

基于小波变换的高分辨率图像数字隐水印微信小程序开发

马佳琪*, 米义嘉, 叶杭东

浙江工商大学, 计算机与信息工程学院, 浙江 杭州
Email: *majiayouernv@126.com

收稿日期: 2021年3月20日; 录用日期: 2021年4月9日; 发布日期: 2021年4月16日

摘要

针对高分辨率数字图像的版权保护问题, 研究并设计了一种基于小波变换的高分辨率图像数字隐水印微信小程序。基于二维离散小波变换实现了高分辨率图像的数字隐水印算法, 算法可实现水印的盲提取, 并由用户用密码进行加密。基于该算法实现的程序被部署在服务器上, 通过WebService的形式提供API接口可供网络上的异构平台软件调用。微信小程序通过调用WebService的各个模块实现水印的嵌入、提取和验证等功能, 以可视化的形式返回结果到用户设备上。实现的图像水印小程序使用方便、占用资源少, 且隐水印不会对图像内容造成影响, 有较高的使用价值。

关键词

图像, 隐水印, 盲提取, 2D-DWT, 微信小程序

Wechat Mini Program Development for Hiding Digital Watermark in High Definition Images Based on Wavelet Transform

Jiaqi Ma *, Yijia Mi, Hangdong Ye

School of Computer Science and Information Engineering, Zhejiang Gongshang University,
Hangzhou Zhejiang
Email: *majiayouernv@126.com

Received: Mar. 20th, 2021; accepted: Apr. 9th, 2021; published: Apr. 16th, 2021

*通讯作者。

文章引用: 马佳琪, 米义嘉, 叶杭东. 基于小波变换的高分辨率图像数字隐水印微信小程序开发[J]. 图像与信号处理, 2021, 10(2): 70-79. DOI: 10.12677/jisp.2021.102008

Abstract

For the demand of copyright protection for high definition images, a Wechat mini program for hiding digital watermark in high definition images is developed based on wavelet transform. The watermarking algorithm for high definition images based on 2D-DWT (Two-Dimensional Discrete Wavelet Transform) is implemented which can blindly extract watermarks and the location that hide watermark bits and watermark itself can both be encrypted with customized password. The program is installed on the server and its APIs is published on the internet through Web Service so that they can be called by software on heterogeneous platforms. Wechat mini program implements the functions of watermark embedding, extraction and verification by calling the web methods of the Web Service, and visualized results are returned on the users' device. The implemented image watermarking mini program is easy to operate, takes less resources and the hidden watermark will not affect the content of the image, and has high practical value.

Keywords

Image, Hidden Watermark, Blind Extraction, 2D-DWT, Wechat Mini Program

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着网络技术和媒体记录设备的发展,越来越多的多媒体信息通过网络和存储设备发布和分享。未经授权的个人或组织可以方便地复制、修改或转发这些信息,这就可能造成侵犯版权问题,迫切需要一种方便快捷的可行技术来保护数字媒体的版权,从而为数字媒体生产者维持健康的生产环境。数字水印是一种将版权信息隐藏在数字媒体中,并可在需要时提取的版权保护技术,从而检测和认证数字媒体的版权是否合法。明水印会影响图像本身的内容,并且其显著的存在性使其易被恶意篡改,透明的隐水印可以对抗版权侵犯并保持媒体的内容完整性。早期的空域水印技术如最低有效位修改(LSB) [1]可以简单有效地在图像中隐藏数据,但对 JPEG 压缩、滤波、加噪等图像处理敏感;一些基于块的方法[2]对 JPEG 压缩和滤波有鲁棒性,但不能实现盲提取。变换域水印技术在离散傅立叶变换(DFT) [3]、离散余弦变换(DCT) [4]、离散小波变换(DWT) [5] [6]等变换域隐藏水印,对 JPEG 压缩及其他信号处理操作有较好的鲁棒性,然而多数变换域水印技术是基于分块和伪随机置乱的,仍然对几何攻击和位置变化敏感。Tsai 等 [7]提出了基于特征的水印方案,可以在多个无交叠的区域嵌入 100 位的水印从而增强鲁棒性。Kang 等 [8]提出的 DWT-DFT 复合水印方案对仿射变换和 JPEG 压缩均有较好鲁棒性,然而其 60 位的水印容量相对较低。近年来还有一些研究者提出了基于深度学习的数字水印方法 [9],对信号处理操作和几何变换攻击都有较好鲁棒性,但只能在有限大小的图像时进行水印的嵌入和提取。随着图像采集技术的发展,图像的尺寸和分辨率正变得越来越大,一个实用的水印方法应该可以应用到高分辨率图像上。微信小程序无须安装,即开即用,用完即走,最大程序减少在手机上所占用的资源,而且微信用户粘性高,对一些偶尔使用的功能,越来越多的用户选择微信小程序来替代专用的 APP,这对资源有限的手机等移动终端而言更为重要。

