

文章编号:1007-9432(2020)04-0508-08

打印扫描不变的多轮廓像素翻转 文本图像水印算法

郭雯¹, 宋春花¹, 关虎², 黄樱², 牛保宁¹, 兰方鹏¹

(1. 太原理工大学 信息与计算机学院, 太原 030024; 2. 中国科学院自动化研究所 数字内容中心, 北京 100190)

摘要:基于打印扫描不变量,提出一种多轮廓像素翻转的文本图像水印算法,其中包含三个措施:提出一种水印压缩预处理方法,通过压缩水印信息,在总嵌入容量一定的情况下,增加水印嵌入的次数,从而提高水印鲁棒性;提出多轮廓像素翻转策略,降低水印错误嵌入的可能性,进一步提高水印鲁棒性;以打印扫描不变量为特征值,结合离散余弦变换和多轮廓像素翻转策略提出一种优化的双域结合的水印嵌入及提取算法,在保证水印不可见性的同时,提高水印抵抗打印扫描攻击的能力。实验表明,本文算法能够在保证水印不可见性的前提下,提高水印对打印扫描攻击的鲁棒性,同时也能够抵抗常见的图像攻击。

关键词:数字水印;文本图像;打印扫描攻击;打印扫描不变量;像素翻转;离散余弦变换

中图分类号:TP301.6 **文献标识码:**A

DOI:10.16355/j.cnki.issn1007-9432tyut.2020.04.004 **开放科学(资源服务)标识码(OSID):** 

Print-scan Invariant Multi-contour Pixel Flipping Text Image Watermark Algorithm

GUO Wen¹, SONG Chunhua¹, GUAN Hu², HUANG Ying², NIU Baoning¹, LAN Fangpeng¹

(1. College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China;
2. Digital Content Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Based on the print-scan invariant, a multi-contour pixel flipped watermarking algorithm for text images was proposed, which contains three measures as follows. First, a watermark compression preprocessing method was proposed. By compressing watermarking information, the number of times of watermark embedding is increased under the condition of a certain total embedding capacity, so as to improve the robustness of the watermark. Second, a multi-contour pixel flipping strategy was proposed, which can reduce the possibility of false embedding of the watermark, and further improve watermark robustness. Finally, characterized by print-scan invariant, combined with discrete cosine transform and multi-contour pixel flipping strategy, an optimized dual-domain combined watermark embedding and extraction algorithm was proposed to improve the ability of the watermark to resist print-scan attack while ensuring the invisibility of the watermark. Experiments show that the proposed algorithm can improve the robustness of the watermark against print-scan attack while ensuring the invisibility of the watermark, and it can also resist common image attacks.

Keywords: digital watermarking; text image; print-scan attack; print-scan invariant; flip pixels; discrete cosine transform

收稿日期:2020-02-13

基金项目:国家重点研发计划“文化内容资源产权交易技术”项目资助(2017YFB1401000);山西省重点研发计划(国际科技合作方面)“数字媒体内容资源版权管理与服务技术及系统研发”项目资助(201903D421007)

通讯作者:宋春花(1965-),女,博士,副教授,主要从事三维场景建模及可视化、图像处理等研究,(E-mail)420160279@qq.com

引文格式:郭雯,宋春花,关虎,等.打印扫描不变的多轮廓像素翻转文本图像水印算法[J].太原理工大学学报,2020,51(4):508-515.

文本图像水印技术是文本水印技术研究的一个重要方向,是文本侵权举证的重要手段^[1]。文本图像在传播过程中,会遇到多种处理攻击,如旋转、加噪、压缩、打印扫描等。其中,打印扫描是文本图像的重要传播途径,同时也是文本信息泄漏和被盗用的常见方式之一。因此,文本图像水印既要能够抵抗常见的图像攻击,又要对打印扫描攻击具有良好的抵抗能力。然而,打印扫描是一个复杂的过程,对水印的攻击受仪器型号、参数影响较大,这给文本图像水印带来了极大的挑战。

当前,文本图像水印方法可以分为变换域方法^[2-3]、空间域方法^[4-9]和双域结合方法^[10-12]。

变换域方法将像素值变换到某种变换域上,如离散余弦变换(discrete cosine transform, DCT)、离散傅里叶变换(discrete fourier transform, DFT)、离散小波变换(discrete wavelet transform, DWT)等,通过修改变换域系数嵌入水印。变换域方法鲁棒性强,但文本图像只有黑白二值,使用变换域方法嵌入水印以后需要将文本二值化,在此过程中水印容易丢失。

