第37卷第2期 中国科学技术大学学报 Vol. 37, No. 2
2007年2月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Feb. 2007

文章编号:0253-2778(2007)02-0147-06

基于位置的 web 搜索索引研究*

周英华,金培权,岳丽华,龚育昌

(中国科学技术大学计算机科学技术系,安徽合肥 230027)

摘要:针对基于位置的 web 搜索需要将网页中位置信息和文本信息相结合进行索引的应用需求,提出了先倒排表再 *R*-tree 索引和先 *R*-tree 再倒排表索引两种混合索引结构,同时处理文本和位置信息. 大规模真实数据集上的实验表明,这两种方法在查询效率上明显优于已有的倒排表和 *R*-tree 相互独立的索引模式.

关键词:基于位置的 web 搜索;位置索引;文本索引

中图分类号:TP391.1,TP311.12 **文献标识码**:A

Research on index of location-based web search

ZHOU Ying-hua, JIN Pei-quan, YUE Li-hua, GONG Yu-chang

(*Department of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027, *China*)

Abstract: For location-based web search, geographic information should be indexed with textual information. Two hybrid index structures were proposed to deal with both textual and geographic information: one of inverted file preceding the *R*-tree and one of *R*-tree preceding the inverted file. Experiments on large real-world web datasets show that the proposed structures have better query performance than the existing index schema of separate inverted file and *R*-tree.

Key words: location-based web search; spatial index; textual index

0 引言

互联网中与位置相关的信息越来越普遍,统计表明,将近 1/5web 搜索的任务是与特定位置相关的[1,2],如"中关村附近的书店"等. 越来越多的商业搜索引擎开始提供位置相关的服务,如本地搜索,本地广告和地图服务等. Google、百度等商业搜索引擎目前只提供基于黄页或其他付费列表的商业位置的搜索,本文针对更一般的、更具有普遍性的位置搜索,即搜索网页中与地理位置相关的内容.

如何有效地索引和检索与位置相关的信息是基于位置 web 搜索中的一个关键问题. 最简单的方式是先用地名表示位置信息,建立类似文本的索引,然后利用关键词匹配的方式进行检索. 这种方式忽略了基本的空间关系,不支持高级的空间查询,因此有必要设计出一种有效并兼顾考虑空间特征和文本特征的索引结构. 这涉及两个关键问题:位置信息的表示和索引模式. 相关研究已有一些成果[3~7].

网页上有很多位置相关的信息,用来索引的只有大家认为与这个网页最相关的地理区域,即这个网页的地理范围(scope)[3]. 网页的地理范围可以通过分析网页的文本内容以及超链接的地理分布得

* **收稿日期**:2006-01-07;**修回日期**:2006-04-21

基金项目:国家自然科学基金青年基金 (604030200)资助.

作者简介:周英华,女,1978 年生,博士. 研究方向:信息检索. E-mail:yhzhou@mail.ustc.edu.cn

通讯作者:龚育昌,教授. E-mail:ycgong@ustc.edu.cn

到. 为了支持空间语义，scope 表示为两维的空间对象，考虑到表示的准确性和计算开销之间的平衡，用基于经纬度坐标的最小外接矩形（minimum bounding rectangle，MBR）表示一个地理区域. 这些 MBR 可以用常见空间索引（如 R-tree[4]）有效地组织起来. 因为一个网页的 scope 可能包含多个空间对象，所以本文将一个网页的 scope 表示为多个 MBR.

已有的索引模式可以分为两类. 一是在文本检索结果的基础上再进行空间处理[5,6]. 这种方法的缺点是文本检索时只返回与文本特征最相关的网页，对于文本排名比较靠后的网页，其 scope 所包含的地理位置可能会被忽略，导致搜索结果不完整. 二是在索引时同时集成文本和地理信息. 文献[7]提出了倒排表和 R-tree 相互独立的索引结构，解决了第一类索引模式中的问题，但是在这种独立的索引结构中，每个网页在倒排表和 R-tree 结构中分别存储，两个独立结构中的网页列表都比较长，导致在磁盘读取和列表合并上消耗了较多的时间.

