

# 使用SVD算法进行图像数字水印加密解密实验报告

姓名： 张旭

学号： MG1633101

日期： 2017.4.22

## 1 基本问题

### 1.1 问题描述

数字水印（Digital Watermarking）技术是将一些标识信息（即数字水印）直接嵌入数字载体（包括多媒体、文档、软件等）当中，但不影响原载体的使用价值，也不容易被人的知觉系统（如视觉或听觉系统）觉察或注意到。通过这些隐藏在载体中的信息，可以达到确认内容创建者、购买者、传送隐秘信息或者判断载体是否被篡改等目的。数字水印是信息隐藏技术的一个重要研究方向。

作为数字水印技术基本上具有下面几个方面的特点：

- 安全性：数字水印的信息应是安全的，难以篡改或伪造，同时，应当有较低的误检测率，当原内容发生变化时，数字水印应当发生变化，从而可以检测原始数据的变更；当然数字水印同样对重复添加有很强的抵抗性
- 隐蔽性：数字水印应是不可知觉的，而且应不影响被保护数据的正常使用；不会降质；
- 鲁棒性：是指在经历多种无意或有意的信号处理过程后，数字水印仍能保持部分完整性并能被准确鉴别。可能的信号处理过程包括信道噪声、滤波、数/模与模/数转换、重采样、剪切、位移、尺度变化以及有损压缩编码等。主要用于版权保护的数字水印易损水印（Fragile Watermarking），主要用于完整性保护，这种水印同样是在内容数据中嵌入不可见的信息。当内容发生改变时，这些水印信息会发生相应的改变，从而可以鉴定原始数据是否被篡改。
- 水印容量：是指载体在不发生形变的前提下可嵌入的水印信息量。嵌入的水印信息必须足以表示多媒体内容的创建者或所有者的标志信息，或购买者的序列号，这样有利于解决版权纠纷，保护数字产权合法拥有者的利益。尤其是隐蔽通信领域的特殊性，对水印的容量需求很大。

## 2 算法流程

使用奇异值分解（SVD）算法对原始图像加入数字水印和数字水印的解密方法是一种比较常见的数字水印技术，下面分别从加密和解密两个过程说明该算法的流程。

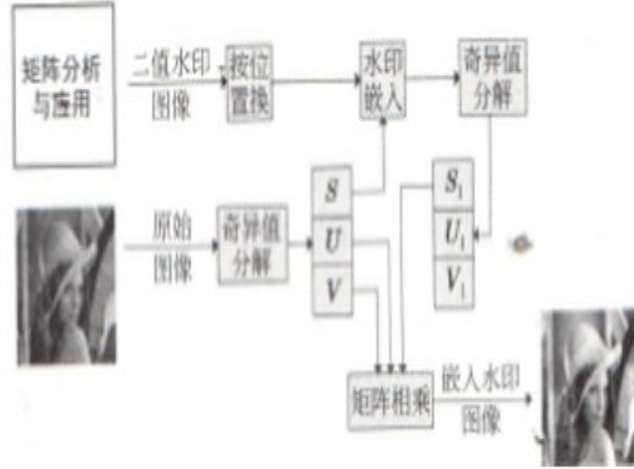


Figure 1: 加密水印算法流程图

## 2.1 加密算法

假设 $A$ 为载体图像，也就是要加密的图像， $W$ 是要嵌入的水印图像， $\alpha$ 是水印强度参数，是一个由用户定义的参数。按照如下公式进行数字水印的嵌入：

