

基于宿主图像差值矩阵的可逆信息隐藏算法^[1]

一、实验名称

基于宿主图像差值矩阵的可逆信息隐藏算法

二、实验目的

学习新的可逆信息隐藏算法，通过实践学习科学的实验方法。

三、算法描述

1. 图像预处理

A. 生成差值矩阵

差值矩阵的元就是原始图像行上相邻像素值的差值，分布在 $[-255, 255]$ 之间，而相邻像素差异很小，所以，得到的差值矩阵的元主要位于“0”附近，一般情况下，大多数元是“0”。

$D=(d)=I(i, 2j)-I(i, 2j-1)$, $d \in [-255, 255]$, 其中, $i=1:M; j=1:N/2$ 。

d_{\max} : 表示差值矩阵 D 中出现次数最多的元的取值;

D_{\max} : d_{\max} 在差值矩阵 D 中出现的次数;

d_{\min} : $[0, 255]$ 区间中不属于差值矩阵的最小整数;

D_{\min} : $[0, 255]$ 区间中不属于差值矩阵的整数个数。

B. 扫描修改差值矩阵

扫描差值矩阵 D ，将值在 d_{\max} 和 d_{\min} 之间的元加 1，到修改后的差值矩阵 D' ：

$$D'=(d'), d' = \begin{cases} d+1 & \text{if } d_{\max} < d < d_{\min} \\ d & \text{otherwise} \end{cases}$$

C. 修改宿主图像

$$I'(i, 2j) = \begin{cases} I'(i, 2j) + 1 & \text{if } d_{\max} < d < d_{\min} \\ I'(i, 2j) & \text{otherwise} \end{cases}$$

2.信息嵌入

A.根据嵌入信息修改差值矩阵

$$D''=(d''), d'' = \begin{cases} d'+1 & \text{if } d'=d_{\max} \text{ \& } W(K)=1 \\ d' & \text{if } d'=d_{\max} \text{ \& } W(K)=0 \end{cases}$$

B. 根据秘密信息修改图像像素值， 嵌入水印

$$I''(i,j) = \begin{cases} I''(i,j)+1 & \text{if } d'=d_{\max} \text{ \& } W(K)=1 \\ I''(i,j) & \text{if } d'=d_{\max} \text{ \& } W(K)=0 \end{cases}$$

$$I''(i,2j-1) = I''(i,2j-1)$$

3.信息提取

A.生成含信息图像的差值矩阵

$$D=(d)=I(i, 2j)-I(i, 2j-1), d \in [-255, 255], \text{其中, } i=1:M; j=1:N/2。$$

B.从差值矩阵提取水印信息

利用与嵌入过程相同的扫描方式扫描差值矩阵 D_x ， 若检测到值为 d_{\max} 的元， 则提取信息比特 0， 若检测到值为 $(d_{\max}+1)$ 的元， 则提取信息比特 1， 秘密信息由此被提取。

$$W(k) = \begin{cases} 1 & \text{if } dx=d_{\max}+1 \\ 0 & \text{if } dx=d_{\max} \end{cases}$$

4.宿主图像恢复

A.对嵌入过程进行恢复

$$I_x'(i,2j)=\begin{cases} I''(i,2j)-1 & \text{if } dx=d_{\max}+1 \\ I''(i,2j) & \text{if } dx=d_{\max} \end{cases}$$

$$I_x'(i,2j-1)=I''(i,2j-1)$$

B.对预处理过程进行恢复

$$I_x''(i,2j)=\begin{cases} I_x'(i,2j)-1 & \text{if } d_{\max}<d\leq d_{\min}+1 \\ I_x'(i,2j) & \text{otherwise} \end{cases}$$

$$I_x''(i,2j-1)=I_x'(i,2j-1)$$

5.溢出问题避免

图像的预处理增大了算法的嵌入量，但嵌入过程可能发生像素值的溢出。如果映射到原始图像中的嵌入位像素值是“255”，那么在嵌入信息比特“1”时就会发生溢出。

$$I''(i,2j)=\begin{cases} I''(i,2j)+1 & \text{if } d'=d_{\max} \& W(K)=1 \& I''(i,2j)\neq 255 \\ I''(i,2j) & \text{if } d'=d_{\max} \& W(K)=0 \end{cases}$$

$$I''(i,2j-1)=I''(i,2j-1)$$

即在嵌入过程中先检测嵌入位，如果其像素值为“255”，则放弃此嵌入位并对其做标记;在提取时，如果遇到像素值为“255”且经过标记的嵌入位，则此位不提取水印。

6.PSNR 计算

假设极端情况，即宿主图像所有偶数列与基数列的差值均等于 d_{\max} ，在水印嵌入过程中，宿主图像的所有偶数列的像素值都被加‘1’，此时，整个宿主图像 1/2 的像素值都被加‘1’。

$$MSE=1/(m\times n)\sum (I(m,n)-J(m,n))^2=1/2$$

$$PSNR=10 \log_{10} (MAX\times MAX/MSE) =10 \log_{10} (255\times 255/0.5) = 51.14$$

7.SSIM 计算

结构相似性，是一种衡量两幅图像相似度的指标。假设极端情况，即宿主图像所有偶数列与奇数列的差值均等于 d_{max} ，在水印嵌入过程中，宿主图像的所有偶数列的像素值都被加‘1’，此时，整个宿主图像 1/2 的像素值都被加‘1’。

