

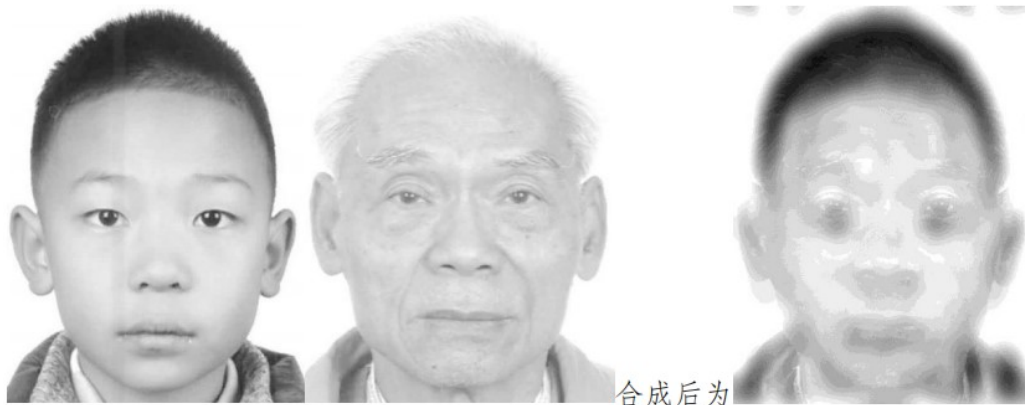
实验摘要：

1. 用所给代码实现不同图片低通和高通的叠加，并理解代码的运行方式和原理。
2. 基于第一题代码的基础上实现图片不同频段的分离，并运用高通和低通滤波器的的工作原理。
3. 将采集到的声音频率经过若干次采样得到不同采样频率下的离散信号，对该信号及其取样信号，分析取样率对信号重构的影响。
4. 实现频域制作数字盲水印和去除数字盲水印。

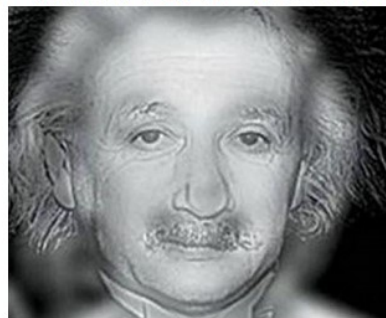
实验题目

1. 图片的高频信息与低频信息

合成图片。找两张轮廓比较像的图片A和B，有一张是你本人。提取一张照片的低频信息，另一张图片的高频信息，结合这两个照片。设置不同的频率门限，组合照片，组合的效果是，放大看是A，缩小看是B。例如以下两张图片



分解图片。把下图爱因斯坦和玛丽莲梦露分开（此图缩小是玛丽莲梦露，截取不同的频段）



部分代码见“图片的合成.m”链接：<http://pan.baidu.com/s/1dFtd0qP> 密码：1oa4

2. 采集一段人说话时的声音(一般最高频率在4kHz左右)，并进一步经过若干次取样，得到对同一段连续信号在不同取样频率下的离散信号，例如最初的取样率是44kHz，经过下取样后可以得到22kHz、11kHz、5.5kHz、2.75kHz等频率下的取样结果。试针对该信号及其取样信号，分析取样率对信号重构的影响。
3. **频域制作数字盲水印和去除数字盲水印** <https://www.zhihu.com/question/50735753>, 看懂, 想想, 有想法写出来, 做一个好玩的东西。

实验内容

一 实验基本原理及步骤

1.(1)

```
I1=imread('11.jpg');%梦露
g1=rgb2gray(I1);
s=fftshift(fft2(g1));
[M,N]=size(s);
n1=fix(M/2);
n2=fix(N/2);

%理想低通滤波器取 d0=10 （15,30）可变
d0=10;
for i=1:M
    for j=1:N
        d=sqrt((i-n1)^2+(j-n2)^2);
        if d<d0
            h=1;
        else
            h=0;
        end
        s(i,j)=h*s(i,j);
    end
end
s=ifftshift(s);
s=uint8(real(ifft2(s)));
figure(1);
imshow(s);
```

%爱因斯坦图片处理

```
I2=imread('22.jpg');%爱因斯坦图片
g2=rgb2gray(I2);
s2=fftshift(fft2(g2));
[M2,N2]=size(s2);
n12=fix(M2/2);
n22=fix(N2/2);

%理想高通滤波器取 d02=5 （15,30）可变
d02=10;
for i=1:M2
    for j=1:N2
        d=sqrt((i-n12)^2+(j-n22)^2);
        if d<d02
            h=0;
        else
            h=1;
        end
        s2(i,j)=h*s2(i,j);
    end
end
s2=ifftshift(s2);
s2=uint8(real(ifft2(s2)));
figure(2);
imshow(s2);
```

%图片合并

```
s3=imadd(s, s2);  
figure(3);  
imshow(s3);  
imwrite(s3, 'he. jpg', 'JPG')
```