空间域方法是直接修改文本像素来嵌入水印的。可以通过修改文本结构嵌入水印,如上下移动行位置^[4]、左右移动字符位置^[5];可以通过修改字符特征嵌入水印,如拆分多结构字符^[6]、调整笔画弯曲程度^[7];可以通过翻转像素嵌入水印,如使用像素翻转策略选择人眼关注度低的像素进行翻转^[8-9]。空间域方法不需要二值化处理,但对各种攻击的抵抗能力较差。

双域结合方法为结合变换域和空间域方法。HE et al^[10]将文本图像进行分块,以每块内黑色像素数量比例作为特征值,结合单个像素的像素翻转策略和 DCT 嵌入水印;但是该算法的特征值在打印扫描前后差异较大,导致水印抗打印扫描的效果不理想。姚寒冰等^[11]使用元文法^[9]提出的打印扫描不变量作为特征值,然后通过像素翻转和 DCT 嵌入水印;该算法只能嵌入 64 位水印,水印容量较低。TAN et al^[12]提出单轮廓像素翻转策略,对连通分量最外层轮廓信息进行 DFT,修改 DFT 系数得到新轮廓,通过新旧轮廓差异选择可翻转像素,然后结合打印扫描不变量翻转像素嵌入水印;该算法视觉效果好,但是单轮廓像素翻转策略可能会嵌入错误水印,降低水印鲁棒性。双域结合方法结合变换域和空间域方法的优点,可以提高水印的鲁棒性,又可用于二值文本图像,但当前双域结合方法对打

印扫描攻击的抵抗能力依然有待提高。

针对当前文本图像水印方法对打印扫描攻击鲁棒性不佳的问题,本文基于打印扫描不变量提出一种多轮廓像素翻转的双域结合文本图像水印算法,在保证水印视觉不可见、能够抵抗常见图像攻击的前提下,提高水印对打印扫描攻击的抵抗能力,其创新点如下。

1) 提出一种水印压缩预处理方法,在嵌入水印前使用跳白块编码压缩水印图像,使得在文本图像总嵌入容量一定的情况下,增加水印嵌入的次数,达到提高水印鲁棒性的目的;

2) 提出多轮廓像素翻转策略,降低单轮廓像素翻转策略嵌入错误水印的可能性,提高水印鲁棒性;

3) 基于打印扫描不变量,结合 DCT 和多轮廓像素翻转策略,设计一种优化的双域结合的文本图像水印嵌入及提取算法,进一步提高水印鲁棒性。

1 相关工作

1.1 打印扫描不变量

在打印扫描过程中,文本会遭遇多种攻击,例如:旋转、压缩、加噪、缩放等,并且不同型号、不同参数的打印机和扫描仪给文本带来的影响也不同。这使得无法使用精确的模型模拟打印扫描过程^[12]。为了使得水印能够抵抗打印扫描攻击,可以找到文本中在打印扫描前后保持不变的部分,利用此部分嵌入水印。元文法^[9]提出一种文本图像打印扫描过程中的不变量。

定义 打印扫描不变量。在打印扫描过程前后,文本中单个字符包含的黑色像素数量和平均每个字符包含的黑色像素数量比值保持不变。表达式如下:

$$\frac{\int_I F(x) dx}{M} = \frac{\int_I F_{PS}(x) dx}{M_{PS}} \quad (1)$$

式中: $F(x)$ 和 $F_{PS}(x)$ 分别表示打印扫描前后的文本图像; I 表示字符区域; x 表示黑色像素数量; M 和 M_{PS} 分别表示打印扫描前后平均单个字符的黑色像素数量。

1.2 跳白块编码

跳白块编码是经典有效的二值图像无损压缩方法。为了增加水印嵌入次数,本文使用跳白块编码压缩水印图像。在跳白块编码中,白色像素表示为“0”,黑色像素表示为“1”。将图像分为多块,每块包含 N 位像素。如果块内都是白色像素,则使用“0”

编码该块;如果块内包含黑色像素,则使用标志位“1”和块内 N 位像素的直接编码共同表示该块。解码时,若识别“0”,则恢复 N 位白色像素;若识别“1”,则按其后的 N 位编码恢复 N 位像素^[13]。

