

基于度量空间高维索引结构 VP-tree 及 MVP-tree 的图像检索

王志强¹ 甘国辉¹ 程起敏²

¹(中国科学院地理科学与资源研究所,北京 100101)

²(中国科学院遥感应用研究所,北京 100101)

E-mail:wangzq@igsnrr.ac.cn

摘要 首先分析了度量空间高维索引结构的研究意义及具体应用,然后在仔细探讨典型的基于距离的度量空间高维索引结构 VP-tree 及其变种 MVP-tree 的基本思想、构造和搜索算法的基础上,通过具体的实验对其在图像检索中的性能进行了全面的分析,给出了通过实验得到的结论,最后指出了有待进一步研究的方向。

关键词 度量空间 高维索引结构 相似性索引 VP-tree MVP-tree 基于内容的图像检索

文章编号 1002-8331-(2004)36-0041-04 文献标识码 A 中图分类号 TP391

Image Retrieval Based on High-Dimensional Index Structure for Metric Spaces VP-Tree and MVP-Tree

Wang Zhiqiang¹ Gan Guohui¹ Cheng Qimin²

¹(Institute of Geographical Sciences and Natural Resources Research,CAS,Beijing 100101)

²(Institute of Remote Sensing Application,CAS,Beijing 100101)

Abstract: The significance and concrete applications of high-dimensional index structure for metric spaces is analyzed firstly. Then the basic idea, constructing algorithm and search algorithm of a typical distance-based index structure VP-tree and its variation MVP-tree are discussed in detail. On this basis, this paper analyzes the performance of VP-tree and MVP-tree in content-based image retrieval synthetically through concrete experimentation. Conclusions and further directions are given finally.

Keywords: metric spaces, high-dimensional index structure, similarity index, VP-tree, MVP-tree, content-based image retrieval

在广义的度量空间中,如何通过距离函数进行高维向量之间相似性检索的问题,在包括模式识别、多媒体数据库、图像检索等在内的众多领域都有着十分广泛的应用需求。基于距离的度量空间高维索引结构正是为了解决计算机科学相似性查询问题中高维度空间的高额距离计算代价、提高检索效率而提出来的,已经成为基于内容的图像检索中一项重要的关键技术,这是因为:

- (1)图像的可视化特征往往采用高维向量表示。
- (2)基于内容的图像检索是一种典型的相似性查询问题。
- (3)查询图像和目标图像之间的距离计算基于广义的度量空间。

该文首先介绍了目前度量空间高维特征向量相似性检索的常用做法,分析了采用包括 R 树族在内的主流索引结构进行索引的不足之处,归纳了现有的度量空间索引结构;在此基础上,从基本思想、构造算法和搜索算法等方面详细探讨了

一种典型的真正基于距离的静态度量空间高维索引结构 VP-tree (Uhlmann, 1991) 及其变种 MVP-tree (Bozkaya, 1997), 通过具体实验对其性能做了详细的分析。实验结果表明在基于内容的图像检索的可视化特征相似性索引中采用 VP-tree 和 MVP-tree 有利于提高检索效率;最后指出了进一步完善和研究的方向。

1 研究背景

1.1 传统方法及其不足之处

高维特征向量的距离索引问题是基于内容的图像检索的一项关键技术。目前经常采用的解决办法是首先对高维特征空间做降维处理,然后采用包括二叉树、k-d 树、R 树族等在内的主流多维索引结构,这种方法的出发点是:目前的主流多维索引结构在处理维数较低的情况时具有比较好的效率,对于维数很高的情况则显得力不从心(即所谓的“维数危机”),实验结果

基金项目:国家“十五”科技攻关计划项目(编号:2001BA513B01)

作者简介:王志强(1976-),男,博士研究生,主要研究方向:GIS、农业信息化。甘国辉(1944-),男,研究员,博导,主要研究方向:GIS、农业信息化,

区域发展。程起敏(1977-),女,博士,主要研究方向:网络GIS、图像处理。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

