

MATLAB的使用 与信息隐藏实验

助教：屠天扬

2021/4/26



目录 content



01 MATLAB介绍



02 MATLAB安装使用



03 空间域信息隐藏



04 变换域信息隐藏



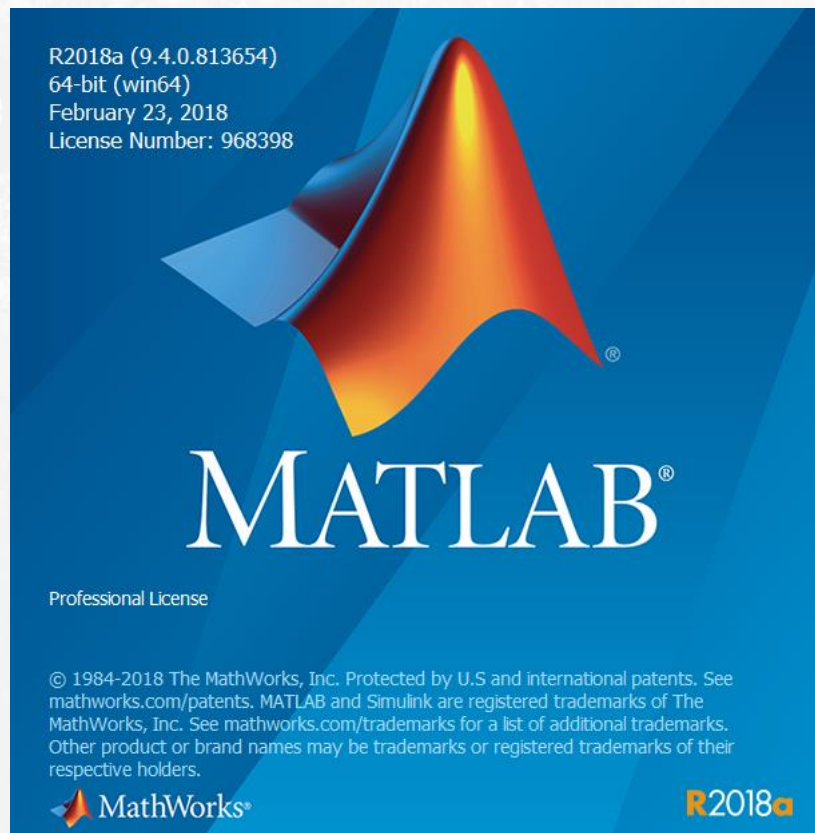
05 语音信息隐藏



01

MATLAB介绍





MATLAB 是美国MathWorks公司出品的商业数学软件，用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。

MATLAB可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序。

主要应用于工程计算、控制设计、信号处理与通讯、图像处理、信号检测等领域。



MATLAB语言

基于C++语言的基础上，语法特征与C++语言极为相似。可移植性好、可拓展性极强，更加符合科技人员对数学表达式的书写格式。



MATLAB函数

包含一个强大的算法集合，所用的算法都是科研和工程计算中的最新研究成果，可以直接调用来代替底层的编程语言，减少编程工作量。



图形处理

方便的数据可视化功能，以将向量和矩阵用图形表现出来，并且可以对图形进行标注和打印。包括二维，三维图像的可视化，和GUI界面的制作。



模块工具

功能强大的模块集和工具箱toolbox，可以直接使用工具箱进行学习包括神经网络，小波分析，优化算法等等。

02

MATLAB安装使用





下载安装：东南大学网络与信息中心→下载专区→东南大学MATLAB校园版安装指南
<https://nic.seu.edu.cn/2019/1017/c23507a322935/page.htm>



东南大学 网络与信息中心

请输入关键字

首页 部门概况 服务流程 网络管理 网络安全 规章制度 部门党建 成果展示 下载专区

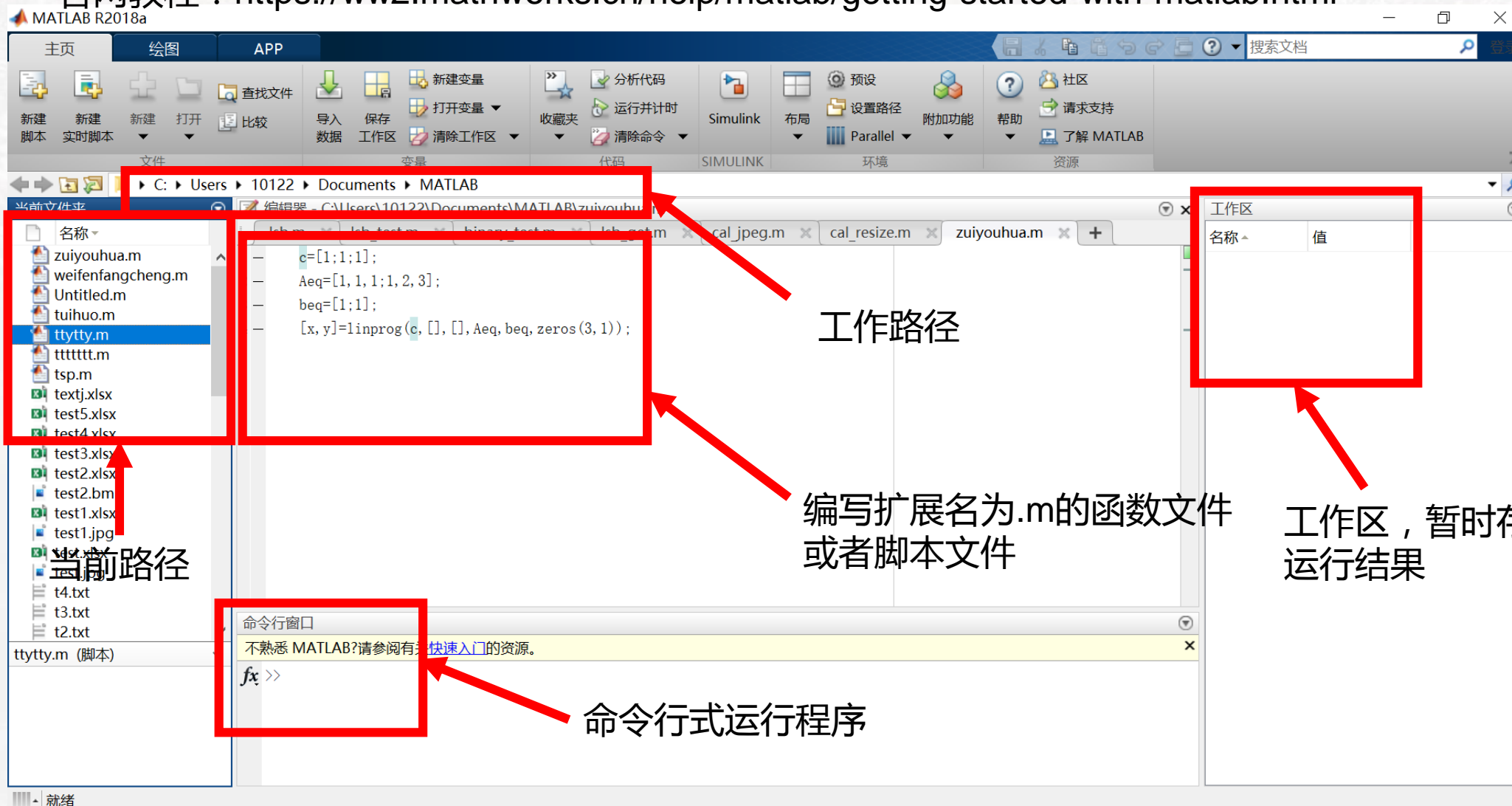
当前位置：首页 | 下载专区

东南大学MATLAB校园版安装指南

发布者：苏聪 发布时间：2019-10-17 浏览次数：1231



官网教程：<https://ww2.mathworks.cn/help/matlab/getting-started-with-matlab.html>



The image shows the MATLAB R2018a interface with several key components highlighted by red boxes and labeled with Chinese text:

- 工作路径 (Current Path):** Located at the top of the file browser, it shows the current directory: `C:\Users\10122\Documents\MATLAB`.
- 当前路径 (Current Path):** A label pointing to the file browser on the left, which lists files like `zuiyouthua.m`, `weifenfangcheng.m`, `Untitled.m`, `tuihuo.m`, `ttytty.m`, `tttttt.m`, `tsp.m`, `textj.xlsx`, `test5.xlsx`, `test4.xlsx`, `test3.xlsx`, `test2.xlsx`, `test2.bm`, `test1.xlsx`, `test1.jpg`, `t4.txt`, `t3.txt`, and `t2.txt`.
- 编写扩展名为.m的函数文件或者脚本文件 (Edit function files or scripts with .m extension):** A label pointing to the code editor in the center, which contains the following MATLAB code:

```
c=[1;1;1];
Aeq=[1, 1, 1;1, 2, 3];
beq=[1;1];
[x,y]=linprog(c, [], [], Aeq, beq, zeros(3,1));
```
- 工作区, 暂时存储运行结果 (Workspace, temporarily store execution results):** A label pointing to the workspace window on the right, which is currently empty.
- 命令行式运行程序 (Command-line execution):** A label pointing to the command window at the bottom, which shows the prompt `fx >>`.