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

【说明】：其中 μ_x 是 x 的平均值， μ_y 是 y 的平均值， σ_x^2 是 x 的方差， σ_y^2 是 y 的方差 σ_{xy} 是 x 和 y 的协方差。 $c_1 = (k_1L)^2$ ， $c_2 = (k_2L)^2$ 是用来维持稳定的常数。 L 是像素值的动态范围。 $k_1 = 0.01, k_2 = 0.03$ 。结构相似性的范围为 0 到 1。当两张图像一模一样时，SSIM 的值等于 1。作为结构相似性理论的实现，结构相似度指数从图像组成的角度将结构信息定义为独立于亮度、对比度的，反映场景中物体结构的属性，并将失真建模为亮度、对比度和结构三个不同因素的组合。用均值作为亮度的估计，标准差作为对比度的估计，协方差作为结构相似程度的度量。

四、算法提出背景原因

A..已有算法缺陷

传统的基于直方图修改的可逆信息隐藏算法^[2]，即通过寻找图像直方图的峰值，移动和修改原始图像在峰值范围内的灰度值，获得嵌入位置，完成信息嵌入；接收方提出秘密信息并可以精确恢复出原始图像。本实验算法对比以上传统的基于直方图修改的可逆信息隐藏算法，避免对图像本身的过度依赖性能，较大提升的是容量指标。

同样，提出一种标志点算法的改进方案^[3]。通过标志点和移动点相对位置的比较和像素值的修改，来生成嵌入信息的空位，同时标志点的设定可以有效避免灰度值溢出现象的发生，但仍有影响。

文献[2]算法简介：该算法的思路是：在原始图像的直方图中寻找某个零值点作为标志点，再将图像分成适当大小的块，用密钥信息决定块中将要修改的像素值。在分块直方图中，将要修改的像素灰度值与标志点进行比较来决定移动方向，从而在分块图像中生成可以嵌入水印的空间。同时，标志点的设定可防止像素值取值在“0”和“255”处的移动可能发生的上溢或者下溢情况，而密钥的参与则使得水印的篡改增加了难度标志点的确定依赖于图像整体直方图中的零值点位置，而在应用中可能出现直方图中没有零值点的特例。对于这种情况，可用最小值点来代替零值点，同时记下最小值点的位置信息，将其作为压缩扩展错误信息供以后在恢复图像时使用。

可逆信息隐藏技术研究主要包括如下 2 个方面：

(1)在保证图像视觉质量(PSNR)的前提下，提高嵌入容量的问题。基于直方图的可逆信息隐藏算法的嵌入容量受到图像直方图的最大值和零点值或最小值的制约，图像直方图最大值所对应的像

素值数目越多，其嵌入量越大。同时，并非所有图像的直方图都存在“零点值”，只能找出“最小值”来替代“零点值”，为保证算法“无损”，在嵌入过程中必须记录“最小值”的确切信息，建立 Locationma 记录最小值点的确切位置，而记录这些确切信息势必浪费空间，降低水印嵌入量。

(2)算法是否会出现像素值溢出的问题。如果像素值溢出，则会在图像中出现噪声点，而且这些噪声点不能被消除，导致算法不可逆。

本实验算法综合考虑水印嵌入容量和像素值溢出问题，对比文献[1]和文献[2]，进行性能提升工作。

本实验算法将原始图像直方图“最值问题”转换为差值矩阵“最值问题”(即预处理的优势)，对于一幅图像，某个像素值重复出现的次数一般小于相邻像素值相等的像素对的个数，同时，差值矩阵分布比较集中，在[0, 255]区间中没有出现的元的数目很多。也就是说，图像中重复出现次数最多的像素值出现的次数远少于图像相邻像素值相等的像素对的个数，且在[0.255]区间中，图像的差值矩阵有更多的“零元”，因此极大地提升了嵌入容量。原始图像直方图的最大值是图像中出现次数最多的像素值出现的次数，原始图像的零值是图像中没有出现或者出现次数最少的像素值。本实验算法中差值矩阵的 Dmax 和 Dmin 分别是原始图像相邻像素值的差值在[0, 255]区间中出现次数最多的元 dmax 的次数和没有出现的元 dmin 的个数。一般情况下，差值矩阵的 dmax=0，表明原始图像所有相邻像素对中相邻像素取值相等的像素对最多。

五、算法测试



图 (0) 图片库 (25 幅灰度图像)

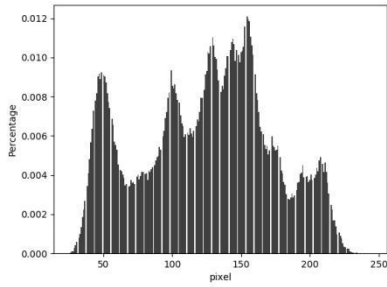
A.信息嵌入、提取测试



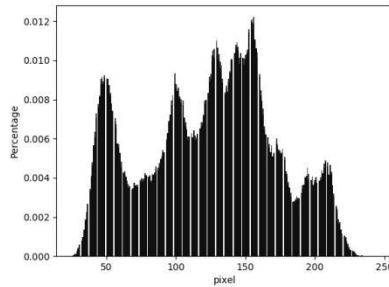
(a) 原图像

(b) 嵌入信息后图像

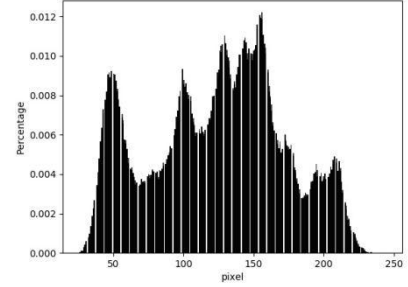
(c) 水印提取后恢复图像



(d)原图像灰度直方图



(e) 嵌入信息后图像后灰度直方图



(f) 恢复图像灰度直方图

```

/Users/cclin/anaconda3/bin/python /Users/cclin/PycharmProjects/project/projec
The embed watermark is fglnpqhswwxveatjtovpmfywowfftjeipuhesfsjprpdesrtgfvn
The extracted watermark is: fglnpqhswwxveatjtovpmfywowfftjeipuhesfsjprpdesrt
There is no difference between original image and recovered image!
There is no difference between embed watermark and extracted watermark!

