

基于位置敏感哈希的网络视频重复检测*

王洪峰, 刘 辛

(周口师范学院 计算机科学与技术学院, 河南 周口 466001)

摘要: 针对当前网络上存在着大量的重复或近似重复的视频问题,提出了一种基于镜头层比较和位置敏感哈希的快速准确的网络视频重复检测方法。通过视频间匹配的镜头数占查询视频总镜头数的比例来判断视频的相似性。除此之外,还利用著名的近似最近邻查找技术——LSH在镜头层来快速查找相似镜头,从而提高检测速度。通过将镜头作为检索单元,把数据库中所有视频的镜头放到一起构建一个新的数据集,将种子(查询)视频的每一个镜头作为一个查询请求,应用基于LSH的近似近邻检索方法,检索出与查询镜头相匹配的所有镜头,最后融合这些返回的结果,得到查询视频的重复或者近似重复的视频集。通过在包含12 790个视频的CC_WEB_VIDEO数据集上的实验结果表明,该方法取得了相比已有方法更好的检测性能。

关键词: 网络视频; 重复检测; 位置敏感哈希; 镜头层比较

中图分类号: TP311 **文献标志码:** A **文章编号:** 1001-3695(2012)05-1954-05

doi:10.3969/j.issn.1001-3695.2012.05.094

Near-duplicate Web video detection based on locality sensitive hashing

WANG Hong-feng, LIU Xin

(School of Computer Science & Technology, Zhoukou Normal University, Zhoukou Henan 466001, China)

Abstract: The amount of video clips on the Web is huge and is still growing very rapidly. However, there are many duplicate or near-duplicate videos on the Web, which affects the efficiency of video retrieval. This paper proposed a fast method to detect near-duplicate videos based on shot level comparison (SLC) and locality sensitive hashing (LSH). It denoted the videos similarity by the ratio of the number of matched shots between two videos, and introduced the LSH in a special way to improve the speed of the detection. That was, instead of calculating the similarity between seed video and all others one by one, it tried to find all shots which were approximately matched with those in query video by LSH technology and a rearrange dataset. Then, it could detect the near-duplicate videos of query video by fusion the returned shot-sets. It applied this method on the CC_WEB_VIDEO dataset include 12 790 videos. The experimental results demonstrate superior performance of this method over other popular state-of-the-arts methods.

Key words: Web video; near-duplicate detection; locality sensitive hashing; shot-level comparison

0 引言

随着视频共享网站(如YouTube、优酷、土豆网等)的出现,互联网上的视频数量和种类飞速增长,用户可以方便地通过网络上传或下载视频。例如,YouTube 2006年7月的统计数据显示,每天大约有65 000个视频上传到该网站,平均每天有超过100万的观看次数,而到2009年5月每分钟上传的视频达20 h之多(每天超过200万个视频),平均每天的观看次数达2亿^[1]。但是在这些海量的网络视频中,存在着大量重复的或者近似重复的视频。Wu等人^[2,3]通过24个关键词(如“the lion sleeps tonight”“evolution of dance”“fold shirt”等)对YouTube、Google Video和Yahoo! Video三个常用视频搜索引擎的搜索结果进行分析,发现在搜索结果中平均存在27%的重复或近似重复的视频。对于个别查询结果,重复率甚至高达93%。这些大量重复的视频不仅浪费了存储空间,也给视频的有效管理带来不便,并且给用户在网络上搜索感兴趣的视频时返回过多的重复结果。因此为了提高网络视频的管理、搜索及浏览效率,高效的视频重复检测方法就显得十分必要。

在现有的基于内容的重复视频检测方法中,有的比较关注检索速度,如基于视频全局签名(global signature)的重复检测方法^[4,5]。这类方法速度快且对于几乎完全相同的视频能够取得很好的效果,但是难以检测到那些在颜色、光照、编辑效果等方面作了一些改变的重复视频。基于局部关键点(如SIFT^[6]、PCA-SIFT^[7]等)的方法^[2,8]在很多计算机视觉相关应用中取得了非常好的效果,特别适合于检测经过较复杂改变的重复的网络视频,但这类方法速度较慢。文献[8]提出了一种结合全局签名和基于关键点(key-point)匹配的关键帧比较的视频重复检测方法(HIRACH),并取得了很好的准确性。但是,由于一个视频往往包含许多的关键帧,而每个关键帧则包含大量的关键点,从而导致基于关键点匹配的方法需要付出极大的计算成本。例如一段场景变化较快的约4 min的MTV,其关键帧数将超过100,同时每个关键帧可能会有几百甚至几千个关键点。为了提高检测速度,Wu等人^[2]针对网络视频提出了一种基于网络视频上下文(缩略图、标题、影片时间长度、观看次数等)和关键点匹配相结合的快速重复检测方法(CONT+CONX)。该方法仅使用了视频缩略图的关键点来进行比较,从而大大地