本文首先基于 2D-DWT 实现了一个高分辨率图像数字水印算法，其次将该程序制作成动态链接库，以便于基于 ASPNET 的 WebService 调用，微信小程序通过 WebService 调用水印嵌入和水印提取、验证等程序，最终以微信小程序为入口，实现在线的水印嵌入、提取和验证。

本文组织如下，第 1 节介绍系统总体设计，第 2 节介绍图像水印嵌入和提取的算法流程。第 3 节介绍微信小程序的设计与实现；第 4 节是实验结果及系统展示；最后是总结和展望。

2. 系统总体结构

系统总体结构如图 1 所示，采用客户端/服务端(C/S)结构，微信小程序搭载在微信 APP 上，运行在客户端，提供用户接口，用户可以在本机上选择图像和水印文件，并输入密码并将图像、水印和密码通过网络发送到服务端 Web Service 搭载于 Windows 系统的 IIS 上，基于 .NET 技术创建，负责与小程序的通信，并调用服务器上的动态链接库执行水印嵌入、提取和验证等操作；动态链接库基于 C++ 和 Opencv 库实现了图像水印的嵌入、提取和验证等核心功能，其中的图像处理函数调用了 Opencv 库中 API，其中运算复杂度比较高的小波变换和逆变换用 CUDA (Compute Unified Device Architecture) SDK 和 GPU 进行 B 并行化加速。

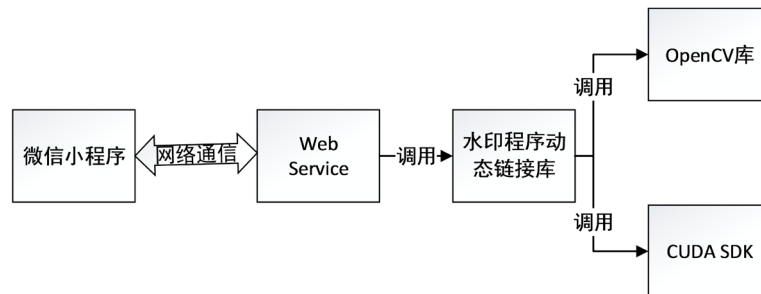


Figure 1. The overall structure of the system
图 1. 系统总体结构

3. 基于 2D-DWT 的高分辨率图像水印算法

3.1. 基于 2D-DWT 的水印嵌入算法流程

基于 2D-DWT 的图像数字水印嵌入算法的流程图如图 2(a)所示。为了提高水印的鲁棒性，水印嵌入于二级小波变换的最后一级的近似系数 LL 中，二级小波变换后的 LL 系数长和宽均为图像的 1/4，假如 LL 中每个系数隐藏一个位的水印，能隐藏的信息量不超过最后一级 LL 系数的数量。因此隐藏水印的第一步是确定所隐藏水印的大小，对水印尺寸进行归一化，保证水印位数小于 LL 系数数量。确定嵌入水印的边长 a 后，对原始二值水印图像进行缩放，使之长边为 a ，并将水印置于左上角填 0 扩充为边长为 a 的正方形，即如果宽大于高，则在下方填 0，否则在右侧填 0，如果原来就是正方形图像，则缩放即可。在嵌入和提取水印时，需要用户提供一个 6 位的数字密钥，以便对水印以及嵌入水印的位置进行加密，其中前两位记为 key1，用作对水印进行 Arnold 变换的变换次数；后四位记为 key2，用作对嵌入位置加密时的伪随机置乱的种子数，以提高水印的安全性。Arnold 变换的计算公式如式(1)：

