

基于多算法融合的 DCT 域盲水印 FPGA 硬件实现

孟煦, 徐金阁, 马新沂, 徐寅

(合肥工业大学 微电子学院, 合肥 230009)

摘要: 本文设计了一种相关数据库和具备 DCT 系数误差校准功能的与改进的线性内插方法相结合的 DCT 域嵌入算法。实验证明: 其对原图破坏较小 (嵌入后图像的 PSNR 值可达 30, NC 值可达 1.0000), 并且对于 JPEG 有损压缩及其他一些恶意攻击都具有良好的鲁棒性。本文在提出一种适于硬件实现的用于图像版权保护的数字盲水印算法的基础上, 对其进行了 FPGA 验证。本设计采用软硬协同的实现方案, 同时硬件采用流水线结构, 进一步提高了处理速度 (1.6 倍), 非常适合高速信号处理的应用。

关键词: 盲水印; DCT 域; 数据相关性; 线性内插; FPGA 硬件实现

中图分类号: TP309.7

FPGA hardware implementation of blind watermarking in DCT domain based on multi-algorithm fusion

MENG Xu, XU Jinge, MA Xinyi, XU Yin

(Hefei University of Technology)

Abstract: A DCT domain's embedding algorithm combining relational database and improved linear interpolation with DCT coefficient error calibration was designed. This design can realize an embedded algorithm based on the relational database and the improved linear interpolation. Experiments show that it has less damage to the original image (the PSNR value of the embedded image can reach 60, and the NC value can reach 1.0000), and it has good robustness against JPEG lossy compression and other malicious attacks. A digital blind watermarking algorithm suitable for hardware implementation for image copyright protection is designed and verified by FPGA. This design adopts a soft-hard cooperative implementation scheme. And the hardware uses a pipeline structure, further improving the processing speed (1.6 times), which is very suitable for high-speed signal processing applications.

Key words: Blind watermarking; DCT domain; Data dependency; Linear Interpolation; Hardware implementation in FPGA

0 引言

随着多媒体技术的快速发展, 数码图像的传播越来越广泛, 与此同时盗版问题也愈发突出。图像水印技术是一种有效的图像认证手段, 也是主流的 IP 保护方法^[1]。人们对鲁棒性数字水印的研究是一个循序渐进的过程, Tirkel 等人提出了基于 LSB (Least Significant Bit) 的数字水印算法^[2], 该算法具有很好的不可感知性, 但任何信道噪声都会给这种方法造成影响。Cox 等人首先提出了水印应该嵌入到人类视觉最重要的地方的观点^[3], 这种嵌入方法具有较好的安全性和鲁棒性, 后来的多数鲁棒性数字水印算法都以其为主要思想, 但缺点是水印的检测过程需要原始图像的参与, 无法实现盲检测。Hsu 等人提出了自适应 DCT 域数字水印算法^[4], 将图像块按照噪声的敏感性进行分类, 在不同的类中分别嵌入不同强度的水印。

作者简介: 孟煦, 出生年月 1989 年 1 月, 男, 博士, 担任电子科学与应用物理学院微电子所讲师, 研究方向: 超大规模集成电路设计. E-mail: xumeng@hfut.edu.cn

上述 DCT 域水印算法虽然具有一定的鲁棒性，但是嵌入的水印都是随机的 0,1 序列，而且水印的嵌入量较低。相对于随机的 0,1 序列而言，人们往往更希望嵌入的水印信息是具有一定直观含义的有意义数字水印。

45 本文针对高信息容量的有意义数字水印，通过对传统的线性内插方法的定量分析，提出了改进的线性内插拆分方案，并在此基础上提出了一种具备 DCT 系数误差校准功能的嵌入方法。实验证明：该方法对于 JPEG 有损压缩以及其它一些恶意攻击都具有较好的鲁棒性。

1 硬件实现部分

1.1 概述

50 1.1.1 各模块 FPGA 综合速度

如下表 1-1 所示：

表 1-1 硬件各模块 FPGA 综合频率
Tab. 1-1 FPGA integrated frequency of each hardware module

模块	DCT 模块	嵌入模块	控制模块	IDCT 模块
频率	228.669MHz	652.614MHz	591.541MHz	229.064MHz

1.1.2 顶层模块接口

55 输入信号：水印数据输入和原始图像数据输入引脚以及相应的使能信号，输入还有 CLK 与 RST_CONTROL 信号，由上表 1-1 可知输入时钟频率应为 228.669MHz。

输出信号：嵌入完成后的数据以及输出使能信号。顶层模块接口如图 1-1 所示：

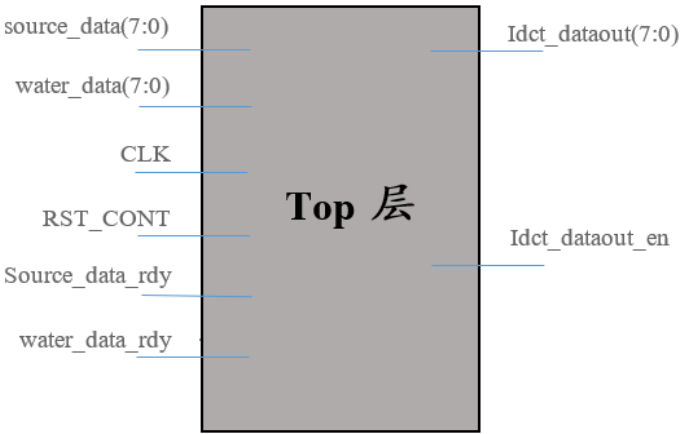


图 1-1 顶层模块接口图
Fig. 1-1 Top module interface diagram

60

1.2 各模块设计方案

1.2.1 DCT 模块

65 一维 DCT 算法的硬件实现框图主要包含串行/并行转换器、翻转触发器、加/减法器、循环移位器和乘法器等模块组成，如下图 1-2 所示。在本算法设计中，数据的输入方式为串行输入，经过串并转换后进行加减法运算，之后进行乘法操作，乘法操作之后为取整操作，