MATLAB中的基本概念



函数

自定义函数：Home→New→function.

文件名:函数名.m

内容: function 输出变量=函数名(输入变量)

将自定义函数保存在工作路径下，可以直接调用。



脚本

新建脚本: Home→New script

文件名扩展名为.m

内容:由用户建立的，通过调用一系列函数来完成计算任务。

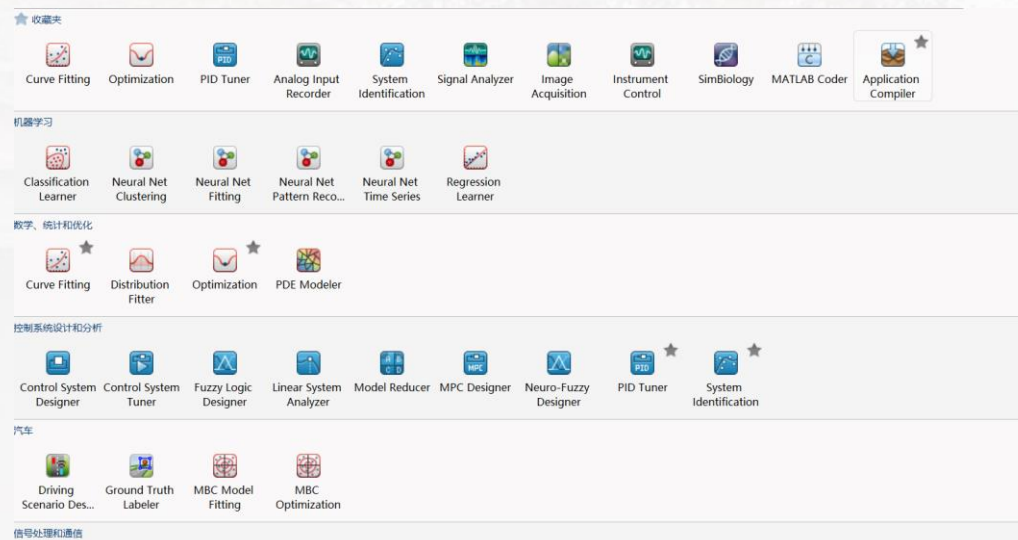
一般保存在工作路径下。



GUI

人机交互界面。

在APPS标签找到所安装的所有已安装的GUI工具。



由GUI打包而成的toolbox



使用注意事项

★良好的编程习惯，增加程序的可读性

MATLAB自带的编辑器可以自动缩进，使程序十分易读。定义变量名要有清晰的含义。

★善于使用断点

当程序出现逻辑错误无法运行的时候，善于使用断点。

★程序模块化

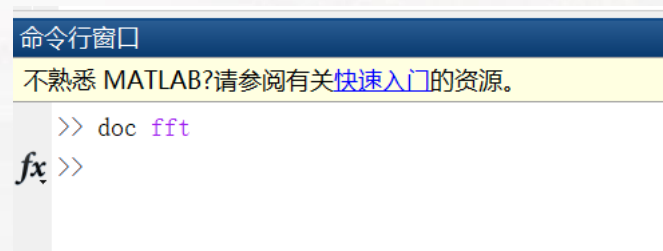
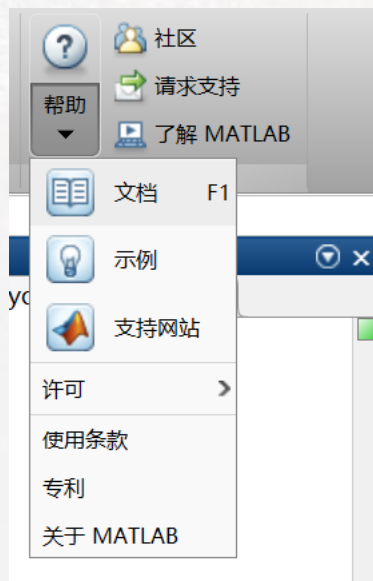
将重复进行的程序写成函数，便于修改和维护。

★学会获取帮助，查阅函数文档



获取帮助的方法

1. Home→Help，获得文档，示例。 2. 在Command Window里输入 doc+函数名 来获得帮助。



3. cn.mathworks.com官网上找到支持。



03

空间域信息隐藏





基于LSB的灰度图像信息隐藏及其抗干扰性研究

实验说明

算法：基于LSB的图像信息隐藏算法。

软件：MATLAB仿真。

载体：320*240灰度图像。

攻击：JPEG压缩攻击。

评价指标：PSNR(峰值信噪比)，误码率。



LSB信息隐藏技术

低比特位(Least Significant Bit, LSB)信息隐藏方法是出现较早的一种信息隐藏技术，其实现比较容易，隐藏时用秘密消息直接替换载体数据最不重要的比特位，提取秘密消息时将最低比特位取出，然后再进行解密等处理，LSB方法虽然抗干扰性较差，但隐藏数据量大，而且对原始数据的修改很小，是一种比较实用的信息隐藏技术，但LSB算法容易被统计检测算法所检测。另外，随着嵌入位数的增加或者选择嵌入的采样点不恰当LSB算法容易失真。

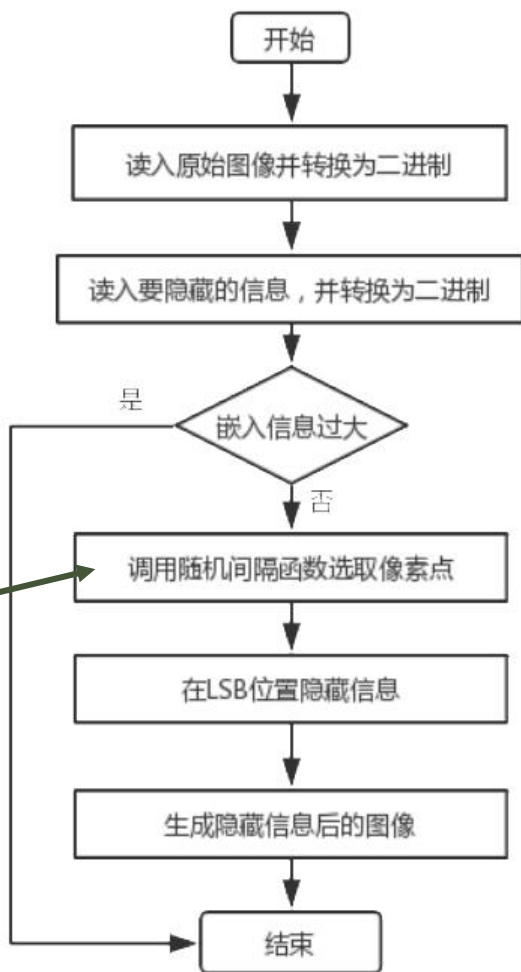
LSB算法是选择一个载体元素的子集 $\{j_1, j_2, j_3 \cdots j_m\}$ ，其中共有 $L(m)$ 个元素，用以隐藏秘密信息的 $L(m)$ 个比特，然后在这个子集上执行替换操作，用秘密信息比特来替换最低位比特。