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}^n \begin{pmatrix} x \\ y \end{pmatrix} \bmod a \quad (1)$$

其中 (x, y) 为水印中某一位的原始坐标， (x', y') 为变换后该位的坐标， $n = \text{key1}$ 为变换次数。 $\bmod a$ 表示用 a 取余，因为坐标超出宽和高的范围将产生越界异常。

为了避免色差变化对图像造成视觉上的失真, 本文将水印隐藏于彩色图像 HSV 空间的 V 通道。首先将图像转换到 HSV 空间, 然后对 V 通道执行二级 2D-DWT。本文基于文献[10]的方法对小波变换进行 GPU 加速, 加速后 GPU 版的 2D-DWT 比 CPU 版的 2D-DWT 快两个数量级, 从而可以保证水印嵌入和提取的实时性, 提升用户体验。首先将以行优先顺序将 LL2 转换为一维向量, 然后以 key2 为种子对一维向量的下标进行伪随机置乱, 提取置乱后的下标序列的前 N 个元素(N 为水印的位数), 以式(2)方式改写这 N 个细节系数。

$$v' = \begin{cases} \left[\text{round}\left(\frac{v}{q}\right) + 0.5 \right] * q & \text{bit} = 1 \\ \text{round}\left(\frac{v}{q} + 0.5\right) * q & \text{bit} = 0 \end{cases} \quad (2)$$

