基于时空域信息融合的视频拷贝检测算法研究

严 聪[1] 纪墨轩[2] 纪庆革[1]
(中山大学数据科学与计算机学院 广州 510006)[1] (东北大学软件学院 沈阳 110000)[2]

摘 要 为了有效利用视频独有的时空域特性来提高视频拷贝检测算法的鲁棒性和精度,提出一种基于时空域信息融合的快速拷贝检测算法。该算法包括基于时空域信息融合的指纹提取算法、基于倒排索引的匹配搜索算法和结合异步滑窗策略的基于匹配状态机的匹配搜索算法。指纹提取算法首先将视频分段形成时空域信息帧,然后对该信息帧进行分块,提取DCT系数后,利用其中值进行阈值化得到视频指纹。基于倒排索引的搜索算法根据指纹的二值性特点建立倒排索引表,然后通过索引表快速查询指纹。结合异步滑窗策略的基于匹配状态自动机的搜索算法,利用与最近邻之间的匹配状态来改变搜索范围和步长,而异步滑窗策略通过对在线和离线过程采用不同的提取策略,减少搜索量,加快搜索速度。实验结果表明,提取的指纹对噪声模糊、添加字幕、空间偏移、旋转、掉帧具有较好的鲁棒性,同时提出的搜索方案在时间效率上也有较大的提升。

关键词 视频拷贝检测,时空域信息融合,倒排索引,状态自动机,异步滑窗
中图法分类号 TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.055

Research on Video Copy Detection Algorithm Based on Spatial-Temporal Domain Informative Fusion

YAN Cong[1] JI Mo-xuan[2] JI Qing-ge[1]
(School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China)[1]
(Software College, Northeastern University, Shenyang 110000, China)[2]

Abstract In order to effectively utilize unique spatial-temporal domain characteristics of video to enhance the robustness and accuracy of copy detection algorithm, this paper proposed a fast video copy detection algorithm based on spatial-temporal domain informative fusion, which includes a fingerprint extraction algorithm based on spatial-temporal domain informative fusion and two kinds of matching search algorithms of which one is based on inverted-file index and the other is based on matching state machine with asynchronous window strategy. The fingerprint extraction algorithm firstly forms the spatial-temporal domain informative frame by video segmentation, then partitions the informative frame into blocks and extracts DCT coefficient with its median value as threshold to obtain video fingerprint. The matching search algorithm based on inverted-file index sets up inverted-file index table by binary characteristics of fingerprint, then quickly queries fingerprint according to the index table. Combining the matching search algorithm based on matching state machine with asynchronous window strategy, we can change search scope and step size by matching state with nearest neighbor. Meantime, asynchronous window strategy can adopt different extract strategies in online and offline process to accelerate the whole search. The experimental results show that our fingerprint extraction algorithm is robust in the case of Gaussian noise, adding subtitles, spatial shift, rotation and frame drop, and the proposed schemes tend to have great improvement in time efficiency.

Keywords Video copy detection, Spatial-temporal domain informative fusion, Inverted-file index, State automation, Asynchronous window

随着互联网技术和多媒体技术的迅猛发展和广泛使用,网络视频数据的采集、存储和分享越来越方便,视频资源呈指数增长。为了适应不同的应用需求和目的,视频通常会经过一些编辑处理(如添加噪声、插入logo/字幕、尺寸变化、对比度/亮度变化、添加黑色边框等),使得网络视频搜索结果中出现大量内容接近甚至完全相同的视频,这给用户的搜索质量和搜索效率带来了巨大的影响。随着这个问题的发展和深化,视频拷贝检测技术的研究逐渐在视频版权保护、视频监控和检索、互联网内容管理和商业媒体追踪等领域有了广阔的发展前景和市场需求,这使得视频识别引起了越来越多的学者和商业人士的研究兴趣和广泛关注。

美国国家标准技术研究所对视频拷贝有如下定义[1]:一个视频或者片段在经过某些编辑处理操作后,得到内容视觉感知相同但表现形式不完全相同的同源视频版本。由于视频拷贝的形式多种多样,因此视频拷贝检测的首要问题是如何能够有效利用视频自身的内容检测出是否为拷贝,即如何提

[1] http://www-nlpir.nist.gov/projects/tv2008/TrecVid2008CopyQueries.pdf

到稿日期:2015-08-05 返修日期:2016-01-11 本文受NSFC-1广东联合基金(U0735001)资助。

严 聪(1991-),男,硕士生,主要研究方向为视频拷贝检测;纪墨轩(1995-),女,主要研究方向为物联网;纪庆革(1966-),男,博士,副教授,CCF高级会员,主要研究方向为计算机图形学、计算机视觉、多媒体、虚拟现实,E-mail:issjqg@mail.sysu.edu.cn(通信作者)。