1.3 像素翻转策略

在像素翻转文本图像水印方法中,像素翻转策略选择文本中翻转后视觉差异小的像素进行翻转,是像素翻转水印方法中平衡水印不可见性和鲁棒性的重要技术。WU et al^[8]提出一种经典的单像素的像素翻转策略,根据每个像素在其 8 邻域内的连通性和平滑性计算该像素的可翻转得分,选择得分高的像素翻转。该策略为量化比较每个像素的翻转级别提供了方案,但是单像素的像素翻转策略容易在字符笔画边界引入凹凸点,降低水印的不可见性^[12]。为了解决该问题,TAN et al^[12]提出单轮廓像素翻转策略。使用傅里叶描述子表示连通分量最外层的轮廓,修改傅里叶描述子得到新连通分量轮廓,根据新旧轮廓的差异选择可翻转像素。由于人眼对轮廓改变的敏感度比单个像素改变的敏感度更低,该策略更符合人类视觉特性。但是该策略容易嵌入错误水印,并且嵌入水印后可能打乱水印顺序,降低水印的鲁棒性。

2 打印扫描不变的多轮廓像素翻转文本图像水印算法

为了提高水印对于打印扫描攻击的抵抗能力,同时使得水印视觉不可见,对常见图像攻击具有抵抗能力,本文基于打印扫描不变量,结合 DCT 和像素翻转方法,提出一种文本图像水印算法。算法包括水印压缩预处理、多轮廓像素翻转策略和水印嵌入及提取算法。

2.1 水印压缩预处理

为了提高水印的鲁棒性,本文提出一种水印预处理方法。在水印图像嵌入到文本图像之前对其进行压缩,使得在文本图像总嵌入容量一定的情况下,根据总嵌入容量动态增加水印嵌入的次数,提取时对提取的多个水印按位取众值,达到提高水印鲁棒性的目的。

典型的二值图像无损压缩方法有跳白块编码、游程编码、二叉树编码等^[13]。其中,跳白块编码通过缩减白色像素压缩图像,适用于白色像素远多于黑色像素的二值水印图像。跳白块编码将图像分为多块进行编码。经实验发现,水印图像分块大小不同时,压缩率不同,图 1 为不同分块大小二值水印

图像的压缩率。从图 1 中可以看出,当分块大小为 2×2 像素时,水印图像压缩率最高。当分块大小为 2×2 像素时所有编码情况如图 2 所示。因此,本文使用跳白块编码设计一种水印压缩预处理方法。将二值水印图像分割为 2×2 像素大小的子块,然后进行跳白块编码,将水印图像压缩为只有 0 和 1 的水印序列 $w = \{w_1, w_2, \dots, w_y\}$, y 为水印序列长度。

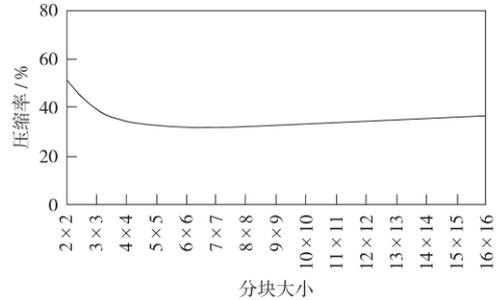


图 1 不同分块大小的二值图像压缩率

Fig. 1 Compression rate of binary images with different block sizes

子块				
编码	0	11000	10100	10010
子块				
编码	10001	11100	11010	11001
子块				
编码	10110	10101	10011	10111
子块				
编码	11011	11101	11110	11111

图 2 分块大小为 2×2 时不同块的编码

Fig. 2 Encoding of different blocks with block size 2×2

2.2 多轮廓像素翻转策略

单像素的像素翻转策略容易在字符边界引入凹凸点,单轮廓像素翻转策略^[12]修改连通分量最外层轮廓翻转像素,新连通分量黑色像素数量的实际值和目标值可能存在误差,导致错误地嵌入水印,并且翻转像素后连通分量的黑色像素数量可能低于连通分量提取阈值 t ,提取时无法识别,打乱了水印顺序。为了更好地平衡水印的不可见性和鲁棒性,本文提出多轮廓像素翻转策略,相比单轮廓像素翻转策略具有以下优点。

1) 降低了嵌入错误水印的可能性。连通分量内所有轮廓由大到小排序。修改一个轮廓后,如果当前连通分量黑色像素数量和目标黑色像素数量误差为 q ,则修改下一个轮廓,下一轮廓待翻转像素数量为 $-q$;直到 q 为 0 或者所有轮廓都已修改。

2) 保证嵌入水印的连通分量在水印提取过程中可以被识别。每次修改轮廓后判断当前连通分量黑色像素数量是否低于阈值 t 。如果低于 t ,则不修

改当前轮廓。

具体翻转步骤如下:

Step 1 连通分量 a_i 的黑色像素数量是 x_i , 目标黑色像素数量为 x'_i . 提取 a_i 中的所有轮廓, 并由大到小排序。

Step 2 修改第一个轮廓。计算该轮廓的傅里叶描述子 $F(p), p \in [0, l-1], l$ 为该轮廓像素点个数。

Step 3 将高频区域 $p \in [l/2-\gamma, l/2+\gamma]$ 的描述子置零, γ 初值为 0. 使用新傅里叶描述子重构得到新的轮廓。

