基于运动矢量的抗几何攻击视频水印方案

戴侃斐[1] 陈真勇[2] 唐 龙[1]

[1](清华大学计算机科学与技术系，北京 100084) [2](北京航空航天大学计算机学院，北京 100083)

摘 要 许多数字图像和视频水印方案都无法抵挡几何变换攻击，即使是轻微的旋转、缩放、平移也将会破坏水印的同步，这就使得水印难以检测出来。为解决此难题，提出了一种新颖的视频水印算法。该算法利用了运动矢量特征的相对关系不易随几何失真变换而改变的特性，使得水印具有内在的抗几何失真的鲁棒性。该算法中，水印嵌入和检测时，不仅可根据平均运动矢量的方向和幅度确定水印模板的位置和方向，而且水印检测不需要原始视频。实验表明，该水印算法对几何变换攻击具有较强的鲁棒性。

关键词 视频水印 运动矢量 几何攻击

中图法分类号：TP309 **文献标识码**：A **文章编号**：1006-8961(2007)09-1682-06

Motion Vector Based Video Watermarking Scheme Against Geometric Attacks

DAI Kan-fei[1], CHEN Zhen-yong[2], TANG Long[1]

[1](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

[2](*School of Computer Science and Engineering, Beihang University, Beijing* 100083)

Abstract For many digital image and video watermarks, geometric distortions are one of the weaknesses. Even slight rotation, scaling and translation can destroy the synchronization and invalidate the detection of blind watermarks. In this paper, we present a novel video watermarking scheme that inserts a watermark intrinsically resistant to these distortions by exploiting the characteristics of the motion vector. We find that the relative relationship between characteristics of the motion vector cannot be altered with the geometric distortions easily. So the position and direction of watermark are determined according to the average values of the magnitude and direction of motion vectors. In addition, another advantage of the novel watermarking scheme is that the original video is not needed in watermark detection process. Experimental results show that the scheme is rather robust to the common geometric distortions including rotation, scaling and cutting.

Keywords video watermarking, motion vector, geometric attack

1 引 言

在原始数字产品中嵌入水印信息是一种可用来证明产品所有权的方法，但是在实际应用环境中，几何攻击很容易使水印失效。因此，如何实现使含水印的媒体(如图像、视频、文本、语音等数字媒体)具有抗几何变换攻击的鲁棒性，是目前数字水印技术急需解决的问题之一。

一种简单的解决办法是，用穷举搜索的方法来检查所有可能的几何失真。然而，由于几何失真变换的可能性太多，难以完全穷举，所以，在不知道嵌入信息的情况下，试图用基于调制类型的穷举法来搜索水印，基本上也是不可能的[1]。即使不考虑时间代价，穷举搜索法也容易产生虚警[2]。在图像水印抗几何攻击方面，如今已经提出了多个针对仿射变换(旋转和缩放)的抵抗方法。例如，选择一种具有旋转和比例变换不变性的水印形状[3]，由于不存在能够抵抗所有明显几何失真的水印形状，因此，这种方法还需要穷举搜索的补充。基于 Fourier-Mellin

基金项目：国家自然科学基金重点项目(60133020)

收稿日期：2005-06-20；改回日期：2005-12-22

第一作者简介：戴侃斐(1980～)，女。硕士研究生。主要研究领域为计算机视觉、信息隐藏及数字水印等。E-mail：tanglong@tsinghua.edu.cn

变换的旋转和比例变换不变域数字水印方法[1]仅仅对未被剪切的均匀缩放、角度旋转和平移等几何变换是有效的,但是,该方法不仅处理时间较长,而且它所用的指数-极坐标非线性映射会引起含水印图像的质量下降。此外,还有一种抵抗图像水印几何失真的方法,那就是在图像(帧)嵌入水印之前,预先嵌入一个 2 维模板(参考标志/同步模板)[4]。通常只有具备足够能量的模板,图像才能在几何失真后恢复过来,可是强能量模板的嵌入,也会对图像造成更大的改变,从而影响图像质量。

