

基于交叉划分的灰度图像篡改检测与恢复算法

马巧梅¹, 邓启森^{1,2}

(1 中北大学 计算机与控制工程学院, 山西 太原 030051; 2 山西运城市统计局, 山西 运城 044000)

摘 要: 针对单像素篡改检测算法和分块篡改检测算法的缺点, 提出了基于交叉划分的篡改检测与恢复算法. 根据图像内容生成的 2×2 分块水印被两次嵌入到不同的对应块中, 两次分块方式采用交叉划分方式. 在水印嵌入时采用混沌映射序列对水印进行了加密. 根据不同的分块位置进行两次篡改检测, 并根据两次检测结果交叉后逐步求精, 得到最终检测结果. 图像恢复采用像素级恢复方式, 先确定水印嵌入区域是否被篡改然后再根据两次嵌入的信息进行恢复. 实验结果表明, 该算法篡改检测定位精度可以达到像素级, 恢复图像质量好, 能够抵抗矢量攻击和噪声攻击.

关键词: 交叉划分; 逐步求精; 篡改检测与恢复; 混沌映射; Torus 映射

中图分类号: TP39

文献标识码: A

文章编号: 1000—7180(2014)09—0096—05

Algorithm for Gray Image Temper Localization and Recovery Based on Intersection Division

MA Qiao-mei¹ DENG Qi-sen^{1,2}

(1 College of Computer and Control Engineering, North University of China, Taiyuan 030051, China;

2 Yuncheng Municipal Bureau of Statistics, Yuncheng 044000, China)

Abstract: Aiming at disadvantages of pixel-level temper detection algorithm and block-level temper detection algorithm, a temper detection algorithm based on intersection division is been proposed in this paper. Watermarking is produced according to image for blocks of 2×2 pixels and embedded in various blocks two times. These blocks of the first division are intersected with these blocks of the second division. Watermarking is been encrypted by chaotic sequence. Twice temper detections are processed according to various block location. The last detection result is obtained using the twice detection results by stepwise refinement method. Image recovery is pixel-level. In order to enhance recovery image quality, temper area is twice recovered using embedded information in watermarking block after the watermarking block is judged to be reality. The proposed algorithm can reach pixel-level temper localization precision. Experiment results show that the algorithm has good recovery performance and may resist noise and vector attack.

Key words: intersection division; stepwise refinement; temper detection and recovery; chaotic map; Torus map

1 引言

1999 年 Fridrich 等人^[1]首次提出了一种灰度图像自嵌入水印算法, 该算法不仅能定位灰度图像被篡改的位置, 而且还能近似恢复被篡改图像块的内容, 文献[2]中研究了彩色图像的篡改与检测, 文

献[3]中对 3D 水印进行了研究. 从定位精度上, 对于灰度图像的篡改和检测算法大致可分为基于单像素的和基于分块的数字图像认证算法. 基于单像素的数字图像认证算法^[4]对不同的像素可能产生相同的信息, 容易产生误判. 基于分块的认证算法^[5-7]可能造成虚警. 被检测到的篡改区域大于实际篡改区

收稿日期: 2013-11-03; 修回日期: 2013-12-30

基金项目: 中北大学 2012 年自然科学基金

域.对误判为篡改的真实图像块“恢复”严重降低了篡改恢复图像的质量.针对上述问题,本文提出的用于图像篡改恢复的认证水印方案分为水印生成和嵌入、篡改定位和恢复三个部分.

2 认证水印生成与嵌入

认证水印的水印与嵌入分两次进行,两次水印嵌入方法基本相似,但是加密的参数和位置不同.

2.1 第一次水印嵌入

水印的嵌入过程示意图如图1所示.