收稿日期: 2011-09-22; **修回日期:** 2011-10-31 **基金项目:** 国家自然科学基金资助项目(61103143)

作者简介: 王洪峰(1981-),男,湖北襄阳人,讲师,硕士,主要研究方向为图像处理、数据库技术、系统仿真和软件工程等(whfwlj202868@163.com);刘辛(1979-),女,河南周口人,讲师,硕士,主要研究方向为图形图像处理、数据库技术等。

减少了关键点数量,与其他基于关键点的方法相比,节省了大量的计算时间,但是视频通常是由很多个镜头组成,仅用一幅缩略图难以较好地代表一个视频,所以检测效果比 HIRACH 方法相差甚多。另一方面,该方法用了视频时间长度作为一个重要的线索来排除那些在时间上相差较大(超过一定阈值)的视频(直接判定其为非重复的视频),然而重复的网络视频中存在着许多长版本和短版本(它们在时间上相差较大,但是内容上一致,或者短版本是长版本的一部分)。这也是导致 CONT + CONX 方法在精度上大大下降的重要原因。由此看来,如何快速并准确地从数据库中检测出重复视频仍然是一个具有挑战性的问题。

值得一提的是,一个针对高维数据的快速近似最近邻检索技术 LSH 方法在 1998 年由 Indyk 等人提出^[9],但是只适用于海明距离。2004 年, Datar 等人^[10]对 LSH 方法作了进一步的改进,将其推广到 $l_p, p \in (1, 2]$ 距离上。对于 l_p 模,在数据规模为 n 和特征维数为 d 时,该方法最坏情况下完成一个查询的时间复杂度为 $O(dn^\rho)$, 其中 $\rho < 1$ 。其基本思想是将相似的对象以很高的碰撞概率哈希到数据库中,这样可以过滤掉大量不相似的对象以避免不必要的比较,从而大大提高检索速度。因此如果将 LSH 技术应用到重复视频的检测当中,将会在很大程度上提高检测速度。

为了能够准确并快速地检测重复视频,本文提出了一种基于镜头层比较和 LSH 的快速视频重复检测方法。

1 基于镜头层比较的重复视频检测

仅使用视频的全局特征来判断两个视频是否重复或者近似重复显然是不够的,尤其是那些具有复杂场景或者变化较快的网络视频。而基于局部关键点匹配的方法需要耗费大量的计算时间。为了能更好地应对那些具有复杂场景的视频重复检测,本章将定义一个新的视频相似性度量。

视频在时间上是由一系列镜头组成的,而根据镜头的定义^[11]可以知道,一个镜头内的每一帧在内容上是相同或者相似的(具有相同的场景或者对象),而镜头之间的切换则通常意味着场景的变化。因此镜头可以看做视频内容的基本单元,若两个视频间相似的镜头越多,则这两个视频越相似。根据这一点,本文提出使用视频在镜头层次的比较来判断两个视频的相似性,两个视频之间匹配的镜头越多则相似度越高。实验结果表明,这样定义的相似性度量准则可以很好地克服复杂场景的视频重复检测问题,同时也不会增加太多的计算时间。

1.1 镜头表示

为了在镜头层次进行相似性比较和提高检测效率,首先要提取镜头的特征来表示镜头。如前面所述,一个镜头内的各帧在内容上是相同或相似的,因此可以采用镜头的一幅关键帧来代表该镜头,然后提取关键帧的特征来表示镜头。在这里可以选择任何紧致(compact)和高效的特征来表示镜头。例如颜色直方图经常被用于视频检索^[4,5],并取得一定的效果。但是颜色特征无法描述视频场景的形状特征,并且对于颜色或者光照发生改变的镜头,颜色特征则无能为力了。而 GIST^[12]作为一种全局特征,能够很好地克服光照的变化,并在场景匹配和分类等应用中取得了良好的性能。因此本文采用 GIST 特征来表示镜头。这样一个镜头可以表示为一个 d 维特征向量 $F = (x_1, x_2, \dots, x_d)$, 在实验中取 $d = 320$ 。