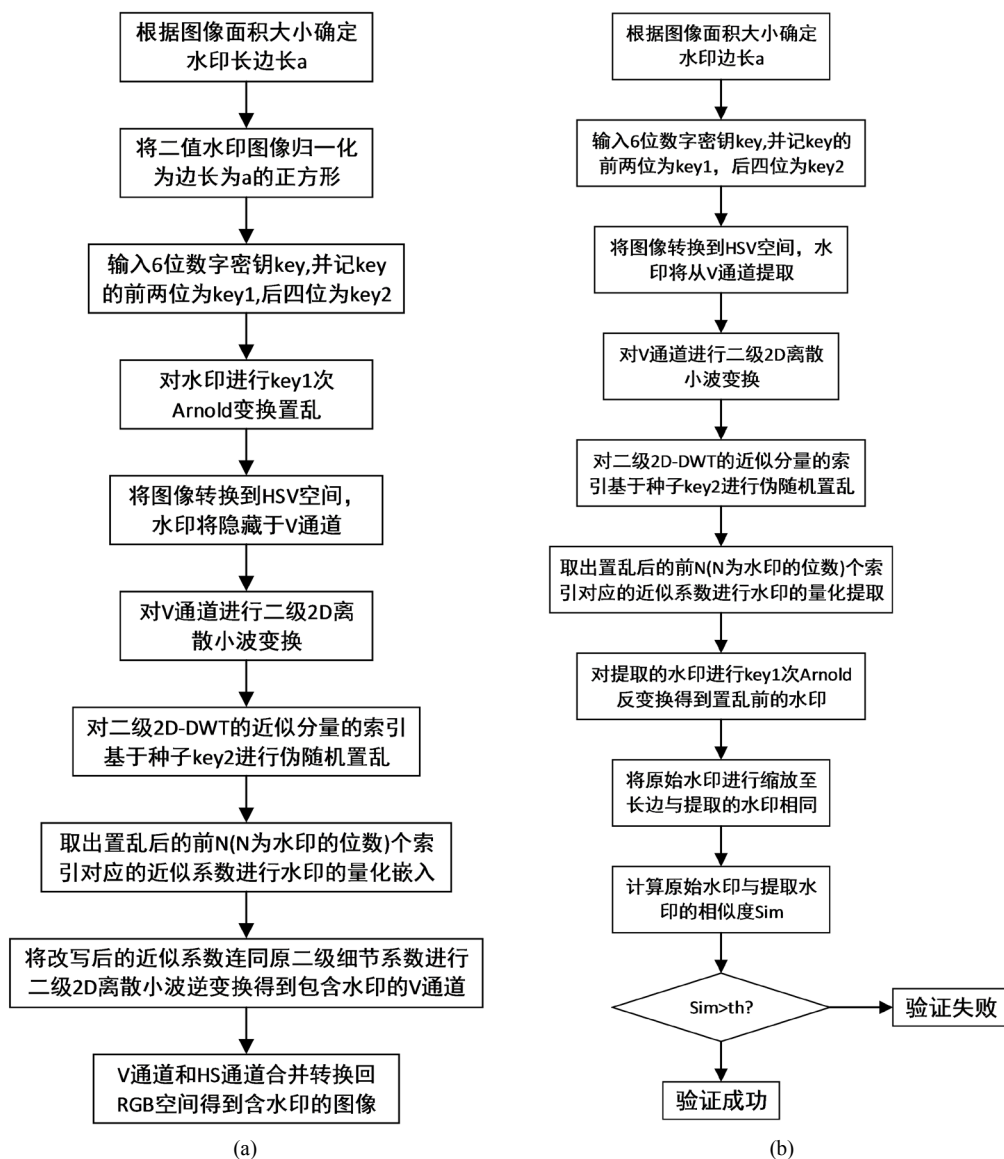


Figure 2. Flow chart of watermark embedding, extraction and verification algorithm based on 2D-DWT
图 2. 基于 2D-DWT 的水印嵌入、提取及验证算法流程图

其中 $round(\cdot)$ 表示四舍五入取整, q 为质量因子, q 越大水印越稳健, 但水印透明性下降, q 越小透明性越好, 但可能在图像经历 JPEG 压缩等信号处理后难以提取到高质量的水印, 本文中取经验值 $q = 50$, 以在水印的稳健性和透明性之间取得良好折衷。 v 和 v' 分别表示改写前和改写后的细节系数, bit 指水印中的某一位。这样, 改写完 N 个指定下标的细节系数后, 执行二级 2D-DWT 逆变换, 得到含水印的 V 通道, 最后将 V 通道与原来 HS 通道合并转换回 RGB 空间即得到了含水印的图像。

3.2. 基于 2D-DWT 的水印提取及验证算法流程

基于 2D-DWT 的图像数字水印提取及验证算法的流程图如图 2(b) 所示。首先与 2.1 节相同的方式确定水印的边长, 并用同样方式将图像转换到 HSV 空间, 将 V 通道用同样的二级 2D-DWT 进行小波变换, 在二级近似系数 LL2 中提取水印。提取水印时, 用户必须输入与嵌入水印相同的密钥 key , 记 key 的前 2 位为 $key1$, 后四位为 $key2$ 。用 $key2$ 对以行优先顺序排列的 LL2 下标进行伪随机置乱, 从前 N 个下标对应的系数中提取水印值, 如式(3)

$$t = v/q - round(v/q) \quad (3)$$

$$bit = \begin{cases} 1 & 0.25 < t < 0.75 \\ 0 & \text{其他} \end{cases} \quad (4)$$

将提取到的 N 位水印按行序排列成正方形, 执行如式(5)所示的反向 Arnold 变换得到置乱前的正方形水印。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}^n \begin{pmatrix} x \\ y \end{pmatrix} \bmod a \quad (5)$$

验证水印需要将原始水印缩放到同一尺度, 然后与提取的水印在左上角对齐, 在两者的交叠区域计算相似度, 计算公式如式(6)。

$$Sim(W, W') = \frac{\sum_{i=1}^N w_i = w'_i = 1}{\sqrt{\sum_{i=1}^N w_i = 1 \sum_{i=1}^N w'_i = 1}} \quad (6)$$