取鲁棒性的视频指纹；视频拷贝检测的第二个关键的问题在于找到一种有效的指纹匹配算法来保证检测精确度和加快搜索速度以及准确定位。

近年来，特征提取的研究可大致分成3类：1）基于空域的特征提取。Hampapur等[1]对当时主流的视频拷贝检测技术进行了归纳分析，并指出序数度量方法用于视频拷贝检测可以取得较好的效果，但对时域攻击和空间几何攻击鲁棒性不强。Lee等[2]提出了基于梯度方向中心的视频指纹算法，即将视频帧分块，计算每块的亮度梯度和方向，生成梯度方向的中心作为视频指纹。2）基于时域的特征提取。Indyk等[3]在视频拷贝检测领域做了先驱工作，提出利用镜头边缘检测计算不同镜头之间的持续时间来构造视频特征。这种方法对镜头变换较多的视频序列比较适用，对于镜头较少的视频序列，检测结果不太理想。Stentiford等[4]提出了一种基于时域序数度量的指纹提取方法，即首先将视频帧分块并求出块的像素平均值，然后定义一个时域窗口，将时域窗口内每一帧的块沿时间轴方向按照块的平均亮度值进行排序，最后把得到的序数矩阵作为视频指纹。3）基于时空域的特征提取。Coskun等[5]将视频序列看作三维矩阵，进行DCT变换后，提取低频系数作为视频指纹。文献[6,7]将时空切片应用到视频拷贝检测中，利用断层扫描的方法形成时空切片来提取视频指纹。另一方面，如何加快特征匹配过程也引起了国内外学者的关注。Oostveen等[8]提出一种运用在视频拷贝检测中的基于倒排索引技术的搜索算法。Zhao[9]提出一种基于统计相似性的搜索，在这种搜索中，统计查询概念是基于相关相似指纹的分布。Barrios等[10]提出一种基于基准的索引结构和近邻相似性搜索的算法，其利用索引极大地减少了匹配过程中的比较计算，并使搜索效率有了较大的提升。

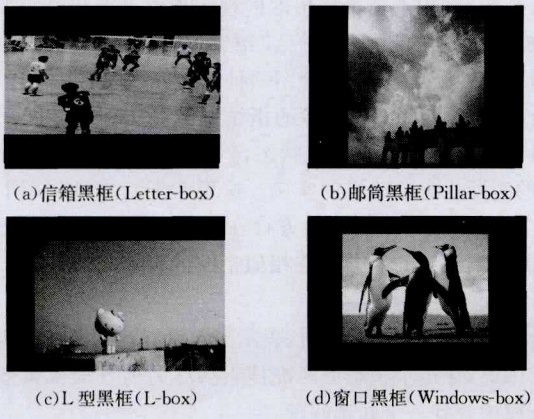
本文的主要贡献在于：1）提出了一种基于时空域信息融合的指纹提取算法，提高了对各种拷贝攻击的鲁棒性；2）给出了基于倒排索引的匹配搜索算法和结合异步滑窗策略的基于匹配状态机的匹配搜索算法，提高了搜索速度和定位精度。

1 指纹特征提取

1.1 预处理

1.1.1 去除黑色边框算法

黑色边框在视频中是很常见的一种几何攻击。视频帧图像中存在的黑色边框的模式一般有4种：信箱黑框（Letter-box）、邮筒黑框（Pillar-box）、L型黑框（L-box）和窗口黑框（Windows-box）。这4种黑框的样例如图1所示。



(a)信箱黑框(Letter-box)　(b)邮筒黑框(Pillar-box)

(c)L型黑框(L-box)　(d)窗口黑框(Windows-box)

图1 各种黑色边框示意图

在包含黑色边框的图像中，从黑色边框到帧画面的实际内容在灰度的梯度值上会有剧烈的变化，根据这个原理提出一种针对各种黑色边框都适用的算法。首先利用水平和垂直Sobel算子求出边缘梯度图像；然后在得到图像的梯度值图像之后，计算梯度图的每行或每列平均值投影成的行向量和列向量，找出此时在行、列方向各个半边的梯度响应的最大值的点作为边缘点；最后提取出真实的帧内容。

1.1.2 预处理流程

（1）对视频的时域方向采用方差为σ_1^2的一维高斯低通滤波器，这样可以避免由于时域降采样对时域产生的混叠现象。

（2）以4帧/秒的帧率对视频序列进行时域降采样操作，通过时域降采样可以避免很多内容冗余帧的计算，加快了指纹的提取速度。

（3）将视频帧进行灰度化，增加其对颜色变化的鲁棒性，这样生成的视频指纹不但对彩色视频适用，而且也能够应用于传统的黑白视频。

（4）根据最外围的像素值的总和是否小于μ来判断是否有黑色边框，然后根据1.1.1节提出的去黑色边框的算法对图像进行去除黑框的操作，得到帧画面的实际内容。

（5）对上述得到的帧内容进行高斯平滑处理，这可以减小噪声影响，同时在一定程度上减小空域降采样对指纹带来的影响。

（6）对获得的帧实际内容图像进行空间降采样处理，将其统一归一化为固定的$W\times H$大小。对视频空间域的降采样处理，可以统一帧的尺度规格，保持视频帧在尺度上的一致性。这使得到的指纹可以对任意分辨率的帧图像具有鲁棒性，同时也给后面的操作提供了统一的标准，减小了因图像大小不一而产生的误差。

（7）将视频序列分成固定长度且有$\tau\%$重叠的片段每个片段包含S帧，之后对这些片段进行指纹提取操作。而有重叠的片段可以减少视频指纹匹配时出现的“同步匹配问题”（也叫做时间偏移，即时间上有偏差对齐匹配）。这里对离线部分和在线部分提取指纹时采用一种异步滑窗策略，以加快搜索速度。异步滑窗策略将在下节进行详细介绍。

这里，对上述时空降采样中采用的高斯核函数的取值做一个简要分析：由于高斯分布的3σ原则，99.73%的值会落在距均值3σ的范围之内，因此为了保持足够的计算精度，应该选取的高斯滤波器模板的大小为$(2\times\lceil 3\sigma\rceil+1)\times(2\times\lceil 3\sigma\rceil+1)$。

如果σ过大，则将过滤大部分的视频动态内容，导致视频指纹的可区分性下降；而如果σ太小，则不能很好地抑制视频内容的高频成分，会影响视频指纹的鲁棒性。这里，对时域的高斯低通滤波器选择$\sigma_1=2$，对帧内容的空域的高斯滤波器选择$\sigma_x=\sigma_y=3$。