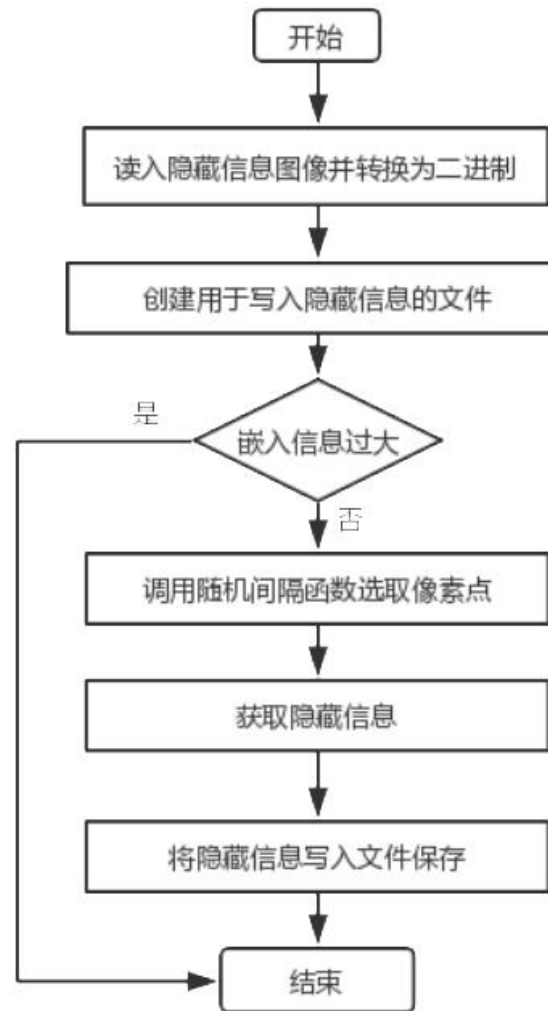
算法流程

Matlab随机数产生器
Randgenerator
基于种子来产生随机序列

在发送端和接收端共享
随机序列种子，就可以
定位秘密信息插入的位
置。



嵌入流程



提取流程



实验结果

原始图像



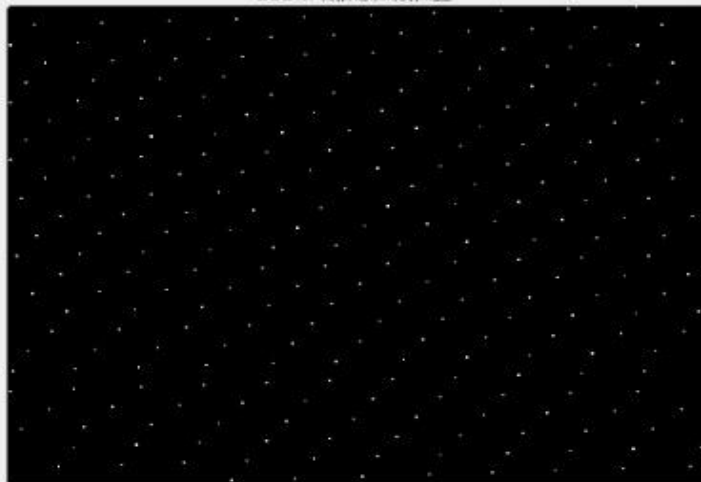
原始图像

隐藏信息的图像



嵌入信息的图像

LSB空域信息隐藏位置



LSB插入信息的空间位置



t2 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V)
网络空间安全与信息隐藏



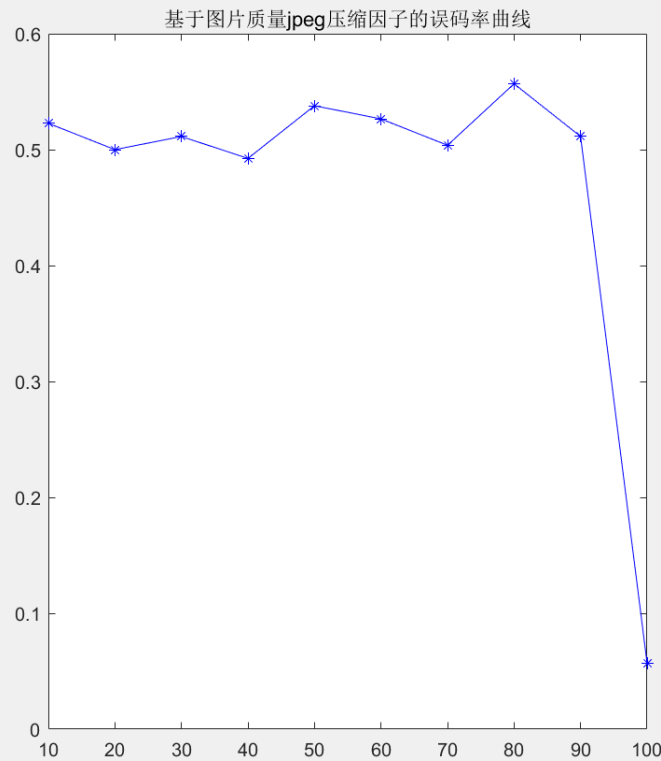
t1 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
网络空间安全与信息隐藏

原始嵌入信息与解密后信息

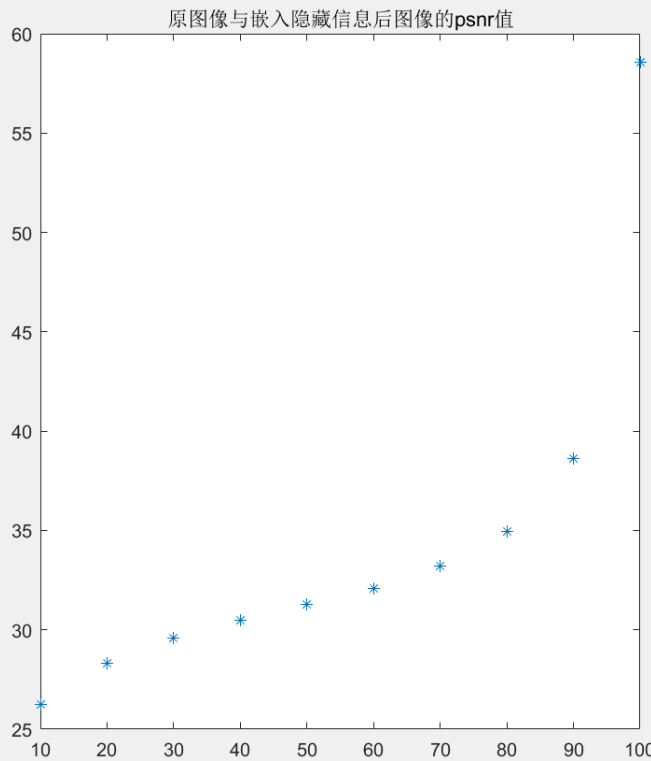


JPEG压缩攻击



基于图片质量JPEG压缩因子的误码曲线

X轴：缩放比例
Y轴：误码率



原图像与嵌入隐藏信息后图像的PSNR值

X轴：缩放比例
Y轴：PSNR值

对伪装对象进行JPEG压缩攻击，计算提取出信息的误码率和伪装对象与原图像的PSNR值。

根据实验结果可知，随着JPEG压缩程度的不断加大，误码率与PSNR值不断增加。



结论

由LSB算法信息隐藏实验可知，LSB算法优点是算法简单，容易实现，并且隐藏容量大，使用私钥隐藏的方法可以提高算法的安全性。伪装对象不易于被肉眼上察觉。

缺点是健壮性弱，在JPEG压缩攻击下，容易擦除隐藏的信息。在人为干扰或者信道噪声的影响下，也容易导致恢复信息的误码率高。



LSB算法改进

传统LSB 嵌入算法的健壮性比较差，不能够抵抗剪切等攻击，由于图像的中心区域一般是图像的最重要区域,也是最不可能被裁剪的区域。因此，在算法中可以通过在载体图像中增加剪切标志，并将秘密信息隐藏在载体图像的中心区域来增强算法的健壮性。



嵌入过程

设秘密信息比特流为 $S(s_0, s_1, s_2)$,它对应的一个载体图像数据字节为 $X(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ 。将载体图像字节中最高 bit 位 x_7 与 $S(s_0, s_1, s_2)$ 进行异或运算, 即 $s' = x_7 \oplus s_0$

- 1) 秘密信息预处理:用密钥 key_1 对秘密信息进行加密处理。
- 2) 根据抗裁剪强度 d 和密钥 key_2 生成抗裁剪标记EmbedMark。
- 3) 将加密后的秘密信息转换成长度为 L 的二进制比特流 $Msg(Msg \in (0,1))$ 。



嵌入过程

4) 在载体图像($M \times N$)的左上角和右下角 $d \times d$ 大小区域的像素最低比特位分别嵌入抗裁剪标记,嵌入规则如下:

