

# 局部化数字水印算法

华先胜<sup>1)</sup> 石青云<sup>2)</sup>

<sup>1)</sup>(北京大学数学科学学院信息科学系, 北京 100871)

<sup>2)</sup>(北京大学信息科学中心视觉与听觉处理国家重点实验室, 北京 100871)

**摘要** 数字水印是一种嵌入到图象、视频或音频数据中的不可见标志, 可以用于多媒体数据的版权保护、认证和标注等。为了提高在频率域嵌入水印抵抗裁剪攻击的能力, 提出了一种局部化的图象数字水印算法, 该算法利用图象中相对稳定的特征点标示水印嵌入的位置, 并在与每个特征点对应的局部区域中独立地嵌入水印。这样, 当只有部分图象时, 仍能通过这些特征点来定位并提取水印。此算法中, 水印的嵌入是在局部图象的小波域中进行的, 并采用对小波系数进行特殊量化的方法来隐藏水印比特, 而水印的提取不需要原始图象参与。实验结果证明, 算法对裁剪有很强的抵抗能力, 同时对压缩、滤波、噪声、StirMark攻击等也有较好的鲁棒性。

**关键词** 信息隐藏 数字水印 局部化

**中图分类号**: TP391.4 TP309.2 **文献标识码**: A **文章编号**: 1006-8961(2001)07-0642-06

## Localized Watermarking Scheme

HUA Xian-sheng<sup>1)</sup>, SHI Qing-yun<sup>2)</sup>

<sup>1)</sup>(School of Mathematical Sciences, Peking University, Beijing 100871)

<sup>2)</sup>(National Laboratory on Machine Perception, Center for Information Sciences, Peking University, Beijing 100871)

**Abstract** A digital watermark is an invisible mark embedded in digital images, video or audio documents, which may be used for a number of different purposes including copyright protection, authentication and captioning. In this paper, we proposed a watermarking scheme that embeds watermarks in the local region of the image. This new idea that we name it local watermarking scheme, inserts watermarks into the wavelet domain of local regions of the image instead of entire image. We select some stable feature points in the image, and then use these points to identify the locations where we insert watermarks. When the image is cropped, the feature points in the remaining piece of the image will not change. We can extract watermark with the help of the remaining feature points. A special quantization procedure is used when we embed watermark bits into the wavelet coefficients of the sub-image, and the watermark can be detected without knowing the original image. Experimental results show that the watermark scheme proposed in this paper is robust to cropping, and also has good robustness to compression, filtering, noise adding, and StirMark attack.

**Keywords** Data hiding, Digital watermarking, Localization

## 0 引言

数字水印技术近年来虽发展比较快, 但离实际应用尚有一段距离, 有很多实际问题尚有待解决。例

如, 水印对裁剪、仿射变换以及随机几何形变等的抵抗能力很有限, 而且许多算法在一些简单的攻击下就难以生存<sup>[1]</sup>。虽然嵌入水印的方法很多, 但其中的大多数方法均可大致归入空间域方法和频率域方法两类。一般来说, 空间域的方法对压缩、噪声、滤波等

基金项目: 国家重点基础研究发展规划资助项目(G1998030600)、自然科学基金重点项目(69735020)

收稿日期: 2000-06-15; 改回日期: 2001-05-25

处理的鲁棒性较差,但如果在整幅图象的频率域嵌入水印,则很难抵抗裁剪攻击,因为如果只有部分图象,且在无法获得原始图象的大小,以及这部分图象在原始图象中的位置的情况下,很难定位水印的嵌入位置。

本文试图探讨能抵抗裁剪的图象数字水印算法,即局部化的水印嵌入方案。这里需要说明的是,一些文献中提到的裁剪(或剪切),是指在原图中去掉一部分图象后,在剩下的部分图象中进行水印检测,而这时原始图象的大小和剩余图象的位置都是知道的;本文所指的裁剪,是指在裁剪下来的这部分图象中进行水印检测,这时原图大小和这部分图象在原图中的位置都是未知的。

## 1 数字水印的局部化

所谓图象的局部化数字水印就是将水印信息嵌入图象的多个局部位置。水印检测器通过局部图象就可以提取水印,并判断水印的存在性。这种局部化的水印既可以作为能抵抗裁剪的鲁棒水印使用,也可以用作局部化的标注水印,利用水印的方法在图象的不同位置,不可见地嵌入该位置图象所表示内容的说明信息。

