Web页面自动化设计中布局挖掘和样式匹配算法

任胜兵,王志健,王 宇
REN Shengbing, WANG Zhijian, WANG Yu

中南大学 软件学院,长沙 410075
School of Software, Central South University, Changsha 410075, China

REN Shengbing, WANG Zhijian, WANG Yu. Layout mining and pattern matching algorithm on automatic Web page design. Computer Engineering and Applications, 2018, 54(3):227-232.

Abstract: There are much of similarity in page structures among Web applications when they have similar functions. Aiming at the current existence of high complexity and low developing efficiency in the process of Web pages' development, a method of mining existing Web pages' layout structures which share similar functions is proposed. The technique fully uses the features on Web pages' layout structures and applies the stage treatment. It firstly uses the page segment algorithms combined with similarity calculation to mine out the code blocks with high degree of similarities from layout structures, then parses style files and matches the stylesheets corresponding to the set of nodes quickly by the way of RoSunday and establishes the tree of document model. The combinations of each submodule can be used to realizethe automatic Web page design. According to the application example, the method can design and generate pages dynamically, it will improve the development efficiency effectively.

Key words: page design; segment algorithms; similarity; layout mining; pattern matching

摘 要: 具有相似功能的Web应用,其页面样式和布局往往存在很大的相似性。针对当前Web页面开发复杂度高且效率低的情况,提出一种挖掘现有页面布局结构和样式属性的方法来实现Web页面自动化设计。该方法充分利用Web网页布局结构上的特点,采用分级处理的方式,首先利用页面分块算法思想通过相似度计算挖掘出具有相似性的代码块,其次通过结合RoSunday方法解析样式文件快速匹配出节点集合对应的样式表并建立文档模型树结构,各个子模块之间的相互组合可以实现页面的自动化设计。通过应用实例表明,该方法能动态地设计并生成页面,有效提升Web页面开发效率。

关键词: 页面设计;分块算法;相似度;布局挖掘;样式匹配
文献标志码: A **中图分类号:** TP311.51 doi:10.3778/j.issn.1002-8331.1608-0405

1 引言

页面是Web应用的重要组成部分,用户通过页面与软件系统进行交互,完成需要的功能。页面设计的好坏直接影响用户体验[1]。

网站页面设计涉及到网络技术、视觉效果和网页经济学等多个领域。网页样式和布局的设计往往参照最新的趋势、网站用途以及产生的视觉效果各个指标。但设计者很难把握需求和准确定位到Web版面设计的最新趋势和风格。文献[2]提出一种自动挖掘同类功能网站页面结构的方案,帮助开发者高效设计出网页。文献[3]提出一种根据用户约束与布局要求自动设计出网页的技术。

相似功能的页面设计通常都有着相似的组成、布局、风格等特征,重复性的代码工作严重降低了工作效率,同时也降低了企业或者组织的快速迭代能力,页面结构的重用[4]可以有效提高Web应用的开发效率。

Web页面代码重用[5]可以由模板技术与组件技术实现。通过模板[6]生成代码可以降低Web开发过程中数据

基金项目: 国家自然科学基金面上项目(No.61272151);2016年中南大学硕士生自主探索创新项目(No.2016zzts385)。
作者简介: 任胜兵(1969—),男,硕士研究生导师,副教授,主要研究领域为软件工程、嵌入式系统、数字图像处理;王志健(1990—),女,硕士研究生,研究领域为软件工程,E-mail:506465921@qq.com;王宇(1992—),男,硕士研究生,研究领域为软件工程。
收稿日期: 2016-08-12 **修回日期:** 2016-10-09 **文章编号:** 1002-8331(2018)03-0227-06
CNKI **网络优先出版:** 2017-02-27, http://kns.cnki.net/kcms/detail/11.2127.TP.20170227.1058.020.html

与视图之间的耦合性。模板引擎可以让(网站)程序实现界面与数据分离,业务代码与逻辑代码的分离,解决了把动态变化的数据插入到页面文件中的问题[7]。文献[8]提出了一种基于Velocity模板引擎的代码自动生成技术研究,一定程度上解决了特定业务领域的信息管理系统的页面的设计与代码自动生成[9]。但目前模板方法不够灵活,改动模板会导致相关联的模块都随之改动,影响开发效率,因此既可以实现重用功能又比较灵活的基于组件[10]的Web重用到更多程序员的青睐。文献[11-12]设计并实现了一套页面组件库,面向可复用的组件编写页面,实现页面布局的自动设计与业务逻辑之间的无缝调用。该方法着重研究页面布局的统一性与动态数据的绑定,忽略了页面设计的多元化与用户体验。且组件开发具有局限性,需要特定的开发人员编写程序,其效率依然较低。