1.2 基于时空域信息的视频指纹提取

1.2.1 时空域信息代表帧

时空域信息代表帧[11]为对一段沿时间轴方向连续的固定数目的帧进行指数加权累加得出的图像，包含了视频的时空域信息，可以对时空域信息进行很好的表达。时空域信息代表帧的计算公式如下：

$$F'(m,n)=\sum_{k=1}^{S}\omega_k f(m,n,k) \tag{1}$$

其中，$f(m,n,k)$为所累加的片段中第k帧的(m,n)像素位置的亮度值；S为所提取片段的固定帧数目；ω_k为指数加权系数，$\omega_k=\lambda^k$，这里λ取值为0.645。

1.2.2 视频指纹生成

视频指纹生成的具体步骤如下：

(1)利用指数加权从分割成包含 S 帧的片段中生成时空域信息帧。

(2)将时空域信息帧分割成大小为 $2\omega\times2\omega$ 的重叠块,每个块与水平方向和垂直方向上的邻近块各自有 50%的重叠。块重叠是为了用足够的空间信息得到更加准确的检测结果。重叠块要遵循以下公式:

$$B(u,v)=\{F'(m,n)\mid x\in u\omega\pm\omega,y\in v\omega\pm\omega\} \quad (2)$$

其中,$u\in\{0,1,2,\cdots,W/\omega-1\}$,$v\in\{0,1,2,\cdots,H/\omega-1\}$,当在计算图像边缘时,给图像的外部填充为 0。

(3)对每个块进行二维 DCT 变换,提取两个最接近 DC(直流)系数的 AC(交流)系数作为特征,这两个 AC 系数代表块水平和垂直方向的纹理特征,可以通过以下等式计算:$\alpha_{u,v}=V^{\mathrm{T}}B(u,v)E$和$\beta_{u,v}=E^{\mathrm{T}}B(u,v)V$,其中,$\alpha_{u,v}$和 $\beta_{u,v}$表示对应块的各自的第一个垂直和水平的 AC 系数,V 为一个包含 2ω 个元素的列向量,$V=[\cos(0.5\pi/2\omega),\cos(1.5\pi/2\omega),\cdots,\cos(\pi-2\omega)]^{\mathrm{T}}$,$E$ 为一个长度为 2ω 且每项元素值全为 1 的列向量。这里不用完全展开 DCT 变换就可以得到这两个 AC 系数,极大地降低了计算量。

(4)最后,将获得的 AC 连接起来并进行二值化处理,将得到的结果作为这段视频片段的指纹。

假设每个时空域信息帧被分成 N 个重叠块,计算所得的 AC 系数的中值 m。视频指纹 h 的计算公式如下:

$$h=\begin{cases}1, & \lambda_i\geqslant m\\ 0, & \lambda_i<m\end{cases} \quad (3)$$

其中,λ_i 为 AC 系数中的第 i 个数。

最后得到一个 0 和 1 数目均衡的指纹作为视频片段的指纹。

1.2.3 异步滑窗策略

为了解决基于窗口的拷贝检测算法中原始视频和其拷贝之间存在的时间错位的问题,并提高检测精度,通常会对参考视频数据库和查询视频同时采用窗口重叠的策略来提取视频指纹。因为查询视频可能是从原始视频中裁剪而来,所以它和它的原始视频之间可能具有不同的起始点,导致查询和原始视频的窗口可能不能对齐($t\neq t'$)。当然这种方式有利也有弊,一方面,算法中窗口重叠部分越大,对时移攻击的鲁棒性就越好;另一方面,增加重叠比例,也会增加指纹数据库大小,从而降低搜索速度。例如窗口重叠比例为 50%~75%时,会导致数据库规模翻一倍。此外,重叠部分越多也会增加不能确定某个指纹的位置的概率,因为相邻窗口重叠的比例越大,就会产生越多的相似指纹,当在处理有重复的场景的视频时,这个问题就会愈加严重。在这种情况下,原始视频可能被正确地检测,但查询结果在时间上可能不是准确对齐。为了解决上述问题,本文对原始视频和查询视频采用不同的滑窗策略,描述如下:

在离线部分处理参考视频数据库时,本文将每个参考视频划分成长 T 秒且互不重叠的窗口,然后将每个窗口序列传递给指纹提取程序产生其指纹。伴随指纹一同产生的相关元数据诸如视频编号和视频内的相应段的确切时间位置也会随指纹存储在指纹数据库中。对于在线部分,从查询视频中提取指纹的过程是相同的,唯一不同的是窗口重叠的策略,即对于给定的查询视频序列,每隔 τ 秒提取一个 T 秒的视频片段。因此,连续的视频片段之间存在 $T-\tau$ 秒的重叠。

异步滑窗策略可以大大减小指纹数据库的大小并降低数据库各项之间的依赖数目,因此生成的指纹数据库可以更有效地存储和检索,同时由于查询视频的指纹间有很大部分的重叠,拷贝检测系统能够更加精确地定位拷贝在数据库视频中的位置。

2 匹配搜索策略

2.1 基于倒排索引的视频指纹搜索

将每个二进制指纹划分成更小的 r 位互不相交的子块,把这些更小的子块叫做"字",由此可知,每个字的十进制值有 2^r 种可能;然后利用数据库指纹中的字来创建一个倒排索引文件,由于所有的指纹长度相等,因此倒排文件用 $2^r\times s$ 大小的表来表示,其中 s 是长度为 L 的指纹的字的个数($s=\lfloor L/r\rfloor$)。这个索引表的水平坐标表示字在指纹中的位置,垂直方向表示字的可能值。为了产生这个索引表,从每个指纹的第一个字开始,将这个指纹的索引加入到对应字的值的第一列,然后继续处理每个指纹的所有的字,并将其映射到倒排文件表的列中。表中的(i,j)项表示指纹的第 j 个字的十进制值为 i,项里存入的是指纹编号 ID。

