doi:10.3788/gzxb20124107.0859

自嵌入双功能图像水印算法

叶天语
(浙江工商大学 信息与电子工程学院,杭州 310018)

摘 要:利用自嵌入技术提出一种同时实现版权保护和内容认证的双功能图像水印算法.在嵌入端,首先将原始图像划分成互不重叠的子块,将各子块每个像素的最低 m 位置 0,然后将最低 m 位置 0 后的子块进行奇异值分解,通过提取奇异值范数的最高位奇偶性产生原始鲁棒特征水印,然后再将原始鲁棒特征水印嵌入各子块每个像素的最低 m 位得到含水印图像.检测端从攻击的含水印图像提取鲁棒特征水印的过程与嵌入端产生原始鲁棒特征水印的过程类似,并且通过计算提取的鲁棒特征水印与原始鲁棒特征水印之间的归一化相关度进行版权鉴别,通过判断提取的鲁棒特征水印与攻击图像各子块每个像素的最低 m 位的一致性实现篡改检测进行内容认证.理论分析和实验结果都表明算法具有非常好的不可见性.实验结果还表明,算法不仅在抵抗添加噪音、剪切、JPEG 压缩、平滑、重采样和几何攻击如随机删除行、向右偏移列、向下偏移行表现出很强的鲁棒性,而且能够精确定位出篡改位置和区分篡改类型.因此,算法具有版权保护和内容认证双重功能.

关键词:数字水印;自嵌入;版权保护;内容认证;篡改检测;恶意篡改
中图分类号:TN911.7 **文献标识码**:A **文章编号**:1004-4213(2012)07-0859-9

0 引言

根据数字水印的鲁棒性,数字水印算法可以分为鲁棒数字水印算法、脆弱数字水印算法和半脆弱数字水印算法.鲁棒图像水印算法[1-11]用于对原始图像进行版权保护;脆弱图像水印算法[12-15]用于对原始图像进行内容认证;半脆弱图像水印算法[16-17]用于在某种程度上同时实现版权保护和内容认证,但往往很难在只嵌入一个半脆弱水印就同时具有很强的抗攻击鲁棒性和很敏感的篡改检测脆弱性.零水印算法[1-3]是一类特殊的鲁棒图像水印算法,在不改动原始图像的前提下提取其稳定特征产生鲁棒零水印实现版权保护.自嵌入脆弱图像水印算法[12-14]是一类特殊的脆弱图像水印算法,其原理是提取原始图像的特征产生脆弱水印然后自嵌入到原始图像,而不是将外在的水印嵌入到原始图像,其特殊之处就在于利用了自嵌入技术.

现有图像水印算法往往存在功能单一的局限性.为了使一个图像水印算法能同时具备版权保护和内容认证双重功能,可以在原始图像中嵌入鲁棒水印和脆弱水印两个水印.脆弱水印因其脆弱性容易遭破坏,所以两个水印的嵌入顺序一般是先嵌入鲁棒水印后嵌入脆弱水印.如果两个水印都是外在的水印,那么一旦嵌入强度控制不好,含两个水印的图像的不可见性可能会比较差.本文利用自嵌入技术可以在只嵌入一个水印就使水印算法同时具备版权保护和内容认证双重功能,其思路是提取原始图像稳定特征产生鲁棒特征水印,然后再将鲁棒特征水印自嵌入到原始图像的最低 m 位使之又具有脆弱性,这样算法就能同时实现版权保护和内容认证,而且又具有非常好的不可见性.鲁棒特征水印通过判断最低 m 位置 0 后的子块奇异值范数的最高位奇偶性产生.版权保护通过计算提取的鲁棒特征水印与原始鲁棒特征水印之间的归一化相关度(Normalized Correlation, NC)来实现,内容认证通过判断提取的鲁棒特征水印与攻击图像各子块每个像素的最低 m 位的一致性来实现.然而,现有的一些自嵌入脆弱图像水印算法[12-14]仅仅具有脆弱性,只具有内容认证的功能.相比之下,本文算法的特点和优势在于只嵌入一个水印就同时具有鲁棒性和脆弱性,能够同时实现版权保护和内容认证双重功能.

1 算法流程

1.1 原始鲁棒特征水印产生及自嵌入

原始图像 OI 的大小设为 $N\times N$.嵌入端原始鲁棒特征水印 W 的产生及自嵌入过程分解如下:

基金项目:浙江省教育厅项目(No. Y201017916)资助
第一作者:叶天语(1982—),男,讲师,博士,主要研究方向为信息隐藏与数字水印.Email: flystu008@yahoo.com.cn
收稿日期:2011-12-01;**修回日期**:2012-02-22

Step1：将 OI 分割成互不重叠的 $n\times n$ 子块，各子块记为 OB_k，k 代表子块的序号，且 $k=1,2,\cdots,(N/n)^2$.