(2)

I1=imread('11. jpg');%合成后的照片

```
g1=rgb2gray(I1);  
s=fftshift(fft2(g1));  
[M,N]=size(s);  
n1=fix(M/2);  
n2=fix(N/2);
```

%理想低通滤波器取 d0=10 (15, 30) 可变

```
d0=10;  
for i=1:M  
    for j=1:N  
        d=sqrt((i-n1)^2+(j-n2)^2);  
        if d<d0  
            h=1;  
        else  
            h=0;  
        end  
        s(i, j)=h*s(i, j);  
    end  
end  
s=ifftshift(s);  
s=uint8(real(ifft2(s)));  
figure(1);  
imshow(s);
```

%再将该图片通过理想高通滤波器并输出分离出的图片

I2=imread('3. jpg');%合成后的图片

```
g2=rgb2gray(I2);  
s2=fftshift(fft2(g2));  
[M2,N2]=size(s2);  
n12=fix(M2/2);  
n22=fix(N2/2);
```

%理想高通滤波器取 d02=5 (15, 30) 可变

```
d02=10;  
for i=1:M2  
    for j=1:N2  
        d=sqrt((i-n12)^2+(j-n22)^2);  
        if d<d02  
            h=0;  
        else  
            h=1;  
        end  
        s2(i, j)=h*s2(i, j);  
    end  
end
```

```
end
s2=ifftshift(s2);
s2=uint8(real(ifft2(s2)));
figure(2);
imshow(s2);
2.
clear;clc
[x,fs]=audioread("G:\FFOutput\temple.wav");%并返回样本数据 y 以及该数据的采样率 fs
% sound(x,fs);%播放语言信号
% n=length(y);%信号长度
% t=(0:n-1)/fs;%时域范围
figure(1)
subplot(211);
plot(x)
xlabel('采样点');
ylabel('幅度');
title('初始信号 48kHz 波形'); %绘出时域波
grid on
y1=fft(x); %快速傅里叶变换 做 length(y) 个点的
w1=fftshift(y1); %频率分量移到坐标中心
subplot(212) %绘出频域频谱
plot(abs(w1))
title('初始信号 48kHz 频谱');
xlabel('频率');
ylabel('幅度');
grid on
% df=fs/length(pyr1);%计算谱线间隔
% f=0:df:(fs/2-df); %频谱范围, 截取前半段 (抽样频率高于最大频率的 2 倍)
% PyrF=abs(pyr1);%幅度响应
% PyrF=PyrF(1:length(PyrF)/2);
% plot(f,PyrF);

figure(2)
y2=resample(y1,1,2);%24kHz 频域下的重采样
x2=real(ifft(y2)); %傅里叶反变化求出, 并取实部, 最大幅度保存信息
%sound(x2,fs/2);
subplot(211);
plot(x2)
xlabel('采样点');
ylabel('幅度');
title('24kHz 信号波形'); %绘出时域波
grid on
subplot(212) %绘出频域频谱
plot(abs(fftshift(y2)))
title('24kHz 信号频谱');
xlabel('频率');
ylabel('幅度');
grid on

figure(3)
y3=resample(y1,1,4);%12kHz 频域下的重采样
x3=real(ifft(y3)); %傅里叶反变化求出, 并取实部, 最大幅度保存信息
%sound(x3,fs/4);
subplot(211);
```

```
plot(x3)
xlabel('采样点');
ylabel('幅度');
title('12kHz 信号波形'); %绘出时域波
grid on
subplot(212) %绘出频域频谱
plot(abs(fftshift(y3)))
title('12kHz 信号频谱');
xlabel('频率');
ylabel('幅度');
grid on
figure(4)
y4=resample(y1,1,8);%6kHz 频域下的重采样
x4=real(iff(y4)); %傅里叶反变化求出,并取实部,最大幅度保存信息
%sound(x4,fs/8);
subplot(211);
plot(x4)
xlabel('采样点');
ylabel('幅度');
title('6kHz 信号波形'); %绘出时域波
grid on
subplot(212) %绘出频域频谱
plot(abs(fftshift(y4)))
title('6kHz 信号频谱');
xlabel('频率');
ylabel('幅度');
grid on

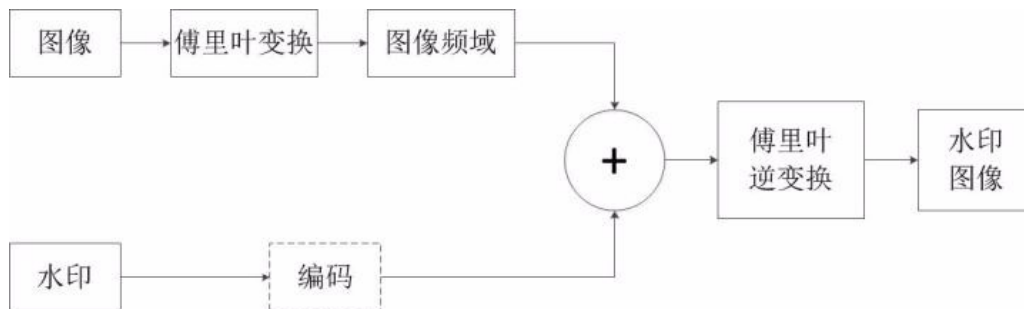
figure(5)
y5=resample(y1,1,16);%6kHz 频域下的重采样
x5=real(iff(y5)); %傅里叶反变化求出,并取实部,最大幅度保存信息
%sound(x5,fs/16);
subplot(211);
plot(x5)
xlabel('采样点');
ylabel('幅度');
title('3kHz 信号波形'); %绘出时域波
grid on
subplot(212) %绘出频域频谱
plot(abs(fftshift(y5)))
title('3kHz 信号频谱');
xlabel('频率');
ylabel('幅度');
grid on

%%信号重构后的误差
figure(6)
subplot(2,1,1)
plot(t,x);
hold on
plot(t1,x2_1);
i = 1;
j = 1;
error = zeros(length(x2_1),1);
error = error(1,:);
```