最后进行四个乘法器输出结果的加法操作，一维 DCT 系数值就开始顺序输出。

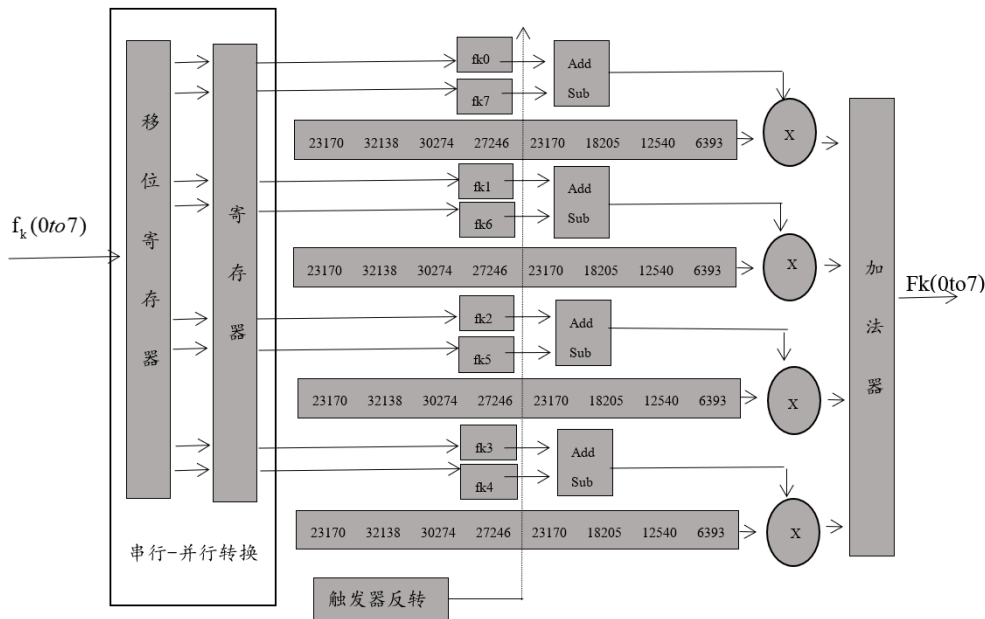


图 1-2 一维 DCT 算法硬件实现框图

Fig. 1-2 One-dimensional DCT algorithm hardware implementation block diagram

70 由于图片是二维矩阵，所以必须进行二维 DCT 模块电路的搭建。本文通过 W.H.Chen 的对称性算法^[5]来进行快速 DCT 变换，需要两次调用一维 DCT 变换来实现二维的 DCT 变换：将输入图像亮度信息作为激励信号，查看其 DCT 变换后的输出结果，并将相同的图像亮度信息经 Matlab 仿真，查看 Matlab 的 DCT 变换后的输出结果，将两者对比来验证本文所设计模块是否可用。用 Modelsim 仿真的二维 DCT 变换的结果如图 1-3 所示，用 FPGA 计算二维的 DCT 变换的结果值如图 1-4 所示，此值和 Matlab 计算的二维 DCT 变换值十分接近，大部分低频分量几乎相同，误差很小并且误差的绝对值小于 1。由此可知，本文设计的二维 DCT 模块可用，设计正确。

75

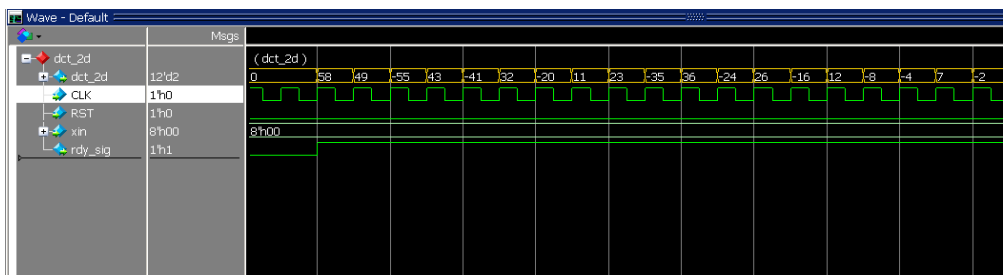


图 1-3 二维 DCT 变换 Modelsim 仿真结果

Fig. 1-3 Modelsim simulation results of two-dimensional DCT transform

8x8 double

	1	2	3	4	5	6	7	8
1	57.5000	23.1176	-4.2299	2.1582	-2.2500	1.1370	-0.4127	0.0140
2	48.6118	-35.6212	6.6127	1.8190	3.7042	0.2010	0.3319	-0.1293
3	-54.4765	36.3068	-2.0910	-1.4710	-3.4645	-0.3657	-0.3232	-0.0666
4	43.0244	-24.0360	3.6098	3.6034	1.4501	-0.1721	0.6695	0.0308
5	-39.7500	26.2535	2.4054	-3.6263	0.5000	-0.8512	-0.7257	0.0661
6	31.1276	-15.9554	-1.0225	3.5602	-0.5087	0.5492	-0.2454	0.1312
7	-20.4602	12.3588	0.6768	-4.9464	1.4351	-0.8830	1.0910	0.7241
8	10.9658	-7.3971	-2.9285	0.6317	0.2052	0.7437	0.0305	-0.5314