Step 4 使用新轮廓填充得到新的连通分量 a''_i . 如果 $x_i < x'_i, a''_i = a_i \cup a''_i$, 若 a''_i 黑色像素数量 x''_i 超过 x'_i , 进入 Step 5; 否则, 扩大傅里叶描述子系数置零范围, $\gamma = \gamma + 1$, 返回 Step 3. 如果 $x_i > x'_i, a''_i = a_i \cap a''_i$, 若 a''_i 黑色像素数量 x''_i 低于 x'_i , 进入 Step 5; 否则, 扩大傅里叶描述子系数置零范围, $\gamma = \gamma + 1$, 返回 Step 3.

Step 5 如果 x''_i 低于阈值 t , 则该轮廓不进行修改; 否则, 若 x''_i 等于 x'_i , 则算法结束, 翻转后的连通分量 $a'_i = a''_i$; 若 x''_i 不等于 x'_i , 则继续翻转下一轮廓, 返回 Step 2.

2.3 水印嵌入及提取算法

2.3.1 水印嵌入

水印嵌入算法基于打印扫描不变量设计, 结合 DCT 和多轮廓像素翻转策略修改文本图像中连通分量的打印扫描不变量嵌入水印。

水印嵌入算法流程如图 3 所示。水印嵌入算法的具体步骤如下。

1) 计算各连通分量的打印扫描不变量。提取文本图像中黑色像素数量高于阈值 t 的连通分量, 使用下式计算各连通分量的打印扫描不变量 e_i :

$$e_i = \frac{x_i}{M} \times A. \quad (2)$$

式中: x_i 为每个连通分量的黑色像素数量; A 为精度调节参数, 如: 1 000, 10 000 等; M 为文本图像中连通分量的平均黑色像素数量。为了避免使用元文法算法调整区域, 增加可以嵌入水印的连通分量数量, 本文使用文本首行的连通分量的平均黑色像素数量作为 M , 计算公式如下:

$$M = \frac{\sum_{i=1}^D x_i}{D}. \quad (3)$$

式中: D 为文本图像首行中连通分量个数。

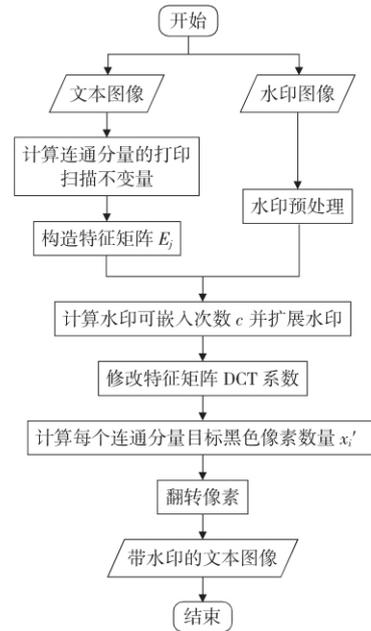


图 3 水印嵌入算法流程

Fig. 3 Watermark embedding algorithm flow

2) 使用打印扫描不变量构造特征矩阵。使用连通分量的打印扫描不变量 e_i , 按照行优先顺序构造多个 8×8 特征矩阵 $E_j (j=0, 1, \dots, k-1, k$ 为特征矩阵的个数)。

3) 计算水印可嵌入次数并扩展水印。为了提高水印鲁棒性, 本文算法将水印重复多次嵌入, 嵌入次数由文本总嵌入容量动态决定。每个特征矩阵 E_j 即每 64 个连通分量中嵌入固定位水印, 则文本中水印可重复嵌入次数 c 计算公式如下:

$$c = \frac{p \times k}{y}. \quad (4)$$

式中: p 为每个特征矩阵内嵌入水印位数; y 为编码后水印序列的长度。

将水印序列扩展 c 次。如: 原水印序列为 010, c 为 3, 水印序列扩展后为 000111000。

4) 修改特征矩阵 DCT 系数得到新特征矩阵。根据水印序列修改特征矩阵 E_j 的 DCT 系数, 然后通过逆 DCT 得到新的特征矩阵 E'_j . 特征矩阵 DCT 系数按逆“Z”字排序, 从第 s 位系数开始修改, 每一位水印修改一位系数。若水印为“1”, 则修改系数为 T ; 若水印为“0”, 则修改系数为 $-T$. T 为 $[1, 5\ 000]$ 范围中的整数。 T 取不同值, 则 E'_j 不同。为了减少连通分量翻转像素数量, 提高水印不可见性, E_j 和 E'_j 的差距应尽可能地小。 E_j 和 E'_j 的差距通过相关系数 R 和最优差分 U 比较。