其中 N 为交叠区域的总像素个数, w_i 和 w'_i 为原水印和提取到的水印对应位的值。用阈值 th 来验证两水印是否匹配。

4. 图像数字水印微信小程序的设计与实现

4.1. 制作动态链接库

因为水印的嵌入和提取涉及运算复杂的小波变换, 无法直接实现单机版的微信小程序, 因此本文将这些复杂运算程序部署在服务器上, 在网络上开放 API 接口供小程序调用。我们把水印的嵌入、提取、以及验证封装成三个程序模块, 并把这三个模块制作成动态链接库, 供服务器的 Webservice 调用。其中水印的嵌入程序接受三个参数, 包括图像文件, 水印文件, 以及用户设置的 6 位数字密钥, 返回整数 1 表明嵌入操作成功, 含水印的图像将替换原来的图像文件, 返回其他整数表示水印嵌入失败; 水印的提取则接受两个参数, 含水印图像以及 6 位数字密钥, 返回整数 1 表示提取成功, 提取的水印将替换原来的含水印图像文件; 水印验证模块接受两个参数, 即原水印和提取的水印, 将原水印归一化到长边与提取的水印相同, 左上角对齐后对交叠部分按式(6)计算相似度, 返回相似度计算结果。所有程序用 C++ 实现, 并调用了 OpenCV 库和 CUDA 库。

4.2. 创建 Web Service

本文基于微软的.NET 技术创建 Web Service, 编程语言为 C#, Web Service 可以把应用的 API 接口发布在网络上, 运行在其他主机上的程序可以通过网络访问这些 API 接口, 跨域调用服务器上开放的应用程序。Web Service 监听网络上的调用请求, 并把运行结果返回给请求方。

Web Service 首先导入上一步生成的动态链接库, 并在导入语句中定义水印的嵌入、提取、以及验证三个模块的入口点, 这样 Web Service 中的 WebMethod 就可以调用这些模块。根据微信小程序的功能需要, Web Service 中定义了 4 个 WebMethod, 除了水印的嵌入、提取、以及验证三个接口外, 另外还定义了一个接收微信小程序上传图片的接口, 以便服务器接收图像和水印。

最后 Web Service 被部署到 Windows 自带的 Internet Information Services (IIS) 上, 微信小程序通过服务器的地址或域名来访问 Web Service 提供的应用服务。

4.3. 图像数字水印微信小程序开发

微信小程序用微信开发者工具进行开发, 本文在小程序中设计了两个页面: 一个提供水印嵌入的页面和一个提供水印提取及验证的页面, 两个页面通过底部的标签实现页面切换。在框架全局文件的公共设置 app.json 文件中对两个页面的路径, 标题栏背景色、标题、文字颜色等窗口表现, 底部标签的显示的文字以及链接页面的路径, 网络超时参数设置等进行配置。每个页面包含文件名相同、但扩展名不同的 4 个文件, 分别是 js 页面逻辑、json 页面配置、wxml 页面结构、wxss 页面样式表。wxml 定义了页面的布局结构, 外观设计通过此文件实现, js 实现业务逻辑。

本文设计的水印嵌入的界面如图 4 所示, wxml 文件中首先是 view 控件, class 属性设置为 labels, labels 定义在 wxss 文件中, 这个 view 控件内包含文字“选择原始图片”, 下面是一个包含了两个 image 子控件的 view 控件, class 属性设置为 set-background, 在 wxss 文件中定义了高度、对齐方式、背景色等属性。两个 image 子控件始终有一个处于隐藏状态, 其中一个显示十字丝印图标, 另一个用于显示选择的图像, 都用 bindtap 绑定了 chooseOriginImage 函数, 这样点击图片就可以调用 wx.chooseImage 在手机相册中选择图像并显示在该 view 内。下面两个 view 跟上面两个类似, 用于选择和显示水印图像, 同时含水印图像也显示在下面的 view 中。下面又是一个 class 为 labels 的 view, 这个 view 包含一个用于显示提示文字 view 子控件和一个用于输入密码的 input 子控件, input 子控件的 type 设置为“number”, password 设置为“true”, 这样密码只能输入数字类型的字符, 输入的密码自动隐藏以提高安全性。最底下是一个 button 控件, 绑定了函数 onClickButton。页面右上角有一个固定大小的 view 控件, class 定义为 add_icon, 因两个页面均定义了这个刷新的 view, add_icon 定义于顶层 app.wxss 文件中, 绑定了 refresh 函数, view 内部包含一个 image 子控件用于显示刷新图标。本文设计的水印提取和验证的界面如图 5 所示, 界面设计基本与图 4 相同, 区别是图 5 底部包含两个按钮, 先提取水印, 由用户决定是否进行验证。