Step2：OB_k 的每个像素的最低 m 位置 0，将得到的各子块记为 OB_k'. 置位过程为

$$OB_k'=\text{bitset}(OB_k,j,0) \tag{1}$$

式中 bitset(·)为置位函数，$j=1,2,\cdots,m$.

Step3：对 OB_k' 进行奇异值分解(Singular Value Decomposition, SVD)，产生的奇异值记为 δ_k^i，i 代表奇异值的序号，$i=1,2,\cdots,n$.

Step4：计算 OB_k' 的奇异值范数，记为 $Norm_k$，即

$$Norm_k=\sqrt{\sum_{i=1}^{n}(\delta_k^i)^2} \tag{2}$$

Step5：通过提取 $Norm_k$ 的最高位奇偶性产生原始鲁棒特征水印 W. 如果 $Norm_k$ 的最高位数字为偶数，则 $W_k=0$；否则，$W_k=1$，W_k 为 W 的第 k bit. 例如，如果 $Norm_k=78$，其最高位是 7，为奇数，则此时 $W_k=1$. 原始鲁棒特征水印 W 保存在认证中心用于版权鉴别和内容认证.

Step6：将 W_k 自嵌入 OB_k' 每个像素的最低 m 位得到含水印子块 OB_k''，OB_k'' 重组后得到含水印图像 OI′. 自嵌入过程为

$$OB_k''=\text{bitget}(OB_k',j,W_k) \tag{3}$$

由上述过程可知，W 的长度为 $(N/n)^2$ bit；每比特 W_k 被自嵌入到 n^2 个像素，每个像素被嵌入 m 位，那么每比特 W_k 总共被自嵌入 mn^2 次. 因此，算法的水印嵌入容量为 mN^2. W_k 自嵌入 OB_k' 每个像素的最低 m 位，对原始像素改变不大，所以算法的不可见性会非常好. 因为 W 不是外在的水印，而是由原始图像自身的特征产生，而且具有抗攻击鲁棒性，本文把 W 称为原始鲁棒特征水印.

1.2 鲁棒特征水印提取及版权鉴别

检测端从攻击的含水印图像 AI 提取鲁棒特征水印 W' 及鉴别版权的过程分解如下：

Step1：将 AI 分割成互不重叠的 $n\times n$ 子块，各子块记为 AB_k，k 代表子块的序号，且 $k=1,2,\cdots,(N/n)^2$.

Step2：AB_k 的每个像素的最低 m 位置 0，将得到的各子块记为 AB_k'. 置位过程为

$$AB_k'=\text{bitget}(AB_k,j,0) \tag{4}$$

其中 bitget(·)为置位函数，$j=1,2,\cdots,m$.

Step3：对 AB_k' 进行 SVD，产生的奇异值记为 $\delta_k'^i$，i 代表奇异值的序号，$i=1,2,\cdots,n$.

Step4：计算 AB_k' 的奇异值范数，记为 $Norm_k'$，即

$$Norm_k'=\sqrt{\sum_{i=1}^{n}(\delta_k'^i)^2} \tag{5}$$

Step5：通过提取 $Norm_k'$ 的最高位奇偶性提取鲁棒特征水印 W'. 如果 $Norm_k'$ 的最高位数字为偶数，则 $W_k'=0$；否则，$W_k'=1$，W_k' 为 W' 的第 k bit.

Step6：计算 W 和 W' 之间的 NC 鉴别版权. NC 定义为

$$NC=\left(\sum_{k=1}^{(N/n)^2}(W_k\times W_k')\right)/\left(\sqrt{\sum_{k=1}^{(N/n)^2}(W_k)^2}\times\sqrt{\sum_{k=1}^{(N/n)^2}(W_k')^2}\right) \tag{6}$$

由上述过程可知，W' 提取时不借助原始图像，达到盲提取.

1.3 篡改检测

检测端对攻击图像 AI 进行篡改检测的过程分解如下：

Step1：将 AI 分割成互不重叠的 $n\times n$ 子块，各子块记为 AB_k，k 代表子块的序号，且 $k=1,2,\cdots,(N/n)^2$.

Step2：提取 AB_k 每个像素的最低 m 位. 提取位的过程为

$$L_k^j(r)=\text{bitget}(AB_k,j) \tag{7}$$

其中 bitget(·)为提取位函数，$L_k^j(r)$ 为第 k 个子块每个像素最低第 j 位组成的比特序列，$j=1,2,\cdots,m$，$r=1,2,\cdots,n^2$.

Step3：通过判断每个子块提取的鲁棒特征水印比特 W_k' 与 $L_k^j(r)$ 之间是否完全一致来检测此子块是否遭到篡改. 只有当 $W_k'=L_k^j(r)$ 对于每个 j 和 r 都成立时，才认为此子块没有遭到篡改；反之，只要 $L_k^j(r)$ 中对于每个 j 和 r 有任意 1 bit 与 W_k' 不一致，则认为此子块遭到篡改，遭篡改的子块用全黑标识.