```

(g) 原图像与恢复后图像像素差值对比



图 (h) 宿主图像与恢复图像差异对比 (512*512 像素值均为 255 表示该像素点原图像与恢复图像无差异)

图 (g) 、 (h) 表明本实验算法能够完全成功提取嵌入信息，并能完全恢复出原始宿主图像。

通过对比图像 (a) (b) (c) ,本实验算法不会造成肉眼可见的噪点，对比原图像，嵌入水印后图像，恢复图像的灰度直方图，直方图的统计数据未发生较大程度的更改，该实验算法透明性很高，且具有很好的抗统计分析特性，

B.水印嵌入过程中差值矩阵直方图

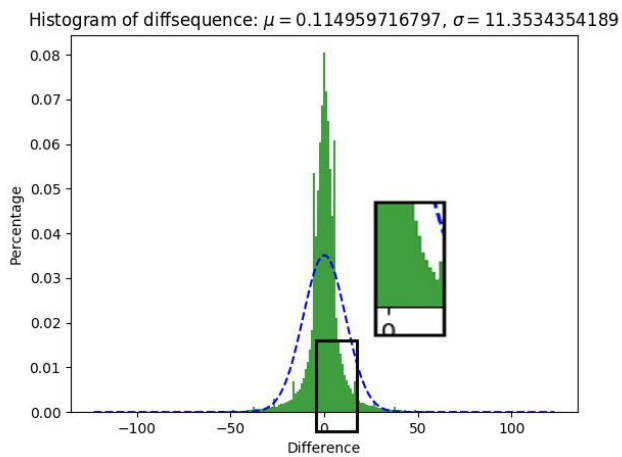


图 (1) 原始图像

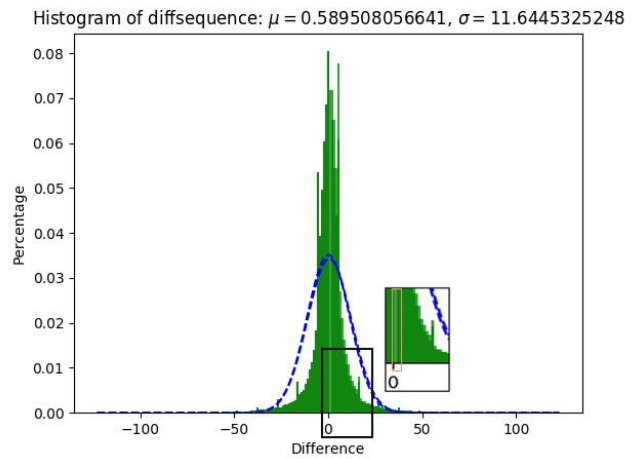


图 (2) 预处理后

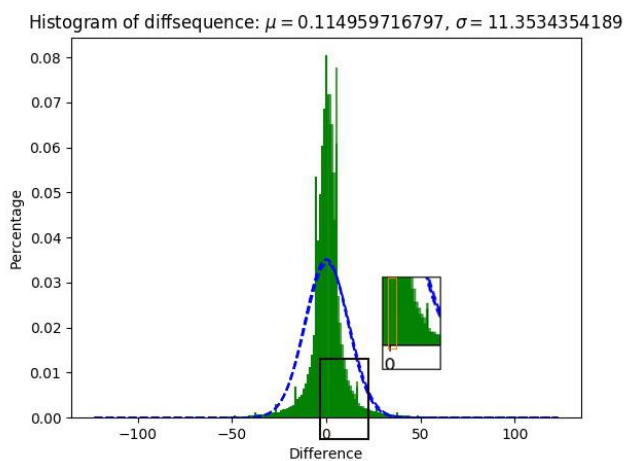


图 (3) 图像嵌入信息后

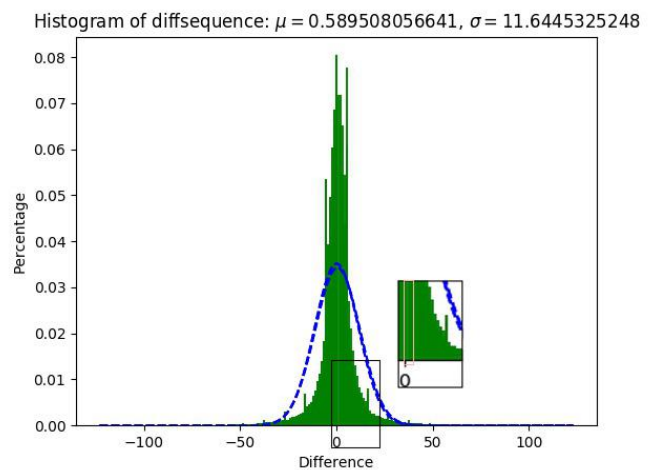


图 (4) 对嵌入过程恢复后

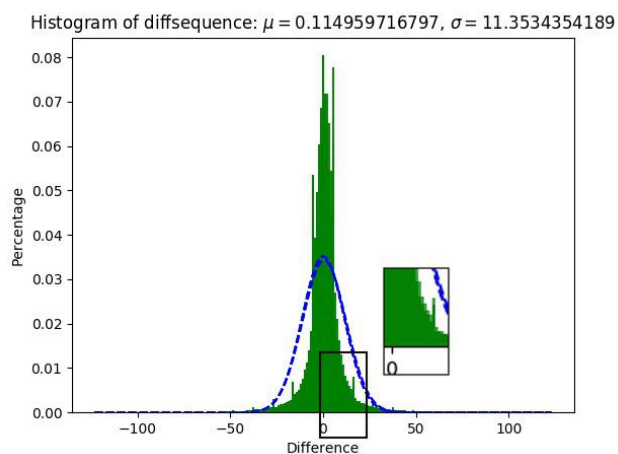


图 (5) 对预处理过程恢复后

同样通过差值矩阵的直方图可以明显观测到图 (1) 和图 (5) 差值矩阵直方图均值和标准差相等, 且差值矩阵直方图数据相同; 图 (2) 和图 (4) 差值矩阵直方图均值和标准差相等, 且差值矩