表明,当特征空间的维数超过 20 的时候,R-tree 效率明显降低 (Otterman, 1992);而可视化特征往往采用高维向量描述,一般情况下可以达到 10^2 的量级,甚至更高。在表示图像可视化特征的高维向量中,各维信息的重要程度是不同的,通过降维技术去除属于次要信息的特征向量以及相关性较强的特征向量,从而降低特征空间的维数。这种方法已经得到了一些实际应用。

然而这种方法存在不足之处:采用降维技术可能会导致有效信息的损失,尤其不适合于处理特征空间中的特征向量相关性很小的情况;另外,主流的多维索引结构大都针对欧氏空间设计,需要利用到欧氏空间的几何性质,而图像的相似性计算很可能不限于基于欧氏距离。

这种情况下,人们越来越关注基于距离的度量空间高维索引结构,可以直接应用于高维向量相似性查询问题,尽管目前在图像检索中的应用还比较有限,但是可以预见其前景必然十分广阔。

1.2 现有的度量空间高维索引结构综述

度量空间中对象之间的距离度量只能利用“三角不等式”性质,而不能利用其他几何性质,向量空间可以看作由实数坐标串组成的特殊度量空间。目前针对度量空间的高维索引问题提出的索引结构有很多种,大致可以作如下分类:

(1)BK-tree(Burkhard and Keller, 1973)及其变种 FQ-tree (Baeza-Yates, 1994)、FHQ-tree (Baeza-Yates, 1997)、FQA (Chavez, 1999);

(2)VP-tree(Uhlmann, 1991)及其变种 MVP-tree(Bozkaya, 1997)、VPF(Yianlios, 1999);

(3)BS-tree(Kalantari, 1983)及其变种 GH-tree(Uhlmann, 1991)、GNA-tree(Brin, 1995)、VT(Dehne, 1987);

(4)M-tree(Ciaccia, 1997);

(5)SA-tree(Navarro, 1999);

(6)AESA(Ruiz, 1986)及其变种 LAESA(María, 1994)。

以上列出的各种度量空间高维索引结构的时空复杂度、构造复杂度、搜索复杂度以及额外的 CPU 时间可参见文献[1]和[2],但是其中某些值属于经验值或仅在有限条件下成立,因此各种结构在具体应用中的性能目前还缺乏综合、客观的分析,很难评价优劣性。

2 算法描述和数据结构定义

VP-tree(Uhlmann, 1991)是一种真正的基于距离的度量空间上的索引结构^[3],是一棵静态的基于连续距离函数的二叉平衡树,其构造和搜索算法都非常直观,基本思想是将二分查找用于只有距离信息的多维度量空间中,即采用特征空间的目标点集的点与 Vantage Point 之间的距离信息对特征空间进行划分。VP-tree 的构造复杂度为 $O(n \log n)$,搜索复杂度理想情况下可以达到 $O(\log n)$ ^[4]。MVP-tree^[5,6]是对 VP-tree 的改进,通过增加 Vantage Point 的个数、增加结点输出能力达到减少距离计算次数的目的。MVP-tree 的构造复杂度为 $O(n \log_m n)$,比 VP-tree 提高了 $\log_2 m$,搜索复杂度即使在最坏的情况下,也小于 $O(n)$ ^[5]。

2.1 VP-tree 的构造和搜索算法

2.1.1 VP-tree 的构造和搜索算法

给定一个包含 n 个对象的数据集 $S=\{S_1, S_2, \dots, S_n\}$ 和度量

空间距离函数 $d(S_i, S_j)$, VP-tree 的构造算法为:

(1)如果 $|S|=0$,构造空树,返回;

(2)否则,从数据集 S 中随机选取一个对象 S_v ,作为 vantage point,令 M 为 S 中所有对象与 S_v 的距离中值,与 S_v 的距离小于、大于 M 的数据集分别构成左子树和右子树。即:

$$S_L=\{S_i|d(S_i, S_v) \leq M, S_i \in S, S_i \neq S_v\}, S_R=\{S_j|d(S_j, S_v) \geq M, S_j \in S\}$$