①如果抗裁剪标记的第 i ($1 \leq i \leq d \times d$)个比特 $EmbedMark(i)$ 与载体图像左上角和右下角 $d \times d$ 大小区域的第 i 个像素点 $f_i(x, y)$ 的最高位相同,则保持该点的像素值不变,即:

$$f'_i(x, y) = \begin{cases} f_i(x, y), & f_i(x, y) \geq 128 \text{ 且 } EmbedMark(i) = 1 \\ f_i(x, y), & f_i(x, y) < 128 \text{ 且 } EmbedMark(i) = 0 \end{cases}$$

$f_i(x, y)$ 为载体图像左上角或右下角区域的第 i 个像素点的像素值, $f'_i(x, y)$ 为嵌入标记比特后该像素点的像素值。

②如果抗裁剪标记的第 i 个比特 $EmbedMark(i)$ 与载体图像左上角和右下角大小 $d \times d$ 区域的第 i 个像素点 $f_i(x, y)$ 的最高位不相同,则用抗裁剪标记的第 i 个比特 $EmbedMark(i)$ 替换被选择像素点的最高位。即:

$$f'_i(x, y) = \begin{cases} f_i(x, y) + 128 & f_i(x, y) < 128 \text{ 且 } EmbedMark(i) = 1 \\ f_i(x, y) - 128 & f_i(x, y) \geq 128 \text{ 且 } EmbedMark(i) = 0 \end{cases}$$



嵌入过程

③误差修正:为减小因改变像素点的最高位而引起的图像降质,必须对修改后的像素值 $f_i(x, y)$ 做必要的修正。修正策略如下:

$$f'_i(x, y) = \begin{cases} 130, & f_i(x, y) < 128 \text{ 且 } EmbedMark(i) = 1 \\ 125, & f_i(x, y) \geq 128 \text{ 且 } EmbedMark(i) = 0 \end{cases}$$

这样就尽可能地保证了标记嵌入后的健壮性,且隐秘图像与载体图像的差异最小。

5) 用密钥 key_3 , 在载体图像的中心区域大小为 $(M - 2d) \times (N - 2d)$ 伪随机选择 L 个像素点作为秘密信息的嵌入像素点。

6) 将加密后的秘密信息比特流 Msg 分别与伪随机选择的像素点的最高位进行异或运算, 并将结果嵌入到像素点的最低有效位。



提取过程

- 1) 用与信息嵌入时相同的密钥 key_2 生成抗裁剪标记 $EmbedMark$ 。
- 2) 从隐秘图像 $S(M' \times N')$ 的左上角和右下角分别提取大小为 $d \times d$ 的抗裁剪标记 $ExtractMark1$ 和 $ExtractMark2$ 。
- 3) 分别计算提取的抗裁剪标记 $ExtractMark1$ 、 $ExtractMark2$ 与原始抗裁剪标记 $EmbedMark$ 在不同位置的相关值。当相关值大于给定的阈值 T 时,记下掩膜左上角点在模板矩阵上的坐标,如坐标为 (i,j) ,则隐秘图像上边缘被裁剪的行数为 $r1 = i - 1$,左边缘被裁剪的列数为 $c1 = j - 1$ 。计算 $ExtractMark2$ 与原始抗裁剪标记 $EmbedMark$ 的相关性与上述方法大致相同。



提取过程

4) 根据步骤3得到的 $r1, c1, r2, c2$,可得到秘密图像嵌入区域的大小 m, n 和起始点坐标 (x, y) 。

$$m = M' + r1 + r2 - 2d$$

$$n = N' + c1 + c2 - 2d$$

$$x = d - r1$$

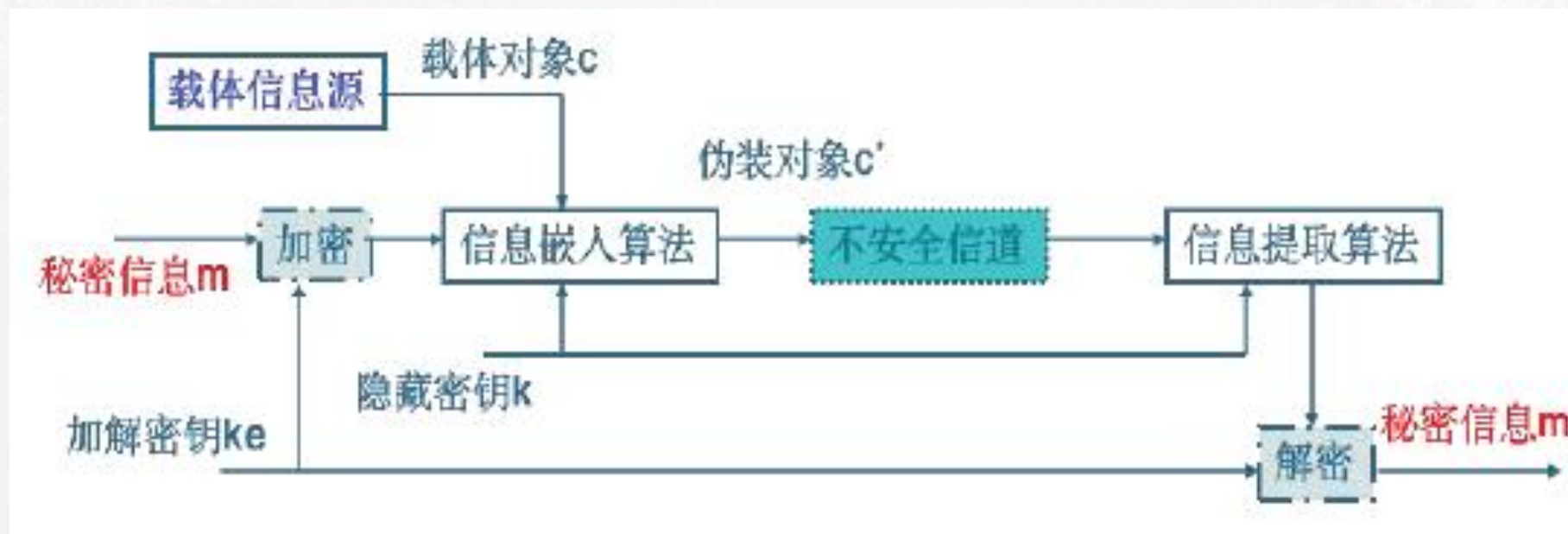
$$y = d - c1$$

5) 用与秘密图像嵌入时相同的密钥 $key3$ 在隐秘图像 S 上以 (x, y) 为左上角坐标,大小为 $m \times n$ 的区域伪随机选择 L 个像素点,并依次提取所选择点的最低位与最高位进行异或运算,即可提取出秘密信息比特流。即： $s_0 = x_7 \oplus s'$

6) 将提取的比特流利用密钥 $key1$ 进行解密即可获得原始的秘密信息。



整体框图





实验结果



上图中，左侧为隐藏前的载体图像，右侧为嵌入秘密信息后的载体图像，肉眼是无法区分他们之间的差别的。

```
Command Window

>> lsb_embed('1.bmp', '194603zmn')
>> Lsb_extract('result1.png')

ans =

194603zmn
```



算法健壮性比较

抗窥探比较

传统的LSB 算法直接在载体图像最低比特位上嵌入秘密信息,所以很容易遭到第三方的攻击和分析。

而本算法将需要进行隐藏的秘密信息比特流与选取的像素点最高比特位进行异或运算,最终将得到的结果嵌入在像素点的最低比特。这样可以有效抵抗攻击者根据提取出来的像素点最低比特集合进行统计分析,因为攻击者无法得知最低比特位代表的含义,直接对其进行处理是无法得出原始秘密信息的。

抗剪切比较

传统的LSB 算法在抵抗剪切等变化的攻击方面的健壮性是比较差的,可以说对隐藏后的载体图像稍作变化就有可能导致信息接收端无法恢复原始秘密信息。

本设计中采用的隐藏算法在抗剪切攻击方面的健壮性相较于传统算法而言是有很大的提升的。在嵌入过程中,将秘密信息隐藏在不容易被剪切的中心区域,且在载体图像的左上角与右下角设置了抗裁剪标记,在提取信息时可以根据这一裁剪标记做出相应的处理,从而保证信息的准确恢复。



算法健壮性比较



剪切后图像

```
Command Window
>> lsb_embed('1.bmp', '194603zmn')
>> Lsb_extract('result1.png')

ans =

194603zmn

>> Lsb_extract('result1_1.png')

ans =

194603zmn
```