创建好倒排索引表之后,利用它找到查询视频的某个指纹在数据库中的匹配指纹。为了找到查询指纹的匹配,首先将指纹划分成字,然后将查询指纹和数据库中所有从相同字开始的指纹进行匹配。指纹的索引号从倒排索引表中对应字的二进制的第一列中找到,之后再计算它们之间的距离。

基于倒排索引表的查询过程如下:

(1)将查询指纹序列的第一个指纹分成 s 个字(每个字 r 位),计算第一个字块的十进制值;然后在指纹数据库中查找所有从相同字块开始的指纹,即从十进制值在倒排文件表中的对应项的第一列中找到这些指纹的索引;最后比较它们之间的相似度,如果指纹之间的相似度高于某个阈值 ThD,则把它作为一个匹配,如果不存在这个匹配,则对指纹的下一个字执行相同的过程。

(2)继续查找指纹的第 i 个字块的值和索引表中的对应值的第 j 列的内容,直到找到一个匹配或者查询到最后一个字,则匹配结束。如果没有找到匹配指纹,则这个查询指纹不属于数据库。

(3)继续对剩下的查询指纹重复步骤(1)、(2)的操作,直至最后一个查询指纹结束。

2.2 基于状态自动机的匹配搜索

由于检测的查询视频一般是从原始参考视频中裁切的一部分视频,利用滑窗策略提取的视频指纹可能和原始视频的指纹具有不同的起始点,导致匹配时会产生错误匹配的结果。针对错位匹配现象,下面介绍一种基于匹配状态机[12](Matching State Machine,MSM)的指纹匹配算法。图 2 展示了一个具有 9 个状态(1 个测试状态,4 个未匹配状态,4 个匹配状态)的匹配状态自动机的实例。这里匹配状态机利用了文献[8]中的思想:越相似的地方存在最接近的指纹的可能性越大,而越不相似的地方存在相似指纹的可能越小。其步骤简略介绍如下:

(1)当处于未匹配态时,表示当前没有找到潜在的匹配指纹。因此,在每个状态下匹配搜索会尽力为查询视频指纹在数据库中找到一个候选匹配。

(2)当任意一个未匹配态找到一个候选匹配时,程序就进入一个中间状态——测试态。当处于测试态时,程序就会进

一步分析这个找到的候选结果是否是一个真实的匹配指纹。

(3)如果上述候选结果被证实,匹配程序就会进入匹配态(从'11'状态开始)。这里,匹配态主要负责查询视频和候选视频的连续片段的匹配以及进一步调整查询视频的匹配窗口的时间位置,以便将查询指纹序列与原始的视频检测到的指纹片段同步。在这个过程中,任何时候匹配失败,都将返回到未匹配态。图 2 比较详细地解释了匹配状态机的操作过程。

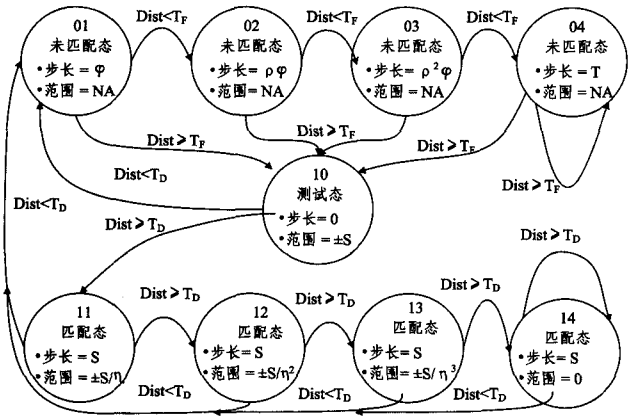
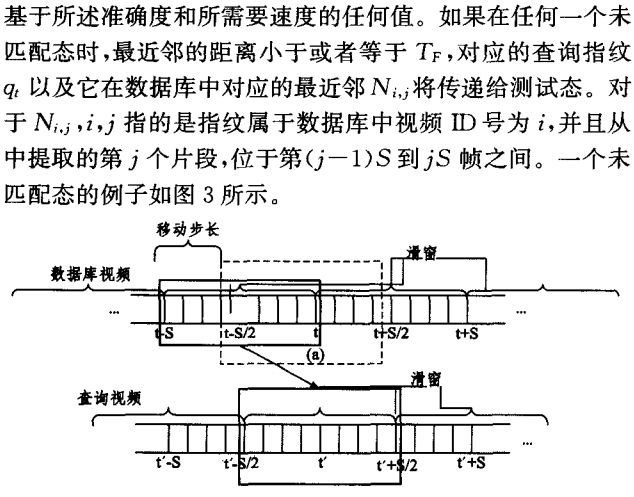


图 2　匹配状态机(MSM)的示例图

接下来对匹配状态自动机的过程进行详细介绍。

对于刚刚进入匹配过程的查询指纹序列指纹,匹配状态机通常从默认状态'01'开始。如果一个指纹处于状态'01',说明它的最近似的结果在指纹数据库中找到。为了找到最近邻,分别采用暴力搜索和上面提到的倒排索引搜索方法。最近邻搜索算法返回指纹数据库中与查询指纹的距离最近的指纹。如果最近邻的距离大于某些预设值 T_F,将进入'02'状态,检测下一个查询序列的指纹(提取 τ 秒后的窗口)。状态'02'将在这个指纹上运行最近邻搜索算法。如果最近邻的距离仍然超过 T_F,程序就进入'03'状态。此时,分析距当前指纹 $\rho\varphi(\rho>1)$的指纹。直到某些匹配找到,程序才进入更高阶的未匹配态。在每个未匹配态,步长将决定下一个要匹配的查询指纹,每次状态机调到更高的未匹配态,步长就乘以 ρ。在最后的未匹配态中,步长不再改变,同时状态保持不变。图 2 中有 4 个未匹配态,S 是最大步长,这会使得整个查询视频至少被扫描一次。然而,未匹配态的数目和最大步长可以有基于所述准确度和所需要速度的任何值。如果在任何一个未匹配态时,最近邻的距离小于或者等于 T_F,对应的查询指纹 q_t 以及它在数据库中对应的最近邻 $N_{i,j}$ 将传递给测试态。对于 $N_{i,j}$,i,j 指的是指纹属于数据库中视频 ID 号为 i,并且从中提取的第 j 个片段,位于第$(j-1)S$ 到 jS 帧之间。一个未匹配态的例子如图 3 所示。