(3)递归构造 VP-tree。

对于一个查询对象 Q 和距离范围 r ,则 VP-tree 的搜索算法为:

(1)如果 $d(Q, S_v) \leq r$,则返回 S_v ;

(2)如果 $d(Q, S_v) + r \geq M$,递归查找右子树 S_R ;

(3)如果 $d(Q, S_v) - r \leq M$,递归查找左子树 S_L 。

2.1.2 MVP-tree 的构造和搜索算法

给定一个包含 n 个对象的数据集 $S=\{S_1, S_2, \dots, S_n\}$ 和度量空间距离函数 $d(S_i, S_j)$,增加的参数 p 用于存储预先计算的距离值, k 为叶结点最大输出能力,另外采用变量 $level$ 记录从根结点到当前子结点的 vantage point 的数目,初始值为 1。MVP-tree 的构造算法为:

(1)如果 $|S|=0$,则构造空树,返回。

(2)如果 $|S| \leq k+2$,那么:

①从 S 中任选一个对象,作为第一个 vantage point,记为 S_{v1} ;

②将 S_{v1} 从数据集 S 中去掉,计算 $d(S_i, S_{v1})|S_i \in S$,存入数组 D_1 ;

③选距离 S_{v1} 最远的对象,作为第二个 vantage point,记为 S_{v2} ;

④将 S_{v2} 从数据集 S 中去掉,计算 $d(S_j, S_{v2})|S_j \in S$,存入数组 D_2 ;

⑤退出。

(3)如果 $|S| > k+2$,那么:

①从 S 中任选一个对象,作为第一个 vantage point,记为 S_{v1} ;

②将 S_{v1} 从数据集 S 中去掉,计算 $d(S_i, S_{v1})|S_i \in S$,若 $level \leq p$,存 $S_v.PATH[level]=d(S_i, S_{v1})$;

③对 S 中的对象按照与 S_{v1} 的距离进行排序;设 M_1 为 $d(S_i, S_{v1})|S_i \in S$ 的距离中值,将 S 根据 M_1 划分为两个子数据集 SS_1 和 SS_2 ;

④从 SS_2 中任选一个对象,作为第二个 vantage point,记为 S_{v2} ;

⑤将 S_{v2} 从数据集 SS_2 中去掉,计算 $d(S_j, S_{v2})|S_j \in SS_1, SS_2$,若 $level < p$,存 $S_v.PATH[level+1]=d(S_j, S_{v2})$;

⑥设 $M_2[1]$ 和 $M_2[2]$ 分别为 $d(S_j, S_{v2})|S_j \in SS_1$ 和 $d(S_j, S_{v2})|S_j \in SS_2$ 的距离中值;

⑦用 $M_2[1]$ 和 $M_2[2]$ 将 SS_1 和 SS_2 分成两个子数据集, $level=level+2$;

⑧递归构造 MVP-tree。

给定查询对象 Q 和距离范围 r ,则 MVP-tree 的搜索算法为:

(1)计算 $d(Q, S_{v1})$ 和 $d(Q, S_{v2})$;若 $d(Q, S_{v1}) \leq r$, S_{v1} 为查询结果;若 $d(Q, S_{v2}) \leq r$, S_{v2} 为查询结果。

(2)对于叶结点:

①分别从 D_1 和 D_2 获取 $d(S_i, S_{v1})$ 和 $d(S_i, S_{v2})$;

②如果满足 $d(Q, S_{v1}) - r \leq d(S_i, S_{v1}) \leq d(Q, S_{v1}) + r$ 和 $d(Q,$

$S_{i2})-r \leq d(S_i, S_{i2}) \leq d(Q, S_{i2})+r$, 那么, 对于 $i, i=1, \cdots, p$, 满足 $PATH[i]-r \leq S_r.PATH[i] \leq PATH[i]+r$, 则计算 $d(Q, S_i)$, 如果满足 $d(Q, S_i) \leq r$, 则 S_i 为查询结果。