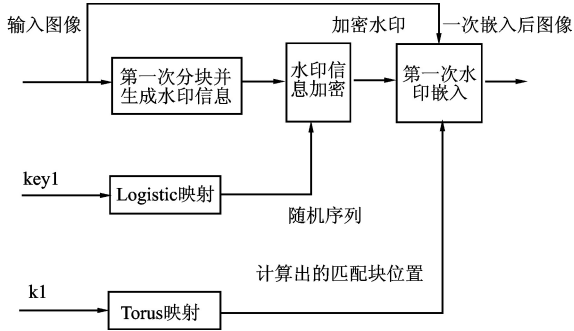


图1 第一次水印嵌入示意图

具体的步骤如下:

(1)首先生成水印信息.

将 $M \times M$ 的原始图像从第1行第1列开始按 2×2 大小不重叠分块 $A_i (i = 1, 2, \dots, M/2 \times M/2)$, 计算每个图像块中4个像素点高5位的灰度均值, 取平均值二值化后的高5位作为该块的水印信息. 第 i 块中的水印信息设为 $A_{ik} (k = 8, 7, 6, 5, 4)$, 并根据 $A_{ik} (k = 8, 7, 6, 5, 4)$ 中为1的个数生成校验位 A_{i3} , 当 $A_{ik} (k = 8, 7, 6, 5, 4)$ 为1的个数为奇数时 A_{i3} 为1, 否则为0. 在密钥 key_1 控制下, 由 Logistic 映射生成二值伪随机序列 $S_{ik} (i = 1, 2, \dots, M/2 \times M/2, k = 8, 7, 6, 5, 4, 3)$, 然后对 $M/2 \times M/2$ 块的水印加密得到水印信息 W , 如式(1)所示:

$$w_{ik} = A_{ik} \oplus S_{ik} \quad (1)$$

式中, $i = 1, 2, \dots, M/2 \times M/2, k = 8, 7, 6, 5, 4, 3$.

(2) 计算用来嵌入每块的水印信息的匹配块

为了避免冲突, 计算每个块的匹配块必须采用双射函数. 根据 Torus 自同构映射一维变换公式计算对应的匹配块, 计算公式如式(2)所示:

$$X' = (k_1 \times X) \bmod N + 1 \quad (2)$$

式中, X, X' 为块序号, N 为块总数, $k_1 \in [0, N-1]$, 必须为素数, 属于私有密钥. 当密钥 k_1 为素数时, X 和 X' 满足一一映射关系, 否则为多对一映射关系.

(3) 水印嵌入

每块的水印信息 W 包括6位 ($W_{ik}, k = 8, 7, 6, 5, 4, 3$), 嵌入到匹配块的1-LSB四位, 剩下的两位嵌入到3-LSB第一行中两个像素位.

2.2 第二次水印嵌入

第二次水印嵌入和第一次水印嵌入的步骤基本相同. 不同是计算水印信息时加密密钥 (key_2) 不同、分块起始位置不同、查找匹配块的密钥 (k_2) 不同以及嵌入水印的位置不同.

(1)第二次分块

第二次分块时与第一次分块的采用交叉划分方式, 第二次分块从原始图像的第二行第二列开始把图像分成 2×2 的图像块. 在第二次分块时图像的第一行和最后一行, 以及图像的第一列和最后一列没有参与分块, 所以第二次的总块数为 $(M/2-1) \times (M/2-1)$. 两次分块形成交叉, 如图2所示.

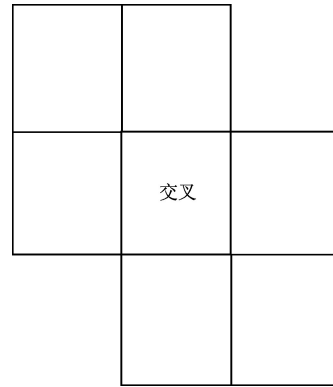


图2 两次交叉分块

两次划分得到的是不同的块, 第一次的第 i 块和第二次的第 i 块形成交叉后交叉结果是单像素, 在篡改定位时提高了定位精度. 而且两次划分的第 i 块是相近块, 在恢复图像时可以相互借鉴, 如果第一次嵌入的水印被修改, 还可以利用第二次嵌入的水印近似恢复这个块.

