

实验六：扩频水印

综合评分：

【实验目的】:

了解扩频通信的概念，掌握扩频水印的原理，实现基于 DWT 的扩频水印算法。

【实验内容】:（请将你实验完成的项目涂“■”）

■一、实现基于 DWT 的扩频水印嵌入和提取算法

【实验工具及平台】:

■ Windows+Matlab

□ 其它：（请注明）_____

【实验涉及到的相关算法】:

- 1、与实验内容选择的项目对应；
- 2、请使用流程图、伪代码、NS 图或文字方式描述，不要完全贴代码

■一、实现基于 DWT 的扩频水印嵌入和提取算法**水印生成和嵌入:**

总体流程步骤如下:

1. 对载体图像 C 进行 DWT 变换并选择低频系数矩阵
2. 对矩阵做 SVD 分解得到 U, S, V
3. 对水印图像 W 做直接扩频得到 W' (Gold 码)
4. 对 W' 做 DWT+SVD 得到 U', S', V'
5. 用 U', V' 直接替换 U, V
6. 重构载密图像

代码如下:

```
function [watermarkimagergb, watermarkimage, waterCA, watermark, S_sigma, S_C, S_S] =  
    wavemarksvd(input, goal, seed, wavelet, level, alpha)  
data=imread(input);  
data=double(data)/255;  
datared=data(:, :, 1);  
%对原始图像进行的 svd 操作  
[row, list]=size(datared);  
[C, S]=wavedec2(datared, level, wavelet);  
CA=appcoef2(C, S, wavelet, level);  
[M, N]=size(CA);  
%低频系数归一化  
CAmin=min(min(CA));  
CAmax=max(max(CA));  
CA=(1/(CAmax-CAmin))*(CA-CAmin);  
d=max(size(CA));  
[U, sigma, V]=svd(CA);
```

```
%R 层作为水印图像
mark = imread('9999.jpg');
mark = double(mark)/255;
mark_R = mark(:, :, 1);
%将水印图像二值化
mark_R = round(mark_R);
[height,width] = size(mark_R);
%计算扩频的系数
times = row*list / (height*width);
%将水印图像转成一维并扩频
for i = 1:height
    for j = 1:width
        for k=1:times
            S_mark((i-1)*width*times + (j-1)*times + k)= mark_R(i, j);
        end
    end
end
%生成随机序列（代替 m 序列）
rand('seed',seed);
sz = size(S_mark,2);
m_sequence = rand(1,sz);
%随机序列二值化
for i=1:sz
    if m_sequence(i)>=0.5
        m_sequence(i)=1;
    else
        m_sequence(i)=0;
    end
end
%异或操作得到 gold 码
goldcode=xor(S_mark,m_sequence);
%还原成 2 维
spread_mark = reshape(goldcode,[row,list]);
%对扩频后的水印图像进行 svd 操作
[S_C,S_S]=wavedec2(spread_mark,level,wavelet);
S_CA=appcoef2(S_C,S_S,wavelet,level);
[M2,N2]=size(S_CA);
CAmin2=min(min(S_CA));
CAmax2=max(max(S_CA));
S_CA=(1/(CAmax2-CAmin2))*(S_CA-CAmin2);