相关系数 R 计算公式为:

$$R = \frac{\sum (E_j - \bar{E}_j)(E'_j - \bar{E}'_j)}{\sqrt{\sum (E_j - \bar{E}_j)^2 \sum (E'_j - \bar{E}'_j)^2}}. \quad (5)$$

最优差分 U 计算公式为

$$U = \sum \sum |(E_j - E'_j)/E_j|. \quad (6)$$

R 越大, U 越小, E_j 和 E'_j 的差距越小。多次迭代寻找 T 值在 $[1, 5\ 000]$ 范围内时 E_j 和 E'_j 的最小差距, 从而确定 T 的最终取值, 此时的 E'_j 为最终特征矩阵。

5) 计算每个连通分量目标黑色像素数量。新特征矩阵 E'_j 的每个分量 e'_i 代表一个连通分量嵌入水印后的打印扫描不变量, 根据 e'_i 可以求得每个连通分量的目标黑色像素数量 x'_i , 公式如下:

$$x'_i = \frac{e'_i M}{A}. \quad (7)$$

6) 翻转像素, 修改打印扫描不变量。根据多轮廓像素翻转策略翻转每个连通分量的像素, 使连通分量的黑色像素数量从 x_i 变为 x'_i , 从而修改各连通分量的打印扫描不变量, 得到嵌入水印的文本图像。

2.3.2 水印提取

水印提取算法与水印嵌入算法相对应, 其流程如图 4 所示。

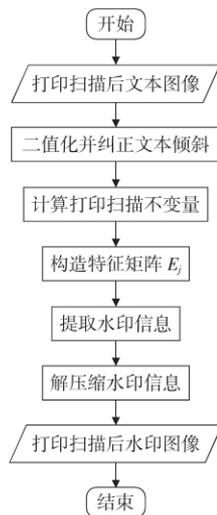


图 4 水印提取算法流程

Fig. 4 Watermark extraction algorithm flow

水印提取算法的具体步骤为:

1) 将打印扫描后的文本图像二值化, 使用霍夫变换纠正打印扫描过程引起的文本倾斜。

2) 提取文本图像中黑色像素数量低于阈值 t 的连通分量, 计算各连通分量的打印扫描不变量。

3) 使用打印扫描不变量以行优先顺序构造 8×8 特征矩阵 E_j 。

4) 根据 E_j 的 DCT 系数提取水印序列。DCT 系数按照逆“Z”字排列后, 从第 s 位系数开始, 若系数为正数, 则提取“1”; 若系数为负数, 则提取“0”。提取序列连续 c 位的众值决定一位水印, 得到水印序列。

5) 解压缩水印序列, 得到水印图像。

3 实验与分析

3.1 实验设置

为了说明本文算法的可行性和有效性, 本节通过实验验证水印的不可见性和鲁棒性。实验数据集为 96 张不同字体不同字号的中英文文本图像, 其中宋体四号、隶书小四、楷体小四、Times New Roman 12pt、Century 11pt、Arial 11pt 各 16 张。嵌入和提取过程在 Visual Studio 2013 中进行。

3.1.1 实验参数设置

为了动态地调节提取连通分量的阈值, t 值取文本首行平均每个连通分量黑色像素数量的 $1/5$ 。为平衡水印的不可见性和鲁棒性, 本文将水印嵌入到特征矩阵的中频区域。每个特征矩阵内嵌入的水印位数 p 为 24, 水印开始嵌入位数 s 为 16, A 取值 10 000。

3.1.2 评价标准

本文使用峰值信噪比 (peak signal to noise ratio, PSNR) 作为水印不可见性的客观评价标准, 使用误码率 (bit error ratio, BER) 作为水印鲁棒性的评价标准。

PSNR (用 R_{PSN} 表示) 通过计算文本嵌入水印前后的差异衡量水印给文本带来的视觉差异。 R_{PSN} 值越大, 说明水印的不可见性越好; 反之亦然。 R_{PSN} 计算公式如下:

$$R_{\text{PSN}} = 10 \log_{10} \left(\frac{g_{\text{max}}^2}{E_{\text{ms}}} \right). \quad (8)$$

式中: R_{PSN} 表示文本图像的峰值信噪比; g_{max} 表示文本图像颜色的最大数值; E_{ms} 为均方差, 计算公式如下:

$$E_{\text{ms}} = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m \|K(i, j) - I(i, j)\|^2. \quad (9)$$