本文提出了两种混合索引结构：先倒排表再 R-tree 索引，和先 R-tree 再倒排表索引. 前者将倒排表中每个网页列表进行空间划分，划分的空间由 R-tree 进行索引；后者将所有的地理区域用一颗 R-tree 索引起来，然后对每个地理区域，找出 scope 包含这个区域的所有网页，再利用倒排表对这些网页进行索引. 本文从理论和实验上，对提出的两种混合索引结构与文献[7]中的倒排表和 R-tree 相互独立索引模式进行了比较. 大规模数据集上的实验表明，这三种索引结构的存储开销大致相同，而两种混合索引在查询时间上都明显优于原有方法，且先倒排表再 R-tree 索引又稍优于先 R-tree 再倒排表索引.

1 混合索引结构

本文的目的是建立集成了网页的文本和位置信息的混合索引结构. 传统的文本索引是倒排表，常见的两维的空间索引有 R-tree 及其变异，如四分树、grid 等. R-tree 使用 MBR 作为空间对象的近似，这与本文采用的网页 scope 的表示是类似的，故选择 R-tree 作为空间索引.

本文提出了两种混合方式：先倒排表再 R-tree 索引和先 R-tree 再倒排表索引，并给出每种方式的描述以及每个结构的代价模型分析. 代价模型中用的符号见表 1.

表 1 符号描述

Tab. 1 The description of symbols

符号	描述
M	地名辞典中 MBR 的数目
G	数据集中 geokeyword 的数目
K	词典中关键字的个数
$g(Q)$	查询 Q 中涉及的 geokeyword 的个数
$P_K(k)$	关键字 k 对应网页列表的长度
$P_M(m)$	MBR m 对应网页列表的长度
$P_G(g)$	geokeyword g 对应网页列表的长度
B_{List}	网页列表的存储空间
$B_R(x)$	含有 x 个元素的 R-tree 的存储空间
$T_{\text{I/O}}$	磁盘访问的时间开销（单位是 ms，下同）
T_{disk}	一次磁盘访问的时间开销
$T_R(x)$	含有 x 个元素的 R-tree 的时间开销
$T_{mg}(x)$	合并 x 个元素的 R-tree 的时间开销

1.1 倒排表和 *R*-tree 相互独立索引

这种结构中，网页被分别索引两次：R-tree 空间索引和倒排表文本索引，所有的 MBR 通过一棵 R-tree 索引起来. 与传统 R-tree 不同的是，某个 MBR（即 R-tree 的一个叶结点）指向了一个网页列表，列表中所有网页的 scope 包含了该 MBR，如图 1 所示. 倒排表则与传统的倒排表是一样的. 这样存在了两种网页列表，其表头分别是 MBR 或者关键字.

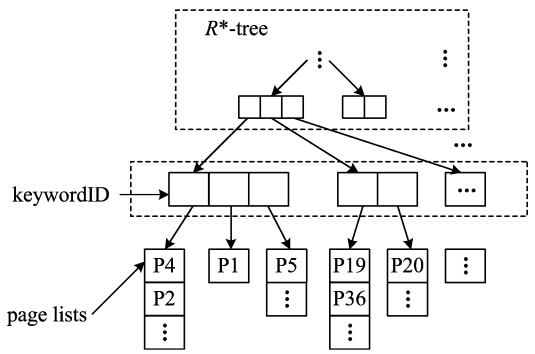

图 1 倒排表和 *R*-tree 相互独立索引结构

Fig. 1 The structure of separate inverted file and *R*-tree

位置相关的 web 搜索包含非空间的关键字和查询区域以及指定的空间查询类型. 非空间的关键字的检索类似传统的倒排表，查询区域和空间查询类型传递到 R-tree. 最终的结果由两种索引网页列表的合并得到，下面给出这种结构的性能.

磁盘上的存储包含这两类网页列表和一个 R-tree. 故 $\text{storage}_1 = B_R^1 + B_{\text{list}}^1$.