阵直方图数据相同。可以对应图像预处理过程、嵌入信息过程、对嵌入过程恢复过程，对预处理过程进行恢复过程。在水印嵌入过程中会给在 d_{max} , d_{min} 之间的差值加 1，同样，图像的像素值也做出相应修改，由于嵌入水印的容量比较小，对于 lena512 仅 4.43%，图像的差值矩阵直方图的变化并不明显。

C.不同图像嵌入信息， 嵌入位置对比



图 1.1 lena512

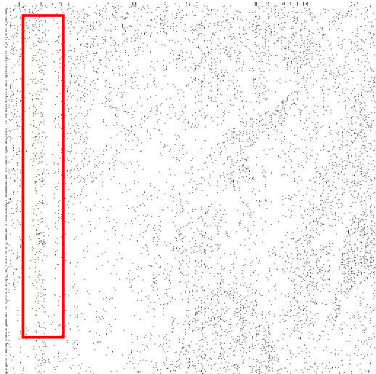


图 1.2 嵌入位置分布

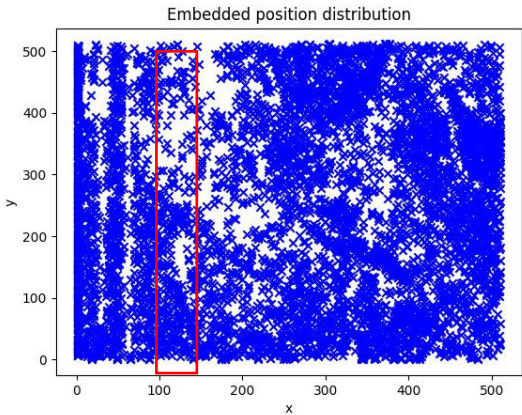


图 1.3 嵌入位置分布 (散点图)

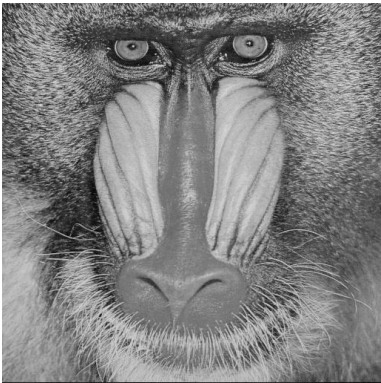


图 2.1 baboon512

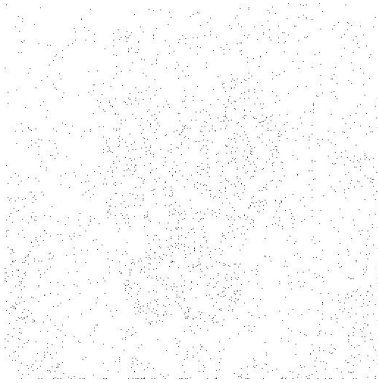


图 2.2 嵌入位置分布

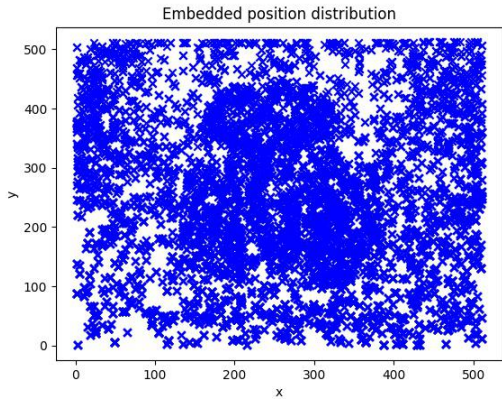


图 2.3 嵌入位置分布 (散点图)

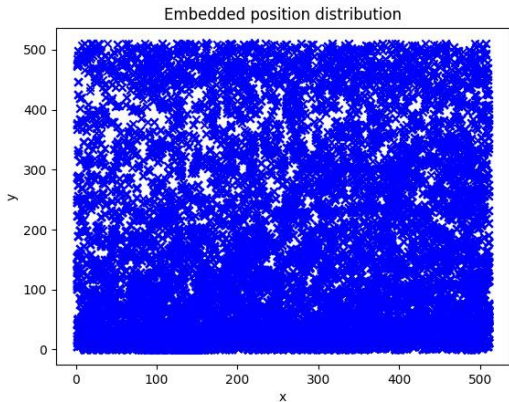
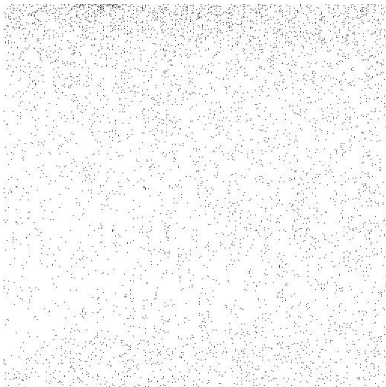