式中: K 和 I 分别是原始文本和嵌入水印后文本; $m \times n$ 为文本大小。

BER 通过计算提取的水印中错误位数的比例来衡量水印鲁棒性 (用 b_{ER} 表示)。 b_{ER} 值越低, 代表水印鲁棒性越高; 反之亦然。其计算公式如下:

$$b_{\text{ER}} = \frac{y_{\text{err}}}{y_{\text{len}}} \times 100\%. \quad (10)$$

式中: b_{ER} 为提取水印的误码率; y_{err} 为水印错误提取位数; y_{len} 为水印总位数。

3.2 水印不可见性

通过举例展示使用本文算法嵌入水印的视觉效果。图 5 为一张二值水印图像,图 6 为一张宋体四号中文文本图像,图 7 为使用本文算法将图 5 嵌入到图 6 的文本图像。为了更直观地展示嵌入水印的效果,将图 6 和图 7 进行放大。图 8 和图 9 分别为原始文本图像和嵌入水印后的文本图像的局部放大图。



图 5 原始水印图像

Fig. 5 Original watermark image

豫章故郡,洪都新府。星分翼轸,地接衡庐。襟三江而带五湖,控蛮荆而引瓯越。物华天宝,龙光射牛斗之墟;人杰地灵,徐孺下陈蕃之榻。雄州雾列,俊采星驰。台隍枕夷夏之交,宾主尽东南之美。都督阎公之雅望,棨戟遥临;宇文新州之懿范,襜帷暂驻。十旬休假,胜友如云;千里逢迎,高朋满座。腾蛟起凤,孟学士之词宗;紫电青霜,王将军之武库。家君作宰,路出名区;童子何知,躬逢胜饯。

时维九月,序属三秋。潦水尽而寒潭清,烟光凝而暮山紫。俨骖騑于上路,访风景于崇阿;临帝子之长洲,得天人之旧馆。层峦耸翠,上出重霄;飞阁流丹,下临无地。鹤汀凫渚,穷岛屿之萦回;桂殿兰宫,即冈峦之体势。

图 6 原始文本图像

Fig. 6 Original text image

豫章故郡,洪都新府。星分翼轸,地接衡庐。襟三江而带五湖,控蛮荆而引瓯越。物华天宝,龙光射牛斗之墟;人杰地灵,徐孺下陈蕃之榻。雄州雾列,俊采星驰。台隍枕夷夏之交,宾主尽东南之美。都督阎公之雅望,棨戟遥临;宇文新州之懿范,襜帷暂驻。十旬休假,胜友如云;千里逢迎,高朋满座。腾蛟起凤,孟学士之词宗;紫电青霜,王将军之武库。家君作宰,路出名区;童子何知,躬逢胜饯。

时维九月,序属三秋。潦水尽而寒潭清,烟光凝而暮山紫。俨骖騑于上路,访风景于崇阿;临帝子之长洲,得天人之旧馆。层峦耸翠,上出重霄;飞阁流丹,下临无地。鹤汀凫渚,穷岛屿之萦回;桂殿兰宫,即冈峦之体势。

图 7 嵌入水印文本图像

Fig. 7 Text image with watermark embedded

即冈峦之体势

图 8 原始文本图像的局部放大图

Fig. 8 Enlarged image of the original text image

即冈峦之体势

图 9 嵌入水印文本图像的局部放大图

Fig. 9 Enlarged image of embedded watermark text image

利用本文所提算法,图 8 中连通分量“口”需要减少黑色像素,在图 9 中部分凸起像素被去掉;图 8 中连通分量“冂”需要增加黑色像素数量,在图 9 中部分凹陷像素被填充。通过对比嵌入水印前后的文本图像,可以发现使用本文算法在文本图像中嵌入水印后,字符失真度非常小,视觉效果良好。

使用 PSNR 值 R_{PSN} 客观评价本文算法的不可见性。使用本文算法、文献[9]、文献[11]和文献[12]算法分别在相同数据集的所有文本图像中嵌入相同的水印图像,通过嵌入前后的文本图像的 R_{PSN} 值对比四种算法的不可见性。不同字体字号的文本图像实验结果如表 1 所示。

表 1 不可见性比较

Table 1 Comparison of invisibility

字体字号	文献[9] R_{PSN}/dB	文献[11] R_{PSN}/dB	文献[12] R_{PSN}/dB	本文 R_{PSN}/dB
宋体四号	26.44	27.64	29.79	29.56
隶书小四	26.94	28.91	31.02	30.71
楷体小四	27.59	28.19	30.97	31.48
TNR 12pt	27.41	26.31	29.10	29.35
Century 11pt	25.84	26.97	28.54	29.10
Arial 11pt	26.50	28.27	25.39	24.68
平均值	26.79	27.72	29.14	29.15