2 不可见性的理论分析

原始图像 OI 与含水印图像 OI′之间的视觉差异用峰值信噪比 (Peak Signal-to-Noise Ratio, PSNR)衡量. OI 与 OI′之间的 PSNR 定义为

$$PSNR=10\lg\left[\frac{255\times255}{\frac{1}{N\times N}\sum_{s=1}^{N}\sum_{t=1}^{N}[OI(s,t)-OI'(s,t)]^2}\right] \tag{8}$$

把原始鲁棒特征水印自嵌入到图像各子块每个像素的最低 m 位，OI 与 OI′的每个像素数值差值绝对值的最大值为 (2^m-1)，此时 PSNR 将取得最小值，有 $\sum_{s=1}^{N}\sum_{t=1}^{N}[OI(s,t)-OI'(s,t)]^2=(2^m-1)^2\times N^2$，PSNR 的最小值为 $10\lg\left(\frac{255}{2^m-1}\right)^2$. 当 $m=2$，PSNR 的最小值等于 38.588 4 dB；当 $m=1$，PSNR 的最小值等于 48.130 8 dB. 文献[18-19]表明，当图像的

PSNR 大于 35 dB 时,水印的不可见性较好. 因此,算法具有较好的不可见性.

3　实验结果

本文以大小为 512 × 512 的 256 灰度级 Barbara、Lena、Elain、Zelda图像为测试图像,见图1.

图 1　原始图像
Fig. 1　Original image

将原始图像分割成互不重叠的 8×8 子块,各子块每个像素的最低 2 位置 0,此时原始鲁棒特征水印 W 的长度为 4 096 bit,算法的水印嵌入容量为 524 288 bit. 产生的含水印 Barbara、Lena、Elain、Zelda 图像见图 2,与相应原始图像之间的 PSNR 分别为 42.095 5 dB、42.374 6 dB、42.208 1 dB 和 40.004 4 dB,大于理论分析得到的 PSNR 最小值,与理论分析吻合. 因此,此时算法具有非常好的不可见性. 从图 2 提取的鲁棒特征水印 W' 与相应原始鲁棒特征水印 W 之间的 NC 都为 1.000 0.

图 2　含水印图像,$m=2$
Fig. 2　Watermarked image, $m=2$

将原始图像分割成互不重叠的 8×8 子块,各子块每个像素的最低 1 位置 0,此时原始鲁棒特征水印 W 的长度为 4 096 bit,算法的水印嵌入容量为 262 144 bit. 产生的含水印 Barbara、Lena、Elain、Zelda 图像见图 3,与相应原始图像之间的 PSNR 分别为 50.535 0 dB、50.836 5 dB、50.778 6 dB 和 48.442 3 dB,大于理论分析得到的 PSNR 最小值,与理论分析吻合. 因此,此时算法具有非常好的不可见性. 从图 3 提取的鲁棒特征水印 W' 与相应原始鲁棒特征水印 W 之间的 NC 都为 1.000 0.

图 3　含水印图像,$m=1$
Fig. 3　Watermarked image, $m=1$

3.1　抗攻击鲁棒性实验

对图 2～3 的含水印图像进行抗攻击鲁棒性实验,用 NC 值来衡量算法抵抗攻击的鲁棒性,用 PSNR 衡量攻击的含水印图像与相应原始图像之间的视觉差异. 随机删除行指从被删除行的下边第一行开始逐行向上移动,空余行补全黑. 向右偏移列指整个图像右移,前几列补全黑,最后几列移出丢失. 向下偏移行指整个图像下移几行,上几行补全黑,最后几行移出丢失. 重采样采用 nearest 插值法. 各表中“/”上方为提取的特征水印与原始特征水印之间的 NC,“/”下方为攻击后的含水印图像与相应原始图像之间的 PSNR.

从表 1 可以看出，对于 Barbara、Lena、Elain、Zelda 图像，不管 $m=2$ 还是 $m=1$，虽然各种攻击给含水印图像造成严重的视觉影响，但是攻击后提取的特征水印与相应原始特征水印之间的 NC 仍然很

