Learning low-level vision [A]. In Proc of IEEE Conf on Computer Vision [C]. Kerkyra, Corfu, Greece, Sep 1999.
- [6] B K Gunturk, A U Batur, Y Altunbasak, M H Hayes, R M Mersereau. Eigenface-domain super-resolution for face recognition [J]. IEEE Transactions on Image Processing, 2003, 12 (5): 597 - 606.
- [7] L Wiskott, J M Fellous, N Kruger, C von der Malsburg. Face recognition by elastic bunch graph matching [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19 (7): 775 - 779.

作者简介:



李 焱 2002 年获清华大学计算机科学与技术系工学学士学位, 目前于清华大学计算机科学与技术系人机交互与媒体集成研究所攻读硕士学位, 研究方向包括计算机视觉、机器学习、模式识别和图像处理. E-mail: li_yang98@mails.tsinghua.edu.cn.

林学 清华大学计算机科学与技术系教授, 博士生导师, 1962 年毕业于清华大学自动控制系, 现研究方向为计算机视觉、图像处理与模式识别, 曾负责与参与多项国家 863 高科技项目, 国家自然科学基金项目, 在国内外期刊、国际学术会议发表论文 100 多篇. Email: lxy-dcs@mail.tsinghua.edu.cn.

(上接第 206 页)

- [6] WANG Y F, Xue Y J, Zhang Y, Zhu S Y, Qian L Q. A matching model for software component classification in faceted scheme [J]. Journal of Software, 2003, 14 (3): 401 - 408.
- [7] Hai-Feng Guo, Miao Liu, Jiexiong Pi. Precise specification matching for automated component retrieval and adaptation [A]. Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration (IRI-2003) [C]. Las Vegas, USA, Oct. 2003.
- [8] Jilani L L, Desharnais J, Frappier M, Mili R, Mili A. Retrieving software components that minimizing adaptation effort [A]. Automated Software Engineering, 1997 Proceedings, 12th IEEE International Conference [C]. Kluwer Academic Publisher, Dordrecht, Netherlands. 1 - 5 Nov. 1997.
- [9] Eduardo Ostertag, James Hendler, Ruben Prieto-Diaz, Christine Braun. Computing similarity in a reuse library system: An AF-based approach [J]. ACM Trans on Software Engineering and Methodology, 1992, 1 (3): 205 - 28.
- [10] M R Gardi, B Ibrahim. Automatic indexing of software artifacts [A]. Proc. 3rd Int'l Conf Software Reuse: Advances in Software Reusability [C]. IEEE Computer Society Press, Rio de Janeiro, Brazil, Nov. 1994. 24 - 32.
- [11] Dieter Merkl, A Min Tjoa, Certi Kappel. Learning the semantic similarity of reusable software components [A]. Proc. 3rd Int'l Conf. on Software Reuse: Advances in Software Reusability, IEEE Computer Society [C]. Rio de Janeiro, Brazil, 11/1994. 33 - 41.
- [12] Tomas Isakowitz, Robert J Kauffman. Supporting search for reusable software objects [J]. IEEE Trans on Software Engineering, 1996, 22 (6): 407 - 423.
- [13] Li Keqin, Guo Lifeng, Mei Hong, Yang Fuqing. An overview of JB (Jade Bird) component library system JBCL [A]. Proc TOOLS Asia '97 [C]. Beijing, China, Sep. 1997.
- [14] W B Frakes, T P Pble. Proteus: A reuse library system that supports multiple representation methods [J]. ACM SIGIR Forum, 1990. 24: 43 - 55.