矩形 NAM 图像表示及其上的连通区域标记算法*)

夏 晖 陈传波 秦培煜 吕泽华
(华中科技大学计算机学院 武汉 430074)

摘 要 既能减少数据量又能直接快速地进行运算是图像表示方法所追求的目标。本文为克服传统的图像层次结构限制条件过多的缺陷,在借鉴 Packing 问题的思想的基础上,提出了非对称逆布局模式表示模型(Non-Symmetry and Anti-Packing Pattern Representation Model,NAM)。NAM 模型的非对称层次结构使其在表示一幅图像时没有过多的限制条件,因此可以获得更高的压缩比,而且它可以直接进行某些图像处理运算,其基于像素块的运算方式使它的运算效率更高,矩形 NAM 图像表示和基于它的连通区域标记算法证明了这一点。

关键词 布局问题,图像表示,连通区域标记

Study on Rectangle NAM Model for Connected Component Labeling

XIA Hui CHEN Chuan-Bo QIN Pei-Yu LV Ze-Hua
(Institute of Computer, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Less data and faster operation are the aim of image representation, for overcoming the shortcoming of traditional hiberarchy of image which has over many limitations, a non-symmetry anti-packing pattern representation model (NAM) is presented with the concept of packing problem. NAM has less limitations when representing an image for its non-symmetry hiberarchy, so it can acquire higher compression ratio and do some image processing directly and more quickly, which is proved by rectangle NAM image representation and connected component labeling with it.

Keywords Packing problem, Image representation, Connected component labeling

1 引言

图像表示方法有很多,根据其目的大致可以分为三类:第一类是传统的二维数组表示方法,它具有直观、算法容易设计等优点,但是所需数据量大是它的缺点;第二类是专为减少数据量而设计的图像表示方法,这类图像表示方法虽然能得到很大的压缩比,但是一般不能直接进行图像运算,而且为了追求高压缩比还会有意损失部分图像信息,JPEG 图像格式就是其中的代表;第三类图像表示方法是一种分层的图像表示方法,它既能够减少数据量也能够直接进行一些图像运算,而且其基于像素块的运算方式一般能够得到比较快的运算速度,其中四元树表示法是这类方法的典型代表。由于更少的数据量和更快速的运算是图像表示方法所追求的目标,因此既紧凑又便于做各种图像处理运算的第三类图像表示方法一直受到人们的关注。

目前以四元树为代表的图像分层表示方法的研究成果有很多,如 J. Elmesbahi[1] 用四元树来计算图像几何属性;M. Shneier[2] 用四元树来计算图像的几何性质,包括计算二值图像区域的面积、质心和两个区域的交、并以及图像的补图像;C. R. Dyer[3] 用四元树计算图像的欧拉数;H. Samet[4] 用四元树给区域的连通部分做标记。虽然这些分层表示法有许多优点,但是由于过分强调分割的对称性,因此不是最优的表示方法。为了寻找分割最大的非对称分割方法,本文借助 Packing 问题的思维提出了一个新的图像表示方法——非对称逆布局模式表示模型(Non-Symmetry and Anti-Packing Pattern Representation Model,NAM)。NAM 模型的非对称层次结构使它与四元树相比在空间上更具紧凑性,而基于像素块的运算方式也使它在一些图像处理运算中比基于像素点的运算方式速度快。

连通区域标记的目的就是要寻找图像中的所有的目标对象,并将属于同一个目标的对象的所有像素用一个唯一的整数值进行标记。目前图像连通区域标记算法有很多,比较常用的算法分别是基于像素的标记算法、基于游程的标记算法和基于四元树的标记算法。本文以二值图像的 8 连通区域标记为例,将基于矩形 NAM 的标记算法分别与这三种算法进行比较,来说明 NAM 图像表示方法在数据量和算法时间上的优势。

2 非对称逆布局模式表示模型(NAM)

2.1 模型思想

Packing 问题可以简单描述为:给定一个容器和 N 个不同形状的物体。将 N 个不同形状物体放入这个容器中,如果客观上放不下,则给出否定的回答;如果客观上放得下,则给出肯定的回答,并且给出具体的放置坐标。NAM 模型要研究的问题是 Packing 问题的一个反问题(Anti-Packing)。具体可以描述为:给定一个模式(容器)和 N 个预先定义的子模式(N 个不同形状的物体),现在要从这个给定的模式(已经摆放好物体的容器)中抽出这些子模式(物体),用这些子模式的组合来表示给定的模式(已经布局好的容器)。