因此两个镜头 S_i 和 S_j 之间的相似性 $d(S_i, S_j)$ 可以使用对应的特征向量的欧氏距离表示:

$$d(S_i, S_j) = d(F_i, F_j) = \sqrt{\sum_{k=1}^d (x_k - y_k)^2} \quad (1)$$

其中, $F_i = (x_1, x_2, \dots, x_d)$ 和 $F_j = (y_1, y_2, \dots, y_d)$ 为两个镜头的特征向量。

1.2 视频相似性度量

如上述定义,一旦视频每一个镜头的特征被提取,就可以比较两个镜头的相似性。而两个视频间相匹配的镜头越多,则越相似。为了更好地定量分析,对视频的相似性作如下定义:

假设 $V_q = \{S_{q1}, S_{q2}, \dots, S_{qm}\}$ 表示查询视频,包含 m 个镜头; $V_d = \{S_{d1}, S_{d2}, \dots, S_{dn}\}$ 表示数据库中的一个视频,包含 n 个镜头。 $S_{qk} \in V_q (k = 1, \dots, m)$ 和 $S_{dl} \in V_d (l = 1, \dots, n)$ 分别表示视频中的镜头特征向量。那么视频 V_q 和 V_d 之间的相似性 $R(V_q, V_d)$ 可以表示为

$$R(V_q, V_d) = \frac{\sum_{S_{qk} \in V_q} \delta(S_{qk}, V_d)}{|V_q|} \quad (2)$$

式(2)表示视频 V_q 和 V_d 之间相匹配的镜头个数与查询视频 V_q 的镜头个数之比。当这个比例超过给定的阈值,则称 V_q 为 V_d 的重复视频。其中 S_{qk} 为视频 V_q 中的任一镜头, $|V_q|$ 是视频 V_q 中镜头的个数,函数 $\delta(S_{qk}, V_d)$ 是个逻辑函数,即视频 V_d 中是否存在与镜头 S_{qk} 相匹配的镜头,若查询视频 V_q 中的镜头 S_{qk} 在视频 V_d 中存在相匹配的镜头,其函数值为 1, 否则为 0, 即

$$\delta(S_{qk}, V_d) = \begin{cases} 1 & \text{if } \min_{S_{dl} \in V_d} d(S_{qk}, S_{dl}) \leq T \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

其中, T 为一个给定的经验阈值(在实验中取 $T = 0.25$), 即当查询视频 V_q 中的镜头 S_{qk} 与视频 V_d 中最相似的(最近邻)镜头 S_{dl} 距离不大于 T 时,则这两个镜头相匹配。

2 基于镜头层 + LSH 比较的快速重复视频检测

有了视频的相似性定义,采用最直接的全搜索方法,可以计算出查询视频与数据库中所有视频的相似性,从而检测出重复的视频。实验结果表明,基于镜头层比较和全搜索最近邻查找的视频重复检测方法(SLC + NNS),在准确性上非常接近 HIRACH^[8]方法,而速度则大大提高。为了进一步加快检索速度,本章将提出一种基于镜头层比较和 LSH 技术^[10]相结合的快速视频重复检测方法(SLC + LSH)。

基于第 1 章的视频相似性定义,可以令 V_q 为一个查询视频, $D = \{V_1, V_2, \dots, V_N\}$ 为包含 N 个视频的数据库,其中每个视频可以表示为一组镜头 $V_i = \{S_{i1}, S_{i2}, \dots, S_{in_i}\}$, 其中 $S_{ik} \in V_i$ 是第 k 个镜头的特征向量。要在数据库 D 中找到所有与 V_q 近似重复的视频,必须计算所有的 $R(V_q, V_i)$, 其中 $V_i \in D (i = 1, \dots, N)$ 。计算 $R(V_q, V_i)$ 的关键是对于 V_q 中的每一个镜头 $S_{qk} (k = 1, 2, \dots, m)$, 在视频 V_i 中找到与 S_{qk} 最相似的镜头。这一问题可以看做是特征空间的最近邻查找问题,其时间复杂度为 $O(dmn_i)$ 。其中 d 为特征维数, m 和 n_i 分别为视频 V_q 和 V_i 的镜头数量。而数据库中共有 N 个视频,要进行重复视频检测则需要重复 N 次前面的操作,因此时间复杂度为 $O(dNmn)$, 这里 $\bar{n}$ 为数据库中视频的平均镜头个数。由于特征的维数及数据集中视频的数量巨大,查找最近邻问题往往是一个困难的问题。为了进一步提高检测速度,笔者认为查找近似近邻是足够