%计算误差

```
while i <= length(x2_1)
    if abs(t1(1,i)-t(1,j))<0.0001
        e = abs(x2_1(i,1)-x1(j,1));
        error(1,i) = e';
        i=i+1
    end
    j=j+1;
end
error = abs(error(1,:));
subplot(2,1,2)
t3 = linspace(0,T,length(error));
plot(t3,error);
```

3.实验原理



实验代码:

```
clc;clear;close all;
alpha = 1;

%% read data
im = double(imread('4.jpg'))/255;
mark = double(imread('5.png'))/255;
imsize = size(im);
%random
TH=zeros(imsize(1)*0.5,imsize(2),imsize(3));
TH1 = TH;
TH1(1:size(mark,1),1:size(mark,2),:) = mark;
M=randperm(0.5*imsize(1));
N=randperm(imsize(2));
save('encode.mat','M','N');
for i=1:imsize(1)*0.5
    for j=1:imsize(2)
        TH(i,j,:)=TH1(M(i),N(j),:);
    end
end

%% encode mark
imsize = size(im);
%random
TH=zeros(imsize(1)*0.5,imsize(2),imsize(3));
TH1 = TH;
TH1(1:size(mark,1),1:size(mark,2),:) = mark;
M=randperm(0.5*imsize(1));
N=randperm(imsize(2));
save('encode.mat','M','N');
for i=1:imsize(1)*0.5
    for j=1:imsize(2)
```

```
TH(i,j,:)=TH1(M(i),N(j),:);
end
end
% symmetric
mark_ = zeros(imsz(1),imsz(2),imsz(3));
mark_(1:imsz(1)*0.5,1:imsz(2),:)=TH;
for i=1:imsz(1)*0.5
for j=1:imsz(2)
mark_(imsz(1)+1-i,imsz(2)+1-j,:)=TH(i,j,:);
end
end
%figure,imshow(mark_),title('encoded watermark');
%imwrite(mark_,'encoded watermark.jpg');

%% add watermark
FA=fft2(im);
%figure,imshow(FA);title('spectrum of original image');
FB=FA+alpha*double(mark_);
%figure,imshow(FB); title('spectrum of watermarked image');
FAO=ifft2(FB);
figure,imshow(FAO); title('watermarked image');
imwrite(uint8(round(FAO)),'watermarked image.jpg');
RI = FAO-double(im);
figure,imshow(uint8(RI)); title('residual');
imwrite(uint8(round(RI)),'residual.jpg');
x1 = 1:imsz(2);
y1 = 1:imsz(1);
[xx,yy] = meshgrid(x1,y1);
```