为了降低页面模板与代码之间的耦合性与组件开发的低效性,本文利用同类功能网站在页面布局与样式上存在的相似性,识别并提取具有相似结构的代码块并通过组合应用到待开发页面中,可以灵活实现页面的自动化设计。

通过总结同类网站中的页面结构特点发现:功能相似页面布局和样式布局也存在很大相似性。对相同性质页面的布局进行挖掘,并通过样式匹配算法可以得到具有相似结构的代码块,不同区域的代码块组合起来可以形成完整页面。由于HTML页面编写复杂,且呈半结构化[13-14],对整个页面的解析就会耗费大量的计算资源,对页面进行分块处理[15]可以很好地解决上述问题。在视觉分割线检测过程中融入相似度计算[16]将分割出的代码块中的可复用区域进一步细分,缩小了页面处理的范围,提高了结构抽取的效率。大部分计算网页相似度的方法都是通过树路径匹配模型来计算相似度,通过计算树之间的转化距离得出网页之间的相似性。文献[17]定义树路径的序列相似度(标签元素)和位置相似度,计算两相似度的加权和获取路径相似度,通过网页间最佳树路径匹配计算结构相似度,在传统基于树路径模型的基础上补充了路径相似度,更能体现网页的差异性。文献[18]使用数字向量比较节点之间的相似度,并设计投影算法来计算位置相似的节点,降低了比较的时间复杂度。文献[19]提出了基于节点统计特征的网页结构相似度度量方法,通过网页标记的频率分布特征得到结构和布局信息并计算网页之间的相似度。文献[20]提出一种将DOM树转换成其节点序列的方法,通过寻找在序列化节点中最大同构的子树序列来比较两个网页的相似性,该方法计算的时间复杂度较低。通过节点相似度算法[21]可以从代码块中提取出重复利用的结构[17]作为模板从而实现页面的自动化设计。

对代码块中的布局结构匹配相应的标签属性需要从页面引入的样式表中筛选出属性标识对应的样式,高效的字符串匹配算法RoSunday[22]可以更加快速地从大量字符中找到目标字符[23]。

将页面布局的设计分割成多个独立的代码块,可以大大提高开发效率并降低页面设计中各个模块之间的耦合性。本文采用基于布局挖掘和样式匹配的页面代码自动化设计方法,利用页面设计中的功能相同其页面结构也存在共性的特点,通过改进的VIPS(Vision-based Page Segmentation)算法挖掘同类网站中的布局结构,通过相似度计算抽取挖掘过程中具有相似结构的代码块,通过RoSunday字符串匹配算法匹配其映射的样式表文件,得到可以重用的代码块,从而实现页面的自动化设计。

2 算法描述

由于功能类别相同的网页在布局结构上存在共性,且通过解析页面发现其对应的DOM树节点也具有相似的样式属性。本文首先采用VIPS算法思想,利用视觉特征对页面进行分块,比较相同功能代码块之间的相似度,根据设定的阈值挖掘出满足要求的代码块并对其进行解析,提取出具有相似结构的代码块,并结合RoSunday字符串匹配算法从样式表中取出代码块中节点对应的样式属性。通过实例验证了页面自动化设计中布局挖掘和样式匹配算法的有效性。

2.1 布局挖掘

2.1.1 页面分块

对于页面分块首先是对块进行抽取。通过视觉特征将Web页面按照规则首先被分割为几个比较大的代码块,同时这几个代码块所组成的层次结构将被记录下来。对于检测出来的每一个大的代码块分块过程又可以继续进行,直到代码块不能进一步分割(达到不可再分的子节点或者已经满足给定的阈值范围)。将分块后的网页进行比较,具有相似布局的划分为同一组,对每一个大的代码块(按照视觉分割线检测)进行细分,将每一次分完后的结果比较相似度,对于满足阈值的每一个节点块划分为同一子集,根据标签中的属性标识符来获取节点信息。