图 1-4 二维 DCT 变换 MATLAB 仿真结果

Fig. 1-4 Two-dimensional DCT transform MATLAB simulation results

1.2.2 多算法结合的嵌入模块

85 多算法结合的嵌入模块结构，如下图 1-5 所示：

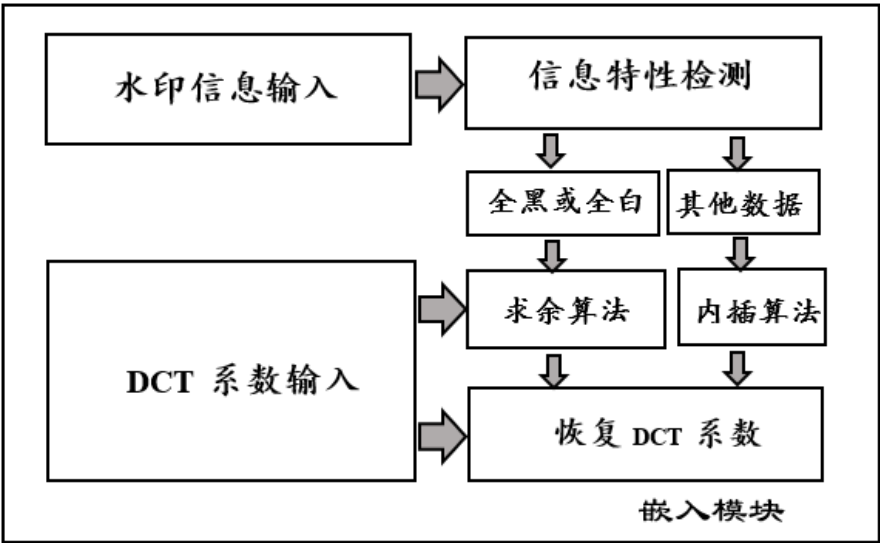


图 1-5 硬件嵌入模块框图

Fig. 1-5 Block diagram of hardware embedded module

本设计水印嵌入过程不占用额外周期，使用 DCT 系数输出的实时结果来操作水印数据与 DCT 系数。介绍了水印算法的硬件实现，并对设计的水印嵌入模块进行 Modelsim 仿真。

1.2.3 水印信息预处理模块

水印信息预处理模块结构，如下图 1-6 所示：

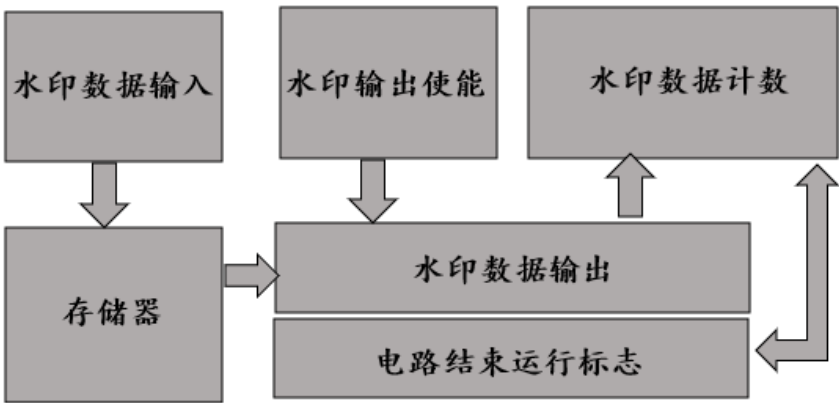


图 1-6 硬件嵌入模块框图

Fig. 1-6 Block diagram of hardware embedded module

95

- (1)无损压缩水印数据：根据水印信息特点进行无损压缩，压缩率在 35%左右。
- (2)分发水印数据部分：水印信息存储在一个 ram 中，当嵌入模块输出一个使能信号 en 时，水印信息输出一个，其中 ram 的前两位存储的是该水印信息总数，ram 最后部分为水印信息嵌入在原始图像的块的地址。
- (3)结束水印嵌入标志：因为水印数据经过无损压缩后个数不定，而整个嵌入过程结束标志为水印信息全部嵌入完毕，所以根据水印个数而产生一个硬件电路结束标志 water_end, 该信号输入顶层模块而结束此次嵌入。

100

1.2.4 原始图像预处理模块

嵌入水印之前需要先对图像进行分块处理,将图像分为互不重叠的 8*8 的块之后对这些块进行二维 DCT 变换。在硬件实现的时候,若按常规的方式对图像进行分割,则需要读入 8 行图像后才能进行 DCT 运算,增大了时间和硬件的开销,所以,为了提高时效一般直接对连续输入的 64 个像素进行处理,该做法虽不同于常规的操作,但对结果的影响较小。因此,选择对连续输入的像素进行操作。

1.2.5 控制模块

因为一张图片需要进行多次嵌入,所以加入一个控制部分来调控整个流程,主要通过复位信号来调控。周期性生成后级复位信号,并且生成一个读取原始图像数据信号给 DCT 模块,使 DCT 模块开始接收信号。在进行完一次 DCT 变换后,数据紧接着输入嵌入模块,无需等待嵌入模块完成而直接输入下一次数据给 DCT 模块,以此达到与嵌入模块同步执行的流水线操作。当一次数据发送完毕后,等待 DCT 结束,之后控制模块输出一个 64 周期的信号给原始图像处理模块控制其发送下一次数据。