一个具有 x 个叶结点的 R-tree 的存储开销[9]是 $B_R(x) = O(x)$.

网页列表的存储依赖与每个列表的长度，单位是一个网页的标识符. 假定表头关键字是 k 的列表长度为 $P_K(k)$，表头是 MBR m 的列表长度为 $P_M(m)$，则这些列表的总长度是 $\sum_{m=1}^{M} P_M(m) + \sum_{k=1}^{K} P_K(k)$，那么

$$B_{\text{list}}^1 = O\Big(\sum_{m=1}^{M} P_M(m) + \sum_{k=1}^{K} P_K(k)\Big).$$

由此可得

$$\text{storage}_1 = B_R^1 + B_{\text{list}}^1 = O(M) + O\Big(\sum_{m=1}^{M} P_M(m) + \sum_{k=1}^{K} P_K(k)\Big) = O\Big(\sum_{m=1}^{M} P_M(m) + \sum_{k=1}^{K} P_K(k)\Big).$$

这说明主要的存储开销取决于列表的总长度.

假定查询 Q 包含 m 个关键字和一个查询区域. 查询计算开销包括：(1)从倒排表中检索 m 个查询关键字对应的网页列表；(2)检索 R-tree，假定得到 n 个 MBR，访问这 n 个 MBR 对应的网页列表；(3)合并这$(m+n)$个列表.

关键字的检索通过哈希函数实现，时间可以忽略. 访问网页列表的时间主要取决于列表的数目和总长度. 合并的过程也取决于这些列表的总长度.

内存中合并 x 个元素是 $T_{mg}=O(x)$，从磁盘读取长度为 x 网页列表的时间是 $T_{\text{I/O}}=T_{\text{disk}} \cdot O(x/B_{\text{section}})$. 其中，$B_{\text{section}}$表示磁盘扇区的大小，具体值依赖于计算机的文件系统，本文取 4 kbytes. 故，

$$\text{time}_1 = T_R^1 + T_{\text{I/O}}^1 + T_{mg}^1 = T_R(M) + \Big(\sum_{i=1}^{m} T_{\text{disk}} \cdot O(P_M(m_i)/B_{\text{section}}) + \sum_{i=1}^{n} T_{\text{disk}} \cdot O(P_K(k_i)/B_{\text{section}})\Big) + O\Big(\sum_{i=1}^{m} P_M(m_i) + \sum_{i=1}^{n} P_K(k_i)\Big).$$

对于不同的查询，影响其性能的因素主要有两个. 一是表头为关键字的 m 个列表和表头为 MBR 的 n 个列表的合并，这取决于$(m+n)$个列表的总长度；二是读入这$(m+n)$个列表的时间. 下面介绍本文提出的两种索引结构.

1.2　先倒排表再 *R*-tree 索引

为方便起见，这种索引结构称之为 IR(inverted file before R-tree)结构.

如图 2 所示，每个关键字指向一棵 R-tree. 在第一种结构中其表头对应的同一个关键字的那些网页根据其地理 scope 分配到不同的 MBRs，在此基础上建一棵 R-tree，这样就可以得到一组网页的列表，其表头是关键字和 MBR 的组合. 如果存在一个网页，它包含了某个关键字且地理 scope 包括了某个 MBR，那么该关键字和该 MBR 组成对，称为 geokeyword.

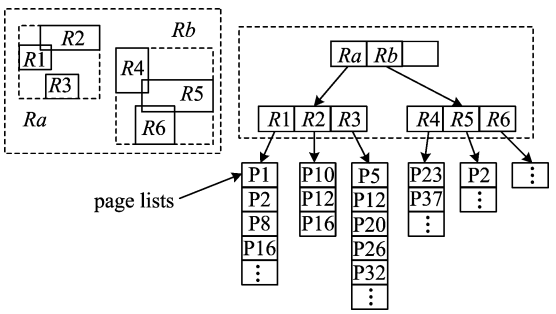

图 2　IR 结构

Fig. 2　First inverted file and *R*-tree index structure

假定一个网页列表的表头是某个 geokeyword，其长度是 $P_G(g)$. 磁盘上的存储包括这些网页列表和由 K 个关键字所指向的 K 颗 R-tree，则有

$$\text{storage}_2 = B_R^2 + B_{\text{list}}^2 = K \cdot O(M) + O\Big(\sum_{g=1}^{G} P_G(g)\Big)$$