VIPS在网页块提取时对DOM树中的每一个节点进行检查,网页块提取的规则多达13条,导致网页块提取过程非常复杂,且需要计算和保存DOM树中的所有节点的视觉信息,这就导致该算法在时间和内存上消耗比较大,使得在处理含有大量节点的网页时性能不高。本文是主要是采用节点之间的层级关系产生分隔条,分块规则充分利用CSS样式中的border属性、节点的background-color属性,margin和padding属性以及长条图片等视觉特征,根据分割出代码块的数量来控制分级的层数,并通过相似度计算可以区分有效代码块和无效

代码块，提高对相似结构提取的效率。

2.1.2　相似度计算

网站的设计者为了保持其网站外观的统一，往往在同一网页内部使用很多的重复的代码（包括节点特征序列及所应用的样式）。相似性表现在以下几方面：

（1）通常具有相同的父节点。

（2）节点内部的HTML标签排列相同。

（3）节点内部各HTML标签的样式相同。

通过分块处理，将网页分割为有效区域与无效区域，有效区域为那些具有相似结构的代码块，无效区域可以去掉，以便提高对页面进行处理的速度。相似度计算主要分三步进行，第一步遍历有效区域的节点，计算节点之间的相似度，第二步，根据计算的相似度值判断两两节点是否满足阈值要求，若满足，根据当前节点中的元素标识查找并按照一定规则存储它的样式表，第三步将同一代码块中相似度值满足阈值的节点进行合并，直到遍历结束。算法具体实现过程如下所示。

算法1　查找并提取代码块中具有相似结构的节点

```
输入：在布局上存在相似性的代码块
输出：具有相似结构的节点集合
Algorithm FindSimilarChildren(Node node1,node2){
    //获得两个Node节点中包含的子节点个数为maxLength和minLength；
    //设定常数K为满足要求的相似节点个数；
1   for(int i=0;i<maxLength;i++){
2     for(int j=0;j<minLength;j++) {
3       if(比较两个节点的结构相似度){
4         相似就添加到nodeSet中
5       } else {
6         if(比较两个节点的位置相似度)
7           相似就添加到nodeSet中，若不相似就删掉该节点
8       }}
9     if(未在Node中找到满足要求的相似子节点)
10      for(每一个childNode)
11        FindSimilarChildren(childNode,node);
12  }
```

算法1只是用来提取代码块之间的相似结构，通过视觉特征得到具有相似特性的代码块，对代码块细分才能得到需要的结构信息。

遍历代码块，将其看做是页面DOM树结构的一个节点。步骤3~5：根据其属性特征计算节点之间的相似度，主要是依据标签排列和标签属性，将得到的结果与事先给定的阈值进行比较，大于阈值说明具有较高的相似度，可以抽取结构。步骤6~8：若相似度小于阈值，则对其位置相似度进行计算，判断其位置上是否与其他节点具有共性，由于每一个网站需要比对的节点数目很多，所以需要删除位置相似性也不满足条件的节点，不对其子节点再做处理。

对于遍历过程中的节点相似性的计算与比较的详细过程下面给予阐述。

（1）当前节点中的结构具有相似性：遍历当前节点，根据相似度计算得出它们具有相同的标签，相似的属性值，所以计算结果大于阈值，继而对代码块中的子节点进行递归遍历。

（2）当前节点中的结构不具有相似性：当前节点在布局相似度计算的结果小于阈值，则对节点的位置进行相似度比较，位置信息主要为节点所占矩形区域的高度、宽度以及长宽之间的比例关系，通过计算将得到的值与阈值比较，若小于阈值，则说明该节点与其他节点不具有相似性，为了避免不必要的计算，则删掉该节点。

因此，需要定义节点的结构相似度SoL（Similarity of Layout）来度量两个节点在结构上的相似程度与位置相似度SoP（Similarity of Position）来度量两个节点在样式上的相似程度。SoL与SoP都是0到1之间的值，越接近1表示两个节点的布局结构越相似。设有两个节点 x,y，则它们的SoL定义为公式（1）所示：

$$SoL(x,y)=\sum_{i=1}^{N}\omega_i\sum_{j=1}^{M_i}\frac{1}{M_i}S_{ij} \tag{1}$$

公式（1）中 N 表示比较的深度，即只比较到第 N 层节点，在本文的计算中，N 的取值很大程度上取决于节点所占区域大小；M_i 表示第 i 层子节点的个数；ω_i 为第 i 层子节点对整体结构布局的贡献系数，一般认为越深层次的节点对宏观布局的影响越小。首先判断两个节点是否为使用了同样的HTML标签，若不同，则 S_{ij} 为0，若相同，则继续比较下一节点。通过实验，若SoL值大于0.9[18]则可认为两个节点相似，此时可以达到较好的识别效果。如果两个节点的SoL值非常小，可通过比较节点的位置关系来确定它们之间的相似度。