1.2.6 IDCT 模块

IDCT 是将 DCT 域中的频域系数转换为空间域的图像像素数据。对于二维 IDCT,利用二维 DCT 中系数矩阵 c 的正交性,所以二维 IDCT 变换公式可记为: $x=c^Tyc$,因此二维 IDCT 的实现可采用与二维 DCT 一样的结构,即首先对 y 进行一维的 IDCT,而后再对一维 IDCT 后的结果的转置进行一维 IDCT,从而得到变换后的结果,二维 IDCT 结构如下图 1-7 所示:

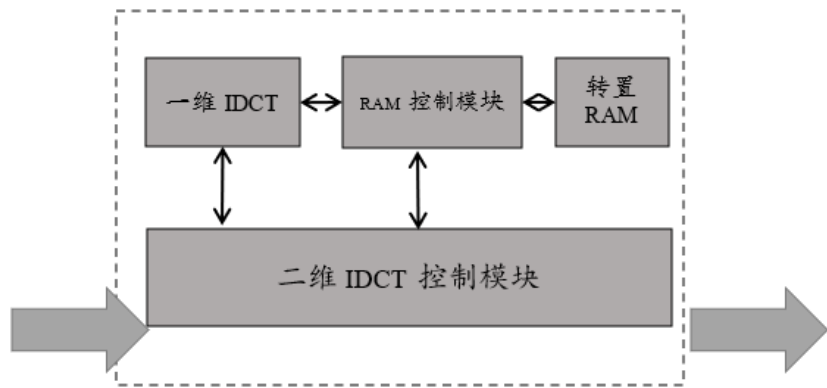


图 1-7 IDCT 模块框图
Fig. 1-7 IDCT block diagram

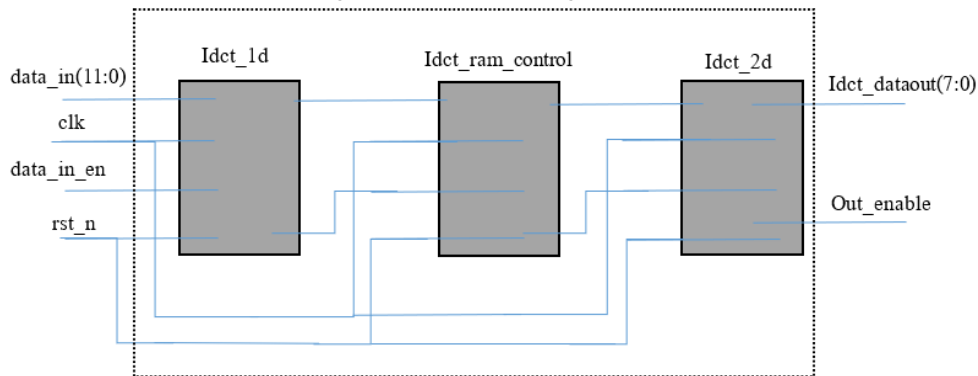


图 1-8 IDCT 电路图
Fig.1-8 IDCT Circuit diagram

利用 Verilog 代码写出如上图 1-8 所示电路各模块，并按图连接相关信号，输入数据 data_in 为嵌入之后输出的数据，该数据送入二维 IDCT 模块后先进行一维 IDCT 变换，之后在 IDCT_RAM_CONTROL 控制下转置一次再次送入一维 IDCT 模块，此时输出的数据进入 IDCT_2D 模块直接给输出数据引脚。

二维 IDCT 由一次 IDCT 以及转置一维 DCT 的结果后再次送入一维 IDCT， datain（黄色数据线）串行输入 64 个数，经过一维 IDCT 后结果由蓝色 OutData_1d 数据线输出，经矩阵转置后，再次输入一维 IDCT，最终结果由 idct_dataout 输出。仿真结果如下图 1-9 所示：

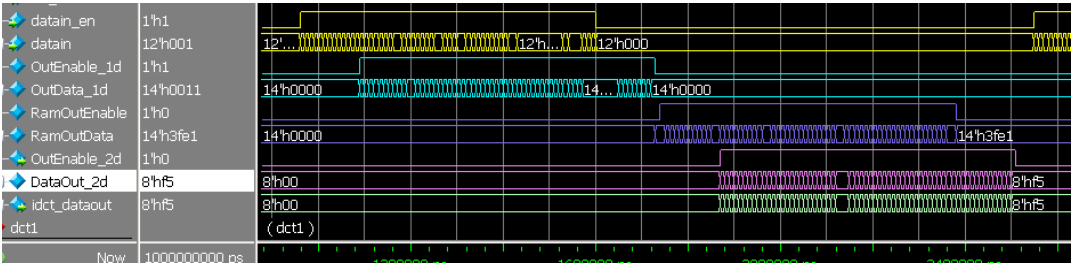


图 1-9 二维 IDCT 内部执行仿真图
Fig.1-9 2D IDCT internal execution simulation diagram

1.2.7 各版本硬件仿真结果对比

各版本硬件仿真结果如下表 1-2 所示，版本 2 和版本 3 的面积报告如下表 1-3 所示：

表 1-2 各版本硬件仿真结果对比

Tab. 1-2 Comparison of hardware simulation results of various versions			
版本号	水印处理速度	面积	主频
1（无损无压缩）	120 picture/100ms	3 路并发	228.669MHz
2（有损压缩）	245 picture/100ms	无需并发操作	228.669MHz
3（无损压缩）	105 picture/100ms	无需并发操作	228.669MHz