实际上，该结构中的一棵 R-tree 所涉及的 MBRs 只是第一种结构中的 R-tree 所索引的 M 个 MBRs 的一个子集，即该结构中的 R-tree 要小于倒排表和 R-tree 相互独立的索引结构. 故其存储主要由这些表头是 geokeyword 的网页列表长度决定.

假定查询 Q 包含 m 个关键字，n 个 MBR，geokeyword 的数目是 $g(Q)$. 查询计算开销包括：(1)首先检索 m 个查询关键字；(2)检索对应的 m 颗 R-tree，假定这些 R-tree 的平均叶结点的个数是 $\overline{M}$，然后找到若干 MBR 以及与其对应的网页列表，列表的个数是 $g(Q)$；(3)合并这 $g(Q)$个列表. 检索 m 个关键字可以通过哈希函数实现，时间可以忽略. 得到

$$\text{time}_2 = T_R^2 + T_{\text{I/O}}^2 + T_{mg}^2 = m \cdot T_R(\overline{M} + \sum_{i=1}^{g(Q)} T_{\text{disk}} \cdot O(P_G(i)/B_{\text{section}}) + O\Big(\sum_{i=1}^{g(Q)} P_G(i)\Big).$$

除了 m 棵 R-tree 的检索，还有两个重要的因素

会影响实时检索. 一是表头为 geokeyword 的网页列表的长度,列表的数目是 $g(Q)$;另一个因素是从磁盘读入这 $g(Q)$个列表的时间. 一个 geokeyword 对应网页列表是,倒排表和 R-tree 相互独立的索引结构中该 geokeyword 涉及的关键字和 MBR 分别对应网页列表的交集. 这样就大大减少了该结构中每个网页列表的长度.

1.3 先 R-tree 再倒排表索引

为方便起见,这种索引结构称之为 RI(R-tree before inverted file)结构.

如图 3 所示,所有网页所涉及的 MBR 都被一棵 R-tree 索引起来. 网页根据其地理 scope 分配给不同的 MBR. 每个 MBR 对应的所有网页根据关键字进行文本索引,这样就可以得到一组表头是 geokeyword 的网页列表.

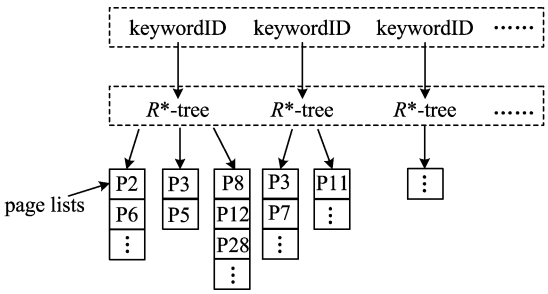

图 3 RI 结构

Fig. 3 First R-tree then inverted file index structure

磁盘上的存储包括这些网页列表和该 R-tree,这样,

$$\text{storage}_3 = B_R^3 + B_{\text{list}}^3 = O(M) + O(\sum_{g=1}^{G} P_G(g)) = O(\sum_{g=1}^{G} P_G(g))$$

这说明主要的存储开销是表头为 geokeyword 的网页列表.

查询计算开销包括:(1)首先检索 R-tree 得到 n 个 MBR;(2)对每个 MBR 和 m 个关键字分别组合,检索其对应的网页列表,列表的个数是 $g(Q)$;(3)合并这 $g(Q)$个列表.

$$\text{time}_3 = T_R^3 + T_{\text{I/O}}^3 + T_{mg}^3 = T_R(M) + \sum_{i=1}^{g(Q)} T_{\text{disk}} \cdot O(P_G(i)/B_{\text{section}}) + O(\sum_{i=1}^{g(Q)} P_G(i))$$

与 IR 结构类似,除了 R-tree 的检索,还有两个因素影响检索时间,即 $g(Q)$个网页列表的总长度和从磁盘读入这 $g(Q)$个列表的时间.