$$A \Rightarrow USV^T \quad (1)$$

$$L \Leftarrow S + \alpha W \quad (2)$$

$$L \Rightarrow U_1 S_1 V_1^T \quad (3)$$

$$Aw \Leftarrow US_1 V^T \quad (4)$$

其中公式(1)和公式(3)均为奇异值分解。

这个算法的流程图如下图所示：

## 2.2 解密算法

提取水印过程中，假设 $P$ 为待检测图像，嵌入过程中 $U_1, V_1, \alpha, S$ 均是要保留的参数，作为密钥使用。

$$P \Rightarrow U_p S_p V_p^T \quad (5)$$

$$F \Leftarrow U_1 S_p V_1^T \quad (6)$$

$$WE \Leftarrow (F - S) \alpha \quad (7)$$

这个算法的流程图如下图所示：

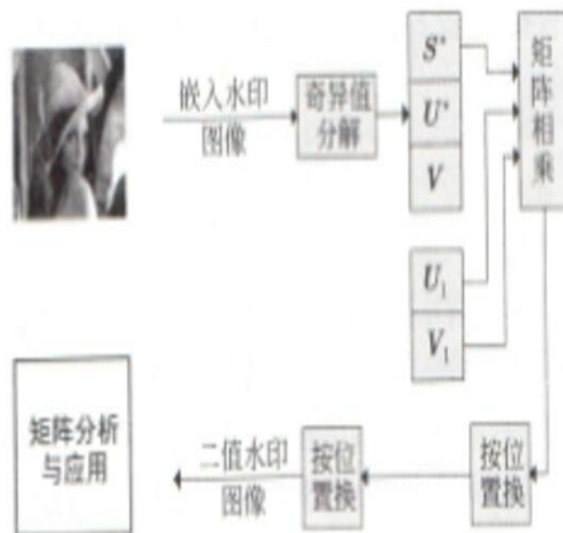


Figure 2: 解密水印算法流程图

### 3 实验

#### 3.1 实验过程与结果

实际实验中，使用python2.7.11作为编程语言，需要numpy、scipy和opencv-python第三方库的支持。其中numpy和scipy主要是完成数值计算部分的任务，如奇异值分解等；opencv主要复杂图像的读取、显示等图像数据处理的任务。

首先，程序读取载体图像和水印图像，并做RGB转灰度处理，使之成为单通道图像，变成二维矩阵易于处理。由于图像大小可能不一致，在此统一使用双线性差值的办法，将图像尺寸统一。这里使用的载体图像是一幅南京大学北大楼的图片，水印图像是学号与姓名。

读取图像后，按照如上所述的SVD加密算法加入水印，为了观察效果，还要在中间显示加入水印后的图像。这里涉及到了数字图像储存格式的问题，处理不当显示效果会有问题，这里采取的处理原则是，计算过程中统一使用32位浮点型计算，计算完毕将数据格式转为无符号8位整型，相当于把图像恢复到原始的数据格式。

加密完成之后，继续解密过程，按照上述算法将加密后的图像作为输入进行解密，最终得到的就是水印图像的近似。

程序结果图像显示如下图示：

#### 3.2 结果分析

实验中采取不同的水印强度 $\alpha$ ，分别取1.0,0.5,0.2和一个极小的值0.01。水印强度不同，最终得到的解密水印图像也不同。按照结果来看，水印强度高，加入水印的图像距离原始图像的相似度会低，对原图的改动也会比较大。解码出的图像大致相同。