表 1 抗攻击鲁棒性实验结果

Table 1 Experimental results of robustness against attacks

		Adding noise				Cropping		JPEG compression		
		Gaussian noise (mean is 0, variance is 0.001)	Gaussian noise (mean is 0, variance is 0.003)	Salt&pepper noise (noise density is 0.003)	Salt&pepper noise (noise density is 0.005)	1/32 upper left corner	1/16 upper left corner	Quality factor		
								20	40	60
Barbara	$m=2$	0.961 9/29.721 6	0.946 1/25.122 9	0.982 6/30.259 4	0.968 4/28.176 3	0.985 7/17.551 3	0.966 2/15.700 5	0.942 8/27.637 5	0.960 5/30.941 0	0.958 4/33.014 7
	$m=1$	0.976 1/29.909 4	0.955 5/25.209 0	0.981 9/30.708 4	0.970 4/28.286 2	0.986 5/17.563 9	0.966 3/15.708 5	0.948 8/27.696 3	0.974 2/31.080 3	0.971 5/33.251 9
Lena	$m=2$	0.961 1/29.774 9	0.951 3/25.132 0	0.988 7/30.574 8	0.979 6/28.240 7	0.981 2/21.584 0	0.969 8/17.274 7	0.954 4/32.394 3	0.961 5/34.427 7	0.961 7/35.602 0
	$m=1$	0.978 7/29.944 9	0.958 6/25.190 1	0.985 9/30.453 7	0.983 8/28.621 6	0.980 7/21.614 2	0.970 1/17.285 5	0.966 0/32.537 7	0.975 5/34.647 4	0.973 1/35.909 5
Elain	$m=2$	0.981 6/29.843 7	0.974 1/25.206 6	0.993 8/30.342 2	0.986 2/28.162 6	0.981 0/23.482 7	0.961 6/19.864 2	0.963 7/31.035 8	0.973 8/32.112 6	0.977 9/32.749 4
	$m=1$	0.988 9/29.981 8	0.979 7/25.281 4	0.992 2/30.318 3	0.988 9/28.557 1	0.981 1/23.531 5	0.961 7/19.884 7	0.975 8/31.119 2	0.989 9/32.231 8	0.986 2/32.900 2
Zelda	$m=2$	0.969 2/29.610 4	0.952 5/25.120 5	0.968 8/29.997 8	0.961 6/27.956 8	0.987 7/14.158 2	0.972 4/12.135 4	0.918 8/31.690 4	0.935 7/33.589 1	0.936 8/34.549 6
	$m=1$	0.969 1/29.946 1	0.950 8/25.236 8	0.976 8/30.600 9	0.962 1/28.223 1	0.989 1/14.167 6	0.974 2/12.141 1	0.934 8/32.032 8	0.965 7/34.063 9	0.954 6/35.184 2
Reference [3]	Barbara	0.906 2/29.974	0.852 4/25.222 9	0.925 1/30.424 6	0.888 3/28.379 2	0.982 2/17.566 1	0.957 7/15.709 9	0.857 7/27.741 0	0.907 0/31.164 8	0.926 4/33.402 6
	Lena	0.924 1/29.965 1	0.879 8/25.246 7	0.931 5/30.990 3	0.890 5/28.362 5	0.985 9/21.619 2	0.960 9/17.287 3	0.904 7/32.633 1	0.947 1/34.800 8	0.955 4/36.126 4
	Elain	0.905 3/30.002 7	0.857 4/25.269 1	0.902 6/30.946 7	0.852 1/28.582 6	0.986 3/23.539 4	0.969 0/19.888 0	0.892 5/31.184 9	0.920 9/32.309 4	0.924 4/32.998 0
	Zelda	0.942 2/29.973 5	0.902 5/25.268 7	0.928 2/30.675 6	0.889 6/28.432 1	0.975 9/14.169 2	0.953 7/12.142 1	0.912 0/32.245 9	0.935 9/34.409 0	0.951 7/35.630 0

		Smoothing		Resampling		Geometric attack			
		Gaussian low-pass filter (window size 3×3 σ=1)	Median filter (window size is 3×3)	First lessen to 80%, then magnify to 125%	First lessen to 50%, then magnify to 200%	Random removal	Row	Right shifting	Downward shifting
						Row number is 1	Row number is 2	Column number is 1	Row number is 1
Barbara	$m=2$	0.941 5/25.770 9	0.962 6/24.544 9	0.916 0/21.156 4	0.937 6/21.825 2	0.963 1/25.739 3	0.939 2/21.388 6	0.917 3/19.355 6	0.947 7/24.275 9
	$m=1$	0.950 7/25.806 2	0.967 2/24.607 8	0.915 4/21.178 6	0.935 1/21.842 8	0.968 2/25.828 0	0.945 2/21.421 6	0.919 4/19.377 7	0.945 6/24.336 5
Lena	$m=2$	0.959 6/32.902 5	0.979 6/34.415 3	0.949 9/25.653 1	0.953 6/27.921 5	0.973 1/28.730 0	0.953 2/24.917 4	0.934 1/25.979 3	0.964 6/27.425 9
	$m=1$	0.972 3/33.033 9	0.982 5/35.007 5	0.949 9/25.725 5	0.956 7/27.985 6	0.975 9/28.892 5	0.952 3/24.985 1	0.934 9/26.067 4	0.960 7/27.551 5
Elain	$m=2$	0.980 1/31.483 1	0.991 4/32.063 0	0.965 1/25.537 0	0.971 7/28.171 9	0.980 7/27.105 2	0.965 0/24.758 5	0.969 1/26.125 5	0.972 0/25.454 3
	$m=1$	0.987 7/31.571 2	0.992 7/32.428 0	0.968 0/25.608 9	0.970 9/28.244 4	0.981 8/27.224 6	0.969 0/24.831 4	0.968 0/26.225 9	0.970 3/25.537 6

