行业动态信息采集系统关键问题解决方案

黎　柯，蔡永香，干佳林，王居远，杨　鼎，胡森勇
（长江大学 地球科学学院，武汉　430100）

摘　要：为了解决行业动态信息采集系统中网页定向爬取、网页清洗、信息检索等关键问题，文章提出一套基于 Heritrix、Jsoup 和 Lucene 的解决方案，并以测绘地理动态信息系统为例进行验证，结果证明该方法能够较好完成测绘地理信息的定向爬取，实现对不同风格网站网页的清洗，并建立索引提供信息检索机制，给测绘行业人士提供准确可靠的信息服务，为相关研究提供参考。

关键词：Heritrix 和 Lucene；信息爬取；网页清洗；全文检索
【中图分类号】P208　　**【文献标识码】**A　　**【文章编号】**1009-2307(2016)03-0161-06
DOI：10.16251/j.cnki.1009-2307.2016.03.032

Key problems and their solutions for industrial dynamic information collection system

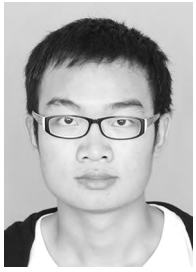
Abstract: In order to solve the key problems on Web page directional capturing, Web filtering and information retrieval for industrial dynamic information collection system, the paper proposed a set of solutions based on Heritrix, Jsoup and Lucene, which was verified through taking the implementation of Surveying and Mapping Dynamic Information System as the example. Result showed that the solutions would help realize the directional capturing, the Web information filtering and information retrieval of Geomatics efficiently, which could provide accurate and reliable information service for related personnel.

Key words: Heritrix and Lucene; information capturing; Web filtering; full-text retrieval

LI Ke, *CAI Yongxiang*, *GAN Jialin*, *WANG Juyuan*, *YANG Ding*, *HU Senyong* (School of Geosciences, Yangtze University, Wuhan 430100, China)

0　引言

专业动态信息的浏览是掌握行业最新动态的必要手段。但行业网站的纷繁复杂和网上信息的良莠不齐使得人们浏览信息的效率大打折扣，建立一个行业动态信息采集系统，抓取准确可靠的行业信息给业内人士提供信息服务是诸多行业的所面临的共同需求。这个系统的实现必须解决以下 3 个方面的问题：1)Internet 上网站众多，网上信息良莠不齐，必须对网站的来源进行控制，确保信息的质量，实现网页信息的定向爬取；2)各网站布局风格不一、种类繁杂，无统一模式和规则，网页上除了包含网页信息，还包含有广告、各种链接等，还有一些非兴趣网页，必须对这些信息进行筛选、清洗来获取兴趣网页正文内容；3)清洗后的网页需要在存储时建立索引机制，才能给业内人士提供行业信息检索服务。本文从主要解决这些问题的角度出发，提出了一套基于 Heritrix、Jsoup 和 Lucene 的解决方案，并以测绘地理信息行业为例，对这套方案进行了实例实现。



作者简介：黎柯(1994—)，男，湖北荆州人，本科生，研究方向：Web-GIS。
E-mail：447996694@qq.com

收稿日期：2015-06-30
基金项目：国土资源部项目(2014Z1317)

通信作者：蔡永香 副教授 E-mail：caiyx2002cn@126.com

1　行业动态信息采集系统实现的技术流程

针对某一行业用户来说，相关的专业网站有很多，提供的信息有交叉也有不同的部分，如果用户为了了解行业动态而挨个到各个网站上去查询浏览，会浪费很多时间和精力。行业动态信息采集系统的建立能够针对某一特定行业为用户提供便捷的信息服务、提高他们的工作效率。建立

一个行业动态信息采集系统的主要技术流程如图 1 所示。

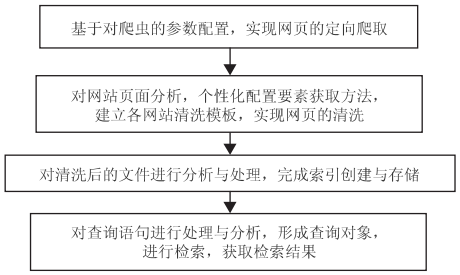


图 1 实现行业动态信息采集系统的主要技术流程

Fig. 1 Workflow of Realization of Industrial Dynamic Information Collection System

2 关键问题及解决方法

2.1 Heritrix 与网页定向爬取

1) Heritrix 原理

Heritrix 是一个由 Java 开发的、开源的网络爬虫，它提供了一个数据爬虫框架，预留了解析、下载、存储等配置接口，用户可以用它抓取网上的资源[1]。它抓取网页的过程如下(见图 2[2])：