Haitsma 和 Yao 等提出了抗几何攻击的视频水印方案,即水印沿时间轴分布,并且每帧仅改变其亮度平均值[5,6]。该算法是利用平均亮度本身不受旋转和缩放等仿射变换影响的特点来抵抗几何攻击。但是,由于人眼对亮度分量比较敏感,而且当水印嵌入数目增多,或者嵌入强度增大时,视频的闪烁现象会比较明显,因此,这个方案水印嵌入的数目和强度都是比较有限的。Wessely 分析了运动矢量的特点后指出,利用运动矢量有助于视频水印在检测时的同步恢复[7],但未结合视频水印方案给出实例及验证。为此,本文在此基础上,提出了一种基于运动矢量的抗几何攻击视频水印方案,在该方案中,水印的检测不需要原始视频。

2　运动矢量与水印同步

对于视频流,其最基本的特征是对象或场景在时间维度上的变化,即使对视频进行处理,这种特征的相对关系也不易被改变。因此,可以尝试利用运动矢量来获得视频流几何失真后的水印同步的可能性。具体分析如下:

(1) 可以假设视频流序列中,在很小的时间间隔内,各个帧受到的几何变换是相同的。例如,一个图像组(group of picture, GOP)中,在 0.5s 的时间间隔内,有 12 ~ 15 帧图像;又如,根据欧洲广播联盟(european broadcasting union, EBU)技术规范,一个水印最小段(watermarking minimum segment, WMS)为 1s 的时间间隔,大约有 24 ~ 30 帧图像。根据以上前提,可以假设一个 GOP/WMS 内,每个帧内的像素沿着时间轴上受到的几何变换是基本一致的[8]。

(2) 由于在一般的视频流中,总是存在对象的运动或者场景和摄像机视点的变化,因此,有可能找到足够的大幅度运动矢量,用来确定水印在视频图像中的位置和方向。即使场景变化较小,也还可以采用之前得到的运动信息来估算水印的位置和方向参数。

(3) 攻击者要想对一段较长时间的视频流进行处理或者攻击,以改变其运动矢量,而又不想引起场景的明显变化几乎是不可能的。

(4) 由于一些常见的非几何攻击(如数字滤波、噪声和帧内的数据压缩)都不会改变视频对象的运动特征或者运动轨迹,因而,也不会对视频图像的运动矢量产生明显的影响。当一段视频流受到平移、旋转或者缩放操作等攻击时,运动矢量也会遵循这种变换。

综上所述,对于几何失真后的水印检测,可借助运动矢量恢复出原始的同步信息。

3　水印的嵌入与检测

3.1　求取平均运动矢量

在水印嵌入与检测之前,先要计算视频流中相邻帧之间的运动矢量(motion vector, MV),嵌入端和检测端采用同样的算法。这可以用传统的块匹配方法来实现,也就是在一个邻域内查找最接近的块。

然后根据每一帧中提取出来的运动矢量来计算得到第 t 帧内的平均运动矢量 $\overline{V}_t^{\text{motion}}$。由于运动矢量的坐标是离散化的,幅度越大的运动矢量,其精度也就越高,因此,在计算 $\overline{V}_t^{\text{motion}}$ 时,幅度大的运动矢量应赋予较高的权重。为得到一个更稳定的计算结果,$\overline{V}_t^{\text{motion}}$ 可以沿时间轴做平均来得到一段视频帧在时间轴上的平均运动矢量 $\overline{V}^{\text{motion}}$。为了防止由于裁剪可能带来的误差,可只选择位于图像中心的运动矢量来计算平均值。否则,会由于裁剪后,边缘区域的丢失而产生较大的影响。一般来说,即使视频帧受到攻击或者改变,图像的中心部分仍然还是存在的。

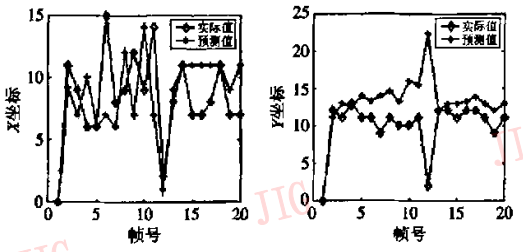
从计算结果的准确性来考虑,对于用于水印同步的运动矢量,应该是真实的运动量,而不能高于设定的一个阈值,提早结束搜索。本文采用了MPEG-4的运动估计算法来求取平均运动矢量,其原因是:

(1) 获取真实运动矢量的时间复杂度巨大,这种巨大的时间复杂度对于本算法是不经济的;