从表 1 中可以看出,本文算法平均 R_{PSN} 值分别在文献[9]、文献[11]和文献[12]的基础上提高了 8.81%、5.16%和 0.03%,说明本文算法的不可见性更好。本文算法和文献[12]算法相比, R_{PSN} 值几乎相等,因为本文算法和文献[12]算法都通过修改轮廓嵌入水印。除在 Arial 11pt 字体字号的文本图像中,本文算法和文献[12]算法的 R_{PSN} 值都高于文献[9]和文献[11]算法。说明除 Arial 11pt 字体,通过修改轮廓嵌入水印给文本带来的视觉损失更小,水印不可见性更高。Arial 字体相对其他字体来说字符边界更加平滑,轮廓可修改地方更少。实验结果证明,本文算法可以保证水印视觉不可见。

3.3 水印鲁棒性

使用误码率 b_{ER} 值客观评价本文算法的鲁棒性。使用本文算法、文献[9]、文献[11]和文献[12]算法分别在相同数据集的所有文本图像中嵌入相同的水印图像,计算经过攻击之后提取水印的 b_{ER} 值,进而验证本文算法的鲁棒性。

3.3.1 打印扫描攻击

使用 HP LaserJet 3055 打印扫描一体机打印文本图像,并以 600 dpi 分辨率扫描成文本图像,然后使用对应的提取算法提取水印,计算 b_{ER} 值。实验结果如表 2 所示。

表 2 打印扫描攻击的鲁棒性比较

Table 2 Comparison of robustness against print-scan attack

字体字号	文献[9] $b_{ER}/\%$	文献[11] $b_{ER}/\%$	文献[12] $b_{ER}/\%$	本文 $b_{ER}/\%$
宋体四号	10.93	7.31	7.41	5.84
隶书小四	12.55	6.55	6.81	5.24
楷体小四	10.91	6.75	6.55	4.58
TRN 12pt	9.93	6.42	7.57	6.00
Century 11pt	11.92	6.88	8.37	5.77
Arial 11pt	11.81	7.35	7.80	6.69
平均值	11.34	6.88	7.42	5.69

从表 2 中可以看出,本文算法平均 b_{ER} 值分别在文献[9]、文献[11]和文献[12]的基础上降低了 49.86%、17.3%和 23.34%,说明本文算法对于打印扫描攻击的抵抗能力更强。这是因为本文算法使用了多轮廓像素翻转策略,还重复嵌入水印,提高了水印鲁棒性。在 6 种字体字号的文本图像中,本文算法在楷体文本图像中表现最佳,说明本文算法更适用于楷体文本中。实验结果证明,本文算法提高了水印对于打印扫描攻击的鲁棒性。

需要说明是,不同的打印扫描分辨率对水印提取的准确率会有一些影响,但从算法效果的对比来看,本文算法与对比算法在不同分辨率下进行 b_{ER} 的比较,呈现的优劣程度基本一致,因此,不失一般性,本文仅就 600 dpi 分辨率下的实验结果进行展示与分析。

3.3.2 其他攻击

本文除了重点针对打印扫描攻击的鲁棒性进行系统实验和深入分析外,也对其它算法对于常见图像处理的抵抗能力进行验证。对使用不同算法嵌入相同水印的文本图像分别进行旋转、噪声、压缩、缩放处理,然后使用对应的提取算法提取水印,计算 b_{ER} 值。实验结果如图 10 所示。

从图 10 中可以看出,对于旋转攻击和噪声攻击,本文算法的平均误码率低于文献[9]、文献[11]和文献[12],说明本文算法的水印对旋转和噪声攻击的鲁棒性更好。这是因为本文算法设置了文本倾斜矫正和过滤过小连通分量,并且使用水印压缩预处理增加水印嵌入的次数,结合 DCT 和多轮廓像

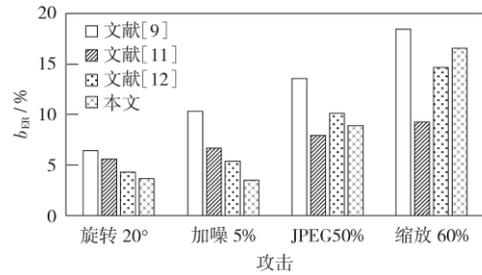


图 10 其他攻击的鲁棒性比较

Fig. 10 Comparison of robustness against other attacks

素翻转策略嵌入水印,提高了水印的鲁棒性。对于压缩攻击,本文算法的表现虽然比文献[11]差,但是优于文献[9]和文献[12]。对于缩放攻击,本文算法的平均误码率高于文献[11]和文献[12],但低于文献[9]。这是因为缩放攻击使得轮廓的细节信息丢失,从而使得特征矩阵的 DCT 系数改变,水印被改变。实验结果证明,本文算法可以抵抗合理范围内的旋转、加噪、压缩等攻击,在抵抗缩放攻击方面,本文算法的性能还有待进一步提高。