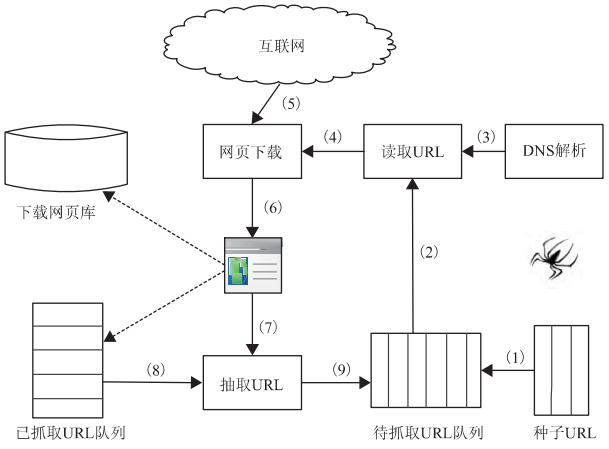


图 2 Heritrix 抓取网页的过程

Fig. 2 The Process of Capturing Web Pages by Heritrix

(1)首先选取一部分精心挑选的种子 URL；(2)将这些 URL 放入待抓取 URL 队列；(3)从待抓取 URL 队列中取出待抓取的 URL，解析 DNS，得到主机的 IP 地址，并将 URL 对应的网页下载下来，存储进已下载网页库中，然后将这些 URL 放进已抓取 URL 队列；(4)分析已抓取 URL 队列中的 URL，分析页面上链接的 URL，将这些链接的 URL 放入待抓取 URL 队列，然后进入下一个循环。

2)网页定向爬取参数控制

开发者可以通过配置 Heritrix 的各项参数以及扩展它的组件来实现自己的抓取逻辑和任务[3]。在网页定向爬取中，我们主要关注 order. xml 中的以下几个参数：

1)时间控制参数：聚焦爬虫需要定向爬取多个网站信息，不同网站所包含的信息量不同。需要根据实际情况，针对不同的网站设定不同的爬取时间。在 order. xml 中，＜long name＝" max-time-sec" ＞0＜/long＞标签值代表抓取网页的最大时间，0 为默认值，表示抓取时间不限。该参数可以根据经验设置，适当调高对信息量多的大型网站的一次抓取时间，调低对相对信息量少的小型网站的一次抓取时间。

2)爬取线程数：为了更加快速、有效地抓取网页的内容，可以采取多线程的爬取方式[4]。order. xml 中的＜integer name＝" max-toe-threads" ＞50＜/integer＞的标签值用来设定网页同时抓取的线程数。默认情况下，Heritrix 开启单线程方式抓取网页，最大值为 50，表示可同时开启最多 50 个线程。多线程抓取能够有效提高网页爬取效率。

3)网页抓取深度：我们放入的种子 URL 一般是各行业网站的首页链接。order. xml 中＜integer name＝" max-hops" ＞20＜/integer＞标签表示网页的爬取深度，Heritrix 默认的爬取深度是 20。如果爬取深度过深可能会导致爬取信息的不聚焦，例如爬取到与种子 URL 关系不紧密的页面。为了提高爬取的准确度，可以适当降低爬取深度。

4)正则表达式指定抓取文件类型：网站上的信息包罗万象，即使是我们感兴趣的行业网站，也有大量与我们兴趣无关的网页。Heritrix 是一视同仁地对每个网页进行抓取，哪怕是链接的广告信息。要想提高行业信息定向爬取的有效性，必须对网站上的网页进行甄别筛选后进行爬取。一般的情况下，只需要抓取有正文内容的文件类型，比如带有 . html、. shtml 和 . htm 等后缀的文件，而 pdf、jepg、css、asp 等格式的文件往往与我们的兴趣无关。我们采用正则表达式来指定抓取文件类型，在 order. xml 中配置标签：＜string name ＝" allow-by-regexp" ＞(. ＊(/ | \ . (html | htm | shtml)) $)＜/string＞就可以只爬取带有 . html、. shtml 和 . htm 后缀的文件。

5)增量爬取设置：各网站每天都在动态更新，网页抓取也需要及时更新，以保证抓取信息的实效性[5]。如果每次爬取网站信息时都要将该网站的所有网页都重新爬取，就会严重影响到抓取效率。增量爬取指每次只抓取新增网页或者更新过的网