的。尽管使用近似近邻可能会带来某些错误,但对结果不会导致太大的影响。因此可以引入著名的近似最近邻查找技术 LSH 来进行重复视频检测。

2.1 近似最近邻查找技术——LSH

为了解决最近邻查找的时间瓶颈问题,近年来,研究人员提出了一些通过查找近似最近邻来减少查找时间的方法^[9,10,13~15]。与最近邻查找不同,这些方法将返回给定查询的一个近似最近邻,该近似最近邻到查询点的距离最多不超过查询到最近邻的距离的 c 倍, $c > 1$ 叫做近似因子(代表了近似程度)。在很多情况下,找到近似最近邻是足够的,微小的距离差距对结果不会造成太大影响。

LSH^[10]是一种著名的近似最近邻查找技术,它能快速地对以很高的概率查找到种子点的 c -近似 R -近邻((c, R) -NN):给定一个 d 维空间的数据集 $\mathcal{P}$ 和参数 $R > 0, \delta > 0$, 构造一个数据结构,对于给定的查询 q , 如果 $\mathcal{P}$ 中存在一个 q 的 R -近邻 p ($\|q - p\| \leq R$), 那么算法将以概率 $1 - \delta$ 返回一个 q 的 c -近似 R -近邻 p' ($\|q - p'\| \leq cR$)。其基本思想是使用一组哈希函数将每个点哈希到若干桶(buckets)中,而每个哈希函数必须满足:两个点距离越近则碰撞(被哈希到同一个桶中)的概率越高。然后,查询过程是将查询点哈希到前面已经建立的若干桶中,通过只搜索这些桶来达到快速检索的目的。对于 $l_p, p \in (1, 2]$ 模,这种方法最坏情况下的时间复杂度为 $O(dn^\rho)$, 其中 $\rho < 1/c, c > 1$ 。LSH 算法已经被成功地应用到了很多领域中,包括网页聚类^[16]、计算生物学^[17]、计算医药学^[18]等。

2.2 结合 LSH 的快速视频重复检测

基于上面的讨论,如果在视频重复检测中引入 LSH 技术,则可以有效地提高检测速度。然而使用文献[10]中的 LSH 技术的基本要求是距离的定义必须是 l_p 模。因此对于视频相似性定义,不能直接在视频层次使用 LSH。幸运的是,镜头间的距离恰好为 l_2 模,因此可以考虑在镜头层次使用 LSH。 E^2 LSH^[19]是基于 l_2 距离的文献[10]中近似近邻查找方法的一个实现。通过对数据库的预处理,给定一个查询 q 和参数 $R > 0, \delta > 0$, 通过文献[10]所述方法 E^2 LSH 可以得到每个 c -近似 R -近邻,然后扔掉所有的近似近邻而保留 R -近邻。因此 E^2 LSH 能以概率 $1 - \delta$ 在子线性时间内返回每一个 R -近邻。可以利用 LSH 来提高视频重复检测的速度。尽管使用 LSH 技术可能会由于漏检个别 R -近邻而给视频重复检测带来某些错误,但是产生这种错误的概率很小,对视频重复检测的结果不会造成太大的影响。