d=max(size(S_CA));
[S_U,S_sigma,S_V]=svd(S_CA);
V2=S_V;
U2=S_U;
```

```
sigma_tilda=alpha*flipud(sort(rand(d,1)));

%-----生成嵌入水印图，替换 u 和 v.

S_CA = S_U*S_sigma*S_V';
S_CA=reshape(S_CA,1,S_S(1,1)*S_S(1,2));
S_C(1,1:S_S(1,1)*S_S(1,2))=S_CA;
spread_mark3 = waverec2(S_C,S_S,wavelet);
spread_mark3 = round(abs(spread_mark3));
goldcode = reshape(spread_mark3,[1,row*list])
%-----异或操作得到扩频信号
S_mark=xor(goldcode,m_sequence);
sz = size(S_mark,2);
n = 0
flag = 0
for i = 1:sz
    flag = flag + 1;
    if flag == 9
        n = n + 1;
        flag = 0;
        mark(n) = S_mark(i);
    end
end
sz2 = size(mark,2);
for i=1:128
    for j=1:128
        spread_cache(i,j)=mark((i-1)*128+j);
    end
end
figure(2)
mark(:, :, 1) = spread_cache*255
imshow(spread_cache)

watermark=U2*diag(sigma_tilda,0)*V2';
%重构生成水印的形状，便于直观认识，本身无意义
watermark2=reshape(watermark,1,S_S(1,1)*S_S(1,2));
waterC=S_C;
waterC(1,1:S_S(1,1)*S_S(1,2))=watermark2;
watermark2=waverec2(waterC,S_S,wavelet);
%成嵌入水印后的图像
CA_tilda=watermark+CA;
%系数调整，将过幅系数与负数修正
```

```
over1=find(CA_tilda>1);
below0=find(CA_tilda<0);
CA_tilda(over1)=1;
CA_tilda(below0)=0;
%系数还原到归一化前范围
CA_tilda=(CAmax-Camin)*CA_tilda+Camin;
%记录加有水印的低频系数（仅保存）
waterCA=CA_tilda;
%重构
CA_tilda=reshape(CA_tilda,1,S(1,1)*S(1,2));
C(1,1:S(1,1)*S(1,2))=CA_tilda;
watermarkimage=waverec2(C,S,wavelet);
watermarkimagergb=data;
watermarkimagergb(:, :, 1)=watermarkimage;
imwrite(watermarkimagergb,goal,'BitDepth',16);
watermarkimagergb2=imread(goal);
%图片展示
% mark(:, :, 1) = mark_R*255;
figure(1);
subplot(321),imshow(watermark2*255);title(' 水印形态图');
subplot(322),imshow(mark);title(' 水印图像');
subplot(323),imshow(data);title(' 原始图像');
subplot(324),imshow(watermarkimagergb2);title(' 嵌入水印后的 rgb 图像');
subplot(325),imshow(datared);title(' R 层图像');
subplot(326),imshow(watermarkimage);title(' 嵌入水印后的 R 层图像');
```