页，能大大提高抓取效率。Heritrix 的 CrawlController 在每次网站爬取后会生成一个名为 recover. gz 的日志文件，这个日志文件会记录本次爬取的所有网站链接。将 order. xml 中＜string name ="recover-path"＞＜/string＞标签中填入上一次爬取该网站的带路径的日志文件 recover. gz，就能跳过上一次网站已经爬取过的网页，从而实现对该网站的增量爬取。

2.2　网页清洗

行业信息动态采集系统利用 Heritrix 将各网站的网页信息爬到本地后，必须经过清洗提取出网页的标题、时间、来源和正文，用于后续的索引建立。这些内容都存在于网页的 HTML(超文本标记语言)中。Jsoup 是一款 Java 的 HTML 解析器，它提供了一套完整的 API，可以直接解析 HTML 文本内容[6]。网页清洗时主要利用 Jsoup 的标签属性查找(doc. getElementsByClass(ClassName))、标签名查找(doc. getElementsByTag(TagName))和特殊变量值查找(doc. getElementsByAttributeValue(Var，Value))3 个方法。

但由于各网站布局风格不一，网页中的每个属性项所在的位置各不相同，需要人工分析各网站的 HTML 格式，提炼出标题、时间、来源和正文的获取方法，从而建立灵活的网站清洗模板。

为了能够建立灵活的网站清洗模板，我们逐个分析了 20 多个网站的网页。网页标题、网页建立时间、来源和正文等属性项是建立网页索引的必备要素。按照这些要素在网页中的位置状况，可以得到相应的要素提取方法：

1)标题要素：网页标题一般会在头文件的＜title/＞标签中。Jsoup 有专门获取文章标题的方法(doc. title(). toString())，但标题中往往会夹杂有网页来源或者位置等其他信息，我们需要将这些其他信息去除。网页标题一般有标题单独列出、标题前带有其他信息、标题后带有其他信息等 3 种情形，相应的提取方法见表 1。

表 1　网页标题要素提取方法

Tab. 1　Extraction Methods of Web Page Titles

标题方法类型	Title_1	Title_2	Title_3
情况说明	标题在 Title 标签中单独列出，如＜title＞中国首次对外正式发布中国海区国际标准电子海图＜/title＞	标题在 Title 标签中，标题前带有其他信息，如＜title＞中国石油新闻中心——国家环保部到中国石油检查调查＜/title＞	标题在 Title 标签中，标题后带有其他信息，如＜title＞陕西局地理国情普查工作纪实-中国测绘新闻网＜/title＞中，“中国测绘新闻网”是与标题无关的其他信息。
获取方法	Title 标签中直接获取信息	取 Title 标签中分隔符后面的信息	取 Title 标签中分隔符前面的信息

2)时间要素：HTML 中的时间往往会有固定的格式，如：yyyy-mm-dd、yyyy/mm/dd 和 yyyy 年 mm 月 dd 日。时间的提取需要找到时间在 HTML 中的位置，再根据这些特定的格式，利用正则表达式将时间提取出来。时间要素所在的位置情形及相应的要素提取方法见表 2。

表 2　网页时间要素提取方法

Tab. 2　Extraction Methods of Web Page Time

时间方法类型	Time_1	Time_2	Time_3
情况说明	时间包含在某个 class 属性值的标签中，如 ＜div class="left"＞ ＜span＞2015-09-15 10：45：58 ＜/span＞ ＜/div＞ 时间就在 div 标签中，其 class 属性值为“left”。	时间包含在某个标签的变量中，如 ＜script type="text/javascript"＞ var source = '黑龙江测绘地理信息局'; var akeyworks = '黑龙江'; var tm = '2015-09-08 16：12：45'; ＜/script＞ 时间在 script 标签的“tm”变量中。	时间包含在某个标签的特定属性值中，如 ＜meta name='Maketime' content='2015-03-27 14：06：58'＞ ＜meta name='subsite' content='中国海洋石油公司'＞ 时间在 name 变量为“Maketime”的 meta 标签中。
获取方法	输入标签名及其 class 值，用正则表达式去匹配获取	输入标签名和变量名，用正则表达式去匹配获取	输入标签名、变量名和变量值，用正则表达式匹配获取