如果每次计算查询视频 V_q 和数据中的任一个视频 V_i 的相似性 $R(V_q, V_i)$ 时,在镜头层使用 LSH 方法,即对于 V_q 中的每一个镜头 $S_{q_k} (k = 1, 2, \dots, m)$, 利用 LSH 从 V_i 中找到与 S_{q_k} 近似最相似的镜头(即 $d(V_q, V_i) \leq T$), 则时间复杂度为 $O(dm_i^\rho)$, $\rho < 1/c$ 。若要计算所有的 $R(V_q, V_i)$, 则需重复 N 次前面的过程,那么时间复杂度约为 $O(dNm\bar{n}^\rho)$, 这里 $\bar{n}$ 为数据库中视频的平均镜头个数。尽管相对于全搜索,查询时间有所减少,但是收益仍然有限。为了进一步加快检测速度,在整个数据集上使用 LSH 得到查询视频每个镜头的近似重复镜头集,最终融合这些返回的镜头集,使得本文方法在时间复杂度为 $O[dm(N\bar{n})^\rho]$ 内计算出查询视频与数据库中每个视频相似性。首先将数据库中所有的视频镜头放在一起构建一个新的数据集:

$$D_s = \{V_1, V_2, \dots, V_N\} = \{S_{11}, S_{12}, \dots, S_{1n_1}, \dots, S_{Nn_N}\} \quad (4)$$

把查询视频 $V_q = \{S_{q1}, S_{q2}, \dots, S_{qm}\}$ 中的每一镜头 S_{qk} 作为一个查询请求。因此,对于每一个给定的查询镜头(点) $S_{qk} \in V_q$ 和阈值 T , 通过 E^2 LSH 将返回一个近似近邻集(整个数据集中与 S_{qk} 相匹配的镜头) $A_k = \{S_1, S_2, \dots, S_{lk}\} (k = 1, \dots, m)$, 其中 $A_k \subset D_s$, 并且对于每个 $S_i \in A_k, d(S_{qk}, S_i) \leq T$ 。然后通过如下方法计算 V_q 与数据库中任一个视频 V_i 的相似性:

$$R(V_q, V_i) = \frac{\sum_{k=1}^m T(V_i, A_k)}{|V_i|} \quad (5)$$

$$\text{其中: } T(V_i, A_k) = \begin{cases} 1 & \text{if } V_i \cap A_k \neq \emptyset \\ 0 & \text{if } V_i \cap A_k = \emptyset \end{cases} \quad (6)$$

$V_i \cap A_k \neq \emptyset$ 意味着数据库中视频 V_i 中至少存在一个镜头与查询视频 V_q 中的第 k 个镜头 S_{qk} 距离不大于 T 。那么必然有 S_{qk} 到视频 V_i 的最相似镜头距离不大于 T , 即 $\min d(S_{qk}, S_{il}) \leq T$ 成立。亦即式(6)等价于 1.2 节定义的视频相似性度量准则式(3)。这样可以通过一次遍历所有返回的近似最近邻集 $A_k (k = 1, \dots, m)$ 一次性计算出查询视频 V_q 与数据库中所有的视频 V_i 的相似性 $R(V_q, V_i)$ 。因此本文方法的计算复杂度为 $O[dm(N\bar{n})^\rho]$, 其中 $\rho < 1/c, c > 1$, 相比在每次计算 $R(V_q, V_i)$ 时使用 LSH, 更大地减少了查询时间。

3 实验结果及评价

为了评价本文方法的性能,在本章中将给出本文方法及其他一些其他现有方法^[2,4,8]的实验结果,包括视频数据集和评价准则的介绍。实验表明,本文方法在准确性上非常接近目前最好的基于局部关键点比较的 HIRACH 方法^[8],而在时间上有着最高数万倍的提高。

3.1 视频数据集及评价方法

实验中使用了香港城市大学的 CC_WEB_VIDEO 数据集^[3]。该数据集是使用 24 个常用查询(如“the lion sleeps tonight”“evolution of dance”“fold shirt”等)在 YouTube、Google Video 和 Yahoo! Video 上查询的结果。数据集共包含 12 790 个视频,具有不同的格式(其中 85.4% 为 FLV 格式,13.4% 为 AVI 格式和 1.2% 的其他格式)。查询关键词和与查询视频重复的视频个数如表 1^[3]所示。例如对于查询“the lion sleeps tonight”,一共有 812 个视频,其中有 334 个视频与查询视频重复。

表 1 利用 24 个查询从 YouTube、Google Video 和 Yahoo! Video 上搜索到的视频集

ID	Query	Near-Duplicate		
		视频个数	重复视频个数	重复率/%
1	the lion sleeps tonight	812	334	41
2	evolution of dance	483	122	25
3	fold shirt	436	183	42
...	...	...	...	...
24	India driving	287	26	9
	total	12 790	3 481	27