本算法提取结果

```
Command Window
>> lsb_embed('1.bmp', '194603zmn')
>> Lsb_extract1('result1.png')

ans =

194603zmn

>> Lsb_extract1('result1_1.png')

ans =

0$
```

传统算法提取结果

可以看出传统算法并不能抵抗裁剪攻击，若隐藏有秘密信息的载体图像被攻击者破坏，则很难从中恢复出秘密信息，健壮性比较差，而本文采用的隐藏算法具有较好的健壮性，即使图像受到裁剪处理，仍然可以恢复出秘密信息。

04

变换域信息隐藏





基于DCT隐秘信息的嵌入算法研究

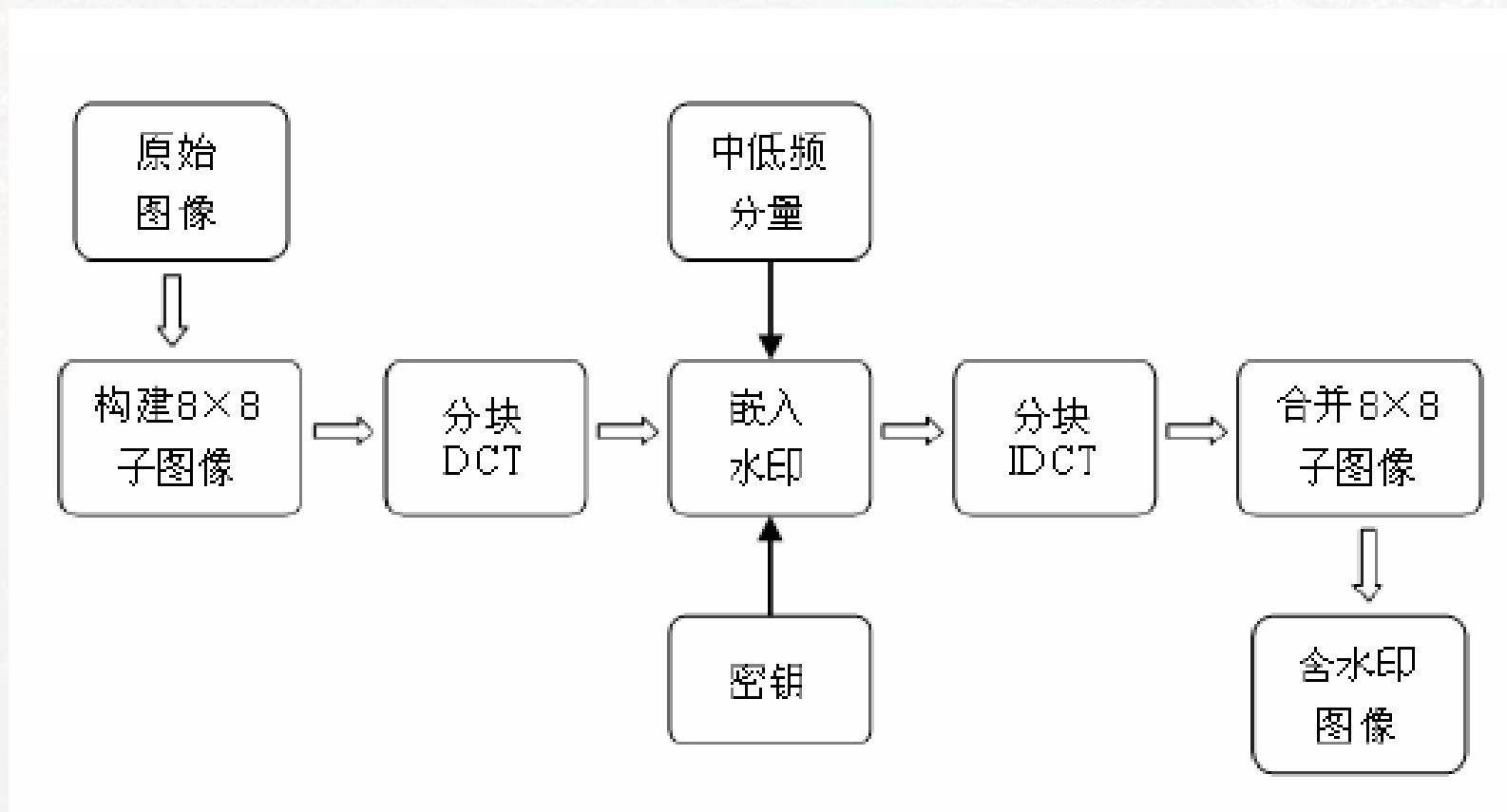
离散余弦变换，又称DCT变换，除了具有一般的正交变换性质外，它的变换阵的基向量能很好地描述人类语音信号和图像信号的相关特征。

通过变换将时域图像信号映射到空间频率域，使得图像在时域所表现出的能量发散形式变换为频域能量相对集中的形式，以便于对图像信息进行各种处理。

为了将隐藏的信息与载体图像的视觉重要部分绑定，一般都将隐藏信息嵌入在载体的中频部分，达到既不引起视觉变化，又不会被轻易破坏的目的。



基于DCT隐秘信息的嵌入算法研究





嵌入算法

- 图像分块：将载体图像分成 $8*8$ 块，并进行二位DCT变换。
- 秘密信息嵌入位置的选取：选择在固定位置的中频系数中叠加秘密信息。

$$x'(i, j) = x(i, j) + \alpha m_i$$

```
cda1(x+1,y+8)=cda0(x+1,y+8)+alpha*k(1);  
cda1(x+2,y+7)=cda0(x+2,y+7)+alpha*k(2);  
cda1(x+3,y+6)=cda0(x+3,y+6)+alpha*k(3);  
cda1(x+4,y+5)=cda0(x+4,y+5)+alpha*k(4);  
cda1(x+5,y+4)=cda0(x+5,y+4)+alpha*k(5);  
cda1(x+6,y+3)=cda0(x+6,y+3)+alpha*k(6);  
cda1(x+7,y+2)=cda0(x+7,y+2)+alpha*k(7);  
cda1(x+8,y+1)=cda0(x+8,y+1)+alpha*k(8);
```

- 嵌入图像容量的说明

载体图像 $N*N$ ，则嵌入水印图像最大为： $MAX = (N-8) / 8 + 1 \geq rm$ ，所能嵌入的最大水印图像大小为 $MAX*MAX$ 。 $(512-8) / 8 + 1 = 64$ 。



嵌入信息的提取

➤ 信息嵌入的逆过程:

对嵌入信息的图像同样进行二维**DCT**变换，比较每一块中约定位置的**DCT**系数数值，根据其相对大小，得到隐藏的比特串，从而恢复出嵌入图像。

➤ 比较不同嵌入深度，不同图像样本的情况下，该算法信息隐藏的效果。

鲁棒性测试





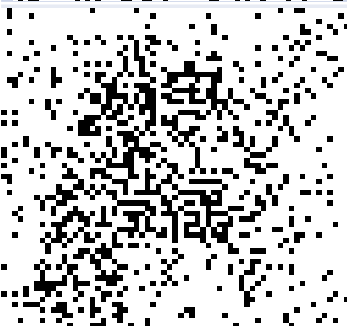


- 添加白噪声
- 高斯低通滤波
- **JPEG**压缩
- 图像剪切
- 旋转**10**度

图像质量评价

- 主观评价
- 客观评价
 - 水印不可见性评价：
峰值信噪比（**PSNR**）
 - 水印鲁棒性评价：
归一化相关系数（**NC**）






不同 α （嵌入深度）下水印嵌入/提取实验结果（lena图像）

载体图像	水印	α	嵌入水印后的图像	提取水印	PSNR	NC
	信息 隐藏	0.5			62.5208	0.6099
		0.7			57.4851	0.8084
		5			39.7979	0.9896


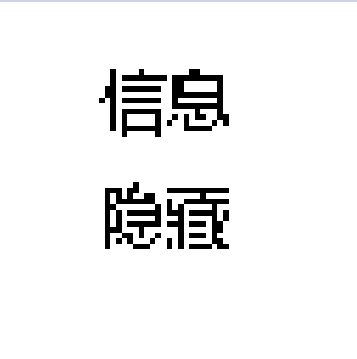

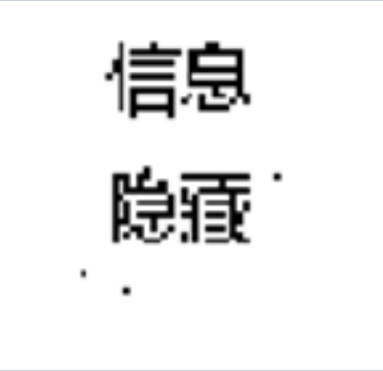




不同 α （嵌入深度）下水印嵌入/提取实验结果（lena图像）

载体图像	水印	α	嵌入水印后的图像	提取水印	PSNR	NC
	信息 隐藏	10		信息 隐藏	33.8743	0.9977
		15		信息 隐藏	32.9924	0.9995