Zelda	$m=2$	0.938 3/ 33.372 7	0.981 9/ 34.101 7	0.925 2/ 27.078 2	0.933 7/ 29.298 0	0.959 1/ 31.024 1	0.927 3/ 27.329 5	0.911 9/ 26.158 0	0.924 6/ 29.154 9
	$m=1$	0.964 8/ 33.763 2	0.982 2/ 35.055 5	0.923 0/ 27.204 0	0.938 9/ 29.456 9	0.959 4/ 31.537 7	0.928 1/ 27.550 7	0.908 9/ 26.313 7	0.924 2/ 29.475 8
Reference [3]	Barbara	0.921 0/ 25.803 0	0.930 9/ 24.618 1	0.910 3/ 21.148 2	0.914 1/ 21.809 2	0.968 3/ 25.843 2	0.949 3/ 21.427 1	0.918 2/ 19.382 1	0.955 0/ 24.347 0
	Lena	0.955 6/ 32.999 1	0.969 7/ 35.111 0	0.946 3/ 25.713 5	0.944 6/ 27.967 9	0.981 3/ 28.922 0	0.966 8/ 24.998 4	0.943 4/ 26.081 9	0.961 1/ 27.573 1
	Elain	0.940 6/ 31.537 8	0.953 7/ 32.489 1	0.915 9/ 25.633 2	0.914 6/ 28.265 3	0.968 9/ 27.246 2	0.941 2/ 24.846 0	0.918 8/ 26.245 1	0.948 8/ 25.552 5
	Zelda	0.962 1/ 33.768 1	0.965 7/ 35.230 9	0.946 5/ 27.235 2	0.950 7/ 29.498 7	0.980 5/ 31.638 3	0.963 2/ 27.596 6	0.940 8/ 26.340 8	0.971 1/ 29.527 6

高.因此,不管 $m=2$ 还是 $m=1$,对于 Barbara、Lena、Elain、Zelda 图像而言,算法在各种攻击中表现出很强的鲁棒性.另外,对比表 1 的 $m=2$ 栏和 $m=1$栏数据可以发现,它们在各种攻击后得到的 NC 不相上下.这是因为特征水印嵌入在各子块每个像素的最后 2 位还是最后 1 位,对于该子块的奇异值范数最高位影响不大.

将本文算法的抗攻击鲁棒性与文献[3]的鲁棒零水印算法进行比较.文献[3]算法对原始图像进行 8×8 分块,对每个子块进行 SVD,然后对每个子块的奇异值矩阵进行离散余弦变换(Discrete Cosine Transformation,DCT),通过比较相邻两个子块的直流(Direct Current,DC)系数大小产生零水印序列.文献[3]算法的抗攻击鲁棒性实验结果见表 1 的"文献[3]"栏.与文献[3]算法相比,对于 Barbara、Lena、Elain 三幅图像,本文算法的抗攻击鲁棒性总体上强于文献[3]算法;对于 Zelda 图像,本文算法在抵抗添加噪音、剪切、JPEG 压缩、平滑的鲁棒性总体上强于文献[3]算法,在抵抗重采样和几何攻击的鲁棒性差于文献[3]算法.

3.2　恶意篡改检测实验和篡改类型区别

3.2.1　恶意篡改检测实验结果

根据篡改的主观故意性,篡改可分为无意篡改和恶意篡改.恶意篡改一般指剪切-粘贴、叠加等操作,无意篡改一般指常规信号处理操作.这部分以 Barbara 图像作为测试图像进行恶意篡改检测实验.用原始 Barbara 图像作为背景,算法检测到的遭篡改子块用黑色标识.限于篇幅,只列出 $m=2$ 的实验结果.

实验一:将图 4 的 64×64 Kids 灰度图像替换图 2(a)的(64∶127,123∶186)区域的内容,篡改图像见图 5,篡改检测图像见图 6.可见,算法可以准确检测出篡改区域.

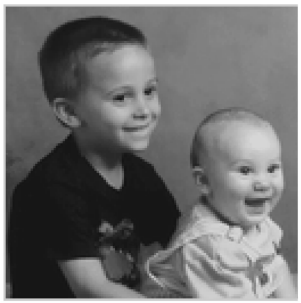

图4　儿童
Fig.4　Kids

图5　实验一的篡改图像
Fig.5　Tampered image of experiment one

图6　实验一的篡改检测图像
Fig.6　Detection of tampered image of experiment one

图7　精武
Fig.7　Jingwu

实验二:将图 7 的 32×32 精武二值图像叠加到图 2(a)的(69∶100,45∶76)区域,篡改图像见图 8.图8中,精武二值图像的像素值为0处没有发生

图8　实验二的篡改图像
Fig.8　Tampered image of experiment two

图9　实验二的篡改检测图像
Fig.9　Detection of tampered image of experiment two

篡改，像素值为 1 处发生了篡改. Barbara 是 256 灰度级图像，所以将精武二值图像叠加到图 2(a)后不能看出“精武”两个字. 篡改检测图像见图 9. 虽然此时只对图 2(a)做轻微改动，但仍可以检测出篡改区域，所以算法对篡改非常敏感.