综上所述,对于倒排表和 R-tree 相互独立的索引结构,其在线检索性能取决于 m 个关键字和 n 个 MBR 分别对应网页列表的总长度以及读入这($m+n$)个列表的时间. 本文所提出的 IR 和 RI 两种结构除了进行 R-tree 的检索之外,还主要取决于 $g(Q)$个网页列表的总长度以及读入这 $g(Q)$个列表的时间,因此,这两种结构的性能差不多. 实际情况中,$g(Q)$很小,故所提出的索引结构的查询性能要优于倒排表和 R-tree 相互独立的索引结构. 对于所提出的两种结构,IR 结构中需要检索 m 棵 R-tree,RI 结构中需要检索一棵 R-tree,但是 IR 中的 R-tree 平均有 $\overline{M}=G/K$ 个叶结点,而 RI 结构中的 R-tree 有 M 个. 实际上,每个关键字与某些 MBR 组成的 geokeyword 的数目是有限的,$\overline{M} \ll M$,因此 IR 结构中 R-tree 的规模远远小于 RI 结构. 基于文献[9]的分析,对于同样的查询,R-tree 的规模是影响查询 R-tree 时间的主要因素;最坏情况下,所有的叶结点都将被访问到. 对一个查询而言,在 IR 结构中任意一棵 R-tree 中查询的时间将远小于 RI 中的 R-tree. 考虑到 m 的值也很小,IR 将略微优于 RI.

2 实验

为了评估所提出的混合索引结构的性能,下面将讨论实验的设置和数据集,作为实验环境的基于位置搜索引擎的框架,然后分析实验的结果.

2.1 实验设置和数据集

本实验采用了. gov 的数据集,由 TREC2003 收集的顶级域名是. gov 的主要美国政府网站. 该数据集涉及美国绝大部分地理区域. 通过对数据集的分析,约 18.78%网页是本地网页. 实验主要针对这些本地网页以突出基于位置的网页搜索的索引性能,相关统计信息如表 2 所示.

表 2 数据集的统计信息

Tab. 2 Statistics of our dataset

统计信息	值
所有网页的数目	1 053 111
本地网页的数目	197 775
本地网页中 MBR 出现的次数	197 988
地理辞典中 MBR 的数目	26 090
关键字总数 L	2 684 633
geokeyword 的数目 G	3 535 505
geokeyword 中出现的 MBR 的个数 M	4 246
geokeyword 中出现的关键字的个数 K	758 717

实验环境是，一台拥有 Intel Xeon 3.05 GHz CPU，2GB 内存的机器，操作系统是 Microsoft Windows server 2003.

2.2　基于位置的搜索引擎框架

本文采用图 4 所示的基于位置的 web 搜索引擎原型来测试所提出的混合索引结构的效率. 该原型系统主要包括：抽取模块、索引模块、排序模块和界面模块四个部分. 搜索过程为：抽取网页的 scope，根据 scope 和文本特征建立索引. 当提交一个与位置相关的查询时，检索位置相关网页、排序，并将结果返回给用户. 其中，抽取模块主要功能是得到网页的地理 scope，传送给索引模块. 网页 scope 的抽取和检测采用了文献[8]中的方法，通过分析网页的文本内容和超链接等得到用地名表示的 scope，然后根据地名辞典转换为多个 MBR 表示的 scope.