设 Γ 为要表示的图像模式 $P=\{p_1,p_2,\cdots,p_n\}$ 为子模式集合,$p=\{v,A|A=(a_1,a_2,\cdots,a_m)\}$ 为子模式集合中的一个子模式,其中 v 为子模式 p 的值,A 是子模式 p 的结构参数,$a_i(1\leqslant i\leqslant m)$ 是具体的参数值。重复利用子模式集合 P 中的元素来构成一个模式的过程叫做 NAM 编码过程,表示为:$\Gamma'=T(\Gamma)=\bigcup p(v,A)$。合理选择子模式集合 P,可以使编码

*)国家 863 项目(编号:2004AA420100)资助。**夏 晖** 博士研究生,主要研究方向为图像处理;**陈传波** 教授,博士导师,主要研究方向为图像处理与机器智能及计算机应用;**秦培煜** 博士研究生,主要研究方向为计算机图形学和软件工程;**吕泽华** 博士生研究生,主要研究方向为计算机模式识别。

后的模式 Γ' 具有更简单的表达，便于模式的存储和分析。从编码后的模式中恢复原模式的过程叫做 NAM 的解码过程，表示为：$\Gamma=T^{-1}(\Gamma')$。

2.2 二值图像矩形 NAM 编码算法

矩形 NAM 图像表示方法的子模式集合 P 只包含一种子模式，即矩形子模式 $p_{rec}=\{v,A|A=(x_b,y_b,x_e,y_e)\}$，其中 v 为矩形块的像素值，(x_b,y_b) 为矩形块的起点坐标，(x_e,y_e) 为矩形块的终点坐标。矩形 NAM 图像表示相对其他的 NAM 来说具有结构简单、算法实现简单的特点。

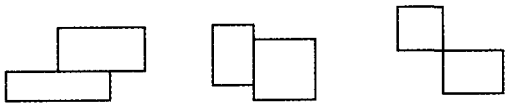
对于二值图像来说，我们可以只对一种像素进行编码。例如我们只考虑 Black 像素点，则其编码算法如下：①将待编码图像全部设为未标记状态；②按照光栅顺序从左至右、从上至下搜索下一个起始点（未标记点的 Black 像素点）$SP(x_b,y_b)$；③从起始点 SP 起匹配面积最大的 Black 像素矩形，确定该矩形的终点为 (x_e,y_e)；④标记由 $SP(x_b,y_b)$ 和 (x_e,y_e) 构成的矩形 R 所包含的区域，并将 R 加入到输出队列 Q 中；⑤重复②～④直到找不到起始点为止。

设图像的像素总数为 M，通过分析容易得知矩形 NAM 图像编码算法复杂度为 $O(M)$，空间复杂度也为 $O(M)$。二值图像的 NAM 算法输出队列 Q 中的存储单元结构为 (x_b,y_b,x_e,y_e)。对于一个 $2^n\times 2^n$ 的图像来说，(x_b,y_b,x_e,y_e) 的二进制码长度都为 n。然而在实际存储的时候，(x_b,y_b) 可以用其跟前一个 (x_b,y_b) 的差值来表示，(x_e,y_e) 可以用其与 (x_b,y_b) 的差值来表示，同时对大小进行限制。在统计意义下，(x_b,y_b,x_e,y_e) 的二进制码长都可以限制为 $n/2$，因此二值图像的一个子模式可以用 $2n$ 位的二进制码存储。假设 $2^n\times 2^n$ 的图像模式用线性四元树来表示，需要 N_T 个 Black 节点，对于线性四元树来说，存贮一个节点占 $3(n-1)+2$ 位[5]，则线性四元树的总数据量为 $H_T=(3n-1)N_T$。假设该图像模式用矩形 NAM 表示需要 N_R 个子模式，则其数据量为 $H_R=2nN_R$。由于线性四元树是一种特殊的矩形 NAM，而且它强调对称性，因此一般情况下矩形 NAM 的子模式数比线性四元树的节点数少很多，即 $N_R\leqslant N_T$。因此 $H_R\leqslant\frac{2}{3}H_T$，也就是说矩形 NAM 的压缩比至少是线性四元树压缩比的 1.5 倍以上。