根据分块后求得水印信息, 从安全性角度考虑, 在采用由 Logistic 映射生成二值伪随机序列时采用 key_2 进行加密.

(2)计算匹配块

根据 Torus 自同构映射一维变换公式计算匹配块, 素数 k_2 采用与第一次嵌入不同的素数进行匹配块的计算, 这样避免了两次划分的相近第 i 块的水印信息同时被修改.

(3)第二次水印嵌入

每块的水印信息包括6位, 第一次嵌入已经占用了图像的1-LSB和3-LSB的两位, 所以第二次

的水印信息嵌入到匹配块的 2-LSB 四位,剩下的两位嵌入到 3-LSB 第一行中两个像素位。

由于两次划分的起始位置不用,两次水印嵌入完成后,在 1-LSB 位平面上和 3-LSB 的奇数行是第一次嵌入的信息,在 2-LSB 位平面上和 3-LSB 的偶数行是第二次嵌入的信。

3 篡改检测

为提高篡改恢复质量,水印算法需准确判定图像块的真实性和真实性。因为篡改块未被检测或真实图像块被误判为篡改,将导致恢复图像中存在未被恢复或被错误恢复的图像块,从而降低图像的恢复质量。本文根据交叉划分嵌入时的嵌入方法和参数,进行两次检测,既提高了定位精度,有提高了检测准确性。

3.1 第一次检测

将 $M \times M$ 的待检测图像 G 从第 1 行第 1 列开始按 2×2 大小不重叠分块,依次判断各个块是否被篡改。一般的检测是根据第 i 块生成的水印信息和嵌入在匹配块 i' 中的水印信息是否相等进行检测,如果 i' 被篡改,也会被判定第 i 块被篡改,造成了虚警。由于为了降低虚警率,本文在判断时加入了对匹配块中水印信息是否可信的判断和多次去除孤立点的判断,具体的检测算法如下:

(1) 对第 i 块图像根据前面生成水印的方法和参数 key_1 求得该块的水印信息 W ;

(2) 并根据密钥 k_1 求得对应的匹配块 i' ,从匹配块中提取对应的嵌入水印 W' ;

(3) 根据 W 和 W' 进行判断,并把检测结果写入到检测结果图 C 中:

如果 W 和 W' 相同,则第 i 块的四个像素点都标志为 0,确定为没有被篡改;

如果 W 和 W' 不同,再根据 W' 的最低位 W'_3 和 $W'_8, W'_7, W'_6, W'_5, W'_4$ 中为 1 的个数是否相符判定匹配块的嵌入水印 W' 是否被修改,如果相符则可以用来检测第 i 块是否被篡改,则 W 和 W' 不同是由于第 i 块被篡改引起的,第 i 块的四个像素点都标志为 255,表示被篡改;

如果 W 和 W' 不同,而且 W' 的最低位 W'_3 和 $W'_8, W'_7, W'_6, W'_5, W'_4$ 中为 1 的个数不符,那么 W 和 W' 不同可能是第 i 块被篡改,也可能是第 i' 块被篡改,也可能是噪声引起 W' 误差,也可能是两块都被篡改,所以第 i 块的四个像素点都标志为 128,表示为未确定。

根据以上算法,生成大小和原图像大小相同的

检测结果图。为了降低漏检率,对于检测结果图再次扫描检测,对没有被篡改的块进行是否孤立块的判断。扫描所有的标志为 0 的块,如果该块的 8 邻域都是 255,则该块为漏检块,则修改该块中的四个像素标志都为 255,最后得到第一次检测结果图 C 。

3.2 第二次检测

将 $M \times M$ 的待检测图像从第 2 行第 2 列开始按 2×2 大小不重叠分块,依次判断各个块是否被篡改。第二次检测算法和第一次检测算法相同,只是参数不同。对第 i 块图像根据前面第二次嵌入的参数生成水印的方法求得该块的水印信息 W ,并根据密钥 k_2 求得对应的匹配块 i' ,从匹配块中提取对应的嵌入水印 W' 。具体的检测算法与第一次检测相同,生成大小和原图像大小相同的检测结果图 D 。