(2) 算法中提取的参量是运动矢量的统计值(帧间平均运动矢量),而不是单个块的运动矢量。

这种采用MPEG-4的运动估计算法所得到的运动矢量,是统计的计算结果,其虽与实际值存在一些偏差,但这种偏差还是可以接受的,后面的实验也证明了这一点。

文献[7]提到先寻找一个与平均运动矢量的方向最接近的矢量,然后用与该矢量对应的块所在的位置和运动矢量的方向作为水印的基线和方向。然而,实验表明,由于单独块的运动矢量不是一个稳定变化的量,会受到各种几何攻击而发生非平衡的改变,因此,用这种方法来确定水印的基线是不可靠的。如图 1 所示(测试视频为 CIF 格式的视频流 Foreman,视频长度为 20 帧,所受到的几何攻击为旋转 5°),从块的坐标可以看出,所查找到的块与预测块的位置有较大的偏差。因此在本文算法中,不将特定的块作为水印的基线。



(a) X 坐标 (b) Y 坐标

图 1 Foreman 视频的块位置偏差

Fig. 1 Deviation of block position(in Foreman)

用平均运动矢量的方向确定了水印的方向后,还要确定水印模板的 1 维同步信息。这就需要计算两个参数,一个是缩放因子 α,另一个是位置补偿。如果把水印嵌入到一个平移不变的域中,比如说傅里叶变换系数的幅度就只需要检测缩放因子 α,这也就会简单得多了。不同于文献[7]采用的穷举搜索方法,本文用平均运动矢量的幅度值来确定图像的缩放因子 α。水印模板 $M_{watermark}$ 的大小 L 直接与平均矢量的幅度 m 成比例。这样,即使不知道图像的缩放比例,也不会影响水印的提取。

3.2 有关函数和变量的定义

在讨论水印算法之前,对算法中涉及到的函数及变量先做一些必要的定义。

假设原始视频序列为

$$\boldsymbol{F}=\{f(x,y,t)\quad x\in[0,W],y\in[0,H],t\in[0,K]\}$$

其中,W 为视频图像的宽度,H 为视频图像的高度,K 为视频序列帧数。在算法中,第 t 帧视频图像记为 $\boldsymbol{F}_t$。

要嵌入的水印信息为 $\boldsymbol{W}=\{w[n]\in\{1,-1\},1\leqslant n\leqslant N)\}$。$N$ 为水印的位数,即长度。在算法中,$N=K$。

嵌入水印后的视频序列为

$$\hat{\boldsymbol{F}}=\{\hat{f}(x,y,t),x\in[0,W],y\in[0,H],t\in[0,K]\}$$

嵌入水印后的第 t 帧视频图像记为 $\hat{\boldsymbol{F}}_t$。

算法使用的水印模板为方向平行于 $\overline{\boldsymbol{V}}^{motion}$、边长等比例于 $\overline{\boldsymbol{V}}^{motion}$、幅度为 m 的正方形(如图 2 所示)。

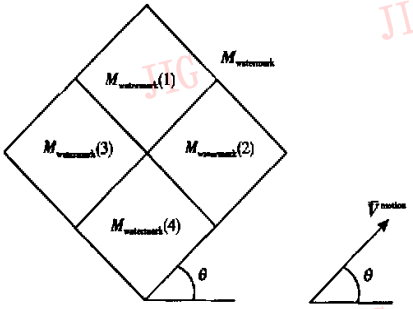


图 2 水印模板定义

Fig. 2 Watermark template

3.3 水印嵌入算法

水印嵌入算法的具体过程如下:

(1) 读入 n 帧视频图像 $\boldsymbol{F}_t,n\geqslant 1$;

(2) 在每一帧视频图像中心区域,选取幅度较大的 $k(k\geqslant 1)$ 个运动矢量,并计算得到其帧内平均运动矢量 $\overline{\boldsymbol{V}}_t^{motion}$,对 n 帧视频图像的 $\overline{\boldsymbol{V}}_t^{motion}$ 在时间轴上做平均,得到 n 帧视频图像在时间轴上的平均运动矢量 $\overline{\boldsymbol{V}}^{motion}$,$(1\leqslant m\leqslant L/n)$;