js 文件调用到的几个微信小程序接口方法包括:

1) wx.chooseImage: 在手机相册中选择图像, 参数包括图像数量, 本文设置为 1, sizeType 设置为 original 不压缩, sourceType 设置为 album 从相册选择, 选择完成后将显示在相应的 image 控件中, 并且调用 wx.uploadFile 接口上传到服务器。

2) wx.uploadFile: 上传手机中的本地文件到服务器, 参数 url 指定 Webservice 的域名以及获取上传图片对应的方法名, 形如 https://域名/Service.asmx/Get_photo, 其中 Service.asmx 是 Webservice 的默认文件, Get_photo 为对应的 WebMethod 名称; 参数 filePath 指定本地文件路径; 参数 name 指定上传文件的键, 服务端根据这个键来获取文件; 参数 header 里指定“content-Type”为“multipart/form-data”。文件上传成功返回该文件在服务器上的路径, 因 Webservice 以 XML 格式返回数据, 调用 SAX (simple API

for XML)库函数从返回的 XML 文件中解析出路径信息。

3) wx.request: 在进行水印嵌入、提取或验证时均需要用到此接口。参数 url 指定 Webservice 的域名及要调用的 WebMethod, 形式同 wx.uploadFile 的 url 参数; 水印嵌入、提取或验证对应的 WebMethod 均需要传入参数, 参数以键值对的方式在 wx.request 的 data 参数中指定, 其中的键必须与服务端 WebMethod 的形式参数同名; method 参数设置为“POST”; 参数 header 里指定“content-Type”为“application/x-www-form-urlencoded”; 返回值由 SAX 库解析。对于水印嵌入和提取, 返回值为整数 1 时执行成功, 执行成功后将调用 wx.downloadFile 从服务器下载含水印的图像或提取的水印;

4) wx.downloadFile: 参数 url 形如 https://域名/二级目录/文件名, 下载成功后临时文件路径在 res.tempFilePath 中, 可调用 wx.saveImageToPhotosAlbum 将图像文件保存至相册;

5) wx.saveImageToPhotosAlbum: 参数 filePath 指定待保存至相册的临时文件路径, 调用成功后该图像文件可在相册中找到。

5. 实验结果

5.1. 基于 2D-DWT 的图像数字水印算法性能评估

本文用 100 张高分辨率图像和两个自定义水印对基于 2D-DWT 的图像数字水印算法性能进行了测试, 图像的大小从 15,451,537 到 60,164,016 不等。用峰值信噪比(PSNR)来评价含水印图像的质量, PSNR 用来衡量嵌入水印后的图像与原始图像相比较质量受影响程度, PSNR 越大则质量越好, PSNR 较低则表明添加水印对图像画质有较大影响; 用归一化相关系数(NC)来评价提取的水印质量, 该系数用于衡量提取的水印与原始水印的一致性, 为[0, 1]间的实数, NC 越接近 1 两者一致性越高(NC = 1 时两者完全相同), NC 越接近 0 则两者差别越大。

$$\text{PSNR}(C, S) = 10 \log_{10} \frac{3M \cdot 255^2}{\|C - S\|^2} \quad (7)$$

$$\text{NC}(W, W') = \frac{\langle W, W' \rangle}{\|W\| \|W'\|} \quad (8)$$

其中 C 和 S 表示载体图像和含水印图像; W 和 W' 归一化的原始水印及提取到的水印。 $\langle \cdot, \cdot \rangle$ 表示内积运算。 M 和 N 分别为图像和水印的像素数量。程序用 C++ 基于 OPENCV 库实现, 小波变换和逆变换用 GPU 用 CUDA 进行了并行化加速。在测试图像集上的统计结果如表 1 所示。