表 1-3 版本 2&版本 3 面积报告

Tab. 1-3 Version 2 & Version 3 Area Report			
Device Utilization Summary(estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	3952	408000	0%
Number of Slice LUTs	5204	204000	2%
Number of fully used LUT-FF pairs	3427	5729	59%
Number of bonded IOBs	3	600	0%
Number of BUFG/BUFGCTRLs	2	32	6%
Number of DSP48E1s	24	1120	2%

2 软件实现部分

2.1 整体架构

软件部分主要是利用 Python 调用 OpenCV，完成宿主图像信息和水印图像信息的提取，并以二进制码流格式输出给硬件，以及将已嵌入水印的图像信息恢复成完整图像，并从中提

取出水印，同时协助嵌入后图片的抗攻击性验证工作。

2.2 具体实现

2.2.1 图像的批量操作及通道分离

为了后续能够处理一系列图像，利用 io 库中 ImageCollection 函数遍历图像文件夹中所有图片并保存，当前需要处理的图片可以由用户输入自由选择。

24 位真彩文件包含三种颜色信息通道：红、绿和蓝(RGB)；32 位真彩文件具有额外的 8 位 alpha 通道(RGBA)，可以指定每个像素的透明度或不透明度。

为了使图像处理对于 24 位和 32 位的真彩位图文件均适用，读取图像时使用函数 cv2.IMREAD_UNCHANGED 实现。利用 shape()函数获得所读取图片通道数，进而判断其类型(RGB 或 RGBA)以方便后续的图像合成恢复工作。

2.2.2 宿主图像处理模块

在作品版本 1、版本 2 和版本 3 中，对于包含 800*800 个像素点的宿主图像，顺序取出对应后续水印信息个数的若干像素点(版本 1 中需顺序提取 227456 个以上像素点数据，版本 2 中需顺序提取 38016 个以上像素点数据，版本 3 中需顺序提取 86080 个数据)

2.2.3 水印图像处理模块

在版本 1 中，为了实现水印的无损嵌入，切割过后的 alpha 通道像素点数据不做处理，即无损无压缩，仅转换为 8 位二进制数据作为水印信息；在版本 2 中，将切割过后的 alpha 通道像素点数据转化为二值(0 值仍为 0，非 0 值转换为 1(255))，为了提高处理速度，同时考虑到嵌入之后图片的鲁棒性，在该版本中我们将已经转换为二值的 alpha 通道数据每 6 个一压缩得到 6 位的二进制数据作为水印信息；在版本 3 中，为了实现水印的无损嵌入，减少硬件嵌入量，提高处理速度，我们对 alpha 通道进行了压缩：将若干个连续的 0 值压缩成 1 个 9 位的二进制数，用 100000000(256)表示，同理，用 100000001(257)表示若干个连续的 255 值，非 0 和非 255 的数据仅零扩展成 9 位二进制数作为水印信息即可。

2.2.4 图像恢复模块

读取硬件完成嵌入后生成的.txt 文件 data_out.txt，从文件中获得被嵌入像素点，并根据各个版本宿主提取时的位置，将其与宿主图像其他部分信息进行合成，得到完整的已嵌入水印信息的 R 通道，然后将其与其他通道合并即可得到嵌入水印后的完整图像 pic_after_emb.png。

3 MATLAB 验证工作

3.1 性能指标

峰值信噪比 PSNR 与数字水印归一化互相关系数 NC 作为现在数字水印最常用的检测指标，尽管部分图像质量评价时可能与主观感知的质量产生较大的偏差，但在图像水印领域这

两项指标对多数图像质量评价仍是有效的,同时被大多数研究者所采用,用以查看提取出水印的性能指标。

3.1.1 峰值信噪比 PSNR 函数

峰值信噪比 PSNR 是一个表示信号最大可能功率和影响它的表示精度的破坏性噪声功率的比值的指标。由于许多信号都有较宽的动态范围,峰值信噪比常用对数分贝单位表示。在图像处理中,要对图像进行客观的评价,常常需要计算 PSNR。PSNR 是衡量图像失真或噪声水平的客观标准。两个图像之间 PSNR 值越大,则越相似。普遍基准为 30dB,30dB 以下的图像劣化较为明显。

PSNR 的定义为:

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (3-1)$$

PSNR 其中 MAX 表示图像颜色的最大数值,8bit 图像最大取值为 255。

MSE 为均方差,定义为:

$$MSE = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m ||K(i,j) - I(i,j)||^2 \quad (3-2)$$

其中, I 和 K 分别是原始图像和处理后的图像, m*n 为两图像的大小。

对于嵌入水印后的图像的破坏性以及对于在对嵌入水印的图像加入噪声后再提取的水印图像和原水印图像的差别,都是通过该函数计算得来的。

3.1.2 数字水印归一化互相关系数 NC

归一化互相关匹配算法是图像匹配算法中较为经典的匹配算法。它是一种相似性度量或者匹配程度的表征,而不是一种图像匹配的完整方法,但是把互相关的思想作为度量测度,在许多匹配算法里都会用到。通过计算参考图和模板图之间的互相关度量值确定二者的匹配程度,度量值大小反应了参考图和模板图之间的相似程度。度量值 NC(i,j) 值越大,则表示搜索子图上的(i,j)位置和模版越相似。当 NC(i,j) 等于 1 时,为匹配位置。由于图像噪声的影响,因此很难找到 NC 为 1 的位置。通常只需要在参考图上找到最大度量值 NC 的位置,为最佳匹配位置。互相关法度量值的计算形式如下:

$$NC(i,j) = \frac{\sum_{m=1}^M \sum_{n=1}^N T(m,n) S^{ij}(m,n)}{\sqrt{\sum_{m=1}^M \sum_{n=1}^N T^2(m,n) \sum_{m=1}^M \sum_{n=1}^N (S^{ij}(m,n))^2}} \quad (3-3)$$