(3)对于中间结点:

①如果 $level \leq p, PATH[level]=d(Q, S_{v1})$;
如果 $level < p, PATH[level+1]=d(Q, S_{v2})$;

②如果 $d(Q, S_{v1})-r \leq M_1$, 则:
如果 $d(Q, S_{v2})-r \leq M_2[1], level=level+2$, 递归查询第一个分支;
如果 $d(Q, S_{v2})+r \geq M_2[1], level=level+2$, 递归查询第二个分支;

③如果 $d(Q, S_{v1})+r \geq M_1$, 则:
如果 $d(Q, S_{v2})-r \leq M_2[2], level=level+2$, 递归查询第三个分支;
如果 $d(Q, S_{v2})+r \leq M_2[2], level=level+2$, 递归查询第四个分支。

2.2 数据结构定义

根据 VP-tree 和 MVP-tree 的构造和搜索算法描述, 该文定义了 VP-tree 和 MVP-tree 的结点数据结构 tagVPTNode 和 tagMVPTNode, 如下所示:

```
typedef struct tagVPTNode {
    float *vp;           //vp 点
    bool bFlag;          //判断是否为叶结点
    float M;              //距离中值
    tagVPTNode *left;    //左子树
    tagVPTNode *right;   //右子树
} VPTNode, *VPTP;

typedef struct tagMVPTNode {
    float *vp1, *vp2;    //两个 vp 点
    bool vp2Flag;        //判断是否存在第二个 vp 点
    bool flag;           //判断是否为叶结点
    float M1, M21, M22;  //距离中值
    float *D1, *D2;      //距离值
    float **data;         //数据点
    int DataCount;        //数据点的个数
    tagMVPTNode *left1, *right1; //第一、二分支
    tagMVPTNode *left2, *right2; //第三、四分支
} MVPTNode, *MVPTP;
```

3 实验及分析

3.1 实验环境及结果

实验数据来源于一个由 4096 幅大小为 64×64 像素的灰度

图像组成的图像库, 通过对图像的灰度值进行不同程度的采样, 获取 3 组由个数相同(都等于 4096)但维数不同(分别等于 16, 64 和 256)的特征向量组成的数据集, 测试当采用 VP-tree 和 MVP-tree 对维数不同的特征向量之间的距离值进行索引时的性能, 测试 MVP-tree 的性能时, 参数 k 和 p 分别取不同的值。

特征向量之间的距离值采用公式(1)计算得到:

$$Dist(X, Y) = \sqrt[p]{\sum_{i=1}^N (|X_i - Y_i|)^p}$$
 (1)

公式(1)中, $Dist(X, Y)$ 表示特征向量之间的距离值, N 为特征向量的个数, X_i 和 Y_i 表示待比较的两个特征向量的第 i 个分量(像素)的灰度值, p 定义了距离函数的类型, 当 $p=1$ 时, 为城市街区距离, $p=2$ 时, 为欧氏距离, 实验中为了便于理解, 仍采用欧氏距离函数。需要说明的是: 采用向量空间欧氏距离函数并不影响对度量空间索引结构的研究, 目前对于广义度量空间的距离定义还没有统一的、比较直观的表达方式; 同时, 实验中仅仅利用了特征向量之间的距离这一几何性质, 与研究度量空间索引结构的前提条件一致。此外, 查询方式为按例查询, 即选取数据集中任意一个特征向量作为查询点; 相似性查询方式采用范围查询。

实验环境: Pentium IV、2GHz、512M 内存。

表 1 至表 3 列出了当查询范围取不同值时, 从三组维数不同的数据集查询到的与随机选取的查询点相似的特征向量的个数。

表 1 维数等于 16 时一次搜索结果

| | | | | | | |
|------|------|------|------|------|------|------|
| 查询范围 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.20 |
| 邻近个数 | 1 | 2 | 7 | 13 | 36 | 70 |