为了评价重复检测方法的性能,采用 TRECVID 所使用的标准评价方法^[20]。本文采用平均查准率(average precision, AP)来评价各种方法的性能。平均查准率是在一个查询的所有相关对象被返回后的平均准确率:

$$AP = \sum_{r=1}^N (P(r) \times R(r)) \quad (7)$$

其中: r 是排列次序; N 为返回结果的个数; NR 是数据集中所有相关结果的个数; $R(\cdot)$ 为一个二值函数(当第 r 个返回结果为重复视频时为1,否则为0); $P(\cdot)$ 为前 r 个返回结果的查准率。

为了评价系统的整体性能,采用了 MAP(mean average precision)来比较各种方法的整体性能。

$$\text{MAP} = \frac{1}{M} \sum_{i=1}^M \text{AP}(i) \quad (8)$$

其中, M 是总的查询个数, $\text{AP}(i)$ 是第 i 个查询的平均查准率。

3.2 实验结果

本文的实验在一台 Core-2 双核 3.0 GHz CPU、内存 4 GB 的机器上进行。将本文方法(SLC + NNS 和 SLC + LSH)与下列现有方法作了比较:基于全局签名的方法(Sig-CH)^[4]、基于局部关键点的层次比较方法(HIRACH)^[8]和基于上下文的方法(CONT + CONX)^[2]。另外针对本文方法还比较了使用近似最近邻查找的 LSH 技术(SLC + LSH)和全搜索(精确最近邻查找)技术(SLC + NNS)的性能。实验结果表明,使用 LSH 技术在准确性上与全搜索技术非常接近,而在检索速度上有最大 5 倍的提高。而且无论是 SLC + NNS 方法还是 SLC + LSH 方法,在精确性上都更接近当前最好的 HIRACH 方法,但是在速度上有甚至高达数万倍的提高。与其他方法相比更是有不可比拟的优势。

图 1 和表 2 分别展示了各种视频重复检测方法在精确性和查询时间上的性能。从图 1 中可以看到,本文方法在性能上非常接近目前最好的 HIRACH 方法,而 Sig-CH 和 CONT + CONX 方法的准确性相对较低。表 2 比较了各种方法完成每个查询所用的时间。可以看到,虽然 HIRACH 方法在性能上最好,但所需的计算时间是非常大的,甚至达 2 h 之久(如查询 15),无法实现实时检索。Sig-CH 方法虽然计算速度最快,但是准确性上不是很令人满意。本文的方法在性能接近 HIRACH 方法,大大优于 CONT + CONX 方法,并且时间上相比 HIRACH 和 CONT + CONX 方法都有很明显的提高,能够实现实时的检索。

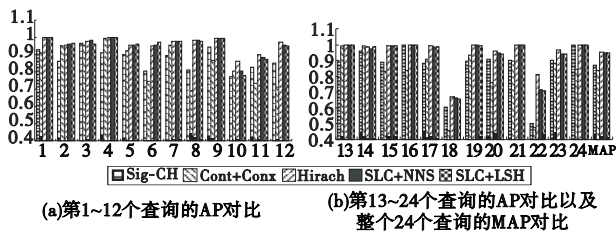


图1 各种视频重复检测方法AP及MAP比较

从图 1 中可以看到,基于颜色直方图的全局签名(Sig-CH)方法,对那些简单场景或者变化很少的视频取得了较好的结果,如查询 9、16、24;而本文的方法取得了更好的或者一样的结果。但是对于那些复杂场景的视频,基于全局签名的方法在性能上则出现明显降低,因为不同的视频也可能具有相似的颜色分布。与 HIRACH 方法相比,本文方法取得了相接近或者一致的准确性。然而,基于关键点匹配的关键帧比较方法(HIRACH)需要耗费非常多的时间,如表 2 所示,这种方法耗时是最多的,对于某些查询甚至超过 2 h 之久(如查询 15)。而本文的方法均能够在不到 1 s 的时间内完成查询。因为 CONT + CONX 方法是用了视频的时长来判断两个视频是否重复,然而数据库中存在大量时间长度不同的重复视频,所以从图 1 中可以看到,该方法所取得的性能是所有方法中最差