通过视觉观察，比较直观的因素为节点所占矩形框大小、色块信息等，所以本文将位置信息的计算单独抽取出来判断两个节点是否具有相似性的依据。有两个匹配节点 x_i、y_i、i 为待比较的第 i 个节点，已有节点为 x_0、y_0，则它们的SoP定义为公式（2）所示：

$$SoP(x,y)=\frac{x_i}{x_0}\cdot\frac{y_i}{y_0}\cdot\frac{x_i/y_i}{x_0/y_0},i\in Z^+ \tag{2}$$

其中 x_i、y_i 为 i 个匹配节点所属区域的长度与高度，i 为正整数，x_0、y_0 为已有节点的长度与高度。通过SoP公式计算出待比较节点与已有节点区域之间各项比值，得出的结果为两个节点之间的位置相似度。通过多次实验可以得出SoP的阈值。若求出的比值大于阈值，则表明两个节点相似，可以对节点进行进一步的解析与布局挖掘，若值小于阈值，则证明该节点所属区域与其他

节点之间不具有共性,应当去除以页面结构的抽取质量与效率。对于有些节点的位置信息难以获取的情况,根据其父节点、兄弟节点、子节点以及相对位置排版,推算出其位置信息。

2.2 样式匹配

根据相似度计算可以得到具有相似布局的节点,对于这些节点,需要根据其需要从页面中抽取其样式特征,并将节点中的标签标识符与在代码块中的层级关系映射到相应的存储结构中,本文采用哈希表来存储标签与样式表之间的映射关系,key为每一个标签节点的唯一标识,value值代表标签层级。且每一个节点集合中的样式表都会被存储为一个新的css文件,方便自动构造页面时样式表的引用。具体算法如下所示。

算法2 对代码块中的节点进行样式匹配

```
输入:具有相似结构的代码块
输出:节点内部结构信息以及样式表文件
Algorithm HtmlToTree(Node node){
    通过分块算法得代码块
    遍历列表中的每一个节点
    使用正则表达式切割字符串
1   for(int i=1;i<tags.length;i++){
2     去掉tags[i]中内容,得到标签tag与样式style
3     if(tag为自封闭标签)
4       下一标签为当前标签的兄弟节点
5       if(tags[i].contains("End")) {
6         分割tag[i]字符串,得到endTags数组
7         for(int j=1;j<endTags.length;j++) {
8           使用正则表达式分割endTags[j]
9           记录当前位置以及结束标签endTag
10        if(!endTag.contains(tag))
11          Compare(tag,endTag)
12          RoSunday算法;
13          将将当前标签的style元素与节点层级
14          放入哈希表中
15} } } }
```

通过分块算法得到HTML页面文件,首先需要对页面文件进行标准化处理,目的是保证页面中的标签文件具有准确的开闭对应关系,减少标签的书写不规范导致的分层错误问题。解析文件后根据页面分块算法得到具有可复用结构的代码块,解读代码块中的节点信息,通过遍历节点得到它们之间的层级关系以及样式表信息。利用解析工具可以得到代码块,利用正则表达式可以其分为层级清晰的节点集合。

步骤1、2:利用for循环遍历代码块中的节点,根据正则表达式中的规则匹配过滤出标签与样式表信息。步骤3~9判断当前标签是否为自闭和标签,若为该类标签,则证明当前节点不存在子节点,下一个遍历节点为其兄弟节点,否则判断该节点中是否包含子节点以及闭合标签标记。步骤11通过compare方法判断最近的该节点匹配的“End”标签。闭合标签的提取也需要匹配一定的规则,提取出闭合标签后,便可以梳理出该节点的层级关系,以及利用标签属性提取该节点的样式表信息。标签中的id、class以及style属性与样式表存在一一对应的关系,且容易获取,可以将其作为节点中的样式标识符。