其中, T(m,n) 为模版图像的第 n 行,第 m 个像素值; S(i,j) 为模版覆盖下的部分,称为搜索子图。

3.2 Matlab 验证工作

3.2.1 模拟硬件的 DCT 快速算法的实现

用 Matlab 实现 DCT 快速算法:用同样的矩阵运算将一维 DCT 变换,输出时再将数据除以 2^{16} ,得到一维 DCT 变换矩阵,方便检验硬件输出数据的正确性。如下图 3-1 所示:



图 3-1 左图为快速算法的输出，右图为 Matlab 自带函数的输出

Fig. 3-1 The left picture is the output of the fast algorithm, the right is the output of Matlab's own function

215 3.2.2 嵌入水印前后差异化测试

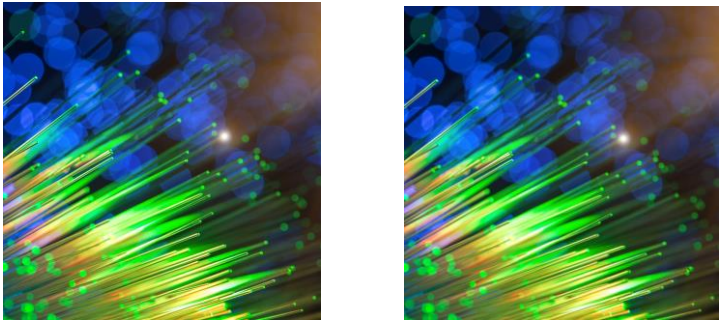


图 3-2 版本 1 嵌入水印前后对比图

Fig. 3-2 Version 1 before and after embedding watermark comparison chart

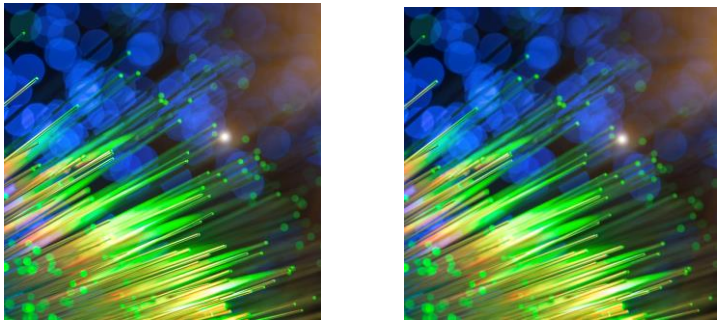


图 3-3 版本 2 嵌入水印前后对比图

Fig. 3-3 Version 2 before and after embedding watermark comparison chart

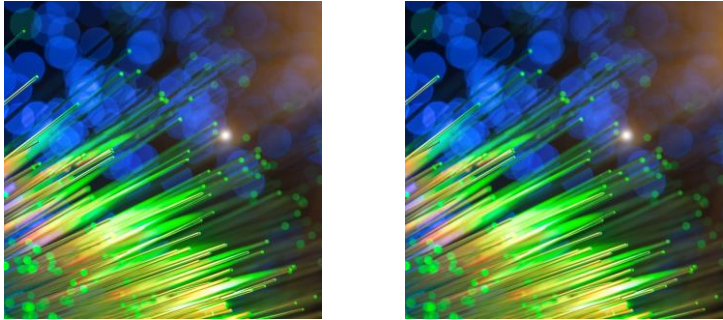


图 3-4 版本 3 嵌入水印前后对比图

Fig. 3-4 Version 3 before and after embedding watermark comparison chart

表 3-1 各版本嵌入水印前后差异化统计表

Tab. 3-1 Differentiated statistics table before and after embedding watermark in each version

版本 1		版本 2		版本 3	
PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
59.1652	1	67.026	1	58.965	1

各版本对比如上图 3-2、3-3、3-4 所示，在表 3-1 中可以看出，版本 2 嵌入水印后对原图的破坏最小，版本 1 和版本 3 相对大点，但总体而言三个版本嵌入水印前后差异都较小。

3.2.3 所提取水印与水印原图差异化测试

将嵌入水印的数据读入 Matlab 中，将软件提出的水印与原始图像做对比，查看提取情况。左图为原始水印图片,右图为从嵌入后的载体图像中提取出的水印图像。峰值信噪比 PSNR=34.0765dB，NC=0.9992。



图 3-5 版本 1 所提取水印与原水印对比
Fig. 3-5 Comparison of the extracted watermark with the original watermark in version 1



图 3-6 版本 2 所提取水印与原水印对比
Fig. 3-6 Comparison of the extracted watermark with the original watermark in version 2



图 3-7 版本 3 所提取水印与原水印对比
Fig. 3-7 Comparison of the extracted watermark with the original watermark in version 3

表 3-2 各版本所提取水印与水印原图差异化统计表
Tab. 3-2 Statistics table of the difference between the extracted watermark and the original watermark

版本 1		版本 2		版本 3	
PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
34.0765	0.9992	17.2337	0.9675	55.8732	1