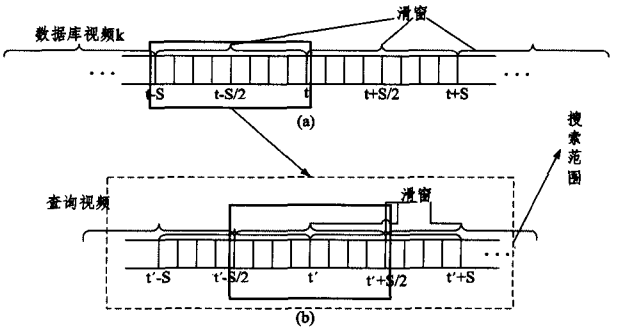


(a)实线框表示数据库中对应的指纹,虚线框表示下一步移动的位置;
(b)实线框表示查询指纹

图 3　未匹配态的滑窗示意图

在测试态时,q_t 和距离它 S 秒的近邻 $q_{t'}$($\{q_{t'}\mid t-S<t'<t+S\}$)用来与检测到的最近邻 $N_{i,j}$ 进行比较。如果与其中任意一个指纹的距离有一个小于或者等于预定阈值 T_D,与数据库中对应位置的匹配就确定了。注意,为了最小化错位对齐的量,应该使 $T_D\leqslant T_F$。现假定它发生在 t^* 时刻,之后程序会进入到一系列的匹配态。匹配态是通过原始视频(第 i 个片段)和查询视频的连续片段来扩展检测的匹配。在处理后面的延续部分时,同样也最小化了未对齐的匹配数量。

接下来,假定 q_{t^*} 和 $N_{i,j}$ 是从测试态转移到'11'状态下的查询指纹和最近邻。在'11'状态时的所有的查询指纹$\{q_{\hat{t}}\mid t^*-S/\eta<\hat{t}<t^*+S/\eta\}$,都是距 $q_{\hat{t}}$ 的 S/η。在$\{q_{\hat{t}}\}$中找到具有最小相似度距离的 $q_{t^{**}}$,如果满足距离小于或者等于 σ_D,则把其作为 $N_{i,j}$ 的一个匹配。如果 t^{**} 在状态'11',而且是在距 t^* 的 S/η^2 的范围内,同上述情况相同时,状态机就进入'12'状态;否则,它仍然处于状态'11'。不管处于哪一个匹配态,对于下一个查询比较,查询和潜在的匹配都往前移动 T 秒,接下来就检测 $q_{t^{**}+S}$ 和 $N_{i,j+1}$。图 4 给出一个匹配态的例子。



(a)实线框表示数据库中对应查找的指纹;
(b)实线框表示查询指纹,虚线框表示搜索范围

图 4　匹配态的滑窗示意图

注意,由于视频存在拷贝攻击,查询序列可能不会和对应的原始视频在时间上准确地对齐。这个搜索的时间范围会随着匹配程序进入更高的状态(如'12','13')而缩小,最终,在到达最后一个匹配态时,这个搜索的时间范围会变成 0,这时它会循环地回到自身状态(这里是'14')。匹配态的数目由准确度和所需的检索速度决定。在实验中,采用如图 2 所示的 4 个状态。所有的匹配态执行相同的任务,但是在查询指纹和参考数据库的可能匹配进行比较时使用了不同的搜索范围,这个搜索范围会随着进入更高阶的状态而减小。当查询处在匹配态时,如果具有相似度量最小距离的指纹在当前指纹的一个高阶状态的搜索范围内,状态机就会进入更高阶的状态,否则它将停留在当前状态。在任何匹配态,如果在当前范围内的所有的查询指纹有一个距离大于 T_D 或者当在检测的参考视频(编号 i)中的最后一个指纹被检测(例如这时匹配丢失了),这个程序就会回到未匹配态('01')。当检测完最后一个查询指纹时,匹配状态自动机随之停止。

3　实验结果与分析

数据集:实验数据库中的视频来自于公开的视频数据库 MUSCLE-VCD-2007[13] 和真实的网络数据集。MUSCLE-VCD-2007 数据集中共 101 个视频文件,每个视频30s~2h 不

等，总时长为80h左右。实验过程中，将数据集中的每个视频随机截取5～15min作为参考数据库中的视频，同时在网络中收集110个网络视频，并保证其内容未在视频数据库中出现过，将其作为查询视频中的负样本。然后从数据库中的每个视频中随机截取1～10min的短视频，并加入表1中的不同类型的拷贝攻击。

表1　实验使用的拷贝攻击的参数

	攻击类型	影响	Min	Max
A1	噪声(σ)	$f'(m,n,k)=f(m,n,k)+G(0,\sigma)$	0	100
A2	添加字幕(L)	添加长度L的随机文字于视频中	1	10
A3	对比度(c)	$f'(m,n,k)=f(m,n,k)+b_\mu(k)$	0.5	0.5
A4	掉帧(fd)	随机地丢掉fd%的帧	0	10
A5	添加黑边框(t,d,l,r)	上下左右分别添加长度为t,d,l,r宽度的黑边	5	20
A6	高斯模糊(θ)	高斯模板方差	5	25
A7	旋转(r)	整帧旋转r°	−5	5
A8	空间偏移(sd,sr)	图像向下偏移sd%，向右偏移sr%	−7	7