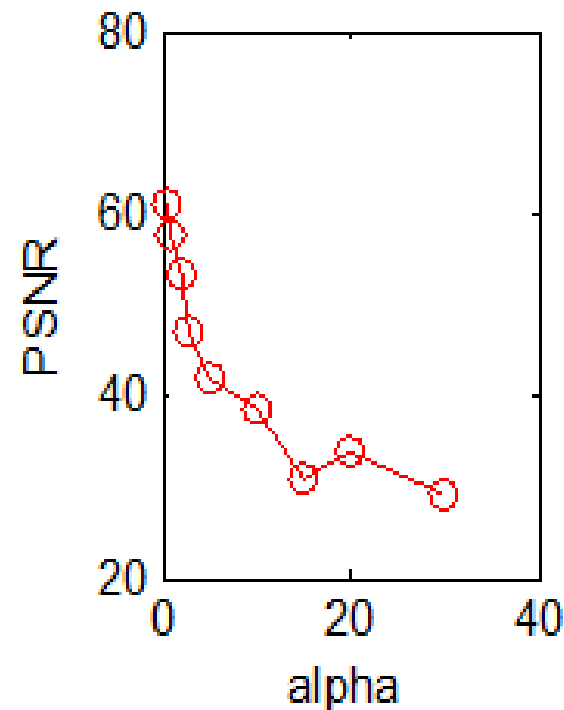
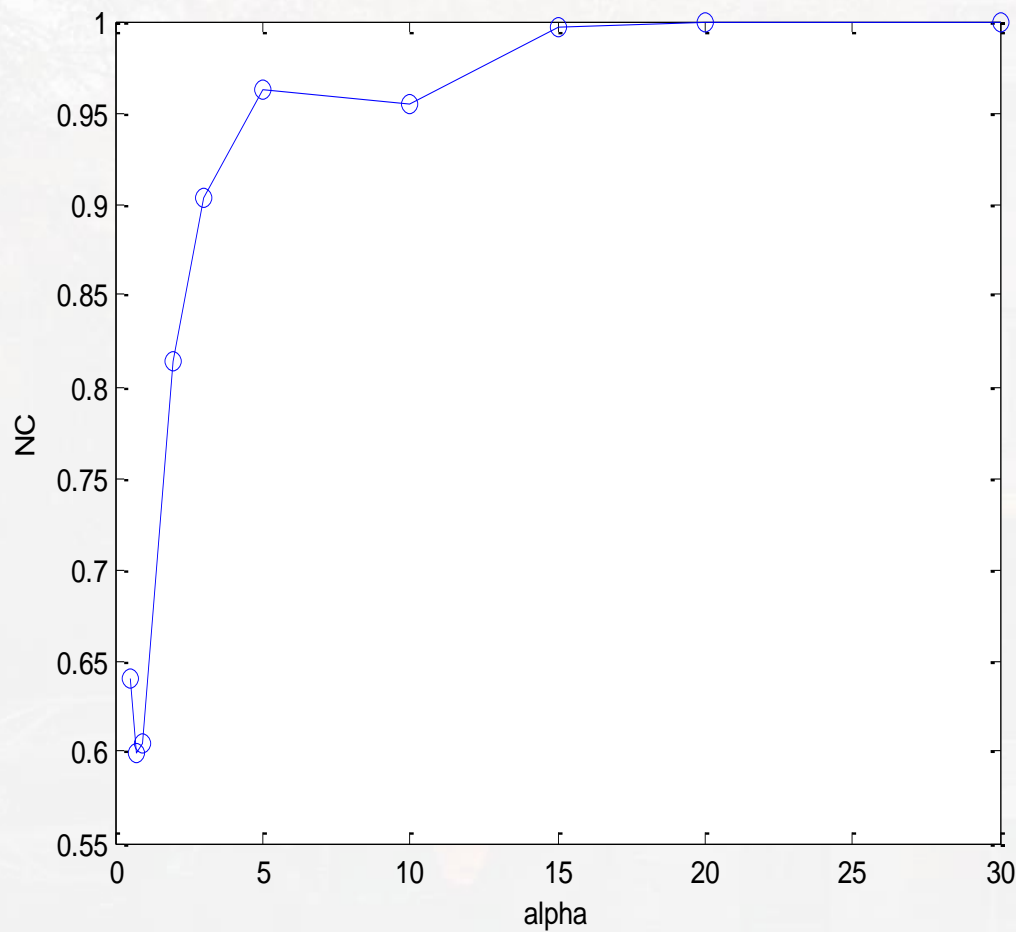
不同 α （嵌入深度）下水印嵌入/提取实验结果（lena图像）

载体图像	水印	α	嵌入水印后的图像	提取水印	PSNR	NC
		20			32.9853	0.9992
		30			25.2435	1 36





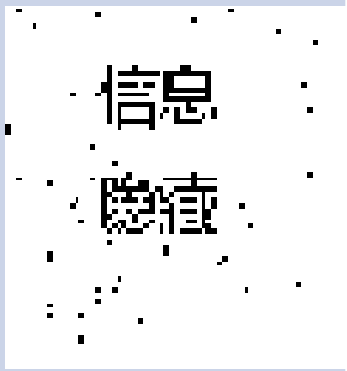



实验结果分析

可以看出，随着 α 的增大 PSNR减小，嵌入信息的不易察觉性越弱。因此选择合理的 α 值是关键。





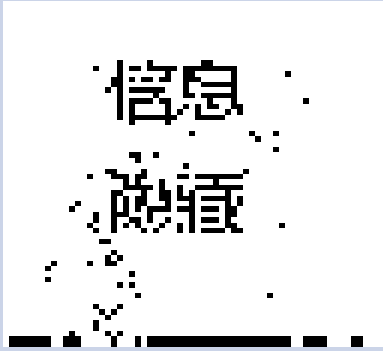


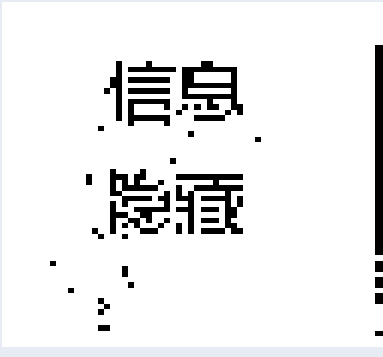


鲁棒性测试——不同攻击下的试验结果表（lena图像）

水印	攻击方式	α	嵌入水印后的图像	攻击后图像	提取水印	PSNR	NC
信息 隐藏	添加白 噪声	30				19.7089	0.9878
		20				21.0224	0.9842






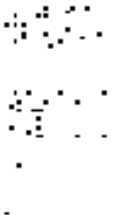


鲁棒性测试——不同攻击下的试验结果表（lena图像）

水印	攻击方式	α	嵌入水印后的图像	攻击后图像	提取水印	PSNR	NC
信息 隐藏	高斯低 通滤波	20				29.7000	0.9741
		45				27.2164	0.9824



鲁棒性测试——不同攻击下的试验结果表（lena图像）

水印	攻击方式	α	嵌入水印后图像	攻击后图像	提取水印	PNSR	NC
信息 隐藏	JPEG 压缩	45			信息 隐藏	4.6716	1
		30			信息 隐藏	4.6725	0.997
		20				4.6750	0.995



鲁棒性测试——不同攻击下的试验结果表（lena图像）

水印	攻击方式	α	嵌入水印后图像	攻击后图像	提取水印	PNSR	NC
信息 隐藏	旋转10度	45				4.6725	0.4773
	对旋转后的图像再 逆向旋转 10度					4.6712	0.9330

即在原始图像参与提取秘密信息的前提下，对旋转攻击具有较好的稳健性。

05

语音信号信息隐藏





基于回声的信息隐藏

人耳听觉的心理声学特性不仅具有频率掩蔽特性（同时掩蔽），还具有时间掩蔽特性。在很近的时间间隔内发出的两个声音会产生暂时掩蔽。一个信号可以被之前发出的噪声或信号掩蔽（前掩蔽），也可以被之后发出的噪声或信号掩蔽（后掩蔽）。当频率差别减小时，同时掩蔽增加；当时间差减小时，暂时掩蔽增加。