(3) 根据 $\overline{\boldsymbol{V}}^{motion}$ 确定每一帧水印模板 $M_{watermark}$ 的大小和方向。

(4) 每一帧嵌入 1bit 水印:$\hat{\boldsymbol{F}}_t=\Phi(\boldsymbol{F}_t,M_{watermark}[n],\overline{\boldsymbol{V}}^{motion})$。$\Phi()$ 为水印嵌入函数。在实验中,先根据 $\overline{\boldsymbol{V}}^{motion}$ 确定水印模板 $M_{watermark}$,然后将 $M_{watermark}$ 等分成 4 个正方形块。对每个块相应位置上的像素做 DCT 变换,根据要嵌入的水印值 $M_{watermark}[n]$ 修改每个块的 DC 系数。为了不改变嵌入水印后的平均运动矢量,实验中,水印是嵌入在视频图像的色度(U)分量中的。

重复步骤(1)~(4),直到嵌入所有的水印。

3.4 水印检测算法

检测算法是嵌入算法的逆过程。假设 $\boldsymbol{G}$ 表示受到几何攻击后的视频流。

(1) 读入 n 帧受到几何攻击的视频图像序列 $\boldsymbol{G}_t,n\geqslant 1$;

(2) 用同样的方法计算 n 帧视频图像的平均运动矢量 $\overline{\boldsymbol{V}}^{motion}$,$(1\leqslant m\leqslant L/n)$;

(3) 根据 $\overline{\boldsymbol{V}}^{motion}$ 确定每一帧水印模板的大小和方向;

(4) 每一帧检测 1bit 水印:$W_t=\Psi(\boldsymbol{G}_t,\overline{\boldsymbol{V}}^{motion})$。

$\Psi(\cdot)$表示检测函数,实际上是嵌入函数 $\Phi(\cdot)$的逆过程。在实验中,先根据 $\overline{V}^{motion}$确定水印模板 $M_{watermark}$的大小和方向,然后将 $M_{watermark}$等分成 4 个正方形块,每个块做 DCT 变换,最后根据 DC 系数的相对大小即可得到该帧所嵌入的水印信息 W_i。

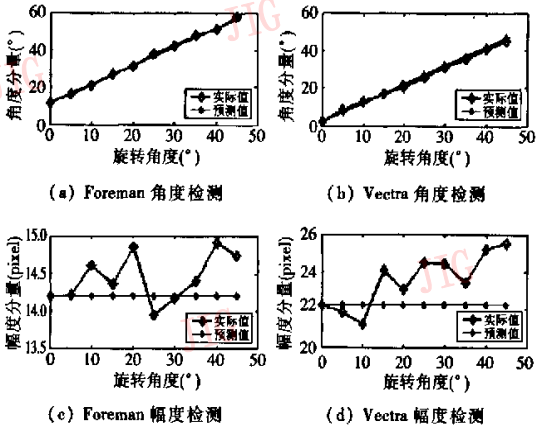
重复步骤(1)~(4),直到提取出所有的水印。

4　实验结果与分析

4.1　旋转和裁剪

选取 352×288 的 CIF 格式的 Foreman 和 Vectra 视频流来进行实验。实验时先对视频分别进行旋转、缩放以及裁剪攻击,并且采用平均运动矢量作为水印嵌入的位置和方向的标志信息。Foreman 视频流的长度为 200 帧,Vectra 视频流长度为 140 帧,每一帧嵌入 1bit 水印信息。

平均运动矢量的幅度分量和角度分量随旋转角度变化而变化的曲线以及和预测值的比较如图 3 所示(横坐标 X,表示视频帧旋转的角度大小,单位是°),图 3(a)和图 3(b)展示的是平均运动矢量角度分量的变化规律;图 3(c)和图 3(d)展示的是平均运动矢量幅度分量的变化规律。其中,所谓的预测值是在理想情况下,根据所给的几何变换参数而得到的运动矢量的角度分量和幅度分量的理论值。



(a) Foreman 角度检测　(b) Vectra 角度检测

(c) Foreman 幅度检测　(d) Vectra 幅度检测

图 3　对几何攻击(旋转和裁剪)后的视频,检测平均运动矢量

Fig. 3　Examining average motion vector in video after geometric attacks(rotation and cropping)