评价指标：采用TRECVID评测标准[14]，主要包含归一化检测代价率(*NCDR*)、拷贝定位精确率和平均拷贝检测时间。

(1)归一化检测代价率：是评测视频拷贝检测系统的漏检率和错报率的加权值，分别用P_{Miss}和R_{FA}表示。最小检测消耗率的计算公式如下：

$$NDCR=P_{Miss}+\beta\times R_{FA} \tag{4}$$

其中，β为损耗系数。

(2)平均检测时间T_p：即系统检测花费的平均时间，指的是从查询视频提交时刻开始到系统返回检测结果时为止的时间开销。

(3)拷贝定位精度*F-score*：是系统的查准率与查全率的调和平均值，计算公式如下：

$$F\text{-}score=\frac{2}{(1/Precision)+(1/Recall)} \tag{5}$$

3.1　指纹鲁棒性分析

为了检测更精确的结果，匹配算法采用暴力搜索得出与查询视频相似度最高的参考视频。

(1)在使用异步滑窗策略和暴力搜索结合的算法时，实验表明指纹间相似度阈值*ThD*取0.08时最佳。在选取相似度阈值时，实验结果如图5所示。当阈值高于0.095时，查准率一直保持为100%，但查全率却在较低水平。而当阈值小于0.08时，查全率上升到98%，查准率却急速下降。而阈值为0.08时，查准率和查全率分别为98.95%和93.6881%，此时的*F-score*结果最佳，为0.9625。故选取0.08作为实验检测的阈值。

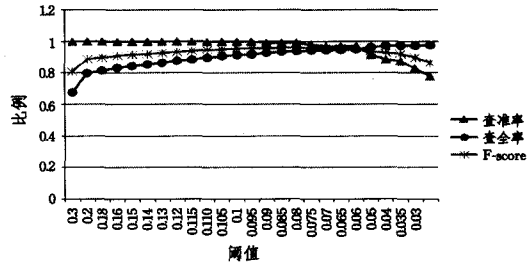


图5　在不同相似度阈值情况下，准确率、召回率以及F值的结果

(2)通过对每一种拷贝变化均进行实验比较，*NCDR*值的结果如图6所示。从实验结果可知，未采用异步滑窗的策略的*NCDR*值在各种攻击下都比较小，综合性能最佳，检测精度都高于其他方法；即使采用异步滑窗策略，检测精度虽然有所下降，但依然很高，表现出很好的性能。文献[5]将视频看作三维矩阵，并采用3D-DCT变换，提取64个低频系数作为视频指纹，在对视频内容变换方面具有很好的鲁棒性，但受掉帧这种时域攻击的影响较大，同时检测的片段是被裁切的，影响其提取片段的准确度。文献[7]采用基于断层扫描的算法虽然也具有很好的鲁棒性，但受旋转和掉帧这两种拷贝攻击的影响较大。

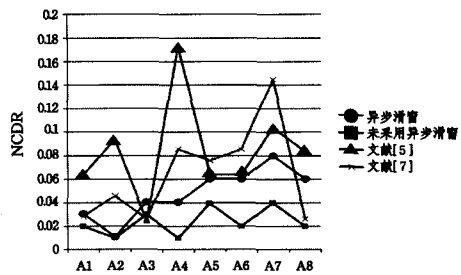


图6　各种拷贝攻击下的*NCDR*值比较

(3)定位精度(F值)、查全率和查准率值如表2所列。从表中看出，未采用异步滑窗策略的指纹提取算法和暴力搜索算法的实验结果最好，平均查全率为98.02%，平均查准率为100%，定位精度也优于其他方案的算法；而采用异步滑窗策略时，平均查全率和查准率均有一定程度的下降。

表2　各种拷贝攻击的查全率、查准率和定位精度的比较(%)

		A1	A2	A3	A4	A5	A6	A7	A8	平均值
异步滑窗	查全率	97.03	99.01	96.04	96.04	94.06	94.06	92.08	94.06	95.30
	查准率	98.99	99.01	98.98	98.98	98.96	98.96	98.94	98.96	98.97
	定位精度	98.00	99.01	97.49	97.49	96.45	96.45	95.38	96.45	97.09
未用异步滑窗	查全率	98.02	99.01	97.03	99.01	96.04	98.02	96.00	98.02	97.64
	查准率	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.00
	定位精度	99.00	99.50	98.49	99.50	97.98	99.00	97.96	99.00	98.80
文献[5]	查全率	94.06	91.09	98.02	83.17	94.06	94.06	90.10	92.08	92.08
	查准率	95.00	94.85	95.19	94.38	95.00	95.00	94.79	94.90	94.89
	定位精度	94.53	92.93	96.59	88.42	94.53	94.53	92.39	93.47	93.42
文献[7]	查全率	98.02	96.04	98.02	92.08	93.07	92.08	86.14	98.02	94.18
	查准率	90.83	93.27	93.40	93.00	93.07	93.00	92.55	93.40	92.81
	定位精度	94.29	94.63	95.65	92.54	93.07	92.54	89.23	95.65	93.45

(4)时间性能。由于匹配方案中的暴力搜索，时间复杂度为$O(NM)$，N为查询指纹个数，M为指纹数据库中的指纹个数。所以这里暂时不与其它两种算法比较时间效率，只比较使用滑窗策略和未使用滑窗策略的检测时间，包含指纹提取

时间和匹配时间，结果如表 3 所列。

表 3　滑窗策略的检测时间比较(s)

平均总时长	基于滑窗策略	未使用滑窗＋暴力搜索
11671.76	7231.97	14040.33

实验结果表明，使用异步滑窗策略的暴力搜索比未使用异步滑窗策略的搜索在时间上节省了近一半。

3.2　匹配算法性能分析

实验中，在基于倒排索引的匹配搜索方案中取每个字的长度为 $n=16$，则 126bit 的二进制指纹就有 7 个字。增加索引会使得准确度降低，为了不使准确率降低过多，倒排索引搜索不会单独和异步滑窗策略进行配合比较。