图 3.1 goldhill512

图 3.2 嵌入位置分布

图 3.3 嵌入位置分布(散点图)

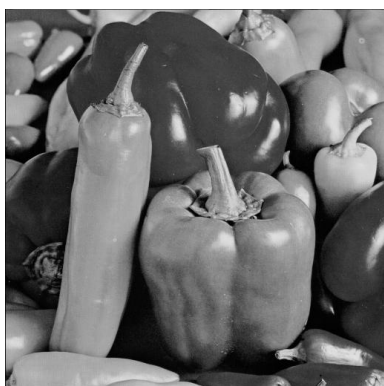


图 4.1 peppers512



图 4.2 嵌入位置分布

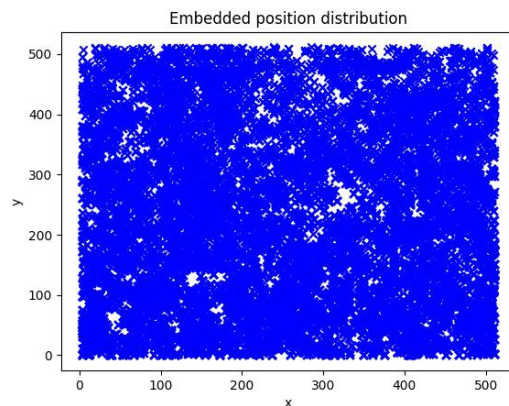


图 4.3 嵌入位置分布 (散点图)

对比宿主图像和嵌入位置分布，嵌入位置主要位于宿主图像平滑区，尤其图 1.1 和图 1.2 对比，宿主图像的边缘地区和嵌入图像刚好吻合，这符合该算法基于差值矩阵，边缘区域相对于内部平滑区域相邻像素值的差值大，嵌入信息的可能小。

对比图 1.1，2.1，3.1，4.1 和 1.2，2.2，3.2，4.2，图像平滑区域面积越大，嵌入容量越大，对比 4 幅图 baboon512 的边缘区域明显多于 lena512，嵌入容量更小。