一般来说,局部化水印算法需要通过图象本身的某些特征来辅助水印的嵌入和(或)提取,特别是辅助确定水印嵌入的准确位置。使用空间上的特征点来定位水印是一个比较自然的想法。通过特征点的定位,在与特征点唯一对应的子图象区域中嵌入和提取水印,就可以实现数字水印的局部化。

为实现有效的局部化水印算法,在图象中选取的特征点应满足以下几个条件:

① 具有足够的稳定性。特征点在压缩、滤波、噪声等处理下应有足够的稳定性,即经这些处理后特征点应仍能被检测到,并应该尽量保持在原位置或原位置附近。

② 分布比较均匀。为使水印能抵抗裁剪,特征点在图象中的分布要比较均匀。

③ 与特征点对应的区域应有足够的容纳水印比特的能力。如果与特征点对应的图象区域结构过于简单,其容纳水印的能力必然很小(即容许不可感知或不易感知地改变的幅度),这时嵌入水印会令图象产生较大的视觉损失。

与特征点对应的区域可以以特征点为中心,或

者其他可以由特征点唯一确定的区域。该区域的形状可以是正方形、圆形,或者是正六边形等。此区域的大小对水印鲁棒性有重要影响,较小的区域使水印更适合抵抗裁剪,但局部水印的鲁棒性却会因此降低;相反,较大的区域虽可提高局部水印的鲁棒性,但也可能会使裁剪下来的部分图象中没有完整的可以提取水印的区域。

已经有研究者提出了局部化数字水印的方案或思想,如 Kuter 等人先利用二维 Mexican Hat 小波来提取图象的特征点,然后以特征点为中心对图象进行 Voronoi 分割,在分割后的各部分中再分别嵌入扩频水印<sup>[2]</sup>;又如 Duric 等文献通过寻找多尺度角点(Corner),来恢复被裁剪和仿射变换后的图象<sup>[3]</sup>,但这种方法在恢复图象时需要原始图象特征点的位置作为参考;文献[4]中虽提出了在图象的兴趣区域嵌入水印的算法,但算法仍然需要对整幅图象作小波变换,从而不具备抵抗裁剪的能力。

本文提出的局部化数字水印算法虽与文献[2]、[3]中的算法有相似之处,但本算法是在空间域上寻找特征点,并在与特征点对应区域的小波变换域中嵌入水印,水印的检测不需要任何关于原始图象的信息。

## 2 特征点的提取

选择图象边缘中的角点(包括叉型点、Y型点和T型点)作为特征点,因为这种点有较好的稳定性。特征点的提取分以下几步进行:

① 用 Canny 算法<sup>[5]</sup>(或其他边缘提取算法)来提取图象边缘;

② 检测边缘图中的角点,作为候选点;

③ 从候选点中,按一定的条件选取一定数量的特征点。

为使选出的特征点符合要求,需要为候选点定义一个“可信度”函数:

$$Cred(p) = k_1 average(p, m) + k_2 var(p, n) + k_3 type(p) \quad (1)$$

其中,  $average(p, m)$  表示  $p$  点的  $m \times m$  邻域内边所有边缘点边缘强度的平均值;  $var(p, n)$  表示  $p$  点的  $n \times n$  邻域内象素值的方差;  $type(p)$  则根据该特征点类型来取值,即当  $p$  为叉型点时取值最大,其次是 Y 型点,而当  $p$  为 T 型点时,  $type(p)$  取值最小;  $k_1$ ,  $k_2$  和  $k_3$  为常数因子。为减小计算量,如果边缘强

度小于最大边缘强度的一半,则不认为其是边缘点.

特征点可按如下方案进行搜索挑选:

① 首先找可信度最大的候选点,作为第1个特征点;

② 在剩下的候选点中,寻找与所有已找到的特征点距离不小于  $D$ ,且可信度最大的候选点,作为下一个特征点.

③ 重复②,直到找到足够数目的特征点,或者无法再找到特征点为止.

其中,  $p, p'$  候选点间距离定义为:

$$d(p, p') = \max(|p_x - p'_x|, |p_y - p'_y|) \quad (2)$$

其中,  $(p_x, p_y), (p'_x, p'_y)$  为候选点在图象中的坐标位置.