实验三：将图 7 的 32×32 精武二值图像每个像素乘以 75 再叠加到图 2(a)的(69∶100，45∶76) 区域，篡改图像见图 10. 此时图 10 可以看出“精武”两个字. 篡改检测图像见图 11. 可见，算法可以准确检测出篡改区域.

实验四：将图 2(a)的(170∶220，135∶185)区域每个像素数值加上 1，篡改图像见图 12. 图 12 中，因为 Barbara 是 256 灰度级图像，篡改区域像素值加 1 对其视觉几乎没有影响，所以从图 12 中将不能察觉到篡改前后的视觉差异. 篡改检测图像见图 13. 虽然此时只对图 2(a)做轻微改动，但仍可以检测出篡改区域，所以算法对篡改非常敏感.

图10 实验三的篡改图像
Fig.10 Tampered image of experiment three

图11 实验三的篡改检测图像
Fig.11 Detection of tampered image of experiment three

图12 实验四的篡改图像
Fig.12 Tampered image of experimentfour

图13 实验四的篡改检测图像
Fig.13 Detection of tampered image of experiment four

实验五：剪切图 1(a)原始图像的(67∶86，106∶125)区域替换图 2(a)的相同区域，篡改图像见图 14. 因为鲁棒特征水印被嵌入到各子块每个像素的最低 2 位，所以从图 14 中将不能察觉到篡改前后的视觉差异. 篡改检测图像见图 15. 可见，算法可以准确检测出篡改区域，对篡改非常敏感.

实验六：将图 2(a)的(69∶124，45∶90)区域替换自身的(121∶176，121∶166)区域，篡改图像见图 16，篡改检测图像见图 17. 可见，算法可以准确检测出篡改区域.

图14 实验五的篡改图像
Fig.14 Tampered image of experiment five

图15 实验五的篡改检测图像
Fig.15 Detection of tampered image of experiment five

图16 实验六的篡改图像
Fig.16 Tampered image of experiment six

图17 实验六的篡改检测图像
Fig.17 Detection of tampered image of experiment six

从实验一到实验六可以看出，遭篡改区域都可以被准确检测出来，所以算法的篡改检测能力很强.

3.2.2 篡改类型区别

定义一个篡改评估函数(Tamper Assessment Function，TAF)定量地度量遭篡改的程度. 算法以 8×8 分块作为一个篡改检测单元，所以篡改评估函数定义为

$$\mathrm{TAF}=\frac{\text{number}}{\text{图像 8×8 分块总数}} \tag{9}$$

其中 number 代表检测到的遭篡改 8×8 分块总数. number 越大，代表检测到的遭篡改总数越多；TAF 越大，代表篡改得越厉害. 表 2 的第二栏和第三栏分

表 2 恶意篡改检测的 number 和 TAF

Table 2 number and TAF for malicious tamper detection

	实验一	实验二	实验三	实验四	实验五	实验六
Number	81	12	12	56	9	36
TAF	0.019 8	0.002 9	0.002 9	0.013 7	0.002 2	0.008 8
Tamper type	Malicious	Malicious	Malicious	Malicious	Malicious	Malicious

别列出实验一至实验六的 number 和 TAF. 从表 2 可以看出,实验一篡改得最厉害,实验五篡改得最少.

同样地,可以利用 TAF 定量地度量表 1 中当 $m=2$时各种常规信号处理无意篡改对含水印 Barbara 图像篡改的程度,见表 3 相应栏的 number 和 TAF.

篡改评估函数 TAF 用于衡量篡改程度. 恶意篡改者往往带有主观故意性,一般针对图像的关键局部区域进行篡改,而且带有隐藏恶意篡改的目的,具有一定的隐蔽性,所以恶意篡改的篡改程度往往不大,其 TAF 较小;无意篡改一般指滤波、加噪等常规信号处理操作,是一种全局性篡改,一般没有主观故意性和隐蔽性,所以无意篡改的篡改程度往往较大,其 TAF 较大. 因此利用 TAF 能够实现区分恶意篡改和无意篡改. 给定一阈值 ϑ,如果 TAF$\leqslant\vartheta$,则认为含水印图像遭到了恶意篡改;反之,则认为含水印图像遭到了无意篡改. 原始载体图像的差异性使得很难从理论上用统一公式确定 ϑ,应由用户根据实际应用的要求采用实验的方法来选定,如果用户在实际应用中对图像真实性要求很高,需把所有恶意篡改都判断出来,此时可以选择一个数值相对较大的 ϑ. 本文选择 ϑ 为 0.025 时可以正确区分表 2 和表 3 各种情况时的篡改类型,见表 2 和表 3 的"篡改类型"一栏.