D.控制宿主图像容量，嵌入不同长度信息

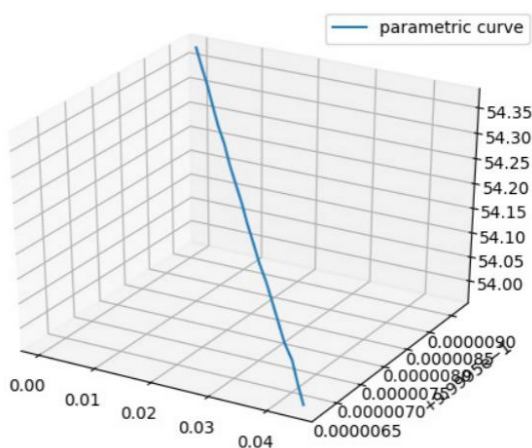


图 A lena512

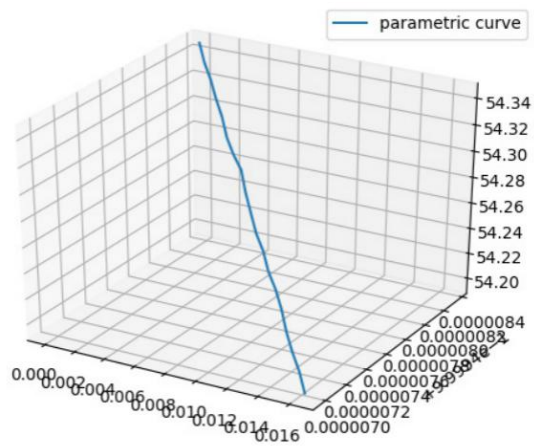


图 B baboon512

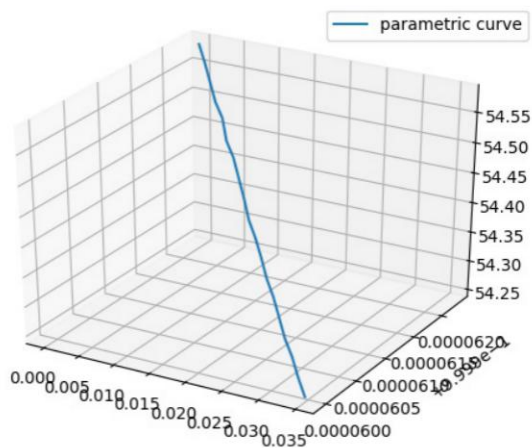


图 C goldhill512

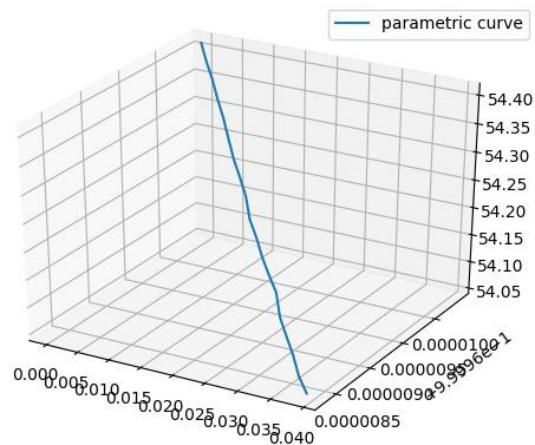


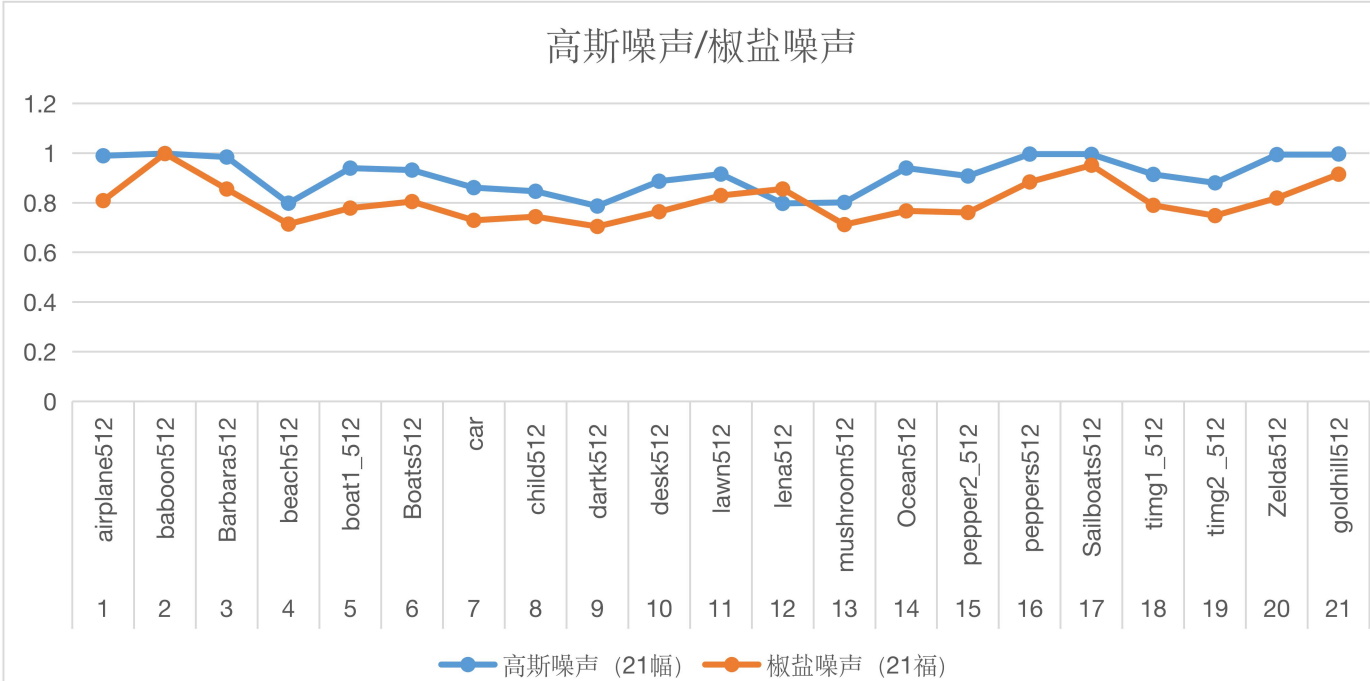
图 D peppers512

实验的四幅图像忽略细节性差异，在总体变化趋势上总是随着嵌入容量的增大，图像的 PSNR 和 SSIM 都随之下落，PSNR 的浮动在 0.35 范围内。同样，SSIM 的浮动在 $6e-1$ 范围内，信息的嵌入不会给图像带来较大结构的改变，肉眼更不可能辨识。

E.鲁棒性测试

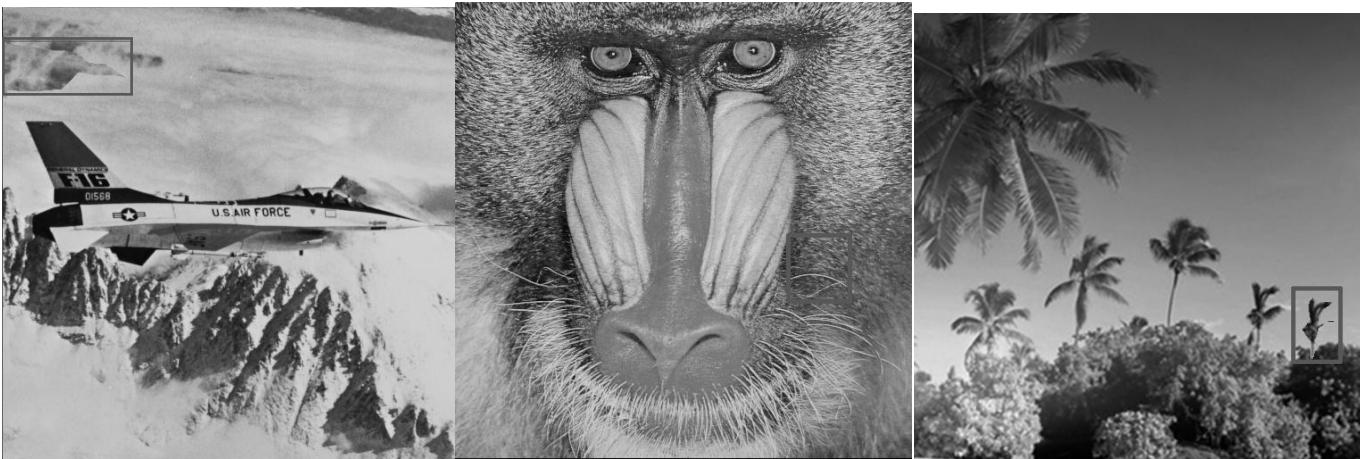
序号	图像名称	高斯噪声 (21 幅)	椒盐噪声 (21 幅)	中值滤波 (8 幅)	区域赋值 (6 幅)
1	airplane512	0.98924	0.80860	0.99784	0.99623
2	baboon512	0.99812	0.99812	0.99812	0.99625
3	Barbara512	0.98413	0.85525	0.99867	-
4	beach512	0.79823	0.71412	1.0	0.99865
5	boat1_512	0.93945	0.77848	1.0	0.99548
6	Boats512	0.93160	0.80476	-	-
7	car	0.86094	0.72925	-	-
8	child512	0.84623	0.74391	-	-
9	dartk512	0.78669	0.70452	-	-
10	desk512	0.88674	0.76386	-	-
11	lawn512	0.91548	0.82919	-	-
12	lena512	0.79738	0.85547	-	0.99862
13	mushroom512	0.80132	0.71198	-	-
14	Ocean512	0.93980	0.76711	-	-