各版本对比如上图 3-5、3-6、3-7 所示，在表 3-2 中可以看出，版本 3 嵌入后所提取的水印与原水印一致性最强，版本 1 次之，版本 2 由于对水印进行了有损压缩，差异相对较明显一些。

3.2.4 各版本抗攻击性测试

对嵌入后图像的抗攻击性测试包括：噪声(椒盐噪声、高斯噪声、乘性噪声)；剪切(上、下、左、右 1/2 部分剪切)；压缩(量化因子为 1 的 JPEG 压缩)。

(1) 版本 1 抗攻击性测试。如下图 3-8、3-9、3-10、3-11、3-12 及 3-13 所示：



图 3-8 版本 1 加入方差 0.003 椒盐噪声后所提取水印
Fig. 3-8 Version 1 watermark extracted after adding variance of 0.003 pepper and salt noise



图 3-9 版本 1 加入方差 0.00005 高斯噪声后所提取水印
Fig. 3-9 Version 1 extracted watermark after adding variance 0.00005 Gaussian noise



图 3-10 版本 1 加入方差 0.0001 乘性噪声后所提取水印

Fig. 3-10 Version 1 extracts watermark after adding variance 0.0001 multiplicative noise

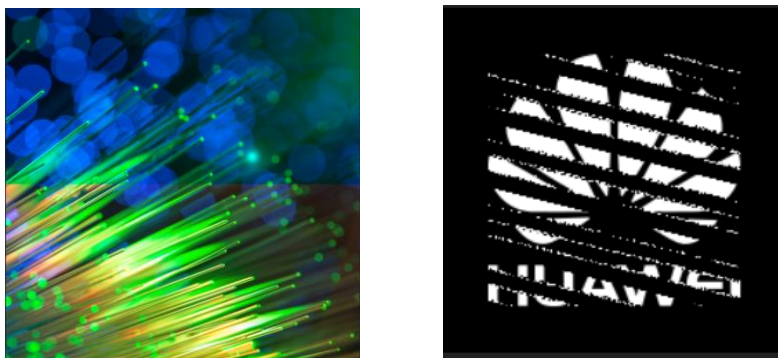


图 3-11 版本 1 剪切上 1/2 部分后所提取水印

Fig. 3-11 Version 1 Watermark extracted after cutting the above 1/2 part

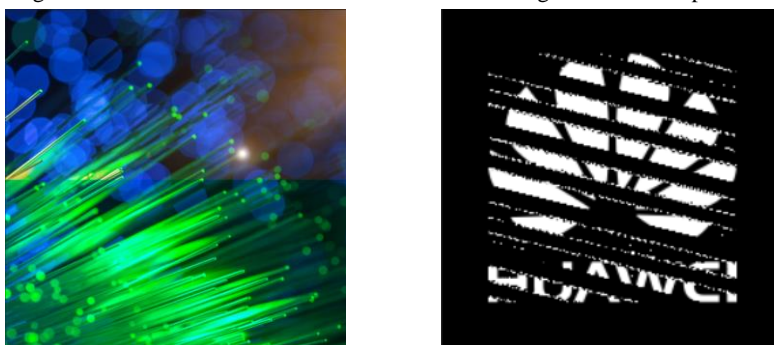


图 3-12 版本 1 剪切下 1/2 部分后所提取水印

Fig. 3-12 Version 1 Watermark extracted after cutting the fellow 1/2 part

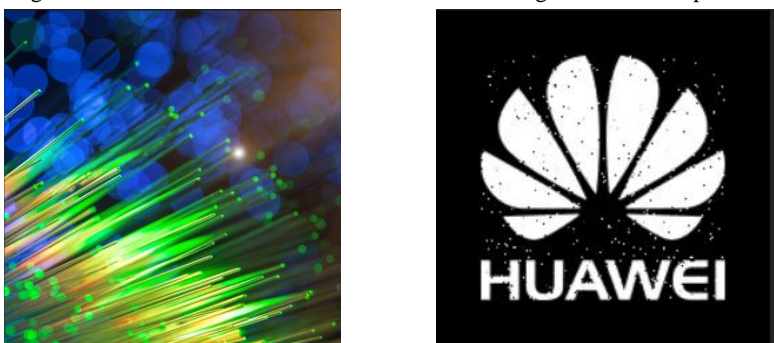


图 3-13 版本 1 量化因子为 1 的 JPEG 压缩后所提取水印

Fig. 3-13 Version 1 watermark extracted after JPEG compression with quantization factor of 1

(2) 版本 2 抗攻击性测试。如下图 3-14、3-15、3-16、3-17、3-18 及 3-19 所示:



图 3-14 版本 2 加入方差 0.003 椒盐噪声后所提取水印

Fig. 3-14 Version 2 watermark extracted after adding variance of 0.003 pepper and salt noise

275



图 3-15 版本 2 加入方差 0.00005 高斯噪声后所提取水印

Fig. 3-15 Version 2 extracted watermark after adding variance 0.00005 Gaussian noise

280



图 3-16 版本 2 加入方差 0.0001 乘性噪声后所提取水印

Fig. 3-16 Version 2 extracts watermark after adding variance 0.0001 multiplicative noise