3 基于矩形 NAM 表示的连通区域标记算法

3.1 算法思想

假设要标记的图像的矩形 NAM 表示队列为 Q，标记的过程就是给 Q 中的每一个矩形一个整数值标记，并使其中任意两个相连通的矩形标记相同而不连通的矩形标记不同，因此整个过程就是对 Q 中全部矩形进行连接关系判断处理的过程。假设 $p_1(x_{b1},y_{b1},x_{e1},y_{e1})$、$p_2(x_{b2},y_{b2},x_{e2},y_{e2})$ 为两个矩形子模式，p_1、p_2 之间的连通关系可以分为 3 类，如图 1。



(a) 垂直相连　(b) 水平相连　(c) 对角相连

图 1　矩形子模式的 3 类连通关系图

其中(a)、(b)、(c)3 类连接关系依次对应式(1)、(2)、(3)，对于 4 连通图像只存在(a)、(b)类连接关系，而 8 连通图像则可以包含上述全部连接关系。

$$\begin{cases} y_{b1}=y_{e2}+1 \text{ or } y_{e1}+1=y_{b2} \\ x_{b1}<=x_{e2}+1 \\ x_{e1}+1>=x_{b2} \end{cases} \tag{1}$$

$$\begin{cases} x_{b1}=x_{e2}+1 \text{ or } x_{e1}+1=_{b2} \\ y_{b1}<=y_{e2}+1 \\ y_{e1}+1>=y_{b2} \end{cases} \tag{2}$$

$$\begin{cases} x_{b1}=x_{e2}+1 \text{ or } x_{e1}+1=x_{b2} \\ y_{b1}=y_{e2}+1 \text{ or } y_{e1}+1=y_{b2} \end{cases} \tag{3}$$

以 8 连通算法为例，根据上面的分析，寻找一个矩形的相连矩形的操作分成两个部分，即垂直相连部分和水平相连部分，至于对角相连的情况因为它既是垂直相连的又是水平相连的，所以可以放到两个部分中的任意一个部分中一起处理。

3.2 算法描述

假设 Q 为图像的矩形 *NAM* 队列，根据矩形 *NAM* 编码算法可知 Q 中矩形互相不重叠，且其顺序是按各矩形起始点的光栅顺序排列的。在进行连接关系判断之前首先需要对 Q 中数据进行排序，以使连接关系判断时进行的比较操作次数最少。假设排序后生成 4 个队列 Qxb、Qxe、Qyb 和 Qye，其中 Qxb(i)、Qxe(i)、Qyb(i)和 Qye(i)分别表示记录了 Q 中全部 xb=i、xe=i、yb=i 和 ye=i 的矩形的索引的子列队，详细的排序过程如下：

```
for (index = 0; index < Q.length(); index++){ /*length()表示读取队列长度*/
    (xb, yb, xe, ye) = Q(index); /*读取Q中第index个矩形*/
    Qxb(xb).add(index), Qxe(xe).add(index), Qyb(yb).add(index); /*将index分别添加到对应子队列的末尾*/
}
for (i = 0; i < Qxe.length(); i++){
    for (j = 0; j < Qxe(i).length(); j++){
        index = Qxe(i).get(j); /*读取Qxe第i个子队列的第j个元素*/
        (xb, yb, xe, ye) = Q(index);
        Qye(ye).add(index);
}}
```

根据 Q 中矩形排列顺序和规则，很容易证明经过上述排序过程后，Qxb(i)、Qxe(i)、Qyb(i)和 Qye(i)子队列中的索引值是按照索引对应矩形的 ye、ye、xe 和 xe 顺序排列的，这样排列的目的是为了在进行后面的连接关系判断时，一个方向一个矩形平均只需要参与比较两次即可。