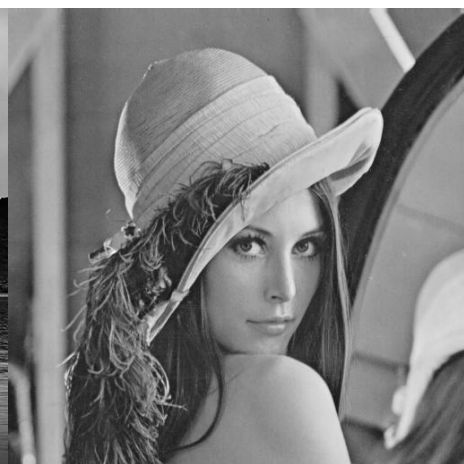
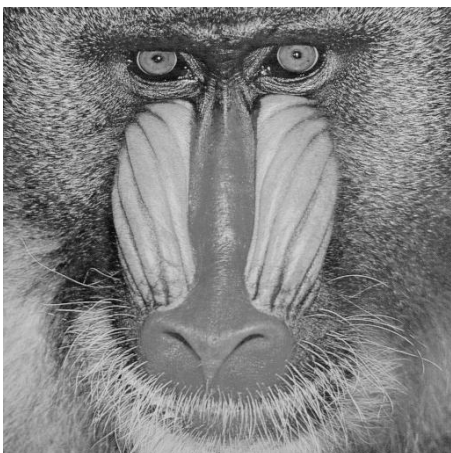
15	pepper2_512	0.90768	0.76065	-	-
16	peppers512	0.99606	0.88364	-	-
17	Sailboats512	0.99480	0.95119	-	-
18	timg1_512	0.91414	0.78971	0.99867	-
19	timg2_512	0.88014	0.74812	0.99812	-
20	Zelda512	0.99394	0.81903	0.99614	-
21	goldhill512	0.99649	0.91498	-	0.99824
均值		0.91232	0.80628	0.99845	0.99725



折线图 1 不同图像嵌入信息后图像受高斯噪声、椒盐噪声影响

以下列出图像区域复制的区域（上方为区域复制图像）：





该算法为可逆水印嵌入算法，对于鲁棒性并没有较高的性能要求，但通过以上实验再一次证明该算法的鲁棒性较差，难以抵抗高斯噪声、椒盐噪声、中值滤波，图像区域复制等攻击。基于大量测试，根据测试结果寻找提升该算法性能的途径。