3.3 逐步求精检测

两次检测完成后,结合两次检测结果,得到最后的检测结果。

(1) 首先合并两次检测的结果 C 和 D 得检测图像 E ,具体处理算法如下:

如果像素点 $C(i, j)$ 为 255 且 $D(i, j)$ 为 255,则 $E(i, j)$ 为 255;

如果像素点 $C(i, j)$ 为 255 且 $D(i, j)$ 为 0,则 $E(i, j)$ 为 0;

如果像素点 $C(i, j)$ 为 255 且 $D(i, j)$ 为 128,则 $E(i, j)$ 为 255;

如果像素点 $C(i, j)$ 为 0 且 $D(i, j)$ 为 255,则 $E(i, j)$ 为 0;

如果像素点 $C(i, j)$ 为 0 且 $D(i, j)$ 为 0,则 $E(i, j)$ 为 0;
如果像素点 $C(i, j)$ 为 0 且 $D(i, j)$ 为 128,则 $E(i, j)$ 为 0;

如果像素点 $C(i, j)$ 为 128 且 $D(i, j)$ 为 255,则 $E(i, j)$ 为 255;

如果像素点 $C(i, j)$ 为 128 且 $D(i, j)$ 为 0,则 $E(i, j)$ 为 0;

如果像素点 $C(i, j)$ 为 128 且 $D(i, j)$ 为 128,则 $E(i, j)$ 为 128。

(2) 然后对图像 E 中不确定的点(检测结果为 128)和可能误判的点(检测结果为 255 的孤立点)根据它的邻域进行逐步求精判断。判断检测值为 255 的像素点,如果该点为孤立点(周围 8 邻域都是 0),改变检测结果为 0;对于不确定像素点可能是该像素被篡改,也可能是水印匹配的像素块被篡改,也可能是噪声引起误差,也可能是两块都被篡改。由于算法是基于密钥随机生成图像块的水印嵌入位置,如果是由于水印匹配的像素块导致该像素点不确定,则周围被篡

改图像块的个数小于对应匹配块周围被篡改图像块的个数,否则周围被篡改图像块的个数大于匹配块周围被篡改图像块的个数.通过这一方法,把不确定像素进行确定,得到最终检测结果图 F .

4 篡改区域恢复

由于检测结果图 F 是两次块检测后进行逐步求精处理,检测篡改结果已经精确到像素级,所以恢复时以像素为单位进行恢复.

4.1 第一次恢复

假设待检测图像为 G ,一次恢复后图像为 H ,恢复结束后同时生成一个与恢复图像大小相等的图像 P ,对于没有修改的像素点和第一次恢复过的篡改点标志位 0,没有恢复的篡改点标志位 255,等待二次恢复.

将 $M \times M$ 的检测结果图像 F 从第 1 行第 1 列开始逐个对所有的像素点进行分析和恢复,具体的恢复算法如下:

如果 $F(i, j)$ 为 0,恢复图像中的像素 $H(i, j)$ 等于待检测图像的像素 $G(i, j)$,并标注 $P(i, j)$ 为 0;

如果 $F(i, j)$ 为 255,而且根据 k_1 求出的 F 中对应的匹配块中四个像素点都没有被篡改,则在待检测图像 G 中提取匹配块中水印信息并解密,恢复图像中的像素 $H(i, j)$ 的高 5 位等于水印中提取的解密信息前 5 位,并标注 $P(i, j)$ 为 0;

如果 $F(i, j)$ 为 255,而且根据 k_1 求出的 F 中对应的匹配块中四个像素点有被篡改,恢复图像中的像素 $H(i, j)$ 等于待检测图像的像素 $G(i, j)$,并标注 $P(i, j)$ 为 255;这个像素点需要在第二次恢复中进行处理.