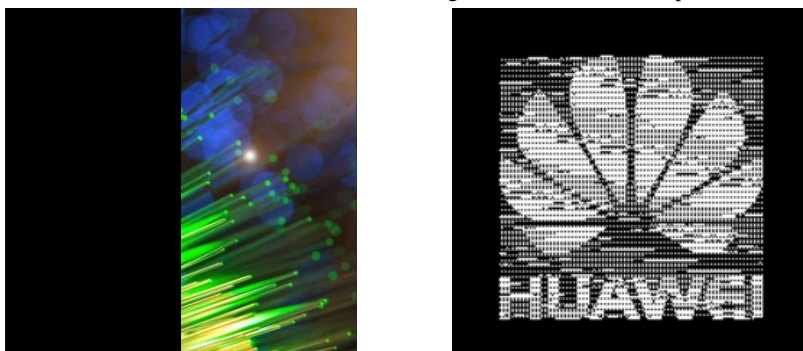


图 3-17 版本 2 剪切左 1/2 部分后所提取水印

Fig. 3-17 Version 2 Watermark extracted after cutting the left 1/2 part

285



图 3-18 版本 2 剪切右 1/2 部分后所提取水印

Fig. 3-18 Version 2 Watermark extracted after cutting the right 1/2 part



图 3-19 版本 2 量化因子为 1 的 JPEG 压缩后所提取水印

Fig. 3-19 Version 2 watermark extracted after JPEG compression with quantization factor of 1

290

(3) 版本 3 抗攻击性测试。如下图 3-20、3-21 及 3-22 所示：



图 3-20 版本 3 加入方差 0.005 椒盐噪声后所提取水印

Fig. 3-20 Version 3 watermark extracted after adding variance of 0.005 salt and pepper noise



图 3-21 版本 3 加入方差 0.0006 高斯噪声后所提取水印

Fig. 3-21 Version 3 watermark extracted after adding variance 0.0006 Gaussian noise

295



图 3-22 版本 3 加入方差 0.005 乘性噪声后所提取水印

Fig. 3-22 Version 3 watermark extracted after adding multiplicative noise with variance 0.005

300

由表 3-3 测试统计结果对比可看出，版本 2 抗攻击性最佳，版本 1 次之，版本 3 较差：

表 3-3 各版本抗攻击性测试结果统计表
Tab. 3-3 Statistical table of anti-attack test results of various versions

攻击方式	参数	版本 1		版本 2		版本 3	
		PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
乘性噪声	0.001	16.1424	0.9528	17.2337	0.9675	8.1611	0.7022
高斯噪声	0.0005	11.8861	0.8749	17.2337	0.9675	5.9972	0.8171
椒盐噪声	0.003	13.8944	0.9202	13.3626	0.917	7.3824	0.6447
剪切	上 1/2	8.6675	0.6872	-	-	-	-
	下 1/2	8.7329	0.6939	-	-	-	-
	左 1/2	-	-	8.2089	0.6534	5.9948	0.6626
	右 1/2	-	-	8.1997	0.7219	6.0704	0.6618
JPEG 压缩	量化因子 (1)	21.8301	0.9872	8.8787	0.9045	5.8657	0.6568
乘性噪声	0.005	-	-	12.5814	0.9675	-	-
高斯噪声	0.0006	-	-	13.9633	0.9282	-	-
椒盐噪声	0.005	-	-	13.2337	0.9275	-	-

4 结论

305 本文设计了一种基于数据相关性和线性内插的 DCT 域盲水印算法，具体总结如下：

 (1)提出了一种基于数据相关性和具备 DCT 系数误差校准功能的改进的线性内插方法的 DCT 域嵌入算法，引入 DCT 系数的补偿因子。实验证明，其对原图破坏较小，并且对于 JPEG 有损压缩及其他一些恶意攻击都具有良好的鲁棒性。通过修改该算法中的取模值可改变鲁棒性，本设计将取模值由 15 变为 93 后，鲁棒性提高 1-2 个数量级，而对原图破坏方面，PSNR 310 仅仅从 60 变为 40。此外，采用 DCT 系数嵌入个数适应型方案，根据特征值选择要改变的系数位置，每个 8*8 DCT 系数块分为 6 个固定嵌入点与 18 个备选嵌入点。

 (2)对水印信息进行可恢复性裁剪，将水印信息的有效部分进行矩形状切割，并将切割的矩形区域信息作为水印信息嵌入以便后期可盲文恢复。同时将切割后的水印信息进一步压缩，使嵌入量达到最小。将水印的有效信息进行切割分离，相较于原始水印信息降低了 47% 315 的嵌入量，同时进行了有损压缩(压缩率达到 83%)和无损压缩(压缩率达到 62%)，大大减小了硬件嵌入的压力，提高了处理速度(100ms 内 245 个并发操作)。

 (3)对提出的算法进行了 FPGA 硬件验证。本设计采用软硬协同的实现方案，同时硬件采用流水线结构，进一步提高了处理速度(1.6 倍)，非常适合高速信号处理的应用。

[参考文献] (References)

320 [1] 张伟,冯建华.IP 保护方法研究进展[J].微纳电子与智能制造,020,2(01):95-101.

 [2] George V.The use of watermark in the protection of digital multimedia products[J].Proc IEEE,1997,87(1):1197-1207

 [3] Cox I J,Killian J,Leighton F T,et al.Secure Spread Spectrum Watermarking for Multimedia[J].IEEE Transactions on Image Processing,1997,6(2):1673-1687

325 [4] Hsu C T,Wu J L.Hidden Digital Watermarks in Images[J].IEEE Trans.Image Processing,1999,8:58-68

 [5] CHEN W H,SMITH C H,FRALICK S C.A fast computational algorithm for the discrete cosine transforms[J].IEEE Transactions on Communications,1997,25(9):1004-1009