3)来源要素：网页来源在 HTML 中的位置也可分为 3 种情形，相应的要素提取方法见表 3。

4)正文要素：正文是建立索引的主要数据源，我们用不带标签格式的正文文本来建立索引。但

表 3 网页来源要素提取方法

Tab. 3 Extraction Methods of Web Page Sources

来源方法类型	Source_1	Source_2	Source_3
情况说明	来源在某个 class 属性值的标签中，如 发布日期：2015-09-14 来源：新华社 来源在 div 标签中，其 class 属性值为“info”。	来源包含在某个标签的变量中，如 <script type=" text/javascript" > var source = '黑龙江测绘地理信息局'; var akeyworks = '黑龙江'; var tm = '2015-09-0816：12：45'; </script> 来源在 script 标签的变量“source”中。	来源包含在某个标签的特定属性中，如 <meta name='Maketime' content='2015-03-27 14：06：58'> <meta name='subsite' content='中国海洋石油公司'> <meta name='channel' content='公司要闻'> 来源在 name 变量为“subsite”的 meta 标签中。
获取方法	输入标签名及其 class 属性值，以及匹配字符获取，如“来源：”	输入标签名和变量名获取，如“script”和“source”	输入标签名、变量名和变量值获取，如“meta”“name”和“subsite”

由于不带格式标签的正文在页面上显示时不能自动换行以及表达一些特殊格式，如空格、加粗等，所以我们同时记录带标签的正文，用以在页面上显示。正文在 HTML 中的位置及相应的要素提取方法见表 4。

表 4 网页正文要素提取方法

Tab. 4 Extraction Methods of Web Page Contexts

正文方法类型	Text_1	Text_2
情况说明	正文包含在具有特殊的 class 属性值的标签中，如 <td class =" content_text" > 第一段 第二段 第三段 …… </td> 正文在 td 标签中，其 class 属性值为“content_text”。	正文包含在某个特定名称标签中，如 如 <con> <！——function content ( ) parse begin——> 第一段 第二段 第三段 …… <！——function content ( ) parse end Oms cost！——> </con> 正文在 con 标签中。
获取方法	输入标签名及其 class 值	输入标签名获取

新增加一个网站需要清洗时，人工对网站的风格进行分析，针对网页标题、时间、来源和正文在 HTML 中的编排格式，选取表 1 到表 4 中不同要素的获取方法，形成该网站的清洗模板。

2.3 建立索引与信息检索

Lucene 是一款高性能的、可扩展的开源信息检索(IR)工具库[7]，是一个全文检索引擎的架构。基于 Lucene 建立索引和进行搜索的流程如图 3 所示[8]。

Lucene 首先利用其 Org. apache. lucene. analysis 包(语言分析器，主要用于切分词)进行语法分析与词法处理[9]，将对清洗后的文件切分成字符串(Term)，再由字符串(Term)组成域(Field)，域(Field)组成记录(Document)，记录(Document)组成段(Segment)，段(Segment)组成索引(Index)[10]，其中 Document(文档)是 Lucene 索引和搜索的原子单位[11]。其次，Lucene 利用 Org. apache. lucene. index 包(索引管理，包括索引的建立和删除等)、Org. apache. lucene. document 包(索引存储时的文档结构管理)和 Org. apache. lucene. store 包(数据存储管理)进行索引存储，最后把代表每篇文章的 Document 记录加入至 IndexWriter 对象里面[12]完成索引的创建。

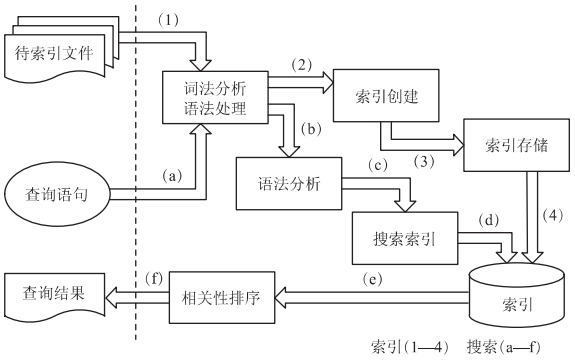


图 3 Lucene 建立检索和进行搜索的流程图

Fig. 3 Workflow of Indexing and Searching with Lucene

针对用户提交的检索请求，Lucene 首先利用 Org. apache. lucene. analysis 包进行词法处理，将检索请求转换成 Term 字符串，再利用 Org. apache. lucene. queryParser 包(查询分析器，实现关键词间的运算)进行语法解析[13]，将 Term 字符串组装成搜索引擎使用的查询(Query)对象，进行检索请求提交。Lucene 可提供多种查询方式，如基于词条查询(TermQuery)、布尔查询(BooleanQuery)、模糊查询(FuzzyQuery)、范围查询(RangeQuery)等，都从 Query 类继承而来[14]。然后利用 Org. apche. lucene. search 包(检索管理，根据检索条件，检索得到结果)进行检索，检索结果返回的是 Hits 对象[15]。