4.2 第二次恢复

根据图像 G 、图像 H 、图像 F 和图像 P ,进行二次恢复.二次恢复后的图像为 I ,恢复结束后同时生成一个与恢复图像大小相等的图像 Q ,对于不需要二次恢复和二次恢复过的篡改点标志为 0,没有被恢复的篡改位标志为 255.

对图像从第 2 行第 2 列到第 $M-1$ 行 $M-1$ 列开始逐个对所有像素进行分析,具体的恢复算法如下:

如果 $P(i, j)$ 为 0,恢复图像中的像素 $I(i, j)$ 等于 $H(i, j)$,并标注 $Q(i, j)$ 为 0;

如果 $P(i, j)$ 为 255,而且根据 k_2 求出的 F 中对应的匹配块中四个像素点都没有被篡改,则在图像 G 中提取匹配块中水印信息并解密,恢复图像中的像素 $I(i, j)$ 的高 5 位等于水印中提取的解密信息的

前 5 位,并标注 $Q(i, j)$ 为 0;

如果 $P(i, j)$ 为 255,而且根据 k_2 求出的 F 中对应的匹配块中四个像素点有被篡改,恢复图像中的像素 $I(i, j)$ 等于待检测图像的像素 $G(i, j)$,并标注 $Q(i, j)$ 为 255;

由于第二次恢复没有扫描第 1 行、第 M 行、第 1 列、第 M 列,对于这些位置的像素 $Q(i, j)$ 等于 $P(i, j)$.

4.3 恢复剩余像素点

观察图像 Q ,对于 $Q(i, j)$ 等于 255 的像素,则 $I(i, j)$ 是篡改后没有被恢复的像素点,对于这些像素点采用 8 邻域的平均值进行近似恢复.

5 仿真实验与性能分析

现有恢复水印算法都能检测一般篡改,其差别主要体现在定位精度、篡改检测性能和篡改恢复图像质量的高低.

5.1 细节修改检测与恢复

首先对图像进行多处字体修改,对于这类修改,对检测和恢复的精度要求较高.为了验证本文算法对于细节修改的检测与恢复性能,分别采用本文方案与文献[5-6]的方法进行篡改检测与恢复仿真实验.

从实验结果可以看出,本文算法对于边缘细节比较曲折的篡改检测结果非常清晰,精确到像素值.与文献[8-9]相比,没有块状结构,漏检率和虚警率低,恢复结果也很好.

5.2 区域篡改检测与恢复

对 plane 图像分别进行 5%、10%、20%、40%、50% 的区域篡改,从检测结果和恢复图像.可以看出,篡改区域为 5% 时,完全能够恢复原图像,随着篡改区域的不断增大,虚警率也在增大,恢复图像会出现不能恢复的像素.篡改区域达 50% 时,有明显的虚警区域,恢复的图像质量也不高,但是由于恢复信息是分两次且分散嵌入在图像中,所以恢复的图像仍然能够看出原图的主要信息.

5.3 拼贴攻击

将为本文与文献[5-6]的检测与篡改结果比较.

从实验结果可以看出,本文对区域篡改具有良好的检测结果和恢复能力.

5.4 噪声攻击

在嵌入水印的图像中,加入椒盐噪声后,再进行矩形方块区域的篡改.观察图像遇到噪声干扰时,篡改检测和恢复仿真结果.