四种对比图如下所示：

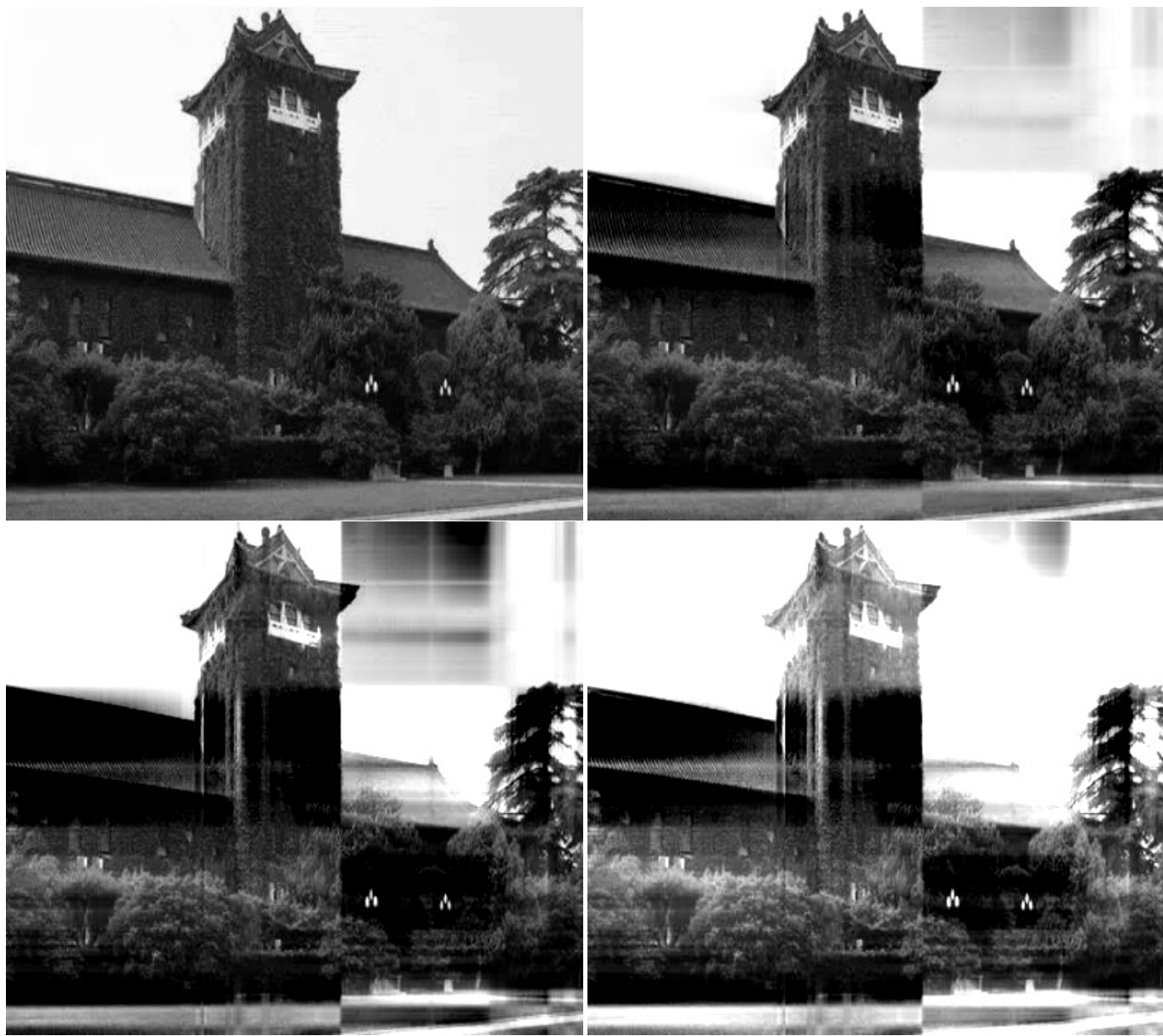


Figure 3: 加入水印图片,  $\alpha$ 分别为0.2,0.5,1.0,0.01

**MG1633101**

**张旭**

**MG1633101**

**张旭**

**MG1633101**

**张旭**

**MG1633101**

**张旭**

Figure 4: 解码后的水印图片,  $\alpha$ 分别为0.2,0.5,1.0,0.01

## 4 总结

实验中使用了svd算法加密解密了数字水印，观察了不同水印强度下的加密和恢复效果，对于SVD分解有了新的认识 and 了解，也对于实际SVD的应用有了更多的操作经验。

## 5 程序清单及运行说明

工程文件包含一个waterMark.py文件，使用python2.7执行该py文件，在当前工作目录下即可生成对应的结果图像文件。