从图 3 可以看出,平均运动矢量基本上反映了图像旋转角度的变化,而幅度值虽然有上下波动,但波动范围都在 1~2pixels 之内。

表 1 所示的水印检测结果显示,其可以作为水印嵌入或者检测位置和方向的参考信息。嵌入水印后受到旋转和裁剪攻击的视频图像两个实例如图 4 所示。

表 1　旋转和裁剪攻击的水印检测结果

Tab. 1　Watermark detection percentage after rotation and cropping attacks

旋转角度(°)	水印检测率(%)	
	Foreman	Vectra
0	100	100
5	100	100
10	100	100
15	100	100
20	100	100
25	100	100
30	100	100
35	100	100
40	100	100
45	100	100

(a) Foreman 第 4 帧(旋转 30°)　(b) Vectra 第 5 帧(旋转 15°)

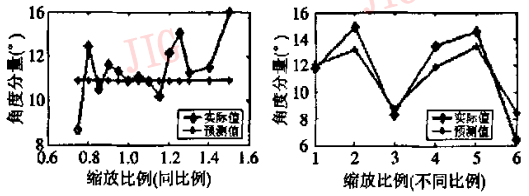
图 4　嵌入水印后视频流的实例

Fig. 4　Watermarked frames(Foreman 4th frame rotated by 30°, Vectra 5th frame rotated 15°)

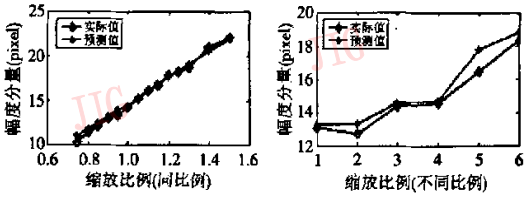
4.2　缩放和裁剪

在缩放实验中,采用的视频流分别为 CIF(352×288)和 SIF(352×240)两种格式,视频流的长度为 200 帧,同样的,每一帧嵌入 1bit 的水印信息。当图像缩放比例大于 1 时,需对图像进行裁剪操作,以使得图像的大小和原始图像保持一致。

平均运动矢量的幅度和角度,一个随 X 轴和 Y 轴缩放比例变化而变化的曲线以及和预测值的比较实例如图 5 所示(以 Foreman 视频为例)。图中横坐标表示视频图像的相对于原始图像的缩放比例值,如同比例缩放中,0.8 表示图像的长度和宽度都缩小到原图像的 0.8 倍。其中,不同比例缩放中,缩放比例(X:Y)分别为(1)0.9:1.0;(2)0.9:1.1;(3)1.0:0.8;(4)1.0:1.1;(5)1.2:1.5;(6)1.3:1.0)。图 5(a)展示的是平均运动矢量角度分量的变化规律。图 5(b)展示的是平均运动矢量幅度分量的变化规律。



(a) 角度检测结果



(b) 幅度检测结果

图 5 缩放和裁减的几何攻击后的 Foreman 图像检测结果

Fig. 5 Examining average motion vector in video(Foreman) after geometric attacks(scaling and cropping)

从图 5 可以看出,平均运动矢量的幅度变化基本上反映了缩放比例的变化,而角度的波动也在 2° 范围内。根据平均运动矢量的方向和幅度检测水印的结果如表 2 所示。嵌入水印后受到缩放和裁剪攻击的视频图像如图 6 所示。

表 2 缩放和裁剪攻击后的水印检测结果

Tab. 2 Watermark detection percentage after scling and cropping attacks

缩放比例	测试视频水印检测率(%)			
	Foreman (CIF)	Vectra (CIF)	Coastguard (CIF)	Stefan (SIF)
0.75:0.75	100	95	100	90
0.80:0.80	100	96	100	96
0.85:0.85	100	95	100	90
0.90:0.90	100	97	100	98
0.95:0.95	100	99	100	97
1.00:1.00	100	100	100	100
1.05:1.05	100	97	100	100
1.10:1.10	100	97	100	100
1.15:1.15	100	90	100	100
1.20:1.20	100	85	100	100
1.25:1.25	100	85	100	100
1.30:1.30	100	85	100	100
1.40:1.40	100	82	100	100
1.50:1.50	100	80	100	100
1.00:1.10	100	98	100	96
1.20:1.50	100	84	100	97
1.30:1.00	96	87	100	84
1.00:0.80	100	94	100	85
0.90:1.00	100	99	100	98
0.90:1.10	100	98	100	93