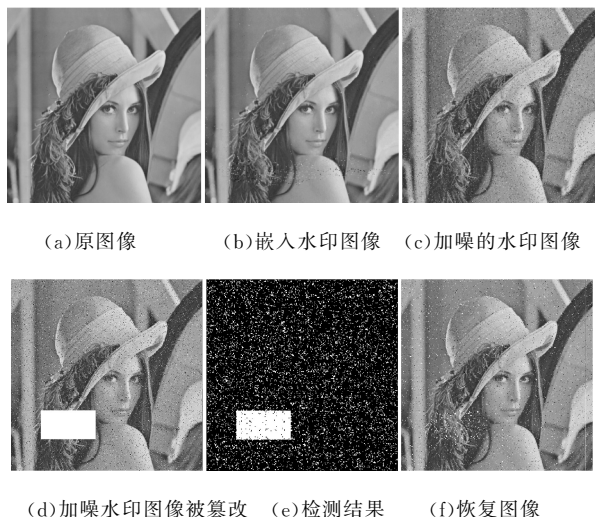


图3 抵抗噪声干扰

从图3可以看出,嵌入水印的图像被噪声干扰后被篡改,尽管虚警率和漏检率存在,仍然能够基本正确地显示篡改的矩形区域和噪声点状。从恢复图像可以看出,尽管不能够完全去除噪声,但是对于大块篡改的面积基本能够恢复。这一算法可以解决网络传输中噪声干扰的问题。

5.5 矢量攻击

矢量量化攻击是通过搜集大量使用同一密钥的作品作为的数据库,并将没有认证的图像分块,从数据库中用最相似的块代替构造新的作品,篡改后的作品仍能通过认证。

本文算法采用经过两次交叉分块,得到的块相互交叉的块。同时采用不同的密钥的混沌序列对图像块生成的水印信息进行加密处理,即利用混沌映射的非周期、不收敛并对初始值非常敏感的特性加密水印信息,对于水印嵌入图像块位置采用Torus映射求解,这些措施使水印信息在不确定的位置上嵌入,攻击者很难采用穷举法获得密钥,可以对抗矢量量化攻击。

6 结束论

本文对图像进行 2×2 分块将水印信息嵌入到原始图像所有像素的3位LSB,为了消除块效应和提高篡改检测与恢复能力,水印的嵌入分两次进行,两次分块的方式采用交叉方式,使得定位精度可以达到像素级,并且在检测时通过两次检测、逐步求精,进一步保证了检测精度。为了提高图像恢复能力,图像恢复时采用的像素级的恢复方式,提高了恢复图像的质量。图像检测认证时不需要原始图像,只需要知道密钥就可以实现篡改检测、定位和近似恢复。本文方案同时还可以

抵抗矢量攻击和噪声攻击。由于检测精度的好坏很大程度上取决于虚警率和漏检率,所以下一步研究的重点仍然是降低虚警率和漏检率。

参考文献:

- [1] Fredich J, Goljan M. Images with self-correcting capabilities[C]// Proc ICIP'99, Kobe: IEEE Press, 1999: 25-28.
- [2] Wang Shi-Ming, Chen Wei-Che. A majority-voting based watermarking scheme for color image tamper detection and recovery[J]. Computer Standards and Interfaces, 2007, 29(5): 561-570.
- [3] 陈永府, 罗易. 3D 网格脆弱水印中孤立水印顶点研究[J]. 计算机应用与软件, 2012, 29(7): 239-242.
- [4] Liu Shao-Hui, Yao Hong-Xun, Gao Wen. An image fragile watermark scheme based on chaotic image pattern and pixel-pairs [J]. Applied Mathematics and Computation, 2007, 185(2): 869-882.
- [5] Tong Xiao-Jun, Liu Yang, Zhang Miao, et al. A novel chaos-based fragile watermarking for image tampering detection and self-recovery[J]. Signal Processing: Image Communication, 2013(28): 301-308.
- [6] Huo Yao-Ran, He Hongjie, Chen Fan. Alterable-capacity fragile watermarking scheme with restoration capacity[J]. Optics Communication, 2013, 28(5): 1759-1766.
- [7] 陈帆, 和红杰, 王宏霞. 用于图像认证的变容量恢复水印算法[J]. 计算机学报, 2012, 35(1): 154-162.

作者简介:



马巧梅 女, (1969-), 博士, 副教授. 研究方向为信息安全, 图像处理.



邓启森 男, (1987-), 中级统计师. 研究方向: 计算机应用.