二 实验结果

1.1 第一题选了两个二刺螈照片进行合成：



合成后的效果如下：



将两张图通过的滤波器反过来后效果如下：

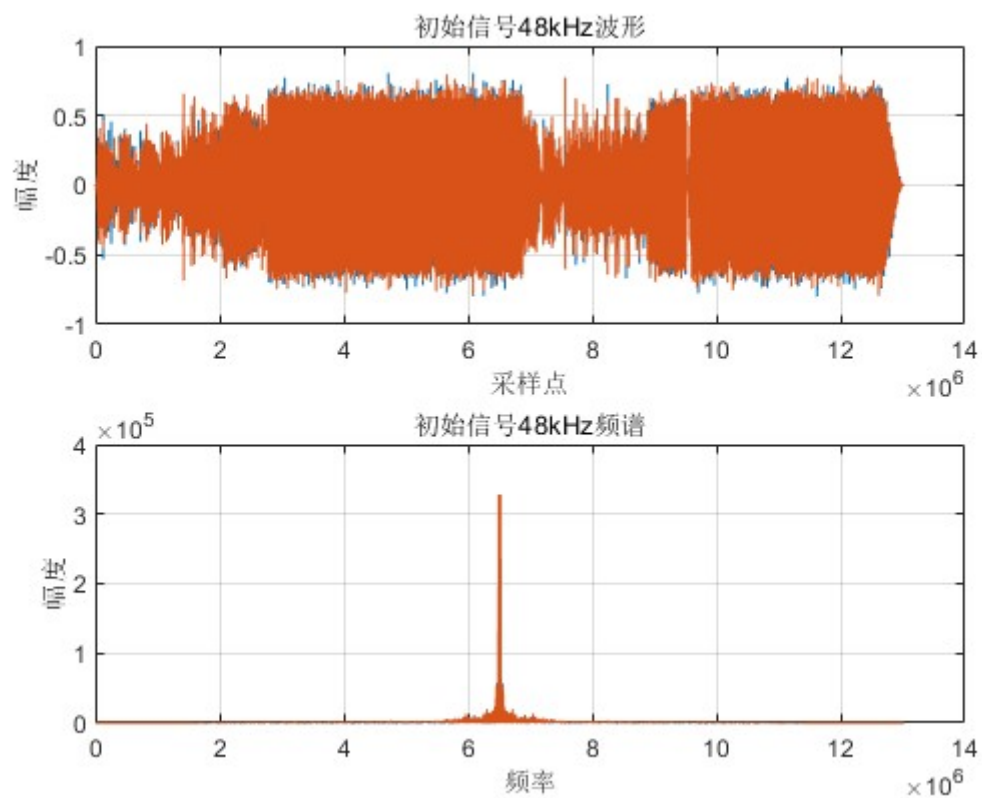


1.2 由第一题可以看出，这张梦露-爱因斯坦图是梦露头像的低频加上爱因斯坦头像的高频，所以在第一题的基础上修改代码即可分离出不同频率的他们的头像；分离后的图像分别为：

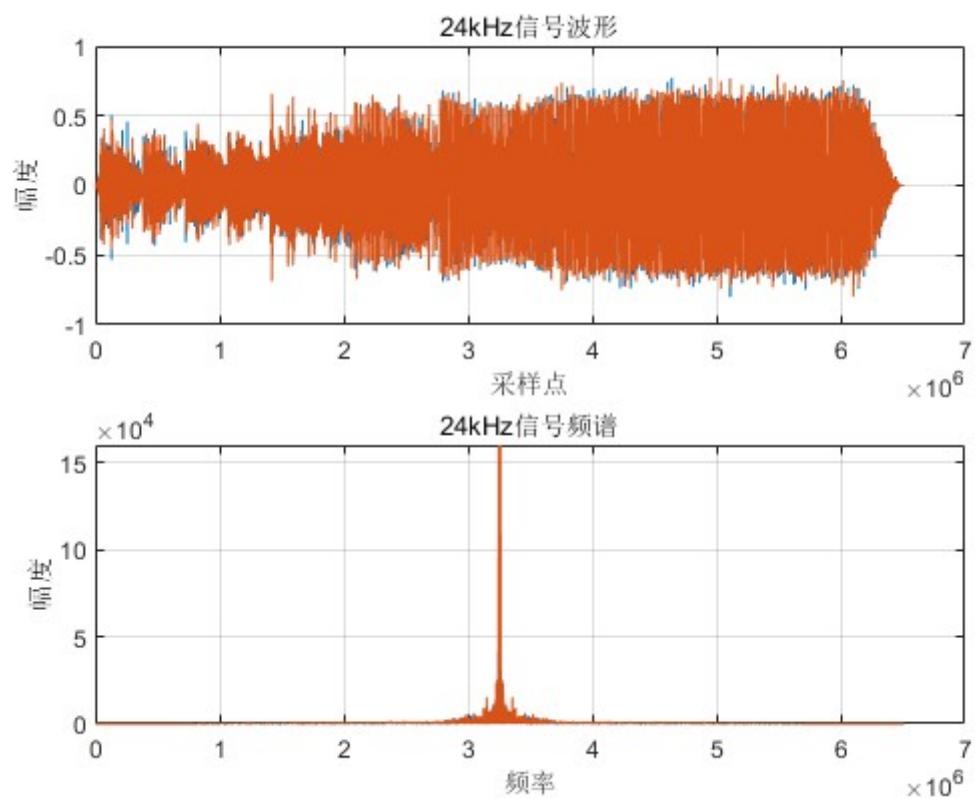


2.