下面来估计一下从整图中任意裁剪出的一个  $H \times W$  矩形(记为  $R$ )中,所包含的特征点及其对应的区域个数. 设整图大小为  $M \times N$ ,与特征点对应的区域是以特征点为中心,大小为  $D \times D$  的正方形区域. 假定特征点均匀分布,特征点个数为  $K$ ,则  $R$  中任意一点  $(x, y)$  是特征点的概率为

$$p(x, y) = \frac{K}{(M-D)(N-D)} \quad (3)$$

其中,  $D/2 \leq x \leq N-D/2, D/2 \leq y \leq M-D/2$ .

这样,  $R$  中包含的与特征点对应的区域个数,就等于该矩形中所包含的至矩形边界距离不小于  $D$  的特征点个数,即

$$P(M, N, D, K, H, W) = p(x, y) \cdot (H-D)(W-D) = \frac{K(H-D)(W-D)}{(M-D)(N-D)} \quad (4)$$

例如,  $P(512, 512, 96, 12, 256, 256) = 1.78$ ,

$$P(512, 512, 64, 40, 256, 256) = 7.35.$$

较小的  $D$  可使得  $P(M, N, S, M, H, W)$  接近或大于 1. 而且此值越大,水印抵抗裁剪的能力就越强,但同时局部水印的鲁棒性会因此而降低.

### 3 局部化水印的嵌入

找到特征点后,即可以分别在以每个特征点为中心的  $D \times D$  大小的子图象区域中嵌入水印. 虽然可以采用各种不同的方法来这一步,但是,由于子图象比原图尺寸小很多,因此需要使用适合在小图象中嵌入水印的算法. 本文采用量化小波域系数的方法来嵌入水印比特,该方法虽与文献[6]中的方法有类似之处,但在量化方案上又有较大的不同.

水印的嵌入在各个  $D \times D$  子图中都是完全相同的,其嵌入过程如下:

① 对  $D \times D$  子图作 3 层 Haar 小波变换;

② 通过密钥控制,随机选择小波域中除直流分量外的部分系数<sup>[6]</sup>;

③ 在选择的系数中依次嵌入水印比特序列;

水印的嵌入是通过对选择的系数进行特殊的量化来完成的:设第  $l$  层的量化间隔为  $\Delta_l$ ,对系数  $f$  进行量化,嵌入水印比特  $w_l$ ,量化后的系数记为  $f^*$ ,量化过程可由下图描述(图 1):

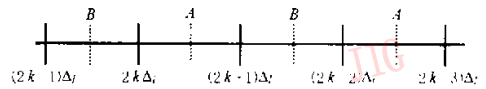


图 1 量化方案

图 1 中虚线所标示的位置可分为 A、B 两类,其对应的值分别为  $A_k = 2k\Delta_l + \Delta_l/2$  和  $B_k = (2k+1)\Delta_l + \Delta_l/2, k = \pm 1, 2, \dots$ . 当  $w_l = -1$  时,  $f$  量化为最与之靠近的 A 类点;当  $w_l = 1$  时,  $f$  则量化为最与之靠近的 B 类点,即

$$f^* = \begin{cases} A_q, q = \arg \min_j |f - A_j|, & \text{if } w_l = -1 \\ B_q, q = \arg \min_j |f - B_j|, & \text{if } w_l = 1 \end{cases} \quad (5)$$

也就是

$$f^* = \begin{cases} \lfloor f/\Delta_l \rfloor \cdot \Delta_l + \frac{\Delta_l}{2}, & \text{if } Q(f, \Delta_l) = w_l \\ \lfloor f/\Delta_l \rfloor \cdot \Delta_l + \frac{3\Delta_l}{2}, & \text{if } Q(f, \Delta_l) \neq w_l, \text{ and } f < 0 \\ \lfloor f/\Delta_l \rfloor \cdot \Delta_l - \frac{\Delta_l}{2}, & \text{if } Q(f, \Delta_l) \neq w_l, \text{ and } f \geq 0 \end{cases} \quad (6)$$

其中,如果  $\lfloor f/\Delta_l \rfloor$  为奇数,  $Q(f, \Delta_l)$  取  $-1$ ; 如果  $\lfloor f/\Delta_l \rfloor$  为偶数,则  $Q(f, \Delta_l)$  取  $+1$ ;

这里的量化嵌入方案与文献[6]中提出的易损水印量化嵌入方案虽有类似之处,但文献[6]中的量化方法只是在原系数上加上或减去若干倍数的量化间隔,而本文的方案则是将小波系数量化在相邻量化间隔的中间位置,即上述 A 或 B 位置,从而使得水印的鲁棒性得以加强.