表 3　无意篡改检测的 number 和 TAF

Table 3　Number and TAF for unintentional tamper detection

		Adding noise				Cropping		JPEG compression		
		Gaussian noise (mean is 0, variance is 0.001)	Gaussian noise (mean is 0, variance is 0.003)	Salt&pepper noise (noise density is 0.003)	Salt&pepper noise (noise density is 0.005)	1/32 upper left corner	1/16 upper left corner	Quality factor		
								20	40	60
Barbara ($m=2$)	number	4 096	4 096	2 371	2 509	128	256	3 920	4 033	4 070
	TAF	1.000 0	1.000 0	0.578 9	0.612 5	0.031 3	0.062 5	0.957 0	0.984 6	0.993 7
	Tamper type	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional

		Smoothing		Resampling		Geometric attack			
		Gaussian low-pass filter (window size 3×3 $\sigma=1$)	Median filter (window size is 3×3)	First lessen to 80%, then magnify to 125%	First lessen to 50%, then magnify to 200%	Random removal	Row	Right shifting	Downward shifting
						Row number is 1	Row number is 2	Column number is 1	Row number is 1
Barbara ($m=2$)	number	4 096	2 099	1 334	365	733	786	1 339	1 169
	TAF	1.000 0	0.512 5	0.325 7	0.089 1	0.179 0	0.191 9	0.326 9	0.285 4
	Tamper type	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional	Unintentional

根据第 3 部分的上述实验结果可知,本文算法具有非常好的不可见性,而且同时具备很强的鲁棒性和很敏感的脆弱性从而很好地同时实现版权保护和内容认证. 然而,文献[1-11]只具有鲁棒性,只实现版权保护功能;文献[12-15]只具有脆弱性,只实现内容认证功能;文献[16-17]也很难同时具有很强的鲁棒性和很敏感的脆弱性.

4　结论

针对现有图像水印算法功能单一的局限性,本文利用自嵌入技术提出一种双功能图像水印算法. 理论分析和实验结果都表明算法具有很好的不可见性. 实验结果还表明,算法不仅在抵抗添加噪音、剪切、JPEG 压缩、平滑、重采样和几何攻击如随机删除行、向右偏移列、向下偏移行上表现出很强的鲁棒性,而且还能够精确定位出篡改位置和区分篡改类型. 因此,算法具有版权保护和内容认证双重功能.

本文算法可以应用于要求图像水印技术既具有非常好的不可见性又同时具备版权保护和内容认证双重需求的场合.

参考文献

[1] WEN Quan, SUN Tan-feng, WANG Shu-xun. Concept and application of zero-watermark[J]. *Acta Electronica Sinica*, 2003, **31**(2): 214-216.
温泉,孙锬锋,王树勋. 零水印的概念与应用[J]. 电子学报, 2003, **31**(2):214-216.

[2] YE Tian-yu. A robust zero-watermarking algorithm against dual print-and-scan process based on discrete cosine transformation[J]. *Acta Photonica Sinica*, 2011, **40**(1):

142-148.
叶天语.离散余弦变换域抗二次打印-扫描鲁棒零水印算法[J].光子学报,2011,**40**(1):142-148.

[3] YE Tian-yu. A robust zero-watermark algorithm based on singular value decomposition and discreet cosine transform [C]. Communications in Computer and Information Science (CCIS), Springer-Verlag, Berlin, Heidelberg, 2011, **137**: 1-8.

[4] LIU Rui-zhen, TAN Tie-niu. SVD based digital watermarking method[J]. *Acta Electronica Sinica*, 2001, **29**(2): 168-171.
刘瑞祯,谭铁牛.基于奇异值分解的数字图像水印算法[J].电子学报,2001,**29**(2):168-171.

[5] SOLANKI K, MADHOW U, MANJUNATH B S, *et al*. "Print and scan" resilient data hiding in images[J]. *IEEE Transactions on Information Forensics and Security*, 2006, **1**(4): 464-478.

[6] KANG Xian-gui, HUANG Ji-wu, ZENG Wen-jun. Efficient general print-scanning resilient data hiding based on uniform log-polar mapping[J]. *IEEE Transactions on Information Forensics and Security*, 2010, **5**(1): 1-12.

[7] LI Xu-dong. Gray-level digital watermarking algorithm based on SVD[J]. *Geomatics and Information Science of Wuhan University*, 2010, **35**(11): 1305-1308, 1359.
李旭东.基于奇异值分解的灰度级数字水印算法[J].武汉大学学报(信息科学版),2010,**35**(11):1305-1308,1359.

[8] LI Xu-dong, ZHANG Zhen-yue. Two-layer partition and singular value decomposition based image watermarking [J]. *Journal of Zhejiang University (Engineering Science)*, 2006, **40**(12): 2088-2092.
李旭东,张振跃.图像双层划分和奇异值分解的数字水印算法[J].浙江大学学报(工学版),2006,**40**(12):2088-2092.