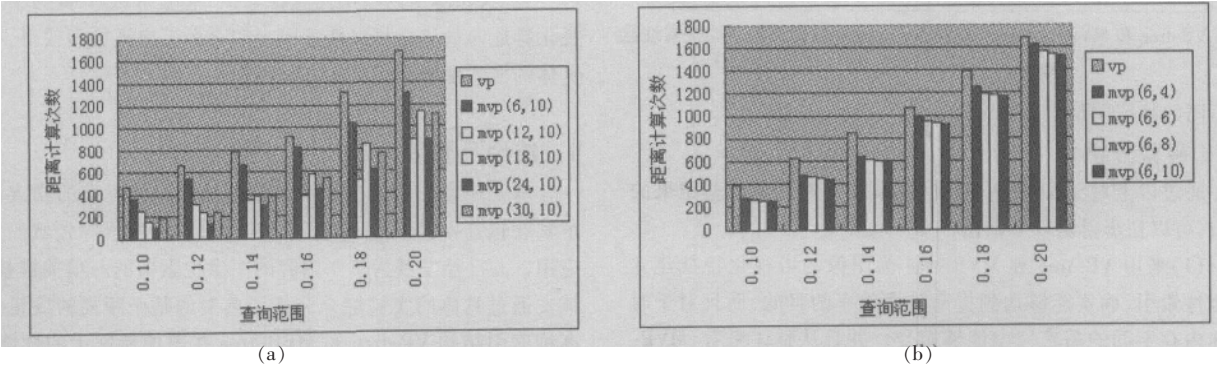
表 2 维数等于 64 时一次搜索结果

| | | | | | | |
|------|-------|-------|-------|-------|-------|-------|
| 查询范围 | 0.350 | 0.400 | 0.425 | 0.450 | 0.475 | 0.500 |
| 邻近个数 | 2 | 6 | 12 | 24 | 46 | 98 |

表 3 维数等于 256 时一次搜索结果

| | | | | | | |
|------|-------|-------|-------|-------|-------|-------|
| 查询范围 | 0.850 | 0.900 | 0.925 | 0.950 | 0.975 | 1.000 |
| 邻近个数 | 2 | 5 | 9 | 31 | 52 | 113 |

实验结果如图 1 至图 6 所示, 图 1 至图 3 分别给出了 VP-tree 和 MVP-tree 应用于对维数不同的三组数据建立相似性索引时的一次搜索性能。图 1 至图 3 中(a)、(b)分别显示了 VP-tree 和 MVP-tree 在参数 k 发生变化($k=4 \sim 10$)而 p 不变($p=10$)时, 以及 k 不变($k=6$)而 p 发生变化($p=6 \sim 30$)时的性能。图 4 至图 6 分别给出了 VP-tree 和 MVP-tree 应用于对维数不同的三组数据建立相似性索引时的 10 次平均搜索性能, 每次都选