3　实例分析

本文以测绘地理动态信息系统为例，对上述技术流程进行了实现。在该系统中对国家测绘地理信息局官网进行定向爬取时，爬虫对该网站的参数配置文件 Order. xml 中的部分参数配置如下：

```
<long name="max-time-sec">3600</long><!—最大抓取时间(秒)，如果抓取时间超过该值，则爬虫将停止抓取—>
<integer name="max-toe-threads">50</integer><!—最大线程数用于同时处理多个URI—>
<integer name="max-hops">20</integer><!—最大跃点数，也就是抓取深度—>
<string name="allow-by-regexp">(.*(/|\.(html|htm|shtml))$)</string><!—对每个 url 都进行判断，匹配该正则表达式的则爬取—>
<string name="recover-path">../recover.gz</string><!—增量爬取上一次爬取的信息—>
```

网页清洗时需要交互性地针对每个网站页面进行分析，个性化配置要素获取的方法，建立起各网站的清洗模板，从而实现网页的清洗。下面是国家测绘地理信息局官网上的一篇文章的 HTML 页面：

```
<title>黑龙江局完成 1∶5 万地形图制图数据更新生产项目首批成果汇交—国家测绘地理信息局</title>
<script type="text/javascript">
var akeyworks = '黑龙江,';
var source = '黑龙江测绘地理信息局';
var tm = '2015-09-08 16:12:45';
</script>
<table class="aarticle">
<br>
<P>近日，黑龙江测绘地理信息局按照年初制定的生产实施方案，
<P>为了更好地完成该生产项目，黑龙江局精心组织，周密部署，缜密计划，狠抓落实，严格要求。
</table>
……
```

该网站属于“标题后带有其他信息”，需采用 Title__3 方法提取；时间在 script 标签的“tm”变量中，需采用 Time__2 方法提取；来源在 script 标签的“source”变量中，需采用 Source__2 方法提取；正文包含在 class 值是“aarticle”的<table/>标签中，需采用 Text__1 方法提取。因此，该网站的清洗模板便为 Title__3＋Time__2＋Source__2＋Text__1，清洗模板设置界面如图 4 所示。

图 4　国家测绘地理信息局官网清洗模板

Fig. 4　The Cleaning Model of the Website of National Administration of Surveying, Mapping and Geoinformation of China

网页清洗的结果直接为建立索引和信息检索提供数据源。图 5 是系统搜索界面，图 6 为以“地理国情”为关键词的搜索结果。

图 5　搜索界面

Fig. 5　Searching Interface

图 6　搜索结果

Fig. 6　Searching Results

4 结束语

本文针对行业动态信息采集系统建立过程中存在的关键问题，提出了一套基于Heritrix、Jsoup和Lucene实现网页信息爬取、网页清洗和全文索引与检索的解决方案，并以测绘地理动态信息系统为实例进行了实现。该实例经过反复测试，能够较好完成测绘地理信息的定向爬取，实现对不同风格网站网页的清洗，并建立索引提供信息检索。结果表明，这种基于Heritrix、Jsoup和Lucene实现网页信息爬取、网页清洗和全文索引与检索的解决方案是切实可行的，可以为类似的应用提供参考。

参考文献

[1] 高伟锋. 基于Heritrix的主题网络爬虫设计与实现[J]. 南宁职业技术学院学报，2011，16(1)：97-100.

[2] 张俊林. 这就是搜索引擎：核心技术详解[M]. 北京市海淀区：电子工业出版社，2012.

[3] 郑如滨，撒力，谢婷. 基于Heritrix与Lucene的垂直搜索引擎研究[J]. 电脑知识与技术，2008，4(2)：350-352.

[4] 刘高军，夏景隆. 基于Heritrix的网络爬虫研究与应用[J]. 软件导刊，2013，12(5)：123-125.

[5] 张皓，周学广. 基于Heritrix的增量式网络爬虫研究[J]. 软件导刊，2013(11)：135-137.

[6] 盛雪丰. Android开发一大神器——Jsoup[J]. 电脑知识与技术，2015(8)：63-65.

[7] GOSPODNETIC O，Hatcher E. Lucene实战[M]. 人民邮电出版社，2011：c4-5.

[8] 薛宇星. 基于Herixtrix和Lucene的Web站内搜索系统[D]. 西安电子科技大学，2008：49-50.