F. 本实验算法与基于直方图修改的可逆信息隐藏算法（后记为算法 A）对比

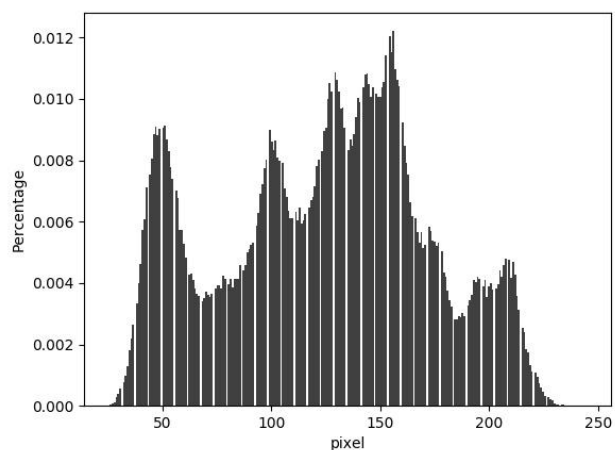


图 1.1lena512 灰度直方图

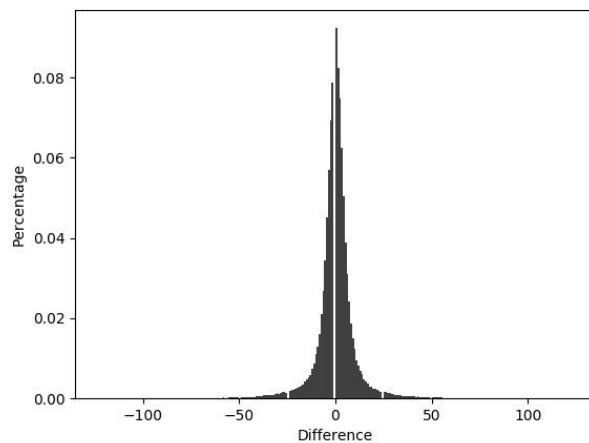


图 1.2 lena512 差值矩阵直方图

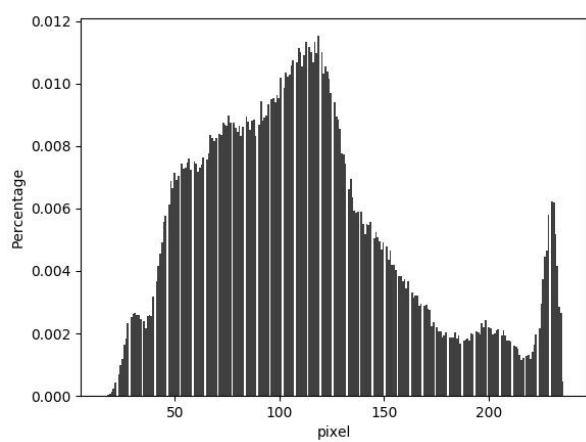


图 2.1 goldhil512 灰度直方图

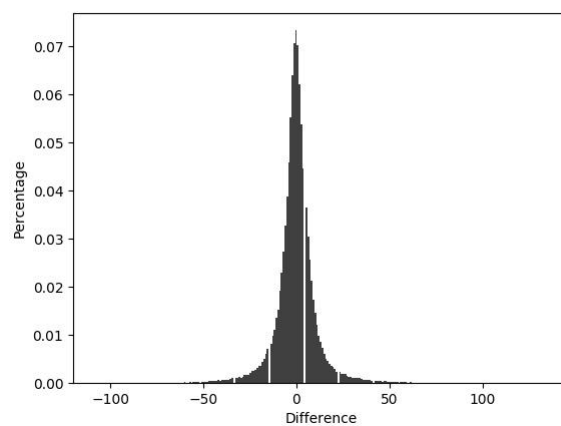


图 2.2 goldhill 差值矩阵直方图

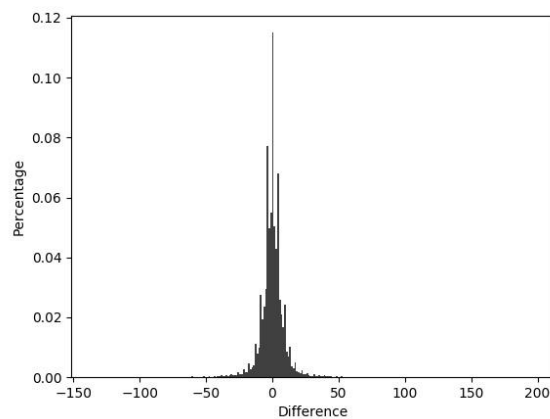
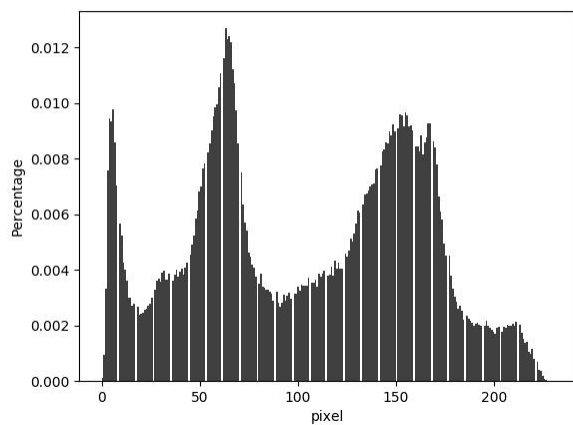


图 3.1 peppers512 灰度直方图

图 3.2peppers 差值矩阵直方图

测试图像	原始图像灰度直方图最 大值个数	原始图像灰度直方图的零 值个数或者最小值个数	差值矩阵的 Dmax/dmax	差值矩阵的 dmin
lena512	2723	0	11628/0	96
baboon512	2759	39	4273/0	108
goldhil512	2618	0	9131/0	86
peppers512	2950	24	10183/0	135

表一 直方图与差值矩阵参量对比(512*512)

测试图像	本实验算法 PSNR (dB)	算法 A PSNR (dB)	本实验算法嵌入容量/ 嵌入率	算法 A 嵌入容量/ 嵌入率	本实验算法 SSIM	算法 A SSIM
lena512	53.97	49.57	11628/4.44%	2723/1.03%	0.99995623	0.99993979
baboon512	54.19	50.44	4270/1.63%	2759/1.05%	0.99994698	0.99992370
goldhill512	54.24	50.25	9131/3.48%	2618/0.99%	0.99995996	0.99993700
peppers512	54.04	49.83	10183/3.88%	2950/1.12%	0.99996834	0.99994637

表二 PSNR、嵌入容量对比(512*512)

基于直方图修改的可逆信息隐藏算法的水印嵌入容量依赖于图像本身的特性，即图像灰度直方图最大值的个数和最小值，从表一可以对比发现，对于测试的多幅图像，其灰度直方图的最大值个数远小于差值矩阵直方图最大值 **dmax** 的个数，从而导致嵌入容量其嵌入容量明显低于本实验算法。

表二通过对比本实验算法和算法 A 的 **PSNR**、嵌入容量和 **SSIM**，本实验算法在明显提高了嵌入容量的同时，提高了 **SSIM**，也保证了较高的 **PSNR**。

六、性能提升思考

本实验算法对性能的提升依赖于图像相邻像素值的相似性，从而极大提高可以嵌入水印的像素的数量，思考一种科学的计算方式可以对图像进行预处理，使相邻像素数值相等的像素值对数目增加，在与原图像进行对比时，不会有肉眼可见的差异，并且尽可能减小进行统计学分析（卡方分析）的检测分析结果可能性。

- 1.混沌系数。
- 2.预测分析。

[1] 强 媛, 高宝建, 孟妍妍.基于宿主图像差值矩阵的可逆信息隐藏算法 计算机工程.2009.8 35(16):135-137.

[2] Ni Zhicheng, Shi Yun—qing, Ansari N, et al.Reversible Data Hiding[J].IEEE Transactions on Circuits and Systems for Video Technology, 2006, 1 6(3):354—362.

[3] 李 妍, 张佑生, 张挺.一种基于直方图的可逆数字水印算法 计算机技术与发展.2006.16(10):122—124.