回声隐藏法利用了人类听觉系统的特性，在原始语音中加入不同延时的回声，从而将密文嵌入到明文中。



基于回声的信息隐藏

利用人耳听觉的前向和后向掩蔽特性，构造了一个前向后向回声核

$$h(n) = \delta(n) + \alpha\delta(n - \Delta) + \alpha\delta(n + \Delta)$$

$h(n)$: 回声内核

α : 能量衰减系数

Δ : 回声延迟



基于回声的信息隐藏

嵌入回声的声音如下：

$$y[n]=x[n]*h[n]$$

其中， $x[n]$ 为原始信号， $h[n]$ 为回声核单位脉冲响应。



实验过程

隐藏信息嵌入的步骤：

(1) 本实验取一段声音信号，先将其分成若干包含相同样点数的片段，每个片段时间约为几到几十个毫秒，样点数记为 N ，每段用来嵌入1比特隐藏信息。

(2) 选择 $d=d_0$ ，则在信号中嵌入隐藏信息比特0.选择 $d=d_1$ ，则嵌入信息比特1.

(3) 延时 d_0 和 d_1 是根据人耳听觉掩蔽效应为准则进行选取的。最后，将所有含有隐藏信息的声音信号段串联成连续信号。



实验过程

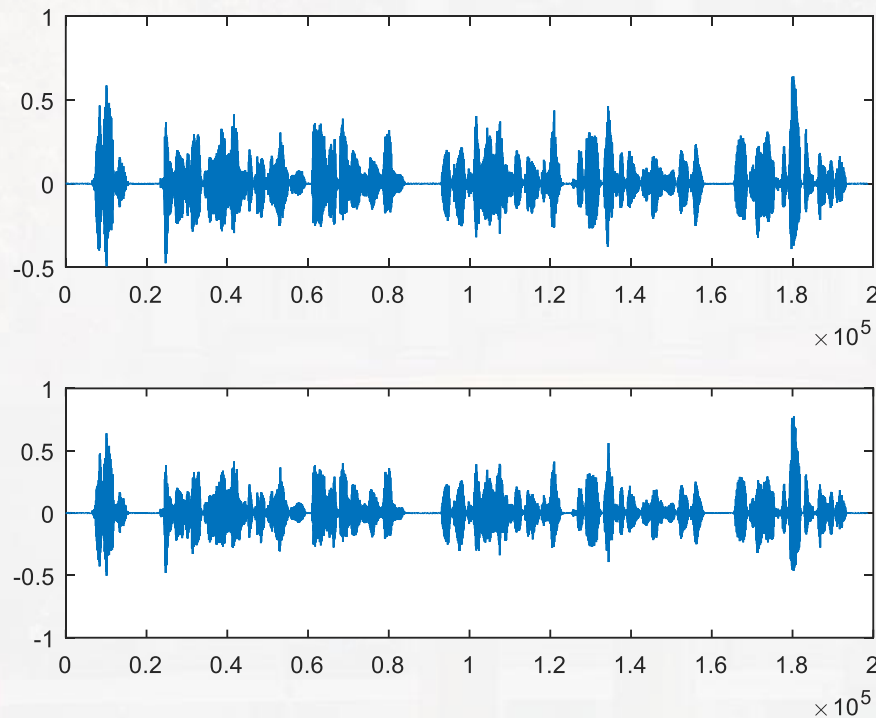
隐藏信息提取的步骤：

- (1) 将携带隐藏信息的语音信号分段，计算其倒谱。
- (2) 根据回声的延迟时间与嵌入信息之间的对应关系，即可判决出隐藏的二进制信息位。
- (3) 完成并-串转换和解密，即得原始隐藏信息。



结果分析

本实验嵌入的隐藏信息是一串随机序列，设定的两个延迟分别为100和200，设定参数为0.3，使用audioread函数将一段音频文件读入MATLAB，采样率为16khz.编程实现回声隐藏算法，得到波形如下：

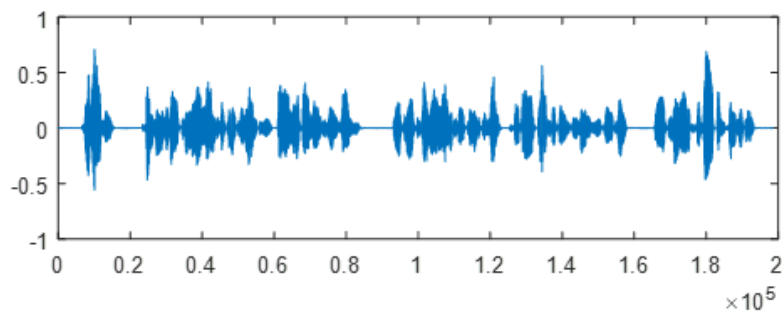
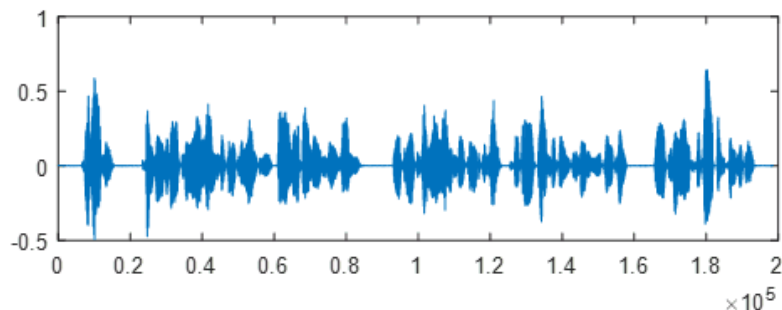




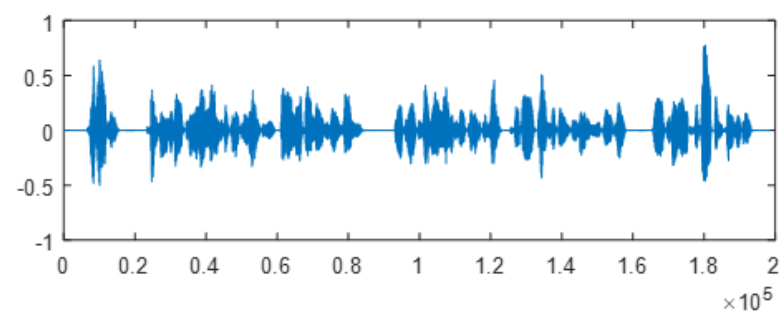
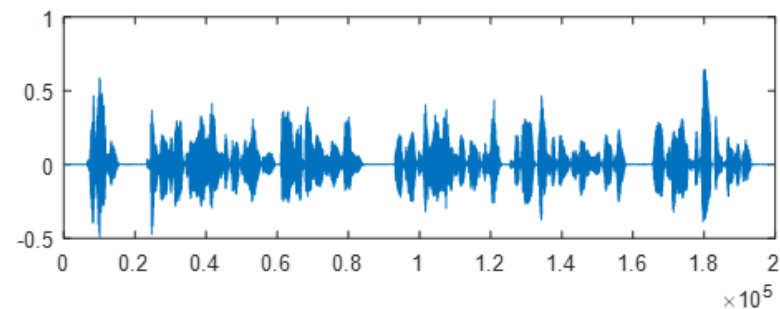
结果分析

·回声延时 d_0 和 d_1 对仿真效果的影响。

讨论同一衰减系数=0.3，同一分段方式同一噪声条件下，回声延时对实验效果的影响。



延时		错误率
d_0	d_1	
10	20	62/120
100	200	10/120
500	1000	18/120





结果分析

· 衰减系数对提取效果的影响。

选取同一段声音，延时采样点数为 $d_0 = 100$ ， $d_1 = 200$ ，改变衰减系数 λ 进行实验。如下表，可以发现当衰减系数取0.7,0.8时，提取效果较好，但是回声隐藏效果较衰减系数为0.3时有所降低。

λ	错误率
0.3	10/120
0.7	8/120
0.8	6/120



结果分析

·隐藏信息容量以及提取错误率比较。

每段长度\比较值	单向回声隐藏		前向后向回声隐藏	
	隐藏容量	错误率	隐藏容量	错误率
500	397	0.26	397	0.09
1000	198	0.15	198	0.05
1600	124	0.06	124	0.03
3000	66	0.03	66	0.01