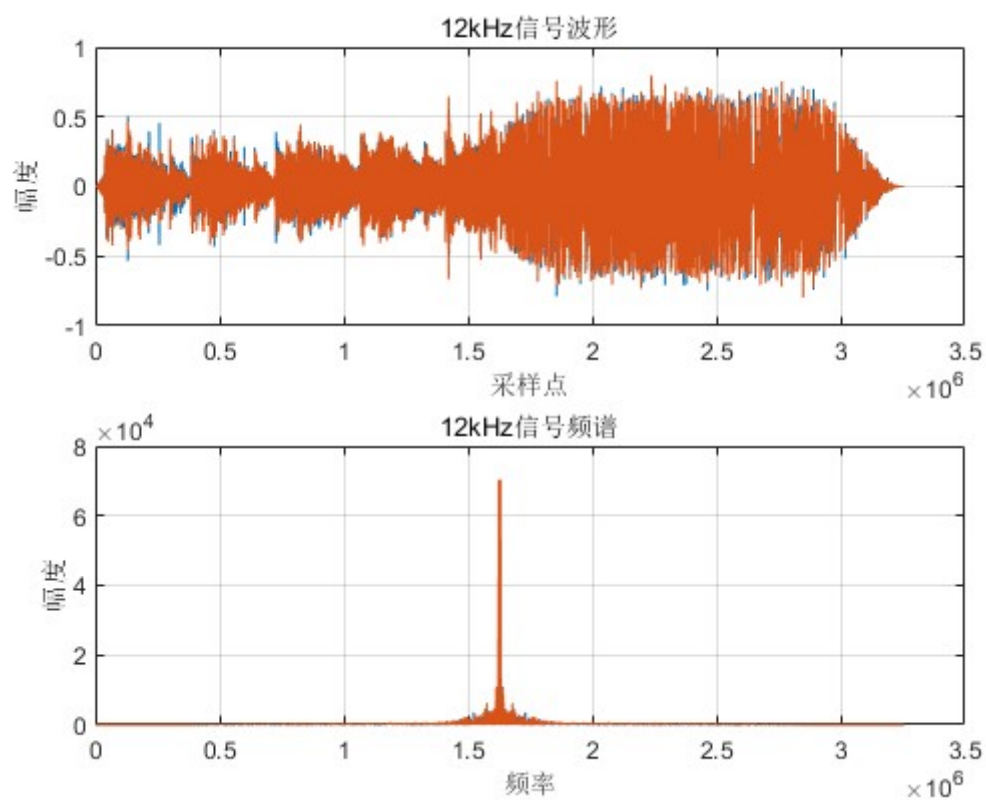
原始信号 48k



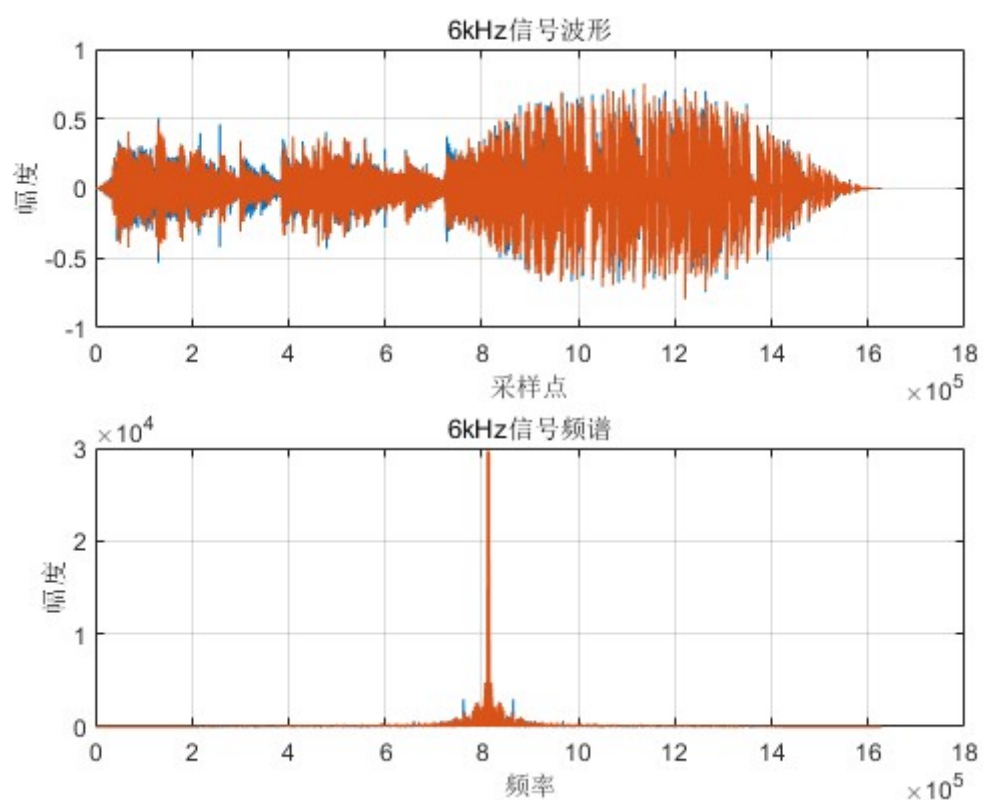
取样信号 24k:



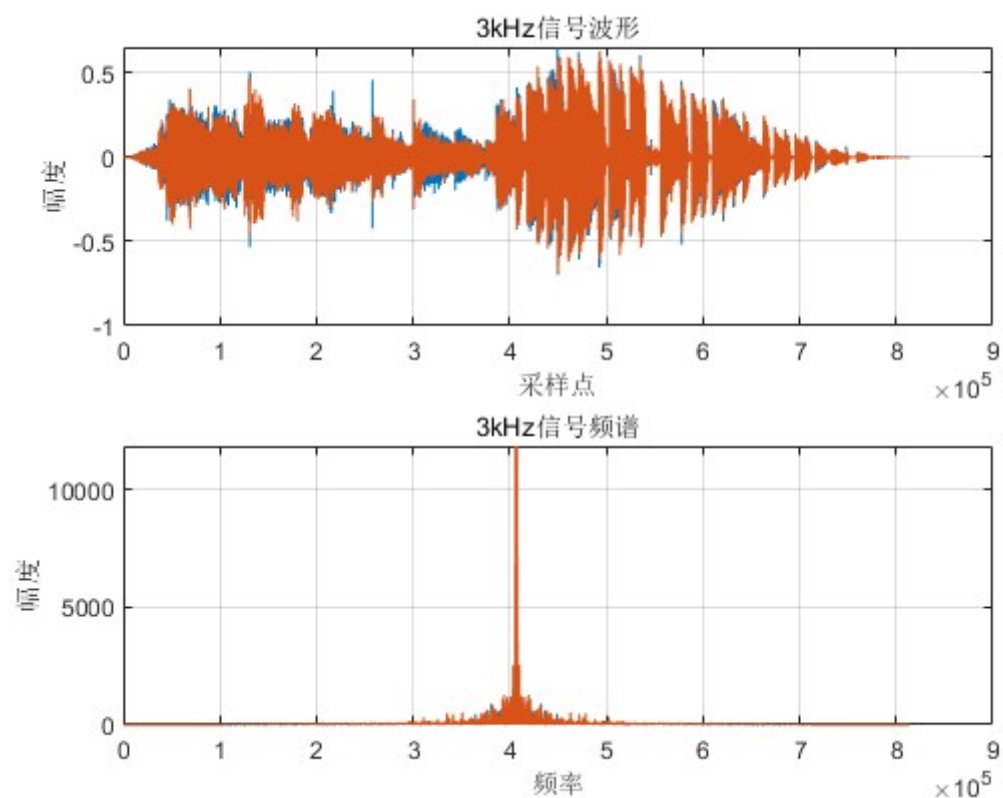
取样信号 12k:



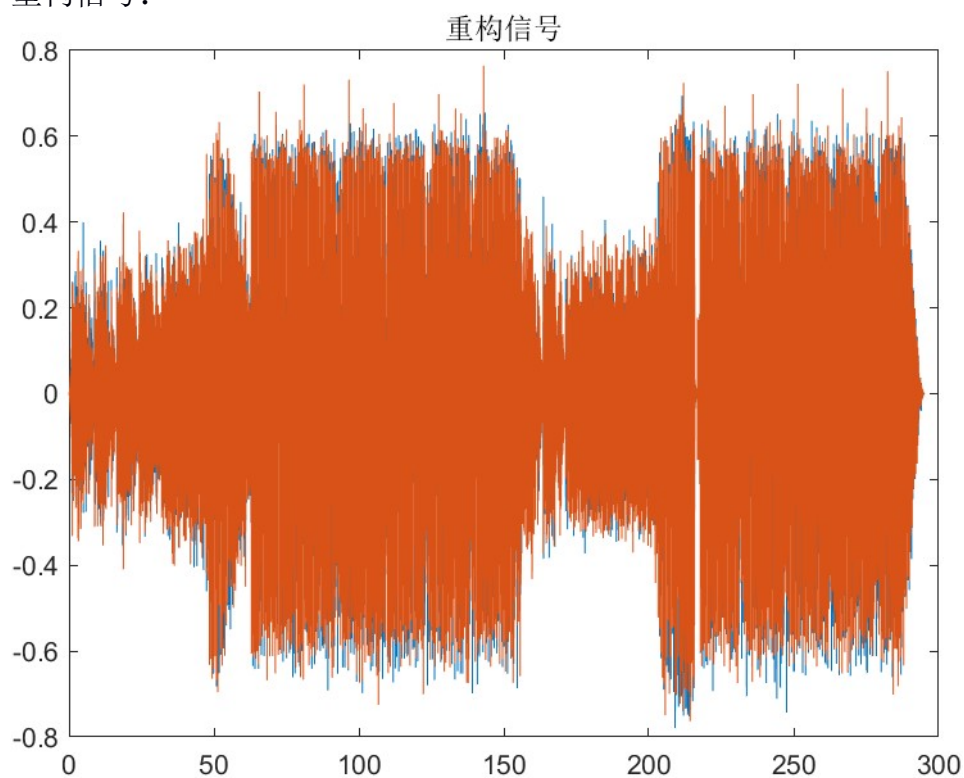
取样信号 6k:



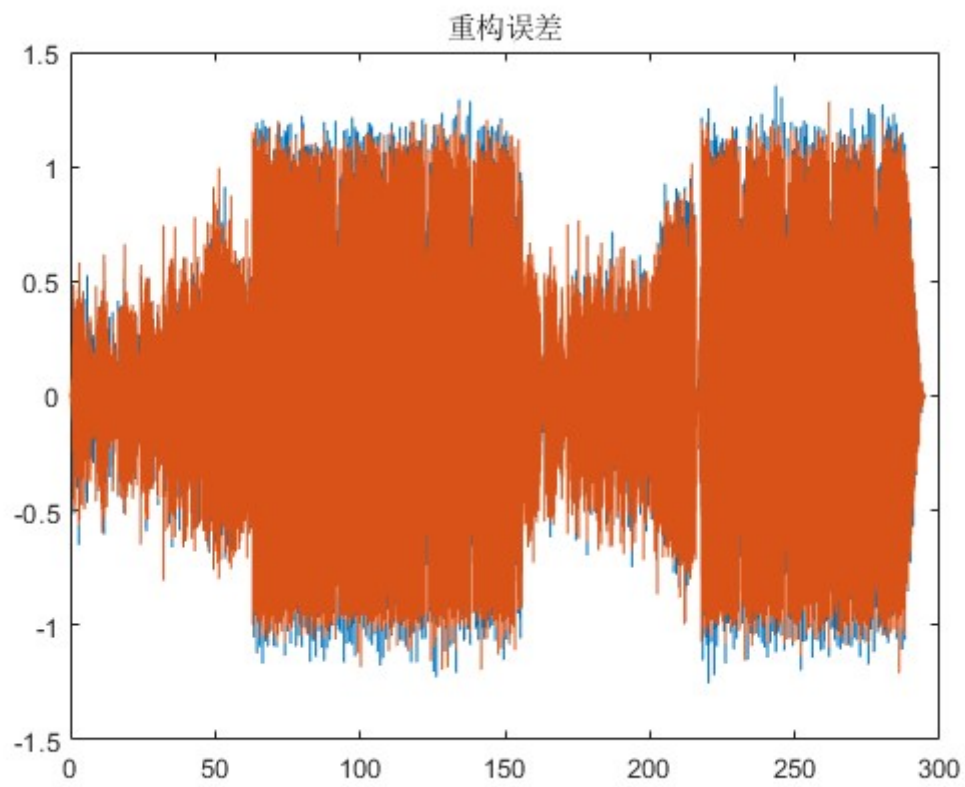
取样信号 3k:



重构信号:



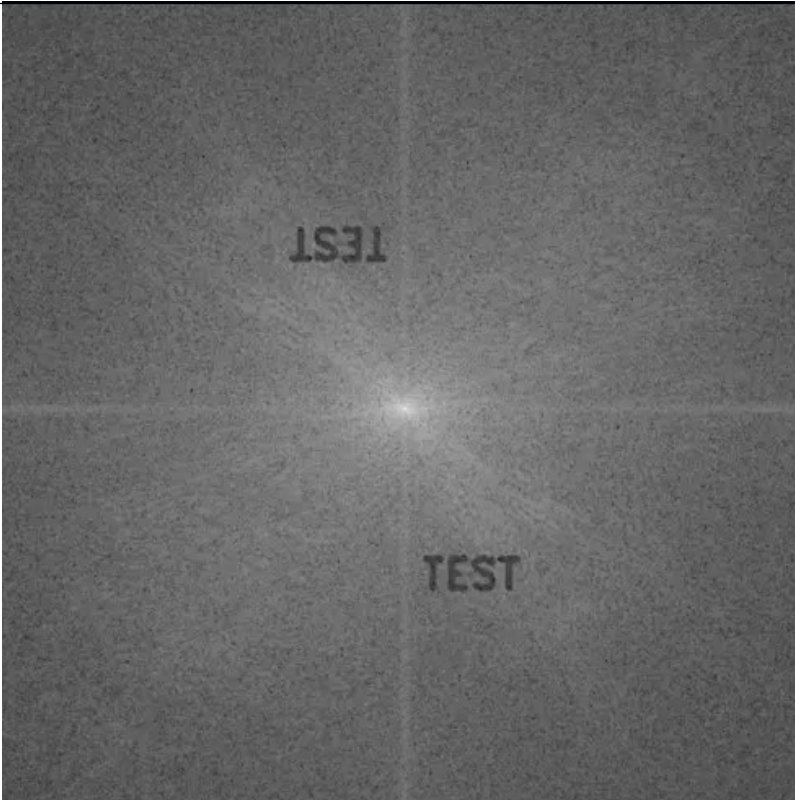
重构误差:



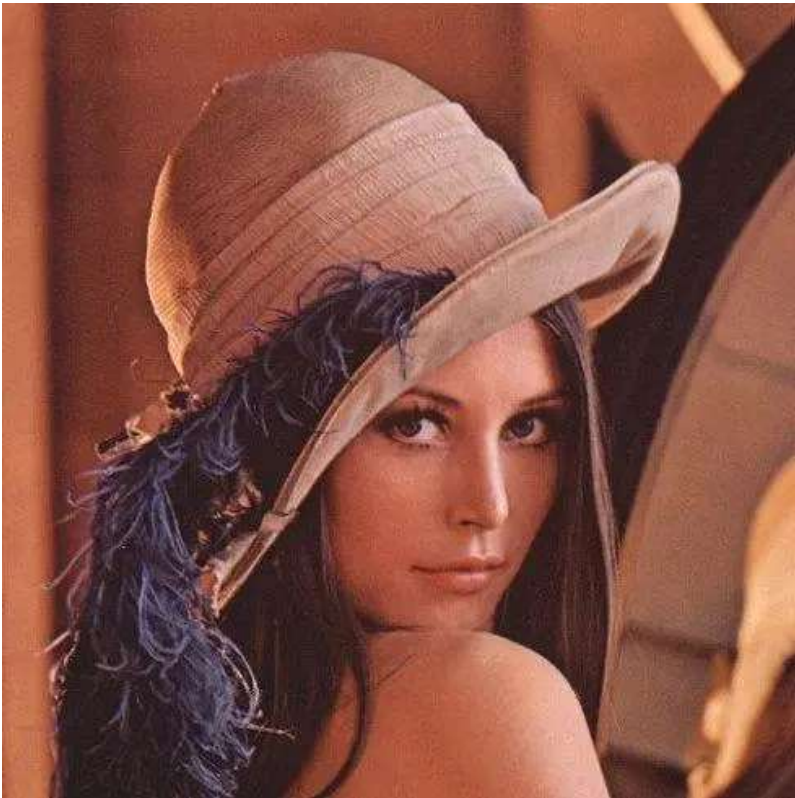
3.
原图:



水印:



加水印后：



三 实验结果的分析

1. 合成图像通常会展示出两张原始图像的某些特征。高频信息与低频信息的结合可能会导致图像在细节上更加清晰或模糊，整体上更加平滑或保留了较多的细节。

如果将一张图像的高频信息与另一张图像的低频信息进行合成，合成后的图像可能会呈现出

两个图像的混合效果，同时保留了一些纹理和细节，但整体上可能比原始图像模糊。观察合成前后频域图像的特征，可以发现高频信息通常包含图像的细节和纹理，低频信息则包含图像的整体结构和低频成分。

2. 对一段人说话时的声音，通过不同的取样频率对该连续信号进行了离散取样。

当初始取样率为 48kHz 时：根据奈奎斯特定理，这个取样率足够高，可以准确地重构原始信号。因此，我们可以期望在这个取样率下获得高质量的声音重构结果。

下取样到 24kHz、12kHz、6kHz、3kHz：通过下取样操作，我们降低了采样频率。对于每个取样频率，我们需要考虑是否满足奈奎斯特定理。

下取样到 24kHz、12kHz 时仍能够准确地重构声音信号，保留较高的质量。下取样到 6kHz、3kHz：这两个取样率都低于最高频率的 2 倍，违反了奈奎斯特定理。在重构过程中，会出现混叠效应，导致高频部分以错误的方式表示为低频成分，从而引起重构信号的失真。因此，在这两个频率下，声音重构结果会丢失较多的高频细节，信号质量会进一步降低。

实验结果表明，当取样频率满足奈奎斯特定理时，我们可以获得准确的声音重构结果，保留较高的质量。随着取样频率的降低，声音重构结果可能会出现失真和细节丢失。

3. 在频域叠加水印很难被察觉，这是因为图像的能量主要集中在低频部分，高频信息是图像的细 节，盲水印叠加在图像的高频区域所有不易被察觉。观察嵌入水印后的图像，分析嵌入水印后图像的质量变化。水印嵌入过程导致图像的失真和质量下降。

实验总结

经过此次 MATLAB 课程设计我学到了很多知识和学习方法。在这次设计中，我更深一步的了解了高/低通滤波器的运行方法和工作原理，掌握如何提取图片的高频信息与低频信息；学到了对信号的采样定理的应用，以及信号的重构，从信号的采样程度进行比较其误差，了解不同采样程度的重构信号和原信号所产生的差异；理解了采样信号恢复原始信号全部信息的原理，实现了奈奎斯特采样定律下信号重构方法。此外还实现了数字盲水印的制作，对频域叠加水印有了深刻的了解。

参考文献

1. 图片加数字盲水印

<https://www.cnblogs.com/jins-note/p/9672893.html>

2. 简介频域制作盲水印技术

<https://www.jianshu.com/p/72cb2e8c6b28>