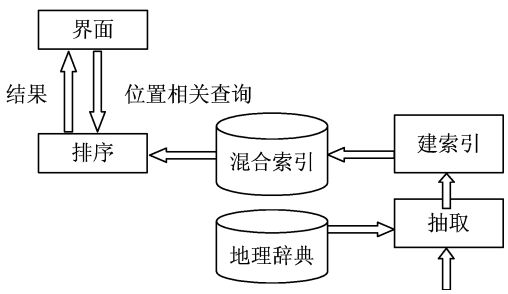

图 4　基于位置搜索引擎原型框架

Fig. 4　The system framework

2.3　实验结果

如表 3 所示，倒排表和 R-tree 相互独立的索引结构的存储是 140.00 +0.83=140.83 Mbytes；IR 和 RI 结构约是 138.95 Mbytes，三种结构的存储开销基本相同. 后两种的差别在 R-tree 的开销，IR 有 K 棵小的 R-tree，RI 结构中只有一棵大的 R-tree，但是 R-tree 的存储相对于网页列表来说是很小的，所以 IR 和 RI 在存储开销方面的差别很小.

表 3　三种结构的存储

Tab. 3　Disk storage for three hybrid index structures

比较项	倒排表和 R-tree 独立		IR 和 RI 结构
网页列表表头	关键字	MBR	geokeyword
列表数目	758 717	4 246	3 535 505
列表总长度	33 481 669	197 988	27 666 384
平均长度	44.13	46.63	7.83
物理存储/MB	140.00	0.83	138.95

为了测试三种结构的查询时间，实验采用了一个由 2 000 个随机产生的查询构成的查询集. 其中 1 000个查询通过在地图上指定一个查询区域，另外 1 000个通过输入地理关键字产生. 空间查询类型随机指定，共有 551 个包含查询，517 个相交查询，514 个被包含查询和 418 个附近查询.

表 4 给出了对应的结果. 显然，IR 和 RI 结构优于倒排表和 R-tree 相互独立的索引. 这是因为倒排表和 R-tree 相互独立的索引结构在 $T_{I/O}$ 和 T_{mg} 方面消耗了太多的时间. 而 $T_{I/O}$ 和 T_{mg} 都取决于网页列表的长度和数目.

表 4　三种结构的平均查询时间

Tab. 4　Average query time for three hybrid index structures

参数	倒排表和 R-tree 独立	IR 结构	RI 结构
T_R/ms	2.34	0.16	2.34
$T_{I/O}$/ms	30.83	7.91	7.91
T_{mg}/ms	17.01	0.73	0.73
查询时间/ms	50.18	8.80	10.98

从表 3 可得到，倒排表和 R-tree 相互独立的索引结构中列表的数目要远小于其他两种(758 717+4 246≪3 535 505)，而其总的列表长度要超过其他两种，所以，倒排表和 R-tree 相互独立的索引结构中列表的平均长度要远大于其他两种结构，如表 5 所示. 同样，倒排表和 R-tree 相互独立的索引结构中需要访问的网页列表的数目也大于本文提出的 IR 和 RI 结构，这是因为为了得到正确的结果，倒排表和 R-tree 相互独立的索引结构需要访问更多的列表，这都增加了查询时间. 另外，IR 结构要访问 m 棵 R-tree，它们的平均叶结点的个数是 3 535 505/758 717=4.6；RI 需要访问一棵 R-tree，其叶结点个数是 M=4 246. 考虑到本查询集中平均每个查询中包含 2.6 个关键字，即 m=2.4，IR 结构在 R-tree 的查询上所花费的时间要小于 RI. 这与上述的分析也是一致的.

表 5　三种结构中的列表长度

Tab. 5　The length of page lists in three hybrid index structures

	倒排表和 R-tree 独立	IR 结构	RI 结构
每个查询所访问网页列表的总长度	38 868.91	122.42	122.42
每个查询所访问网页列表的数目	72.04	4.36	4.36

3　结论

为了高效地进行基于位置的 web 搜索，本文首先将网页的地理 scope 表示为多个 MBR，然后提出

两种集成文本索引和空间索引的混合索引结构，并和已有混合索引进行了比较；另外，还开发了一个基于位置的搜索引擎原型验证提出索引结构的性能. 大规模真实数据集上的实验和分析表明本文所提出的两种混合索引机制在查询时间上明显优于已有索引结构，其中先倒排表再 *R*-tree 混合索引结构具有最佳的效能.