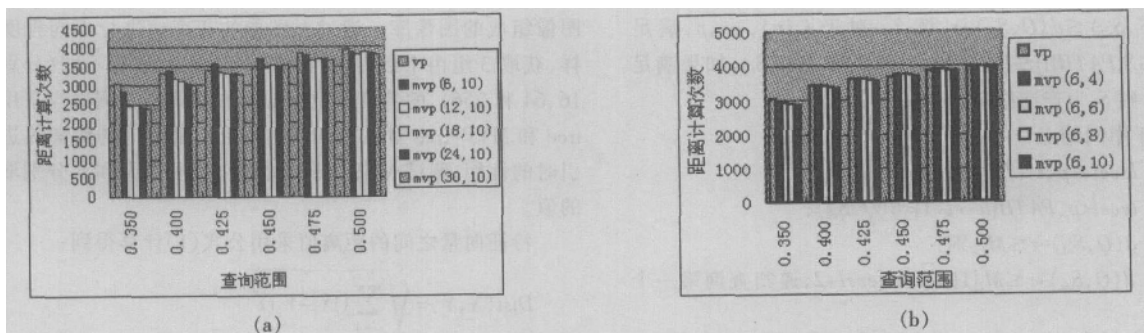


图2 VP-tree 和 MVP-tree 应用于维数等于 64 的数据集的一次搜索性能

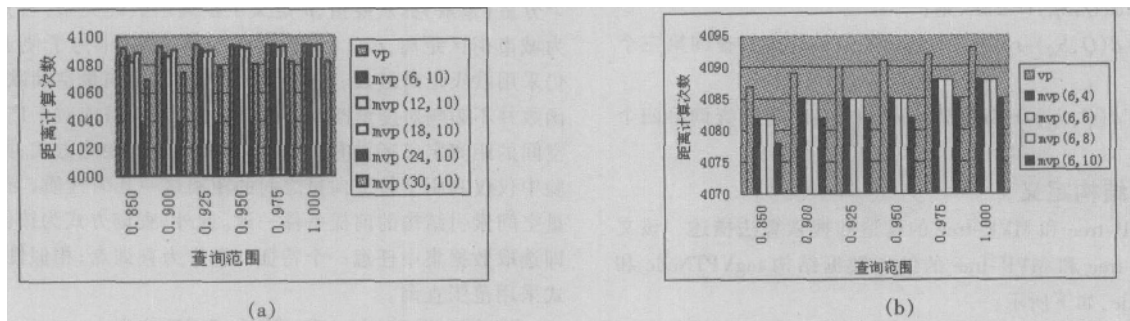


图3 VP-tree 和 MVP-tree 应用于维数等于 256 的数据集的一次搜索性能

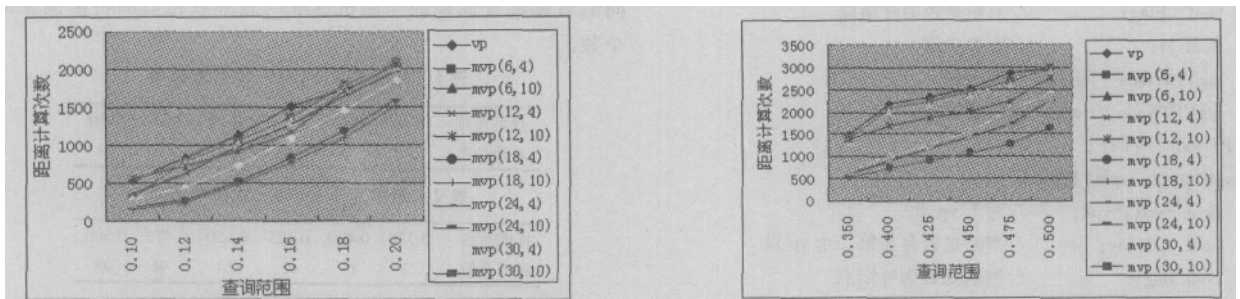


图4 VP-tree 和 MVP-tree 用于维数等于 16 的数据集的平均搜索性能

图5 VP-tree 和 MVP-tree 用于维数等于 64 的数据集的平均搜索性能

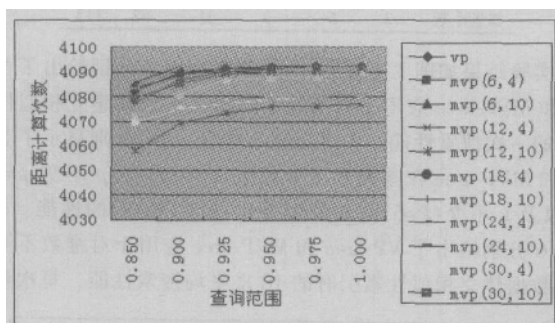


图6 VP-tree 和 MVP-tree 用于维数等于 256 的数据集的平均搜索性能

取不同的特征向量作为查询点。

3.2 实验分析

通过以上对于 VP-tree 和 MVP-tree 搜索性能实验结果的分析,可以初步得出以下结论:

(1)采用 VP-tree 和 MVP-tree 为图像的可视化特征建立相似性索引,确实能够达到提高检索效率的目的,而且对于时间和内存空间的需求完全能够接受,并且从整体来看,MVP-tree 的性能明显优于 VP-tree,尤其对于查询范围和维数较低

的情况。

(2)VP-tree 和 MVP-tree 的搜索性能随着查询范围和维数的增加趋于降低。

(3)查询点的选取会影响到搜索的性能。

(4)在维数相同的情况下,参数 k 和 p 的变化会影响到 MVP-tree 的性能。具体而言, k 的增大会明显提高 MVP-tree 的性能;而 p 的增加则对 MVP-tree 的性能影响不明显。当查询范围很大时,性能趋于一致。

总之,在基于内容的图像检索中,采用 MVP-tree 为图像可视化特征向量建立相似性索引确实有助于提高检索效率,但是具体的搜索性能会受到多方面因素的影响。

4 结论及展望

基于距离的度量空间高维索引结构以其提出的初衷决定了其在包括模式识别、多媒体数据库、图像检索等领域的重要应用,并已经发展为基于内容的图像检索中的一项关键技术。该文通过具体的实验综合分析了典型的基于距离的度量空间高维索引结构 VP-tree 和 MVP-tree 在图像检索中的性能,验

(下转 150 页)

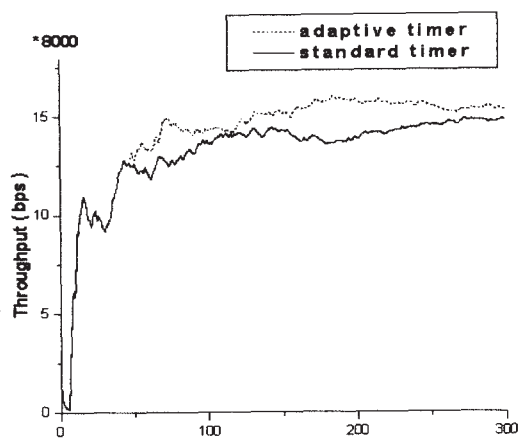


图 14 $N=5$, 8%误码丢包率的吞吐量

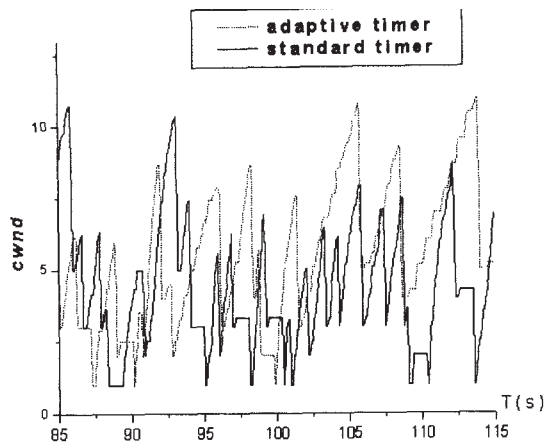


图 15 $N=5$, 8%误码丢包率的拥塞窗口

误码丢包率为 8% 时的吞吐量如图 14, $cwnd$ 如图 15。

由图可以看出: 当总丢包率达到 28% 时, TCP 的吞吐量降低、 $cwnd$ 中空闲等待明显减少; 采用 adaptive timer, TCP 的性能要好于采用 standard timer 算法的 TCP 性能。当 TCP 源的个数 N 大于 5 时的情况与 N 等于 5 的结果相近, 不再详述。

另外该文还仿真了图 1 拓扑中 $N+1$ 节点与 $N+2$ 节点之间突然引入 1 秒时延的情况, 仿真结果表明, adaptive timer 算法的定时器的 TCP 性能仍好于 standard timer 性能。

5 结论

通过对传统 TCP 重传定时器 RTO 设置算法的改进, 改善了无线环境下的 TCP 性能。该文提出的重传定时器算法简单, 自适应于网络状况, 在一定程度上克服了重传定时器的过早超时, RTO 抖动性明显降低。因此, 使用该文的算法能有效改善无线环境下链路误码丢包率在 [0.1%, 11.8%] 时的 TCP 性能。