4 结束语

本文提出打印扫描不变的多轮廓像素翻转文本图像水印算法。为了提高水印鲁棒性,首先在嵌入水印之前使用水印压缩预处理方法压缩水印图像,在总嵌入容量一定的情况下,增加水印嵌入的次数;然后提出多轮廓像素翻转策略,进一步提高水印鲁棒性;最后基于打印扫描不变量,结合离散余弦变换和多轮廓像素翻转策略,设计水印嵌入及提取算法,尽可能地提高水印鲁棒性。在文献[9]、文献[11]和文献[12]的基础上,嵌入水印前后的平均 R_{PSN} 值分别提高了 8.81%、5.16%和 0.03%,打印扫描攻击后的平均 b_{ER} 值分别降低了 49.86%、17.3%和 23.34%。经实验验证,本文算法在保证水印不可见的前提下,提高了水印对打印扫描攻击的鲁棒性,并且水印可以抵抗合理范围的旋转、噪声、压缩等攻击。本文算法在嵌入水印的过程中,多次使用迭代方法,导致算法运行时间长,并且对于缩放攻击的鲁棒性有待提高,下一步将就提高算法运行效率和对缩放攻击的抵抗能力进行深入研究。

参考文献:

- [1] KAMARUDDIN N S, KAMSIN A, POR L Y, et al. A review of text watermarking: theory, methods, and applications[J]. IEEE Access, 2018, 6: 8011-8028.
- [2] ALOTAIBI R A, ELREFAEI L A. Text-image watermarking based on integer wavelet transform (IWT) and discrete cosine transform (DCT)[J]. Applied Computing and Informatics, 2019, 15(2): 191-202.
- [3] LAOUAMER L, TAYAN O. A semi-blind robust DCT watermarking approach for sensitive text images[J]. Arabian Journal

- for Science & Engineering,2015,40(4):1097-1109.
- [4] BRASSIL J T,LOW S,MAXEMCHUK N F,et al. Electronic marking and identification techniques to discourage document copying[C]//Proceedings of INFOCOM 94. Toronto,Ontario:IEEE,1994:1278-1287.
- [5] BRASSIL J T,LOW S,MAXEMCHUK N F. Copyright protection for the electronic distribution of text documents[J]. Proceedings of the IEEE,1999,87(7):1181-1196.
- [6] SUN X M,CHEN H W,LI H,et al. Mathematical representation of a Chinese character and its applications[J]. International Journal of Pattern Recognition and Artificial Intelligence,2002,16(6):735-747.
- [7] YAZDANI V,DOOSTARI M A,YAZDANI H. A new method to persian text watermarking using curvaceous letters[J]. J Basic Appl Sci Res,2013,3(4):125-131.
- [8] WU M,LIU B. Data hiding in binary image for authentication and annotation[J]. IEEE Trans Multimedia,2004,6(4):528-538.
- [9] 亓文法,李晓龙,杨斌,等. 用于信息追踪的文本水印算法[J]. 通信学报,2008,29(10):183-190.
- QI W F,LI X L,YANG B,et al. Document watermarking scheme for information tracking[J]. Journal of Communications,2008,29(10):183-190.
- [10] HE B,WU Y Q,KAI K,et al. A robust binary text digital watermarking algorithm for print-scan process[C]//2009 WRI World Congress on Computer Science and Information Engineering. Los Angeles,CA:IEEE,2009:290-294.
- [11] 姚寒冰,魏木鑫,周俊伟,等. 打印-扫描不变量结合双域的文本水印算法[J]. 华中科技大学学报(自然科学版),2018,46(6):127-132.
- YAO H B,WEI M X,ZHOU J W,et al. Document watermarking algorithm combined with print-scan invariants and double domain[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition),2018,46(6):127-132.
- [12] TAN L N,HU K,ZHOU X M,et al. Print-scan invariant text image watermarking for hardcopy document authentication [J]. Multimedia Tools and Applications,2019,78(10):13189-13211.
- [13] 刘勇. 二值图像压缩编码算法的若干研究[D]. 济南:山东大学,2009.

(编辑:庞富祥)