致谢：感谢微软亚洲研究院谢幸研究员对本文研究工作提供的帮助.

参考文献(References)

[1] Gravano L, Hatzivassiloglou V, Lichtenstein R. Categorizing web queries according to geographical locality[C]// Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management. New Orleans: ACM Press, 2003: 325-333.

[2] Sanderson M, Kohler J. Analyzing geographic queries [C]//Proceedings of SIGIR 2004 Workshop on Geographic Information Retrieval. Sheffield, UK: ACM Press, 2004.

[3] Ding J, Gravano L, Shivakumar N. Computing geographical scopes of web resources[C]//Proceedings of 26th International Conference on Very Large Data Bases. Cairo, Egypt: ACM Press, 2000: 545-556.

[4] Beckmann N, Kriegel H, Schneider R et al. The *R*-tree: an efficient and robust access method for points and rectangles [C]//Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data. Atlantic, NJ: ACM Press, 1990: 322-331.

[5] Markowetz A, Chen Y, Suel T, et al. Design and implementation of a geographic search engine[R]. TR-CIS-2005-03, Polytechnic University, 2005.

[6] Ma Q, Tanaka K. Retrieving regional information from web by contents localness and user location[C]//Asia Information Retrieval Symposium. Beijing, China: Lecture Notes in Computer Science, 2004: 301-312.

[7] Martins B, Silva J M, Andrade L. Indexing and ranking in GEO-IR systems [C]//GIR Workshop of CIKM'05. Bremen, Germany: ACM Press, 2005: 31-34.

[8] Wang C, Xie X, Wang L, et al. Web resource geographic location classification and detection [C]// Proceedings of the 14th International World Wide Web Conference. Chiba, Japan, 2005: 1 138-1 139.

[9] Theodoridis Y, Stefanakis E, Sellis T. Efficient cost models for spatial queries using *R*-trees[J]. IEEE Trans. Knowledge and Data Engineering, 2000, 12(1): 19-32.

(上接第 146 页)

analytical approach[J]. IEEE J. Quantum Electron., 1987, 23(5):539-544.

[4] Themistos C, Rahman B M A, Grattan K T V. Finite element analysis for lossy optical waveguides by using perturbation techniques [J]. IEEE Photonics Technology Letters, 1994, 6(4):537-539.

[5] Tseng S M, Hsu K Y, Wei H S, et al. Analysis and experiment of thin metal-clad fiber polarizer with index overlay [J]. IEEE Photonics Technology Letters, 1997, 9(5):628- 630.

[6] She S X. Propagation characteristics and loss of metal-clad graded index optical waveguides [J]. Optics Communications, 1993, 103:365- 369.

[7] QUAN Hong-jun, KANG Shou-wan. Propagation characteristics of metal-clad graded index planar optical waveguides [J]. Chinese Journal of Quantum Electronics, 1995,12(2): 186-191.
全宏庆，康寿万. 金属包层渐变折射率平板波导传输特性[J]. 量子电子学，1995，12(2)：186-191.

[8] Konrad A. High-order triangular finite elements for electromagnetic waves in anisotropic media[J]. IEEE Trans. Microwave Theory Tech., 1977, 25(5):353-360.

[9] XU Shan-jia, LIU Jian, Mizuno K. Dispersion analysis of birefringent dielectric grating structure for application in pressure and temperature measurements [J]. International Journal of Infrared and Millimeter Waves, 2001, 22(3):407-420

[10] 盛新庆. 边缘元及其应用[D]. 中国科学技术大学电子工程与信息科学系，1995.

[11] XU Shan-jia, SHENG Xin-qing. Coupling of edgeelement and mode-matching for multi- step dielectric discontinuity in guiding structures[J]. IEEE Trans. Microwave Theory Tech., 1997, 45(2): 284-287.

[12] She S X. Characteristics analysis of metalclad and absorptive dielectric waveguides by a simple and perturbation method[J]. Opt. Quantum Electron., 1991, 23:1 045-1 054.