的。因此,时间长短并不是一个非常有效的特征。而本文所提出的基于镜头层比较的方法取得了几乎与 HIRACH 方法一样的结果。通过镜头间的比较,使得本文方法对具有复杂场景变换的视频重复检测依然取得了很好的效果,如查询 13 和 15。对于查询 18 和 22 取得了相对较低的查准率,是因为:查询 18 中,原始视频是通过手机获取的,导致视频模糊不清,从而影响了查询结果;而对于查询 22,有很多不相关的帧被插入到视频的开头和结尾,从而导致较低的相似性。但在性能上仍然优于 Sig-CH 和 CONT + CONX 方法。总之,本文方法在总体性能上与 HIRACH 方法一样取得了令人满意的结果,但对于某些查询检测速度却有高达数万倍的提高。

表2 不同方法所用时间比较 /s

Query ID	Sig-CH	SLC + NNS	SLC + LSH	CONT + CONX	HIRACH
1	0.56×10^{-3}	0.023	0.005	0.48	2.41
2	0.42×10^{-3}	0.133	0.123	1.28	48.95
3	0.38×10^{-3}	0.005	0.005	0.62	1.17
4	0.39×10^{-3}	0.004	0.004	0.40	0.62
5	0.35×10^{-3}	0.161	0.059	3.39	215.60
6	0.57×10^{-3}	0.893	0.192	10.78	1610.92
7	0.35×10^{-3}	0.083	0.057	2.92	115.49
8	0.43×10^{-3}	0.549	0.264	11.06	1291.86
9	0.22×10^{-3}	0.151	0.080	6.24	655.63
10	0.31×10^{-3}	0.494	0.107	9.08	2396.25
11	0.31×10^{-3}	0.030	0.017	1.05	14.28
12	0.64×10^{-3}	0.437	0.299	2.72	197.97
13	0.36×10^{-3}	0.061	0.017	2.45	54.83
14	0.13×10^{-3}	0.027	0.018	1.63	62.25
15	1.19×10^{-3}	7.187	2.410	43.33	8201.56
16	0.25×10^{-3}	0.042	0.031	2.61	94.19
17	0.05×10^{-3}	0.279	0.157	11.56	6161.64
18	0.41×10^{-3}	0.011	0.015	0.16	0.60
19	0.46×10^{-3}	0.386	0.231	5.88	589.71
20	0.24×10^{-3}	1.454	0.487	7.59	5571.87
21	0.38×10^{-3}	0.449	0.338	18.59	3135.72
22	0.37×10^{-3}	0.024	0.015	0.31	3.07
23	0.87×10^{-3}	5.475	1.221	34.42	6214.73
24	0.40×10^{-3}	0.014	0.003	1.75	13.36

图 1 的最后一项显示了各种方法的总体平均查准率(MAP)。很容易发现,HIRACH 方法对所有的查询都取得了最好的结果,而本文的方法所取得性能都非常接近 HIRACH 方法。并且根据对使用精确最近邻查找方法和近似最近邻查找方法 LSH 的比较可以发现,使用 LSH 方法所取得结果与精确最近邻查找所得结果相差无几,因此近似最近邻查找的精度已经足够,对结果不会造成太大的影响。

根据表 2 的时间比较,在计算速度上,本文所提的基于 LSH 的方法却比精确最近邻查找方法有很大的提高,平均有 3 倍的提高。相比 CONT + CONX 方法,在速度上则有几十到上百倍的提高,而对于 HIRACH 方法则有几百到几万倍的提高,而且数据规模越大,提高的幅度越大。例如查询 15,HIRACH 方法需要超过 2 h,CONT + CONX 方法需要 43 s,本文方法即使使用精确最近邻查找也仅需 0.7 s,而本文所提的基于 LSH 的方法则只需 0.24 s。

4 结束语

为解决网络视频重复检测问题,本文提出了一种基于镜头的视频特征,并定义了新的视频相似度量。并提出了一种基于镜头层比较和 LSH 的快速重复视频检测方法。通过在镜头层级提取特征和镜头间的比较,该方法在检测性能上取得了很好的结果,特别是对具有复杂场景的视频也能取得较好的性能,明显优于基于全局特征的 Sig-CH 方法和基于上下文的 CONT + CONX 方法,并非常接近当前最好的基于关键点匹配的方法(HIRACH)。除此之外,通过使用 LSH 技术,本文方法完成视频重复检测任务时最坏情况下的时间复杂度为 $O[dm(Nn)^{\rho}]$,其中 $\rho < 1/c, c > 1$,在检索速度上与 HIRACH 方法相比有高达数万倍的提高,能够实现实时检测。