使用样式信息的方法和位置非常灵活,主要有以下三种情况:(1)使用标签引用文件,标签只能位于签中,数量可以是多个;(2)使用标签定义样式信息,标签的位置不确定,数量可以是多个;(3)通过标签的属性指定样式信息,但并不是每个标签都有属性。利用RoSunday算法,可以快速地通过标识符与样式表的映射关系可以抽取每一个节点的完整信息,并将其作为条有效记录存入代码块的CSS文件中。RoSunday算法是Sunday字符串匹配算法的一种改进,算法在匹配过程中,首先判断模式串的最后一个字符位置在主串 T 中对应的位置上的字符有没有在模式串中出现过。如果出现,按照Sunday算法进行匹配,如果没有出现,则直接跳过整个模式串长度,并从该字符的下一位字符开始进行下一轮的匹配首先按从左向右进行逐个匹配。为了方便页面的自动化设计,将每一个标签的样式标识符与其在代码块中的层级关系通过哈希表关联起来,key为样式标识符,value中存储该节点的层级编号。

3 实验验证

根据调查发现,功能相似或相同的网站在布局结构上大都存在一定的相似性,可以将Web网站分为以下三种形式:门户网站、企业网站和个人网站。开发一个大型网站需要耗费大量的人力物力,特别是在页面的结构设计上,需要精确每一个节点元素的布局位置以及呈现效果,如果能够对已经存在的结构进行提取,并应用到相似功能的网站开发中,会大大减少开发成本,提高效率,且多次复用的代码块可以作为模板或者组件,方便网站的维护与修改。

对于分块后得到的代码块需要进行相似度计算,方法中主要用到的参数有布局相似度阈值,计算布局相似度时需要比较的层数以及每一层对整体布局的贡献系数。其中比较层数 N 选取比较重要,因为其取值会直接影响到计算结果。层数 N 的取值需要平衡匹配耗费时间以及抽取的准确度。层数偏少,比较得过于粗略导致抽取结果的准确度不高;层数偏多,比较得过于细致耗费过多时间。因此,需要综合考虑使得抽取的准确率和召回率都比较高,同时运算处理速度又快。贡献系数的选取要依据以下原则:外层的贡献系数要大于内层,即要满足公式(3)的条件:

$$\sum_{i=0}^{N-1}\omega_i=1,\ \omega_i>\omega_j,\ 0\leqslant i<j\leqslant N \tag{3}$$

在实验中，选择 N =1，2，3，4，5，6来分别计算处理网页之间相似代码块的抽取的召回率和准确率，而贡献系数的选取则根据式(3)，可根据实际情况作具体调整，这里只给出 $N=3$ 时的一组参考值(0.5，0.2，0.1)，此时布局相似度SoL为0.9时效果较好。具体实验结果图1所示。由图1可知，N 的取值会严重影响准确率和召回率。综合考虑，N 的取值为3时抽取精度达到最优。

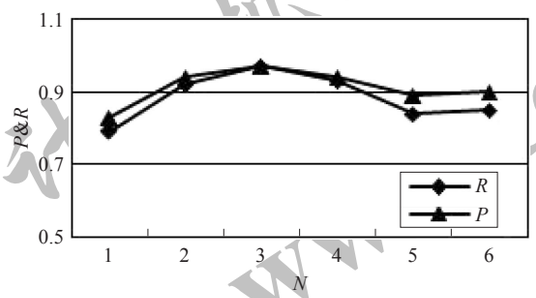

图1　N取不同值时的准确率和召回率

计算样式相似度时，如果布局相似度的值小于阈值需要计算位置相似度，因为它决定匹配节点在页面中的展示效果。如果两个比较的节点之间不具备相同的属性值，可以根据位置相似性来判断是否需要进一步解析节点进行比较。主要需要考虑以下因素：位置相似度相差太大，则说明两个节点所占的矩形区域大小存在较大差异，则代码块的子节点可以提取出相似结构的概率比较小；如果位置相似度差距不大，只要是可以再分的两个节点，都有可能存在相同结构的代码块，则需要进一步计算来判断是否可以挖掘出具有相似性的页面结构。因此需要综合考虑使得匹配的节点在页面中显示不突兀。本文对来自新浪、腾讯、搜狐、网易、凤凰网等门户网站的1 000个代码块进行了测试。通过位置相似性划分的区域进行节点相似度计算，结果可以分为：满意(节点匹配的正确率达95%以上)；有错误但错误可接受(节点匹配的正确率为85%~95%)；有错误且错误不可接受(节点匹配的正确率低于85%)。根据实验数据表明，由于需要考虑计算节点位置所耗费时间与节点抽取的满意度，SoL的值在0.9以上效果最佳。实验数据如表1所示。

