

作业 2——水印说明文档

191250024 丁炳智 软件学院

一、文件结构说明

此次作业涉及的文件主要包括：

- 1) 水印操作原图: /assets/user_image.png
- 2) 效果演示页面: /watermark.html
- 3) 添加水印脚本: /js/invisible_wm.js && /js/visible_wm.js

二、作业效果演示

注：由于 js 文件跨域问题，直接双击打开 watermark.html 将不会显示水印效果图

解决方案：

- 1) 在 IDE 中以本地服务器的方式打开 watermark.html
- 2) (推荐)访问我的页面 <http://101.43.39.60/watermark.html> 查看效果，由于服务器带宽原因，少数情况下可能出现图片加载较慢或只加载水印的情况，刷新页面即可

效果图见下，从左至右依次为：原图、添加可见水印后、添加不可见水印后、不可见数字水印



三、可见水印思路说明

- 1、传入 base64 图片数据，使用 Image 对象加载图片
- 2、使用 canvas 写入图片，再绘制水印文本
- 3、绘制水印文本思路：
 - ①取传入图片宽度和高度中的最大像素值，并根据此值计算出分布水印之间的距离
 - ②取原图中心点作为原点，将水印文本向图像四周扩展，直至铺满整个图像
- 4、绘制好水印文本后，用 canvas.toDataURL() 方法返回包含水印文本的新图像 base64 码。

注：由于 base64 编码较大，采用 Promise 对象封装后将处理的数据异步返回
- 5、相关代码详见/js/visible_wm.js。

其中 addVisibleWM() 方法调用了 drawWM() 方法，drawWM() 方法包含了添加水印文本的主要逻辑

四、不可见水印思路说明

1、加密：

①先向 `canvas` 画布中绘制水印文本信息，再以 `getImageData()` 方法获取添加了文本水印后的图像数据——一个较大的一维数组，4 个元素为一组，对应于 R/G/B/Alpha 四个通道值

②再以 `getImageData()` 方法获取原图像的数据，也是一个较大的一维数组，以 4 个元素为一组。我们对原图像的像素点通道值进行扰动

③对于原图像的像素点，若相应于同一位置的文本水印画布上有文本信息（即 Alpha 通道值不为 0），说明该像素点处有水印，需要对其进行扰动。在本次作业中，对 G 通道进行扰动，扰动规则为：若有文本信息，则将 G 通道值的末位置 1，否则置 0

④遍历①、②中获取到的两个一维数组，直到所有存在文本信息的像素点均得到扰动

2、解密

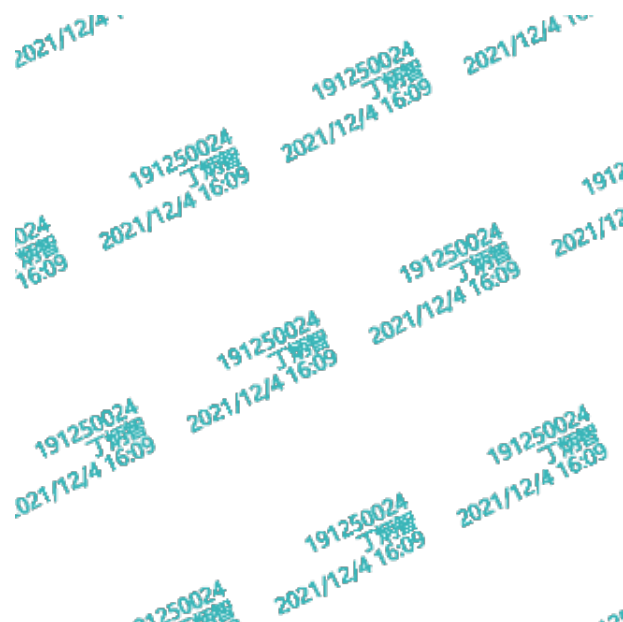
①在加密过程中，我们已经将所有包含文本信息的像素点进行了 G 通道的扰动，即对于水印位置，将 G 通道值末位置 1

②解密时，需要先以 `getImageData()` 方法获取加密图像的图像数据一维数组，然后遍历此数组，G 通道值末位为 1 的像素点即为加密像素点，否则为未加密像素点

③对于所有加密像素点，我们可以为它们设置一个颜色，本次作业中取青色：#3ab6b9 

④对于所有未加密像素点，我们将其设为无像素，即 R/G/B/Alpha 四个通道值均为 0

⑤对所有像素点操作结束后，最终只会保留显示了青色的加密像素点，即加密时添加的水印文本



（加密像素点解密后显形）

⑥相关代码详见 `/js/invisible_wm.js`，其中添加文本水印时，引用了 `visible_wm.js` 中的 `drawWM()` 方法；`addDisturb()` 方法包含主要加密逻辑，`exposeWM()` 方法包含主要解密逻辑