(收稿日期: 2004 年 6 月)

(上接 44 页)

证了采用 VP-tree 和 MVP-tree 为图像的可视化特征建立相似性索引, 有利于提高检索效率。笔者进一步研究的方向包括 VP-tree 和 MVP-tree 的动态更新、MVP-tree 的参数扩展等。

(收稿日期: 2004 年 8 月)

参考文献

1. Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates et al. Search in metric spaces[J]. ACM Computing Surveys (CSUR), 2001; 33(3): 273~321
2. 刘芳洁, 董道国, 薛向阳. 度量空间中高维索引结构回顾[J]. 计算机科学, 2003; 30(7): 64~68
3. 冯玉才, 曹奎等. 一种支持快速相似检索的多维索引结构[J]. 软件学报, 2002; 13(8): 1678~1685
4. Peter N Yianilos. Data structures and algorithms for nearest neighbor

参考文献

1. J. Postel. Transmission Control Protocol[S]. RFC793, 1981-09
2. M. Allman, V. Paxson, W. Stevens. TCP Congestion Control[S]. RFC2581, 1999-04
3. A. Gurtov. Making TCP Robust Against Delay Spike[R]. Technical Report C-2001-53, University of Helsinki, Department of Computer Science, 2001-11
4. P. Sarolahti, M. Kojo, K. Raatikainen. F-RTO: A New Recovery Algorithm for TCP Retransmission Timeouts[R]. Technical Report C-2002-07, University of Helsinki, Department of Computer Science, 2002-02
5. V. Paxson, M. Allman. Computing TCP's Retransmission Timer[S]. RFC2988, 2000-11
6. S. Cen, P. C. Cosman, G. M. Voelker. End-to-End differentiation of congestion and wireless losses. URL: <http://citeseer.nj.nec.com/cen02end-toend.html>
7. G. R. Wright, W. R. Stevens. TCP/IP Illustrated, Volume 2 (The Implementation)[M]. Addison Wesley, 1995: 817~850

search in general metric spaces[C]. In: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, Austin, Texas, United States, 1993-01: 311~321

5. T. Bozkaya, M. Ozsoyoglu. Distance-based indexing for high-dimensional metric spaces[C]. In: ACM SIGMOD Intl Conf on Management of Data, Sigmod Record, 1997; 26(2): 357~368
6. T. Bozkaya, M. Ozsoyoglu. Indexing large metric spaces for similarity search queries[J]. ACM Transaction on Database Systems, 1999; 24(3): 361~404
7. Ada Wai-Chee Fu, Polly Mei-shuen Chan, Yin-Ling Cheung et al. Dynamic vp-tree indexing for n-nearest neighbor search given pairwise distances[J]. VLDB Journal, 2000; 9(2): 154~173
8. Xiangmin Zhou, Guoren Wang, Jeffrey Xu Yu et al. M+-tree: A New Dynamical Multidimensional Index for Metric Spaces[J]. ADC, 2003: 161~168

(上接 111 页)

Morphological Filtering Method for Preserving the Details of Images[J]. Journal of optoelectronics laser, 2003; 14(8)

5. Li Cuihua, Zheng Nanning. Regular Transformation of Image and Fast Adaptive Filtering[J]. ACTA ELECTRONIC SINICA, 1999; 27(8)
6. Jiangxin Wu, Zhi-Hua Zhou. Face Recognition with One Training Image per Person[J]. Pattern Recognition Letters, 2002; 23(14): 1711~1719

7. Zhao-qi Bian, Xue-gong Zhang. Pattern Recognition (Chapter 8, 9)[M]. Publishing House of Tsinghua University, 2000-01

8. Ji-xiang Sun. Feature Extraction in Pattern Recognition and Computer Vision Invariants (Chapter 2)[M]. National Defence Industry Press, 2001-09
9. M. Kirby, L. Sirovich. Application of Karhunen-Loeve Procedure for the Characterization of Human Faces[J]. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1990; 12(1)