表1　SoP阈值选取

SoP值	满意	可接受	不可接受
0.95	0.953	0.032	0.015
0.90	0.921	0.057	0.022
0.85	0.768	0.196	0.360
<0.85	0.210	0.546	0.244

对具有相似结构的代码块进行解析，需要将标签属性标识符对应的样式从样式表中提取出来，本来采用RoSunday算法匹配样式表中的关键字，与其他字符串匹配算法比较具有较好的性能。从800 KB以上的页面文件中匹配实验数据如图2所示。

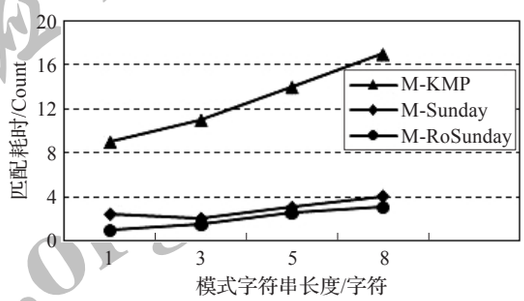

图2　样式匹配算法运行时间

上文中的M-KMP，M-Sunday，M-RoSunday为分别结合了KMP、Sunday、RoSunday字符串匹配算法的样式匹配算法。字符串匹配算法的效率与匹配次数、匹配的串长度等密切相关，根据样式匹配算法中对样式表文件的处理表明，当子串长度小于8个字符时采用M-RoSunday样式匹配算法可以取得较高效率。

为了评价页面中提取代码块的结果，每一种形式的网站随机选择某一主题的网页来运行本文提出的Web页面的布局挖掘和样式匹配算法，通过前面的工作，得到具有相似结构的代码块以及在网页中所占的比重。其中门户网站与企业网站中比较高，可以达到37%以上。

4　结束语

在Web应用系统的开发过程中，页面代码的编写因其数量大、重复多、出错率高等特点已经成为影响整个项目进度的瓶颈。现有页面辅助开发工具只注重对单一页面的图形化操作和生成，开发效率仍然很低，需要重复设计具有相似结构的页面。提出一种方法在Web页面自动化构建过程中挖掘现有页面功能相似的布局结构。通过上述处理过程得到相似代码块，各代码块之间相互组合可以实现页面的自动化设计，且各个代码块独立存在，方便页面的维护与修改。页面相似结构的挖掘与样式匹配算法不仅可以实现页面的自动化设计，其更重要的意义在于如何充分利用已经成熟的页面结构与布局实现代码的重用，进而更进一步地提高Web应用的自动化程度。但本文仅针对页面结构与布局的挖掘做了相关工作，对挖掘后代码块之间如何相互组合自动化设计页面并未做过多研究，约束编程可以很好解决满足约束的条件之间的组合问题，使用约束方程自动化设计页面成为下一阶段工作的重心。

参考文献：

[1] Zhu S，Liu W，Cai J，et al.The research and design of the Web page information system editor[C]//2015 International Conference on Smart and Sustainable City and Big Data，2015：51-55.