垂直连接关系判断是通过 Qyb 和 Qye 两个队列进行的，最终要得到全部矩形的垂直连接关系，假设 Qc 为矩形连接关系队列，Qc(index)表示记录与索引 index 对应矩形相连的全部矩形索引的子队列。由式(1)可知与 Qyb(i)子队列中的矩形垂直相连的矩形在且只可能在 Qye($i-1$)子队列中，根据两个子队列中索引的排列顺序，建立垂直连接关系的详细过程如下：

```
for (i = 1; i < Qyb.length(); i++){
    b = 0, e = 0; /*b、e分别表示Qyb和Qye对应子队列的当前访问位置*/
    while (b < Qyb(i).length() && e < Qye(i-1).length()) {/*循环比较直到任何一个子对列到达末尾*/
        index_b = Qyb(i).get(b), index_e = Qye(i).get(e);
        if (IsConnected(index_b, index_e){ /*判断两个索引对应矩形是否相连*/
            Qc(index_b).add(index_e), Qc(index_e).add(index_b); } /*添加两个索引到对方对应的连接关系子队列中*/
        xe_b = Q(index_b).xe, xe_e = Q(index_e).xe; /*得到两个矩形的xe值,用于判断下一次进行比较的矩形*/
        if (xe_b > xe_e) {e++;}
        else if (xe_b < xe_e) {b++;}
        else { /*两个矩形的xe值相等,此时需要判断是否存在对角连接关系*/
            index_b1 = Qyb(i).get(b+1), index_e1 = Qye(i).get(e+1);
            if (IsConnected(index_b1, index_e)){
                Qc(index_b1).add(index_e), Qc(index_e).add(index_b1);}
            if (IsConnected(index_b, index_e1)){
                Qc(index_b).add(index_e1), Qc(index_e1).add(index_b);}
            b++, e++; }/*将两个矩形分别与对方子对列的下一个矩形进行判断后,然后同时取两个队列的下一个矩形进行判断*/
}}
```

水平连接关系建立过程与垂直连接关系建立过程完全一

样，只是不需要再考虑对角连接关系判断，因为它已经在垂直连接关系建立时已经考虑了。两个方向的连接关系建立过程完成后，全部矩形的连接关系都记录在 Qc 中，最后一步就是根据 Qc 生成标记，这个过程十分简单，仅仅是一个顺序扫描，迭代赋值的过程，因此在此不做详细说明。

3.3 算法分析

假设图像大小为 $N\times N$，矩形 NAM 队列长度为 S，原始数据排序所用时间为 $O(S+a_1N)$；两个方向的连接关系建立因为平均每个矩形只参加 4 次判断(水平方向 2 次，垂直方向 2 次)，所以所用时间为 $O(S+a_2N)$；最后的标记过程因为是对连接关系进行遍历，而连接关系个数要小于建立过程中进行判断的次数，所以所用时间为 $O(S)$。上述与 N 有关的时间是因为遍历过程中要先访问各个子队列，而子队列个数就是 N，但是对比总的时间来看，N 的影响很小，因此总的算法时间复杂度为 $O(S+aN)(a<1)$。根据算法很容易得知算法空间复杂度为 $O(S)$。

4 实验结果

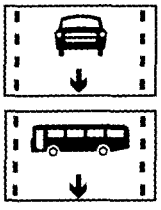


图 2 sign

实验环境为 AMD Athlon 64 Processor 2800＋处理器、1G 内存、Windows XP Professional 操作系统以及 VC6.0 编程环境。图 2、3、4、5 是实验所用的测试图像，大小均为 512×512，另外除了对原始图像进行了实验外还对各自的 1024×1024 放大图进行了实验。实验内容分别包括：计算行程编码、四元树编码和矩形 NAM 编码所用的单元数，这里的单元数分别指的是行程数、黑像素节点数和矩形子模式数；计算行程编码、四元树编和矩形 NAM 编码的压缩比；计算 8 连通区域基于像素标记算法、基于行程标记算法和基于矩形 NAM 标记算法所用的时间，表 1 记录了上述各个数据。

图 3 man

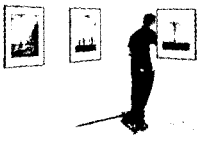


图 4 gallery

图 5 peppers

表 1 矩形 NAM 图像表示数据量以及连通区域标记算法性能比较

图像	图像尺寸	单元数			压缩比			标记算法时间(ms)			
		行程	四元树	NAM	行程	四元树	NAM	像素	行程	四元树	NAM
Sign	512×512	1657	6167	291	3.84	1.63	50.05	4.09	1.04	6.20	0.49
Sign_b	1024×1024	3314	6167	291	6.33	5.86	180.17	16.72	4.61	6.21	0.74
Man	512×512	1129	3834	838	5.02	2.63	17.38	5.91	1.09	3.76	1.14
Man_b	1024×1024	2258	3834	838	9.36	9.43	62.56	24.13	4.49	3.74	1.79
Gallery	512×512	1650	4504	610	4.27	2.24	23.87	3.72	1.02	4.58	0.96
Gallery_b	1024×1024	3300	4504	610	6.99	8.03	85.95	15.11	4.20	4.57	1.42
Peppers	512×512	5328	13711	4177	1.98	0.74	3.49	8.61	1.56	13.55	3.74
Peppers_b	1024×1024	10656	13711	4177	3.34	2.64	12.55	34.85	5.60	13.47	5.23