(a) Foreman (第 4 帧, $X:Y=1.4:1.4$)

(b) Vectra (第 3 帧, $X:Y=0.8:0.8$)

(c) Coastguard (第 2 帧, $X:Y=1.0:0.8$)

(d) Stefan (第 10 帧, $X:Y=1.2:1.5$)

图 6 嵌入水印并经缩放和裁剪攻击后视频图像实例

Fig. 6 Watermarked frames(scaling and cropping attacks)

4.3 性能分析

虽然,检测出的平均运动矢量和预测值之间存在微小偏差,但由于其对水印位置和水印模板大小的影响基本上在 1 ~ 2pixels 范围内,所以,即使对水印模板的匹配产生了影响,也可以用局部搜索的方法快速地检测出水印。

从实验结果可以看出,本文所提出的基于运动矢量的水印算法能够有效抵抗旋转和缩放等几何攻击,特别是在旋转攻击下,水印都能正确检测。但也存在以下两种情况会使得水印检测出现比较大的误码率:其一,X/Y 方向缩放比例严重不等时(如 X 轴放大 1.3 倍,Y 轴保持不变)平均运动矢量的角度误差较大;其二,平均运动矢量的幅度值比较大,虽然其幅度值的相对误差基本不变,但绝对误差增大,而本文算法只是做了 1pixel 范围内的相关检测,这样也就会造成相应的比较大的误码率。即使这样,正确检测率依然在 80% 以上。

5 结 论

本文在文献[7]的运动矢量分析的基础上,提出了一种基于运动矢量的获取同步视频水印的新方法。运动矢量是视频流一个鲁棒的本质特征,它能够可靠地反映图像所受到的几何攻击。实验结果表明,该方法能够有效抵抗几何同步攻击。同时,由于

本文提出的同步方法独立于具体的水印算法,所以,它可以应用到不同水印编码系统中,以提高视频流抵抗几何攻击的鲁棒性。

参考文献(References)

1 Ruanaidh J J K Ó, Pereira S. A secure robust digital image watermark[A]. In: Proceedings of SPIE Europto Conference on Electronic Imaging[C], Zurich, Switzerland, 1998, **3409**: 150 ~ 163.

2 Lichtenauer J, Setyawan I, Kalker T, *et al*. Exhaustive geometrical search and the false positive watermark detection probability[A]. In: Proceedings of SPIE Security and Watermarking of Multimedia Contents V[C], Santa Clara, CA, USA, 2003, **5020**: 158 ~ 161.

3 Fletcher P A, Larkin K G. Direct embedding and detection of RST invariant watermarks [A]. In: Proceedings of 5th International Workshop on Information Hiding, (IH) 2002[C], Noordwijkerhout, The Netherlands, 2003: 129 ~ 144.

4 Kutter M, Bhattacharjee S K, Ebrahimi T. Towards second generation watermarking schemes [A]. In: Proceedings of 6th International Conference on Image Processing ICIP'99[C], Kobe, Japan, 1999, **1**: 320 ~ 323.

5 Haitsma J, Kalker T. A watermarking scheme for digital cinema[A]. In: Proceedings of International Conference for Image Processing [C], Thessaloniki, 2001, **2**: 487 ~ 489.

6 Yao Z, Reginald R L. Video watermarking scheme resistant to geometric attacks[A]. In: Proceedings of International Conference for Image Processing[C], Rochester, NY, USA, 2002, **2**: 145 ~ 148.

7 Wessely U. Synchronization of video watermarks for oblivious detection after geometrical distortions[A]. In: Virtual Goods 2004 [EB/OL]. http://virtualgoods. tu-ilmenau. de/2004/wmsync-VG04. pdf.

8 Niu Xia-mu, Schmucker M, Busch C. Video watermarking based on time-axis template[J]. Acta Electronica Sinica, 2004, **32**(8): 1236 ~ 1238.[牛夏牧, Schmucker M, Busch C. 基于时间轴上模板的动态图像数字水印处理技术[J]. 电子学报, 2004, **32**(8): 1236 ~ 1238.]