[9] 白坤，耿国华. 基于Lucene/Heritrix的垂直搜索引擎的研究与应用[J]. 计算机应用与软件，2009：212-214.

[10] 高玉良，张济强，白瑶. 基于Lucene的多索引搜索的研究与应用[J]. 电脑知识与技术，2012 (7)：1470-1472.

[11] 罗刚. 解密搜索引擎技术实战：Lucene&Java精华版[M]. 第二版. 北京市海定区：电子工业出版社，2014：6-7.

[12] 彭哲. 基于Lucene/XML全文检索系统的跨库应用[J]. 图书情报工作，2008，52(06)：110-110.

[13] 潘亭沥. 基于Lucene的面向商业应用的搜索引擎研究与实现[D]. 电子科技大学，2007：39-41.

[14] 周文勤. 使用Heritrix和Lucence的全文检索解决方案[J]. 甘肃联合大学学报：自然科学版，2012，26(4)：52-56.

[15] 周锦程，王丹，余泉. 基于Lucene的全文检索系统的研究与实现[J]. 计算机技术与发展，2011，21：67-71.

（责任编辑：熊苹）

（上接第170页）

深色为源道路网，浅色为比较道路网。从图中可以看出，本文方法实现了较多路段的成功匹配。虽然本文的实验结果未体现出这种方法的效率优势，但置信水平的设置不同，取得的匹配成果也不同，这将是后期进一步研究的方向。

4 结束语

道路网数据作为空间数据库中的重要组成部分，其相关的更新、融合、集成问题的解决关键在于道路网数据的匹配。本文提出的基于模糊信息处理的匹配方法，与现有方法相比，杜绝了通过设定阈值来决定匹配与否所带来的诸多弊端，把匹配关系的判定要素组成一个模糊集合，由置信水平判定匹配关系。

参考文献

[1] 陈玉敏，龚健雅，史文中. 多尺度道路网的距离匹配算法研究[J]. 测绘学报，2007，36(1)：84-36.

[2] 尹川，王艳慧. 道路网增量更新中基于OSTU的目标几何匹配阈值计算[J]. 武汉大学学报·信息科学版，2014，39(9)：1061-1067.

[3] 栾学晨，杨必胜，李秋萍. 基于结构模式的道路网节点匹配方法[J]. 测绘学报，2013，42(4)：608-614.

[4] 张小国，王庆，万德钧. 基于路网拓扑特性及先验知识的地图匹配算法[J]. 东南大学学报：自然科学版，2006，36(4)：625-629.

[5] 孟樊，方胜辉. 利用模板匹配和BSnake算法准自动提取遥感影像面状道路[J]. 武汉大学学报·信息科学版，2012，1(37)：39-42.

[6] 司毅博，李润生，孟伟灿. 一种改进的道路匹配算法[J]. 测绘科学技术学报，2010，6(27)：438-432.

[7] 赵东保，盛业华. 全局寻优的矢量道路网自动匹配方法研究[J]. 测绘学报，2010，39(4)：416-421.

[8] 郭黎，李宏伟，张泽建，等. 道路信息投影匹配方法研究[J]. 武汉大学学报·信息科学版，2013，9(38)：1113-1117.

[9] 胡云岗，陈军，赵仁亮，等. 地图数据所编更新中道路数据匹配方法[J]. 武汉大学学报·信息科学版，35(4)：451-456.

[10] 王密，郭丙轩，雷霆，等. 车载GPS导航系统中GPS定位与道路匹配方法研究[J]. 武汉测绘科技大学学报，2000，3(25)：248-251.

[11] 张睿，张继贤，李海涛. 基于角度纹理特征及剖面匹配的高分辨率遥感影像带状道路半自动提取[J]. 遥感学报，2008，2(12)：224-231.

[12] 朱长青，杨云，邹芳，等. 高分辨率影像道路提取的整体矩形匹配方法[J]. 华中科技大学学报：自然科学版，2008，36(2)：74-77.

[13] 冯可君，张育之. 模糊数学基础知识及其在地图制图学中的应用讲座[J]. 地图，1986(1)：41-44.

[14] 李红梅. 地理空间实体类型语义相似度计算模型的研究[D]. 武汉：武汉大学，2005.

[15] 万波，宗琴，刘川川，等. 基于骨架化和蜘蛛编码的面状实体匹配方法研究[J]. 测绘科学，2012，5(37)：98-99.

（责任编辑：邓国臣）