Table 1. Performance evaluation results of image digital watermarking algorithm based on 2D-DWT

表 1. 基于 2D-DWT 的图像数字水印算法性能评估结果

PSNR	NC	嵌入时间(秒)	提取时间(秒)
42.41	0.9940	2.812	2.812

该算法对一些常见的信号处理表现出一定的鲁棒性, 表 2 为测试图像集上鲁棒性统计结果, 以提取到的水印的 NC 来表征。图 3 为可视化结果展示。

Table 2. Robustness test results of image digital watermarking algorithm based on 2D-DWT

表 2. 基于 2D-DWT 的图像数字水印算法鲁棒性测试结果

信号处理类型	高斯噪声 0.03	JPEG 压缩 75%	高斯滤波 $\sigma = 0.01$	均值滤波	中值滤波
NC	0.9772	0.8901	0.8134	0.9103	0.9372

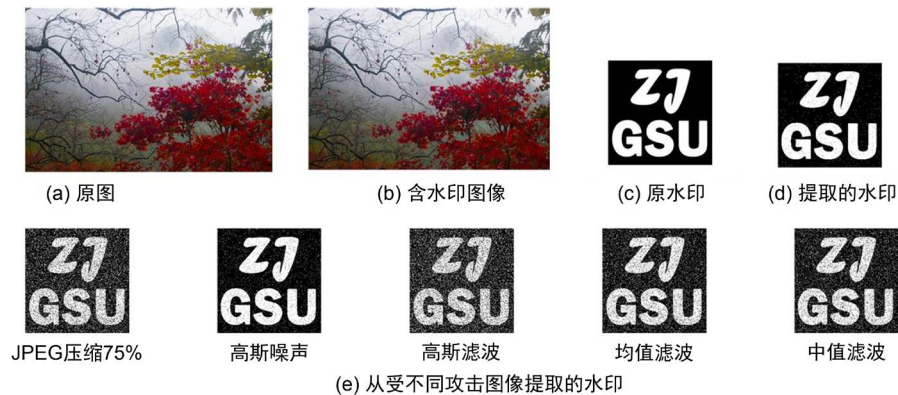


Figure 3. Visualization results of high-resolution image embedding and watermark extraction
图 3. 高分辨率图像嵌入和提取水印可视化结果展示

5.2. 微信小程序结果展示

用微信扫一扫选择微信小程序码图片，或在微信小程序中搜索“图像数字隐水印添加与验证”即可打开小程序并使用。按图 4(a)所示步骤选择图像和水印，输入密码，点击底部按钮，则如图 4(b)原图展示于界面上方，水印展示于下方，水印隐藏完成后含水水印图像将替换原来的水印图像展示于界面下方(如图 4(c))，以便用户比较添加水印前后的图像，点击底部【保存含水水印图像】按钮将含水水印图像保存到相册后界面清空返回图 4(a)所示的初始状态。



Figure 4. Screenshot of the watermark adding interface
图 4. 水印添加界面截图

点击页面右下方的【水印验证】标签，将进入水印提取及验证标签页，如图 5 所示，先在本机中选择含水水印图像，然后在密码输入框输入与嵌入水印时一致的 6 位数字密码，再点击界面左下方的【提取水印】按钮即可从含水水印图像中提取隐藏的水印图像。水印提取完成后会弹出如图 5(b)所示的对话框，点击【确定】按钮将下载提取到的水印图像，并替换原含水水印图像展示于界面上半部分，如图 5(c)所示。