结论 表 1 中的数据表明，分别用行程、四元树和矩形 NAM 表示一幅图像所用的单元数，矩形 NAM 最少，这是因为行程编码强调方块的一维性，四元树编码强调方块对称性，而采用非对称结构的矩形 NAM 则没有这些限制，因此能够用更少的矩形单元来表示一幅图像，从而其压缩比也比行程编码和四元树编码要好。

表 1 中的数据还表明，在进行 8 连通区域标记运算时，采用基于像素的标记算法所需时间最长，这是因为它的算法时间正比于像素个数；矩形 NAM 标记算法在图像块状性较好的情况下要优于行程标记算法，在图像比较复杂的情况下则略逊于对方，另外随着图像的增大，行程标记算法时间增加很快，而矩形 NAM 标记算法时间增加较慢，这是因为行程标记算法时间与行程数成正比，而 NAM 标记算法时间正比于其子模式数，而受图像尺寸影响很小；通过比较还可以发现，矩形 NAM 标记算法比四元树标记算法要好很多，这是四元树标记算法时间正比于节点数，而其节点数比 NAM 子模式数要大很多的缘故。

实验结果及理论分析表明，由于用矩形 NAM 表示图像时减少了很多限制，因此它可以用较少的子模式来表示一幅图像，因此可以减少较多数据量，并且在进行某些图像处理运算时其性能优于基于其它图像表示方法的运算。另外矩形 NAM 图像表示和其编码算法在 NAM 图像表示中还不是最优的，如果能根据不同类型的图像，选取更合适的子模式集和更优的编码算法，那么用 NAM 表示的图像将能在数据量和图像运算速度方面得到更大提高，因此值得进一步研究。

参 考 文 献

1 Elmesbahi J, Bouattane O, Benabbou Z. theta (1) time quadtree algorithm and its application for image geometric properties on a mesh connected computer (MCC). IEEE Transactions on Systems, Man and Cybernetics, 1995, 25(12)
2 Shneier M. Calculations of Geometric Properties Using Quadtrees. Computer Graphics and Image Processing, 1981, 16 (3): 296～302
3 Dyer C R. Computing the Euler Number of an Image From its Quadtree. Computer Graphics and Image Precessing, 1980, 13 (3): 270～276
4 Samet H. Connected Component Labeling Using Quadtrees. Comm ACM, 1981, 28 (3): 487～501
5 Gargantini I. An Effective Way to Represent Quadtrees. Comm ACM, 1982, 25(12): 905～910
6 Chang F, Chen C J, Lu C J. A linear-time component-labeling algorithm using contour tracing technique. Computer Vision and Image Understanding, 2004, 93(2): 206～220
7 Suzuki K, Horiba I, Sugie N. Linear-time connected-component labeling based on sequential local operations. Computer Vision and Image Understanding, 2003, 89(1): 1～23
8 Dillencourt M B, Samet H, Tamminen M. A general approach to connected-component labeling for arbitrary image representations. J ACM, 1992, 39(2): 253～280
9 Shima Y, Murakami T, Koga M, et al. A high-speed algorithm for propagation-type labeling based on block sorting of runs in binary images, In: Proceedings of 10th International Conference on Pattern Recognition, 1990. 655～658
10 Nicol C J. A systolic approach for realtime connected component labeling. CVGIP: Image Understanding, 1995, 61 (1): 17～31

(上接第 175 页)

得出的每一句话，区分汉字和非汉字字符，并对非汉字字符用自动机识别数字和英文字串。汉字串作为自动分词模块的输入序列。

3.2 分词及未登录词识别阶段

自动分词过程中，需要找出 N 个概率最大的切分词串（N 的值可以根据需要通过用户界面进行设定）。计算概率时，由于每个词的概率是一个很小的正数（小于 1），最后词串的概率接近于 0，无法在计算机上表示。为了解决这个问题，我们用费用代替概率，词串的费用按如下公式计算：

$$Fee(W)=\sum_{i=1}^{n}-\log P(w_i) \qquad (23)$$

处理时，首先从左到右扫描输入汉字串，按其在句子中的先后顺序列出所有候选词，并保留在数组中。扫描候选词序列，采用动态规划的方式，计算每个候选词的 N 个最佳前驱词，即累计费用最小的前一词，并计算当前词的费用，保留这 N 个最佳前驱词信息。如果当前词是终点词，通过回退，得到 N 个费用最小（概率最大）的词串。