结果分析

基于回声的信息隐藏

优点：鲁棒性很强，能较好的抵抗环境噪声，并且在一般的信息处理中如压缩等也不会影响隐秘信息的提取。

缺点：算法较复杂，倒谱计算量大，将信息隐藏在时延中，因此储存容量也较小。



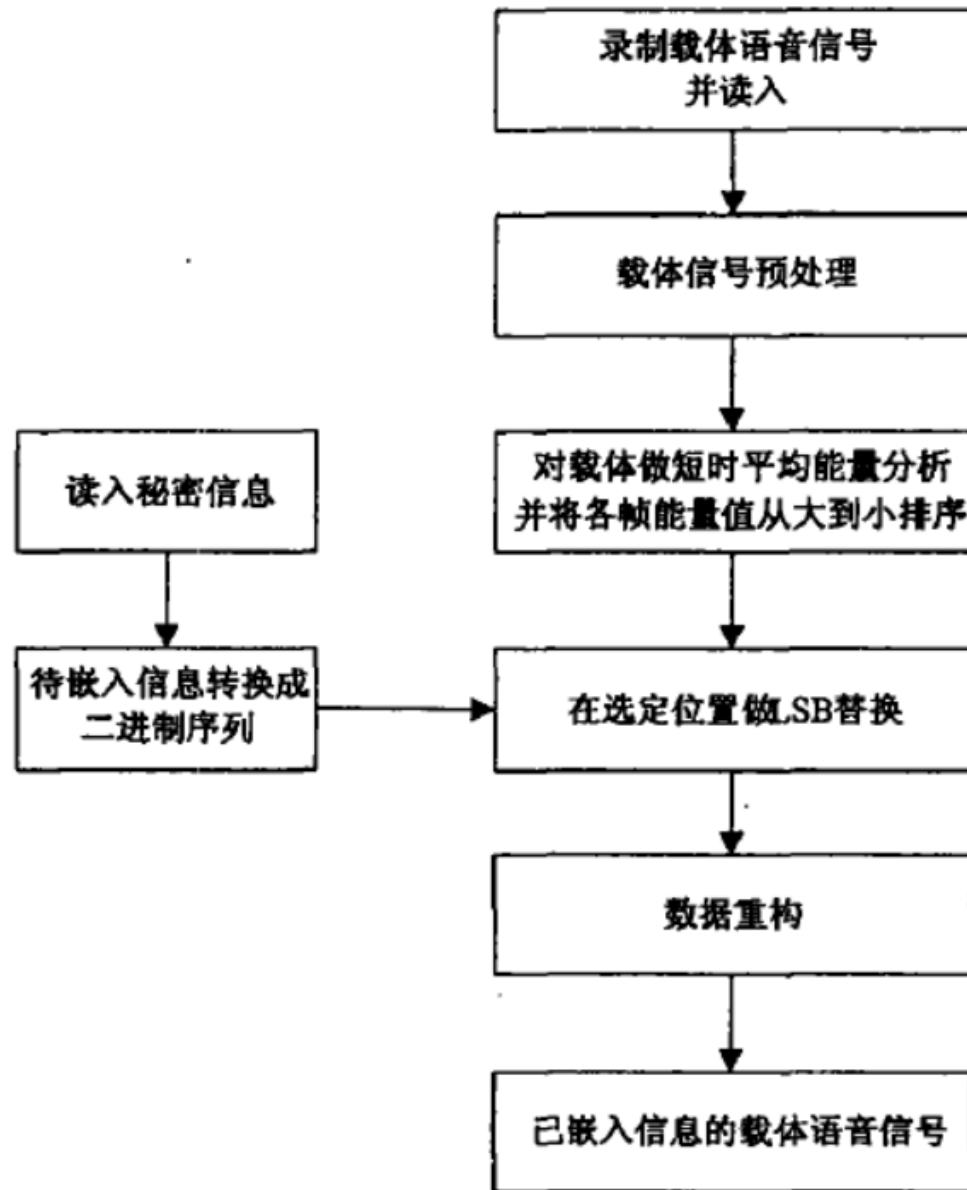
能量自适应LSB

为了改善LSB算法缺点我们将能量计算与抗统计检测的LSB算法相结合，在能量自适应LSB算法中加入了对语音信号的短时能量分析并根据语音信号的能量分布情况选择水印信息嵌入的位置，选择的原理是语音信号帧的能量从大到小的排序，嵌入时从能量最大帧开始嵌入，直到秘密信息全部嵌入到载体信号中为止。



能量自适应LSB

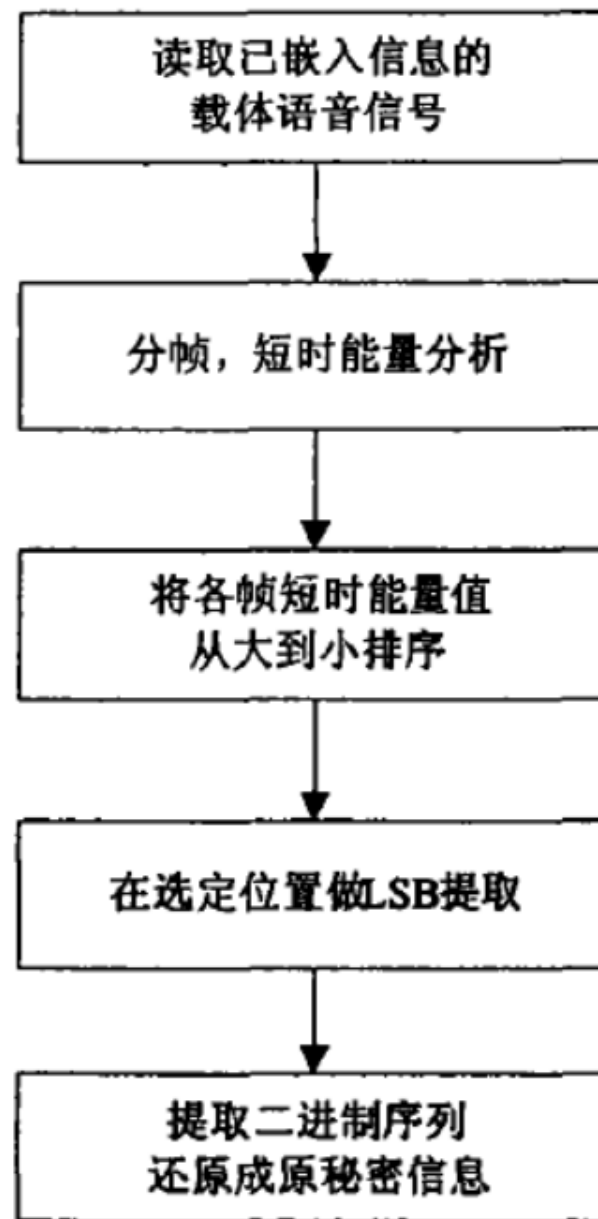
- 1. 录制载体语音信号，读入已经录好的载体语音信号。
- 2. 对载体语音信号进行预处理，包括采样、分帧、短时能量分析等。
- 3. 对载体语音信号各帧进行短时平均能量分析，并从大到小的排序以备嵌入。
- 4. 读入密写信号((2值图像，语音信号等)，并将其转换为二进制序列。
- 5. 在已经选出的符合嵌入条件的位置，参照LSB方法，用密写信号的二进制序列替换掉载体信号数据的最低有效位。
- 6. 将经过替换的数据重构成原来的语音载体信号。





能量自适应LSB

- 1.和嵌入算法一样，先对含有秘密信息的语音数据进行同样的短时能量分析，并把平均能量大的分帧选出来，作为提取位置。
- 2.在提取位置进行数据提取
- 3.如此循环2中的操作，直至所有的秘密信息数据位被提取出来。
- 4.根据嵌入时的转换方法，将提取出来的秘密信息数据还原为嵌入的秘密信息。





实验分析

实验中采用的音频文件大小为464k，在其中隐藏大约12800bit数据后其误码率为0，当嵌入信息13000bit的数据时，误码率为0.0078，也就是刚刚出现提取误差，该改进的基于能量的LSB隐藏方法明显增大了嵌入信息量，降低了误码率。

嵌入量 (bit)	误码率
8000	0
10240	0
12800	0
13000	0.0078
14000	0.0401
18000	0.1443
20480	0.1842
51200	0.3758
102400	0.4352



谢谢观看！