④ 作逆小波变换,得到嵌入水印的子图象.

对所有与特征点对应的子图进行上述处理,就得到了整个嵌入水印后的图象. 需要说明的是,小波变换并不一定必须在特征点的正方形邻域进行,还可以选择在圆形、正六边形邻域上进行,不过,此时应使用适形小波变换<sup>[7]</sup>.

## 4 局部化水印的提取

水印的嵌入、图象压缩以及其他图象处理可能会使图象的边缘产生一定的漂移,因此,在提取水印时,不能只对以特征点为中心的子图提取水印,还需要对所有的候选点及其一定范围的邻域内的所有点尝试提取水印。只要有一个点与特征点或候选点对应的子图中能检测到水印,就可以认为此图中存在水印。

子图中检测水印的过程为:先对子图作小波变换,然后找到可能嵌入了水印比特  $w$  的系数  $\tilde{w}$ ,从中提取的水印比特为

$$\tilde{w}_i = Q(\tilde{f}_i, \Delta_i) \quad (7)$$

水印的存在性一般可通过相似度函数  $Sim(w, \tilde{w})$  来判断。若设  $w = \{w_i, 1 \leq i \leq L\}$  和  $\tilde{w} = \{\tilde{w}_i, 1 \leq i \leq L\}$  分别为原水印和从待测子图中提取的水印,则二者的相似度定义为

$$Sim(w, \tilde{w}) = \frac{1}{L} \sum_{i=1}^L w_i \tilde{w}_i \quad (8)$$

其中,  $L$  为水印比特序列的长度。在这里,  $Sim(w, \tilde{w})$  其实就等于  $w$  和  $\tilde{w}$  的相关系数。

下面分析一下水印检测的虚警概率。易知,两个均匀随机序列的  $Sim$  值的概率分布为

$$P(Sim = k/L) = 2^{-L} C_L^{L-k} \quad (9)$$

其中,  $k$  取  $-L, -L+2, \dots, L-2, L$ 。当  $L$  较大时,上

式可近似看作高斯分布  $N(0, 1/L)$ , 因此

$$P(Sim \geq T) \approx \Phi(-T\sqrt{L}) \quad (10)$$

当  $L=256$  时

$$P(Sim \geq 0.3) \leq 7.93 \times 10^{-7} \quad (11)$$

所以,如果原水印和提取水印的  $Sim$  值不小于 0.3, 就是可以认为待测图象中存在水印。在所有尝试提取水印的子图象中,得到的最大的  $Sim$  值,即为整个待测图象的最后检测结果。

为了得到更可靠的检测结果,还可以只计算最低频 3 个子带中水印比特的  $Sim$  值,因为低频子带的系数具有更好的鲁棒性,即对压缩、滤波和噪声等处理有更强的抵抗能力。

## 5 模拟实验结果

本文对大小为  $512 \times 512$  的 256 级 Lena 灰度图象进行局部化水印嵌入和提取的实验。实验中,采用 Canny 算法<sup>[5]</sup>来提取边缘,特征点最大数目为 16 个,特征点对应区域的大小为  $96 \times 96$ ;采用 Haar 小波变换,嵌入水印时各层小波系数的量化间隔均为 32。图 2 显示了与所选特征点对应的,用来嵌入水印的子图区域(白色方框)。图 3 是嵌入 256 Bits 水印后的图象,峰值信噪比为 43.039。

图 4 是从图 3 中剪取的几个子图象块,直接对这些图象块进行水印提取,其  $Sim$  值均等于 1,即水印被完全准确提取。

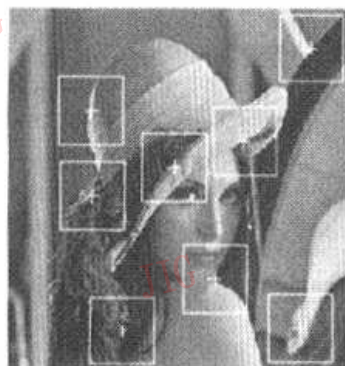


图2 特征点及其对应的子图区域  
(白色方框)



图3 嵌入 256 比特水印后的图象  
(PSNR=43.039)



图4 从已嵌入水印的图象中  
裁剪出来的区域

从理论上讲,只要裁剪下来的部分图象中至少包含一个特征点和与该特征点对应的子图象区域,水印就能被检测到,但是,图象经过压缩和其他图象处理(如滤波、添加噪声等)后,其特征点不一定能全部被检测到,此时可能会造成水印提取失败。