基于状态自动机的匹配搜索算法中，本文选择和算法示例图一样的状态数目的自动机：4 个 No Match 状态，1 个 Test 状态，4 个 Match 状态。实验中的参数如下：$\varphi=0.25$，$\rho=2$，$\eta=2$，$S=2$。

接下来，将实验分成以下几种方案：①倒排索引搜索＋未使用异步滑窗；②异步滑窗＋状态自动机＋暴力搜索；③异步滑窗＋倒排索引＋状态自动机匹配。分别对这几种方案和上面性能最优的未使用异步滑窗策略的暴力搜索算法的实验结果进行比较。

(1)查全率和查准率：实验比较结果如图 7、图 8 所示。结果表明，查全率均较原来的暴力搜索算法得出的查全率有一定程度的下降，但基本都在 90%以上。虽然各种方案检测出的结果略有不同，但方案 2 的查全率在各种攻击结果下都有较好的实验结果。

从查准率上看，3 种方案较未使用异步滑窗的暴力搜索算法同样也有一定程度的下降，但 3 种方案查准率的结果都相当接近，基本在 93%左右。

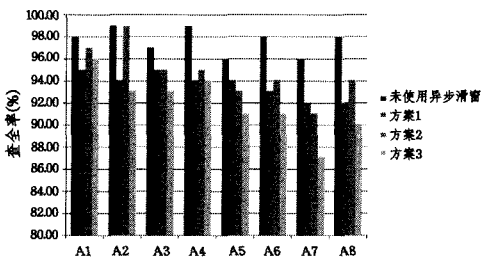


图 7　查全率比较

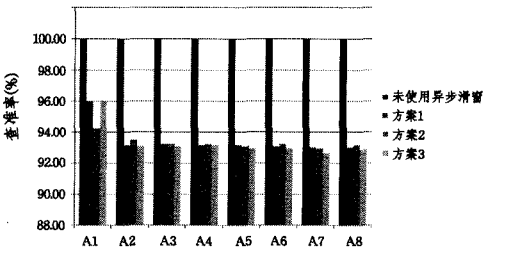


图 8　查准率比较

由图 7 和图 8 可以看出，3 种实验方案的查全率和查准率相较于暴力搜索的算法精度值都有所下降，但仍然保持在较高的精度值水平；同时可以看出方案 2 的各项在 3 种方案中综合性能比较好；方案 1 综合性能比方案 2 略低，但诙谐曲于方案 3；方案 3 的整体性能稍低。

(2)定位精度(F 值)比较：实验结果如图 9 所示，根据结果数据将本文提出的 3 种方案和对比文献中的方案进行比较分析，发现文献[5]和文献[7]对于各种拷贝攻击效果不一，特别是受掉帧的时域攻击影响较大；同时可以看出本文的 3 种方案对各种攻击变换的精度都比较稳定，而且也都在 93%左右，其中方案 2 中对各种拷贝攻击的综合整体效果最好，检测精度比较稳定，同时 F 值保持在较高的水平。

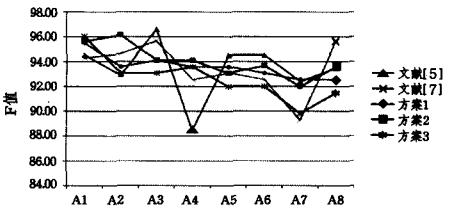


图 9　F 值比较结果

(3)时间性能比较：实验结果如表 4 所列，文献[7]由于通过二步匹配对匹配时间进行了加速，而且提取方法简单高效，搜索时间缩短了不少。从实验结果中也可以看出，方案 1 中的倒排索引的搜索算法的匹配速度是暴力搜索的 5 倍左右，而使用了状态自动机和异步滑窗策略，匹配速度明显上升，是暴力搜索的近 7 倍多，这证明基于倒排索引的匹配搜索方案和结合异步滑窗的状态自动机匹配的搜索方案可以大大缩短搜索时间。

表 4　运行时间比较

	A1	A2	A3	A4	A5	A6	A7	A8	平均值
异步窗口	7206.12	7575.89	8013.02	7117.51	7770.92	6884.5	7365.64	7231.97	7395.70
暴力搜索	14505.20	14903.8	14829.6	14053.7	14557.3	13852.95	14640.33	14040.3	14422.93
文献[7]	2905.04	3010.68	2863.24	2950.24	3045.36	2898.77	2986.52	2987.65	2955.94
方案 1	3065.45	3118.46	2534.01	2856.28	2561.15	2830.70	2986.66	3079.41	2866.52
方案 2	2206.12	2075.89	2101.3	2117.51	2170.92	2084.5	2037.49	2123.19	2114.62
方案 3	772.76	886.04	814.35	872.38	890.48	869.21	906.87	943.48	869.45

结束语　为了提高对各种拷贝攻击的鲁棒性，本文提出了一种基于时空域信息融合的指纹提取算法；为了提高搜索速度和定位精度，给出了基于倒排索引的匹配搜索算法和结合异步滑窗策略的基于匹配状态机的匹配搜索算法。从实验结果可以看出，基于时空域信息融合的指纹提取算法，提取时间较少，而且相比于其他指纹提取方法步骤不繁琐，同时对于各种拷贝攻击具有很好的鲁棒性。同时提出的基于滑窗策略和状态自动机结合，以及基于倒排索引的指纹搜索算法得到的检测结果都较好，检测精度范围的平均 F 值都达到了 0.92 以上，而且对各种拷贝攻击都有较好的抵抗性，同时在搜索时间上也有较大的改进。