未登录词的识别包括姓名识别和地名识别两部分，识别过程如图 3 所示。姓名的识别采用基于概率统计的方法。姓名的费用为姓氏费用与人名费用的和。如果一个汉字串的姓名费用小于一个阈值，就认为它是一个姓名。地名的识别才用基于规则的方式，主要通过检查其后缀来识别。识别出的未登录词保留其词性信息。

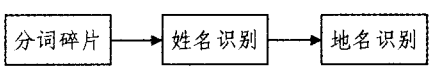


图 3 未登录词识别模块路流程图

3.3 词性标注及评估判优阶段

词性自动标注的对象是一句话的词串，我们的目标是寻找一条概率乘积最大的词性序列。将词串中的每个词的所有词性及其费用添入到该节点中（未登录词的词性信息由前面的处理获得，其余从词典中获得）。从左到右扫描词串，计算该词在某一词性的费用和前一词最有可能的词性。当前词为终点词时，进行回退，得出该词串的词性序列。

对每个分词结果分别进行未登录词识别和词性标注之后，形成了 N 个具有词性信息的词串。我们根据式(22)对每个路径进行打分，费用最小者为最终结果。在这个模块中，我们还进行叠词识别和合并部分未登录词的操作。

4 实验与分析

我们所实现的中文分词及词性标注一体化系统使用了词性标记集为 39 个词性标记的北大标记集，词典中包括 108784 条记录的词语表、114758 条记录的词性表和 2328 条记录的姓名频率表。本文对从 1998 年 1 月人民日报提取的包含 5221 词的文章进行了开放测试，结果如表 1 所示。

从表 1 中数据可看出：(1)该一体化系统的性能良好，开放测试分词准确率和词性标注准确率高于文[9]的 96% 和 94%。(2)引用 N-最短路径法后，分词准确率和词性标注准确率都得到提高，证明前期保留多个粗分结果是合适的。(3)词性信息的引入提高了分词的准确率。(4)随着 N 值的增大，系统性能提升变得不明显。针对这种情况，并且考虑到系统的运行效率，我们一般选取 N 值为 3。

表 1 系统性能测试表

N 值	总词数	自动分词		词性自动标注	
		正确切分词数	准确率 (%)	正确标注词数	准确率 (%)
1	5221	5090	97.49	4920	94.23
2	5221	5190	97.85	4956	94.92
3	5221	5121	98.08	4964	95.07
4	5221	5123	98.12	4964	95.07

结论 本文应用 N-最短路径法，构造了一种中文自动分词和词性自动标注的模型，并实现了一个中文自动分词和词性自动标注一体化处理的中文词法分析器。经测试，该系统具有较高的分词准确率和词性标注准确率，初步证明该方法是有效的。

未登录词的识别和概率字典的构建是影响本系统性能的重要因素。如何处理好这些问题，还有待进一步的研究。

参 考 文 献

1 张华平，刘群．基于 N-最短路径的中文词语粗分模型．中文信息学报，2002，16(5)：1～7
2 ZHANG Hua-Ping, LIU Qun, Zhang Hao, et al. Automatic Recognition of Chinese Unknown Words Recognition. In: First SIGHAN Workshop attached with the 19th COLING, 2002. 71～77
3 何克抗，徐辉，孙波．书面中文自动分词专家系统设计原理[J]．中文信息学报，1991，5(2)：1～14
4 陈小荷．现代中文自动分析．北京语言文化大学出版社，2000
5 Yuan S C, Henry T. Probability Theory. Springer-Verlag New York Inc, 1978. 324～338
6 Manning C D, Hinrich S. Foundations of statistical natural language processing, MIT press, 1999. 197～202
7 Ma Qing, Isshara H, Maoson S A. Mnhi-neuro Tagger Applied in Chinese Texts. In: Proceedings 1998 International Conference on Chinese Information Processing, Beijing, 1998-11-18/20. 200
8 Fine S, Singer Y, Tishby N. The hierarchical Hidden Markov Model: Analysis and applications. Machine Learning, 1998, 32 (1): 41
9 白栓虎．中文词切分及词性自动标注一体化方法[J]．计算语言学进展与应用(JSCL－95)，1995．56～61