实验中,通过图3和图4(a)测试了水印对JPEG压缩、小波压缩,以及对噪声和中值滤波的抵抗能力,其中,子图象和整幅图象经过JPEG压缩后的水印检测结果如图5所示,图中包括在所有小波分解子带提取和在最低频子带提取两组数据。由图5可见,本文提出的局部化水印算法对JPEG压缩有很强的抵抗能力,特别是在只考查低频部分水印比特的情况下。

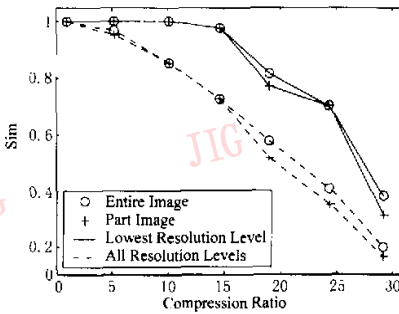


图5 JPEG压缩后的检测结果

表1是整幅图象和子图象经过不同倍数Wavelet压缩后的检测结果(即相似度),其中带\*号的数据是由于子图象中特征点没有被检测到而造成的,即子图象中特征点的丢失使水印的提取失败。进一步的实验表明(表2、表3),本文算法对噪声和轻度滤波也有一定的鲁棒性。

表1 小波压缩后的水印检测结果

压缩倍数	整幅图象		部分图象	
	所有子带	最低频	所有子带	最低频
6.9	1	1	0.984 4	1
11.1	0.890 6	1	0.773 4	1
21.2	0.468 8	0.885 1	0.171 9*	0.241 4*
30.7	0.359 4	0.540 2	0.141 8*	0.264 4*

表2 添加噪声后的水印检测结果

PSNR	整幅图象		部分图象	
	所有子带	最低频	所有子带	最低频
33.6	1	1	1	1
30.1	0.937 5	0.931 0	0.882 8	0.862 1
28.1	0.804 7	0.839 1	0.781 3	0.747 1
24.6	0.406 3	0.540 2	0.351 6	0.517 2

表3 水印对在中值滤波的抵抗能力

滤波器大小	全部图象		部分图象	
	所有子带	最低频	所有子带	最低频
3×3	0.726 6	1	0.601 6	0.977 0
5×5	0.289 1	0.353 6	0.132 8*	0.218 4*

这里,值得提及的是,本文算法在StirMark<sup>[1]</sup>攻击下也有一定的鲁棒性,如整幅图象在缺省参数的StirMark攻击后,从所有子带和最低频子带中提取水印得到的相似度分别为0.570 3和0.793 1,这都明显表示水印存在;对应部分图象,图4(a)在StirMark攻击下得到的Sim值在0.3附近,尚不能明显表示水印存在,而图4(d)得到的相似性则与整图相同。由于本文算法是在全图的多个局部块中嵌入水印,所以在StirMark攻击下,可能存在虽平移了位置但整体变化(MSE)很小的图象块,因此整图对StirMark有一定的鲁棒性,但子图水印的鲁棒性却依赖于子图区域的选取。很明显,如果用来嵌入水印的子图象块的边长 $D$ 越小,那么水印对StirMark攻击的抵抗能力就越强。

## 6 结论

本文提出的局部化嵌入水印的方法对裁剪、压缩、噪声和轻度滤波有很好的抵抗能力,而且整幅图象中的水印对StirMark攻击也有一定的鲁棒性,但子图对StirMark攻击的鲁棒性却有待提高。

还可以从如下3个环节来提高局部化水印的鲁棒性:一是使特征点具有更强的稳定性;二是采用更适合在小幅图象中嵌入水印的算法,以增强局部水印本身的鲁棒性,同时也可能因此减小用来独立嵌入水印的局部区域的大小,以使得抵抗裁剪和StirMark攻击的能力进一步加强;三是设法使水印在能抵抗裁剪的同时,还能抵抗缩放甚至一般的仿射变换。

## 参考文献

- Peticolas F A P, Anderson R J, Kuhn M G. Attacks on copyright marking system. In: Second International Workshop on Information Hiding, Portland, Oregon, U.S.A., April 14~17. Published in: Lecture Notes in Computer Science, 1998, 1525, 218~238.
- Kutter M, Bhattacharjee S K, Ebrahimi T. Towards second generation watermarking schemes. In: Proceedings of 6th