[2] Bajwa I S, Siddique I, Choudhary M A.Web Layout Mining(WLM):A new paradigm for intelligent web layout design[C]//International Conference on Information & Communications Technology,2006:1-2.
[3] Guilherme D, Horta N, Guilherme J.Automatic layout generation of power MOSFET transistors in bulk CMOS[C]// IEEE International Conference on Electronics, Circuits and Systems,2014:606-609.
[4] Budhija N,Ahuja S P.Review of software reusability[C]// Proceedings of the 1st International Conference on Computer Science and Information Technology,2011:113-115.
[5] 王博,林中.可重用构件界面框架的研究与实现[J].计算机工程与设计,2011,32(6):2035-2039.
[6] Campos-Rebelo R, Pereira F, Moutinho F, et al.From IOPT Petri nets to C:An automatic code generator tool[C]// 2011 9th IEEE International Conference on Industrial Informatics,2011:390-395.
[7] Radjenovic J, Milosavljevic B, Surla D.Modelling and implementation of catalogue cards using FreeMarker[J]. Program Electronic Library & Information Systems,2009, 43(1):62-76.
[8] 孔得雨,罗锋,林伟波,等.一种基于 Velocity 的代码自动生成技术研究[J].计算机应用与软件,2014,31(10):20-23.
[9] Han G,Liu H,Zhang Z,et al.Analysis and design of automatic code generation system based on J2EE[C]// 2011 Third International Conference on Communications and Mobile Computing,2011:77-80.
[10] 覃发兵,葛玉辉.基于Java Web组件技术的毕业设计管理系统[J].计算机应用,2010,30(S1):321-323.
[11] 樊国柱.基于页面组件的应用系统快速开发平台:CN, CN101178649[P].2008.
[12] 任保钢.久其DNA界面组件库的设计与实现[D].北京:北京工业大学,2015.
[13] 张乃洲,曹薇,李石君.一种基于节点密度分割和标签传播的Web页面挖掘方法[J].计算机学报,2015,38(2):349-364.
[14] 李文昊,彭红超,童名文,等.基于视觉特征的网页最优分割算法[J].计算机科学,2015,42(11):284-287.
[15] Zhang X, Zhang Y, He J, et al.Vision-based web page block segmentation and informative block detection[C]// International Joint Conferences on Web Intelligence, 2013:265-269.
[16] Hattori G,Hoashi K,Matsumoto K,et al.Robust Web page segmentation for mobile terminal using content-distances and page layout information[C]//International Conference on World Wide Web,Banff,Alberta,Canada, May 2007:361-370.
[17] 廖浩伟,杨燕,贾真,等.一种改进的基于树路径匹配的网页结构相似度算法[J].吉林大学学报:理学版,2012,50(6):1199-1203.
[18] 江鸿.基于视觉的相似性算法在信息抽取中的研究与实现[D].长春:吉林大学,2011.
[19] Cruz I F,Borisov S,Marks M A,et al.Measuring structural similarity among web documents:Preliminary results[C]//Proceedings of the 7th International Conference on Electronic Publishing,Held Jointly with the 4th International Conference on Raster Imaging and Digital Typography:Electronic Publishing,Artistic Imaging,and Digital Typography.[S.l.]:Springer-Verlag,1998:513-524.
[20] Hu Z,Sun F.Measuring similarity of web pages on maximum isomorphic subtree[C]//International Conference on Fuzzy Systems and Knowledge Discovery,Yantai, Shandong,China,2010:2469-2473.
[21] 王允,李弼程,林琛.基于网页布局相似度的Web论坛数据抽取[J].中文信息学报,2010,24(2):68-76.
[22] Sunday D M.A very fast substring search algorithm[J]. Communications of the ACM,1990,33(8):132-142.
[23] 徐珊,袁小坊,王东,等.Sunday字符串匹配算法的效率改进[J].计算机工程与应用,2011,47(29):96-98.

(上接205页)

[6] Yan Y,Sukthankar R.PCA-SIFT:A more distinctive representation for local image descriptors[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition,Washington,USA,2004:506-513.
[7] Mikolajczyk K,Schmid C.A performance evaluation of local descriptors[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2005,27(10):1615-1630.
[8] 戴金波,赵宏伟,刘君玲,等.一种针对于描述子的SIFT简化方法[J].仪器仪表学报,2012,33(10):2255-2262.
[9] 刘佳,傅卫平,王雯,等.基于改进SIFT算法的图像匹配[J].仪器仪表学报,2013,34(5):1107-1112.
[10] 曾峦,顾大龙.一种基于扇形区域分割的SIFT特征描述符[J].自动化学报,2012,38(9):1513-1519.
[11] 程德志,李言俊,余瑞星.基于改进SIFT算法的图像匹配方法[J].计算机仿真,2010,28(7):285-289.
[12] 赵烨,蒋建国,洪日昌.基于RANSAC的SIFT匹配优化[J].光电工程,2014,41(8):58-65.
[13] Lindeberg T.Scale-space theory:A basic too for analyzing structures at different scales[J].Journal of Applied Statistics,1994,21:224-270.
[14] 赵录刚,吴成柯.基于随机抽样一致性的多平面区域检测算法[J].计算机应用,2008,28(12):154-157.
[15] Hartley R,Zisserman A.Multiple view geometry in computer vision[M].Cambridge UK:Cambridge University Press,2003:121-126.