参 考 文 献

[1] Hampapur A, Bolle R M. Comparison of distance measures for video copy detection[C]// Proc. IEEE Int. Conf. Multimedia and Expo (ICME). 2001:737-740

(下转第 314 页)

将进一步优化背景模型自适应的方法，以减少计算的复杂度，提高系统的实时性及准确性。

参 考 文 献

[1] Wang D L, Brown G J. Computational Auditory Scene Analysis: Principles, Algorithms, and Applications [M]. Wiley-IEEE Press, 2006

[2] Espi M, Fujimoto M, Kinoshita K, et al. Feature Extraction Strategies in Deep Learning Based Acoustic Event Detection [C] // INTERSPEECH. 2015: 2922-2926

[3] Plinge A, Grzeszick R, Fink G A. A Bag-of-Features Approach to Acoustic Event Detection [C] // 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2014: 3704-3708

[4] Phan H, Maab M, Mazur R, et al. Random Regression Forests for Acoustic Event Detection and Classification [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(1): 20-31

[5] Parascandolo G, Huttunen H, Virtanen T. Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings [C] // 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016: 6440-6444

[6] Lim H, Kim M J, Kim H. Cross-Acoustic Transfer Learning for Sound Event Classification [M] // 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016: 2504-2508

[7] Atrey P K, Maddage N C, Kankanhalli M S. Audio based Event Detection for Multimedia Surveillance [C] // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2006: 813-816

[8] Zhuang X, Zhou X, Hasegawa-Hohnson M, et al. Real-world Acoustic Event Detection [J]. Pattern Recognition Letter, 2010, 31(12): 1543-1551

[9] Zhang A Y. Using Hierarchical Method to Improve Real Time for Audio-based Surveillance System [C] // International Symposim on Chinese Spoken Language Processing (ISCSLP). 2014: 570-573

[10] Rabaoui A, Davy M, Rossignol S, et al. Using One-Class SVM and Wavelets for Audio Surveillance [J]. IEEE Trans. on Information Forensics and Security, 2008, 3(4): 763-775

[11] Angel D L T, Peinado A M, Segura J C, et al. Histogram Equalization of Speech Representation for Robust Speech Recognition [J]. IEEE Trans. on Speech and Audio Processing, 2005, 13(3): 355-366

[12] Lee C H, Lin C H, Juang B H. A Study on Speaker Adaptation of Parameters of Continuous Density Hidden Markov Models [J]. IEEE Trans. on Signal Processing, 1991, 39(4): 806-814

[13] Kaltenmeier A, Regel P, Trottler K. Fast Speaker Adaptation for Speech Recognition Systems [C] // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 1990: 133-136

[14] Legetter C, Woodland P. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models [J]. Computer Speech and Language, 1995, 9(2): 171-185

[15] Povey D, Yao K. A Basis Method of Robust Estimation of Constrained MLLR [C] // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2011: 4460-4463

[16] Ntalampiras S, Potamitis I, Fakotakis N. An Adaptive Framework for Acoustic Monitoring of Potential Hazards [J]. EURASIP Journal on Audio, Speech, and Music Processing, 2009, 10: 1-15

[17] Kryze D, Rigazio L, Junqua J C. A New Noise-Robust Subband Front-End and Its Comparison To PLP [J]. J. Chem. educ, 2000, 31(15): 269

[18] Young S, Evermann G, Gales M, et al. The HTK Book [OL]. http://htk.eng.cam.ac.uk/docs/docs.shtml

---

(上接第 279 页)

[2] Lee S, Yoo C D. Robust video fingerprinting for content-based video identification[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2008, 18(7): 983-988

[3] Indyk P, Iyengar G, Shivakumar N. Finding pirated video sequences on the internet: Technical report[R]. Stanford University, 1999

[4] Chen L, Stentiford F W M. Video sequence matching based on temporal ordinal measurement[J]. Pattern Recognition Letters, 2008, 29(13): 1824-1831

[5] Coskun B, Sankur B, Memon N. Spatio-temporal transform based video hashing [J]. IEEE Transactions on Multimedia, 2006, 8(6): 1190-1208

[6] Leon G, Kalva H, Furht B. Videof identification using video tomography[C] // IEEE International Conference on Multimedia and Expo. IEEE, 2009: 1030-1033

[7] Ji Qing-ge, Tan Zhi-feng, Lu Zhe-ming, et al. An Improved Video Identification Scheme Based on Video Tomography [J]. IEICE Transactions on Information and Systems, 2014, 97(4): 919-927

[8] Oostveen J, Kalker T, Haitsma J. Feature extraction and a database strategy for video fingerprinting [M] // Recent Advances in Visual Information Systems. Springer Berlin Heidelberg, 2002: 117-128

[9] Zhao Wan-Lei, Ngo Chong-Wah, Tan Hung-Khoon, et al. Near-duplicate keyframe identification with interest point matching and pattern learning [J]. IEEE Transactions on Multimedia, 2007, 9(5): 1037-1048

[10] Barrios J M, Bustos B. Competitive content-based video copy detection using global descriptors[J]. Multimedia tools and applications, 2013, 62(1): 75-110

[11] Esmaeili M M, Ward R K. Robust video hashing based on temporally informative representative images [C] // International Conference on Consumer Electronics (ICCE). 2010: 179-180

[12] Malek Esmaeili M, Ward R K, Fatourechi M. Fast matching for video/audio fingerprinting algorithms[C] // IEEE International Workshop on Information Forensics and Security (WIFS). 2011: 1-6

[13] MUSCLE-VCD-2007 [OL]. http://wwwrocq.inria.fr/imedia/civr-bench/index.html

[14] Awad G, Over P, Kraaij W. Content-based video copy detection benchmarking at TRECVID[J]. ACM Transactions on Information Systems, 2014, 32(3): 1-40