参考文献:

- [1] Wikipedia. YouTube[EB/OL]. <http://en.wikipedia.org/wiki/YouTube>.
- [2] WU Xiao, NGO C, HAUPTMANN A, *et al.* Real-time near-duplicate elimination for Web video search with content and context[J]. *IEEE Trans on Multimedia*, 2009, 11(2): 196-207.
- [3] WU Xiao, NGO C, HAUPTMANN A. CC_WEB_VIDEO: near-duplicate Web video dataset[EB/OL]. <http://vireo.cs.cityu.edu.hk/webvideo/>.
- [4] YUAN Jun-song, DUAN Ling-yu, TIAN Qi, *et al.* Fast and robust short video clip search for copy detection[C]//Proc of the 5th Pacific Rim Conference on Multimedia Information Processing. Berlin: Springer-Verlag, 2004: 479-488.
- [5] LIU Lu, LAI Wei, HUA Xian-sheng, *et al.* Video histogram: a novel video signature for efficient Web video duplicate detection[C]//Proc of the 13th International Conference on Multimedia Modeling. Berlin: Springer-Verlag, 2007: 94-103.
- [6] LOWE D. Distinctive image features from scale invariant keypoints[J]. *International Journal of Computer Vision*, 2004, 60(2): 91-110.
- [7] KE Yan, SUKTHANKAR R. PCA-SIFT: a more distinctive representation for local image descriptors[C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2004: 506-513.
- [8] WU Xiao, HAUPTMANN A, NGO C. Practical elimination of near-duplicates from Web video search[C]//Proc of the 15th International Conference on Multimedia. New York: ACM, 2007: 218-227.
- [9] INDYK P, MOTWANI R. Approximate nearest neighbors: towards removing the curse of dimensionality[C]//Proc of the 30th Annual ACM STOC. New York: ACM, 1998: 604-613.
- [10] DATAR M, IMMORLICA N, INDYK P, *et al.* Locality sensitive hashing scheme based on p-stable distributions[C]//Proc of the 20th Annual Symposium on Computational Geometry. New York: ACM, 2004: 253-262.
- [11] BIMBO A. Visual information retrieval[M]. [S. l.]: Morgan Kaufmann Inc, 1999.
- [12] OLIVA A, TORRALBA A. Modeling the shape of the scene: a holistic representation of the spatial envelope[J]. *International Journal of Computer Vision*, 2001, 42(2): 145-175.
- [13] AILON N, CHAZELLE B. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform[C]//Proc of the 38th Annual ACM Symposium on Theory of Computing. New York: ACM, 2006: 557-563.
- [14] CHAKRABARTI A, REGEV O. An optimal randomised cell probe lower bounds for approximate nearest neighbor searching[J]. *SIAM Journal on Computing*, 2010, 39(5): 1919-1940.
- [15] PANIGRAHY R. Entropy-based nearest neighbor search in high dimensions[C]//Proc of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms. New York: ACM, 2006: 1186-1195.
- [16] HAVELIWALA T, GIONIS A, INDYK P. Scalable techniques for clustering the Web[C]//Proc of WebDB Workshop. 2000.
- [17] BUHLER J. Efficient large-scale sequence comparison by locality-sensitive hashing[J]. *Bioinformatics*, 2001, 17(5): 419-428.
- [18] DUTTA D, GUHA R, JURIS C, *et al.* Scalable partitioning and exploration of chemical spaces using geometric hashing[J]. *J Chem Inf Model*, 2006, 46(1): 321-333.
- [19] ANDONI A, INDYK P. E²LSH: exact Euclidean locality-sensitive hashing[EB/OL]. (2004). <http://web.mit.edu/andoni/www/LSH/>.
- [20] SMEATON A, OVER P, KRAAIJ W. Evaluation campaigns and TRECVID[C]//Proc of the 8th ACM International Workshop on MIR. New York: ACM, 2006: 321-330.