水印验证的功能与水印提取的功能在同一个标签页内完成，水印提取完成后可与原水印图像进行对比验证，验证过程很简单，点击页面下半部分的十字丝印图标在本机上选择原始的水印图像，选择完成后的界面如图 5(d)所示。点击右下方的【验证水印】按钮，服务端进行两水印图像的相似度比较，比较

的结果将弹出对话框显示。验证完成后界面被重置，回到初始状态，在验证和提取过程中，也可以随时点击右上角的重置按钮，重置界面回到初始状态。



Figure 5. Screenshot of the watermark extraction and verification interface
图 5. 水印提取及验证界面截图

6. 结束语

本文基于 2D-DWT 实现了高分辨率图像的数字隐水印算法，包括水印的嵌入、提取和验证等功能，通过 Web Service 技术将实现的程序 API 接口发布在网络服务器上，并设计了微信小程序来调用这些程序功能。基于 CUDA 对算法中的最复杂的 2D-DWT 进行了 GPU 并行运算加速，使程序运行过程中实现实时响应，从而提高用户体验。基于 2D-DWT 的图像数字水印算法虽然对 JPEG 压缩、加噪、滤波等信号处理有一定的鲁棒性，但在旋转、缩放、平移等几何变换攻击下不具备鲁棒性，因此，我们未来的工作是基于深度学习实现对信号处理操作和几何变换攻击均具备较好鲁棒性的算法，并把实现的算法部署到服务器供微信小程序调用。

基金项目

浙江省自然科学基金资助项目(LY20F020002), 浙江省大学生科技创新活动计划暨新苗人才计划(2019R408052), 浙江省大学生科技创新活动计划暨新苗人才计划(2020R408016)。

参考文献

- [1] 李钢, 张国良, 张仁斌. 一种基于分块大容量的 LSB 算法[J]. 合肥工业大学学报, 2006, 29(6): 707-711.
- [2] 谢伟蔚, 肖德贵. 基于块均值调制的彩色图像公开水印技术[J]. 计算机工程与应用, 2007, 43(6): 52-54.
- [3] 王树梅, 张文斌. 一种基于傅里叶变换的鲁棒水印算法[J]. 湖南理工学院学报: 自然科学版, 2019, 32(3): 17-22.
- [4] Cox, I.J., Kilian, J., Leighton, F.T. and Shamoon, T. (1997) Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, **6**, 1673-1687. <https://doi.org/10.1109/83.650120>
- [5] 刘旭. 基于小波变换的数字水印隐藏与检测算法设计[J]. 西安文理学院学报(自然科学版), 2018, 21(4): 42-44.
- [6] 王钢飞, 康琳. 一种新的小波变换域数字水印算法[J]. 太原科技大学学报, 2019, 40(5): 370-375.
- [7] Tsai, J.S., Huang, W.B. and Kuo, Y.H. (2011) On the Selection of Optimal Feature Region Set for Robust Digital Image Watermarking. *IEEE Transactions on Image Processing*, **20**, 735-743. <https://doi.org/10.1109/TIP.2010.2073475>
- [8] Kang, X., Huang, J., Shi, Y.Q. and Lin, Y. (2003) A DWT-DFT Composite Watermarking Scheme Robust to Affine Transform and JPEG Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, **13**, 776-786. <https://doi.org/10.1109/TCSVT.2003.815957>
- [9] Mun, S.M., Nam, S.H., Jang, H., Kim, D. and Lee, H.K. (2019) Finding Robust Domain from Attacks: A Learning Framework for Blind Watermarking. *Neurocomputing*, **337**, 191-202. <https://doi.org/10.1016/j.neucom.2019.01.067>
- [10] Zhu, L.Q., Zhou, Y.D., Zhang, D.X., Wang, D.D., Wang, H.Y. and Wang, X. (2015) Parallel Multi-Level 2D-DWT on CUDA GPUs and Its Application in Ring Artifact Removal. *Concurrency and Computation-Practice & Experience*, **27**, 5188-5202. <https://doi.org/10.1002/cpe.3559>