[9] LI Xu-dong. Public watermarking using matrix norm[J]. *Journal of Computer-Aided Design& Computer Graphics*, 2005, **17**(8): 1857-1861.
李旭东.利用矩阵范数实现的公开水印技术[J].计算机辅助设计与图形学学报,2005,**17**(8):1857-1861.

[10] LI Xin-wei, GUO Bao-long, LI Lei-da. A statistic-quantization based image watermarking algorithm resisting geometric attacks [J]. *Journal of Optoelectronics · Laser*, 2009, **20**(8): 1082-1086.
李新伟,郭宝龙,李雷达.一种基于统计量化的抗几何攻击图像水印算法[J].光电子·激光,2009,**20**(8):1082-1086.

[11] XU Wen-li, LI Lei, WANG Yu-min. Robust digital watermarking scheme resistant to gaussian noise, geometric distortion and JPEG compression attacks[J]. *Journal of Electronics & Information Technology*, 2008, **30**(4): 933-936.
许文丽,李磊,王育民. 抗噪音、几何失真和JPEG压缩攻击的鲁棒数字水印方案[J].电子与信息学报,2008,**30**(4):933-936.

[12] ZHANG Hong-bin, YANG Cheng. Tamper detection and self-recovery of images using self-embedding [J]. *Acta Electronica Sinica*, 2004, **32**(2): 196-199.
张鸿宾,杨成.图像的自嵌入及窜改的检测和恢复算法[J].电子学报,2004,**32**(2):196-199.

[13] HE Hong-jie, ZHANG Jia-shu. Self-embedding watermarking algorithm with robustness against watermark information alterations[J]. *Journal of Software*, 2009, **20**(2): 437-450.
和红杰,张家树.对水印信息篡改鲁棒的自嵌入水印算法[J].软件学报,2009,**20**(2): 437-450.

[14] ZHANG Xian-hai, YANG Yong-tian. Image authentication scheme research based on fragile watermarking [J]. *Acta Electronica Sinica*, 2007, **35**(1): 34-39.
张宪海,杨永田.基于脆弱水印的图像认证算法研究[J].电子学报,2007,**35**(1):34-39.

[15] DING Ke, HE Chen, JIANG Ling-ge, *et al*. A fragile watermark technique based on address code[J]. *Journal of Shanghai Jiaotong University*, 2004, **38**(4): 620-623.
丁科,何晨,蒋铃鸽,王宏霞.基于地址码的脆弱数字水印技术[J].上海交通大学学报,2004,**38**(4):620-623.

[16] SCHLAUWEG M, PROFROCK D, PALFNER T, *et al*. Quantization-based semi-fragile public-key watermarking for secure image authentication[C]. *SPIE*, 2005, **5915**: 41-51.

[17] LI Chun, HUANG Ji-wu. A semi-fragile image watermarking resisting to JPEG [J]. *Journal of Software*, 2006, **17**(2): 315-324.
李春,黄继武.一种抗JPEG压缩的半脆弱图像水印算法[J].软件学报,2006,**17**(2):315-324.

[18] NIKOLAIDIS A, PITAS I. Asymptotically optimal detection for additive watermarking in the DCT and DWT domains[J]. *IEEE Transactions on Image Processsing*, 2003, **12**(5): 563-571.

[19] CHENG Q, HUANG T S. Robust optimum detection of transform domain multiplicative watermarks [J]. *IEEE Transactions on Signal Processing*, 2003, **51**(4): 906-924.

A Self-embedding Image Watermarking Scheme with Dual Purpose

YE Tian-yu
(*College of Information & Electronic Engineering, Zhejiang Gongshang University, Hangzhou* 310018, *China*)

Abstract: An image watermarking scheme with dual purpose is proposed by introducing the self-embedding technology, which can achieve both copyright protection and content authentication. At the embedding side, an original image is split into non-overlapping blocks, and the m least significant bits of every pixel in each block are set to be zero. Then each block is conducted with singular value decomposition, and an original robust feature watermark is derived from judging the parity of the first digit of singular value's norm from each block. Finally, a watermarked image is produced by self-embedding the original robust feature watermark into the m least significant bits of every pixel in each block. At the detection side, robust feature watermark's extraction from an attacked watermarked image is similar to its production at the embedding end. Copyright identification is accomplished by calculating the normalized correlation between the extracted robust feature watermark and the original robust feature watermark. Content authentication is accomplished by tamper detection through judging the consistency between the extracted robust feature watermark and the m least significant bits of every pixel in each block of attacked image. Both theoretic analysis and experimental results illustrate that it has perfect invisibility. Experimental results also show that it not only has strong robustness towards attacks such as adding noise, cropping, JPEG compression, smoothing, resampling and geometric attacks like random row removal, right shifting and downward shifting, but also can accurately locate the tampered region and distinguish tamper type. Therefore, it can achieve dual purpose containing copyright protection and content authentication.

Key words: Digital watermarking; Self-embedding; Copyright protection; Content authentication; Tamper detection; Malicious tamper