水印提取：

总体流程步骤如下：

1. 对原始载体图像 C 进行小波变换得到低频系数矩阵
2. 对原始水印图像 W 进行小波变换得到低频系数矩阵
3. 分别对 C 和 W 做 SVD 得到 USV 和 U' S' V'
4. 重构提取水印 $W' = U * S' * V$
5. 用 Gold 码解码得到原始水印图像
6. 计算提取水印的误码率

提取代码如下：

```
function
[corr_coef,corr_DCTcoef]=detect(test,original,seed,wavelet,level,alpha)
%function realCA=wavedetec(test,original,seed,wavelet,level,alpha,ratio)
dataoriginal=imread(original);
datatest=imread(test);
dataoriginal=double(dataoriginal)/255;
datatest=double(datatest)/65535;
```

```
dataoriginal=dataoriginal(:, :, 1);
datatest=datatest(:, :, 1);
%提取加有水印的图像的小波低频系数
[watermarkimagergb, watermarkimage, waterCA, watermark2, S_sigma, S_C, S_S]=wavemarks
vd(original, 'temp.png', seed, wavelet, level, alpha);
%提取待测图像的小波低频系数
[row, list]=size(datatest);
[C, S]=wavedec2(datatest, level, wavelet);
CA_test=appcoef2(C, S, wavelet, level);
%提取原始图像的小波低频系数
[C, S]=wavedec2(dataoriginal, level, wavelet);
realCA=appcoef2(C, S, wavelet, level);
%生成两种水印
realwatermark=waterCA-realCA;
testwatermark=CA_test-realCA;

%计算相关性
corr_coef=trace(realwatermark'*testwatermark)/(norm(realwatermark, 'fro')*norm(t
estwatermark, 'fro'));
%DCT 系数比较
DCTrealwatermark=dct2(waterCA-realCA);
DCTtestwatermark=dct2(CA_test-realCA);
DCTrealwatermark=DCTrealwatermark(1:min(32, max(size(DCTrealwatermark))), 1:min(3
2, max(size(DCTrealwatermark))));
DCTtestwatermark=DCTtestwatermark(1:min(32, max(size(DCTtestwatermark))), 1:min(3
2, max(size(DCTtestwatermark))));
DCTrealwatermark(1, 1)=0;
DCTtestwatermark(1, 1)=0;
corr_DCTcoef=trace(DCTrealwatermark'*DCTtestwatermark)/(norm(DCTrealwatermark, '
fro')*norm(DCTtestwatermark, 'fro'));

%-----将扩频信号（水印）还原成窄带信号
[S_U, S_sigma, S_V] = svd(testwatermark);
S_CA = S_U*S_sigma*S_V';
S_CA=reshape(S_CA, 1, S_S(1, 1)*S_S(1, 2));
S_C(1, 1:S_S(1, 1)*S_S(1, 2))=S_CA;
spread_mark3 = waverec2(S_C, S_S, wavelet);
spread_mark3 = round(abs(spread_mark3));
goldcode = reshape(spread_mark3, [1, row*list])
%-----异或操作得到扩频信号
S_mark=xor(goldcode, m_sequence);
sz = size(S_mark, 2);
n = 0
flag = 0
```

```
for i = 1:sz
    flag = flag + 1;
    if flag == 9
        n = n + 1;
        flag = 0;
        mark(n) = S_mark(i);
    end
end
sz2 = size(mark, 2);
for i=1:128
    for j=1:128
        spread_cache(i, j)=mark((i-1)*128+j);
    end
end
figure(2)
imshow(spread_cache)
```

接下来计算误码率：

代码如下：

```
seed = 4;
row = 384;
list = 384;
test = 'embed.png';
original = 'girl.jpg';
wavelet = 'db6';
level = 2;
alpha = 0.1;

%-----R 层作为水印图像
mark = imread('9999.jpg');
mark = double(mark)/255;
mark_R = mark(:, :, 1);
%-----将水印图像二值化
mark_R = round(mark_R);
[height,width] = size(mark_R);
%-----计算扩频的系数
times = 384*384 / (height*width);
%-----将水印图像转成一维并扩频
for i = 1:height
    for j = 1:width
        for k=1:times
            S_mark((i-1)*width*times + (j-1)*times + k)= mark_R(i, j);
```

```
        end
    end
end
%-----生成随机序列（代替 m 序列）
rand('seed', seed);
sz = size(S_mark, 2);
m_sequence = rand(1, sz);
%-----随机序列二值化
for i=1:sz
    if m_sequence(i) >= 0.5
        m_sequence(i) = 1;
    else
        m_sequence(i) = 0;
    end
end
%-----异或操作得到 gold 码
goldcode = xor(S_mark, m_sequence);
%-----还原成 2 维
spread_mark = reshape(goldcode, [row, list]);

[S_C, S_S] = wavedec2(spread_mark, level, wavelet);
S_CA = appcoef2(S_C, S_S, wavelet, level);
[M2, N2] = size(S_CA);
CAmin2 = min(min(S_CA));
CAmax2 = max(max(S_CA));
S_CA = (1 / (CAmax2 - CAmin2)) * (S_CA - CAmin2);
d = max(size(S_CA));
[S_U, S_sigma, S_V] = svd(S_CA);
V2 = S_V;
U2 = S_U;
sigma_tilda = alpha * flipud(sort(rand(d, 1)));
watermark = U2 * diag(sigma_tilda, 0) * V2';

%-----
dataoriginal = imread(original);
datatest = imread(test);
dataoriginal = double(dataoriginal) / 255;
datatest = double(datatest) / 65535;
dataoriginal = dataoriginal(:, :, 1);
datatest = datatest(:, :, 1);
%提取加有水印的图像的小波低频系数
[watermarkimagergb, watermarkimage, waterCA, watermark2, S_sigma, S_C, S_S] = wavemarks
vd(original, 'temp.png', seed, wavelet, level, alpha);
```

```
%提取待测图像的小波低频系数
[row, list]=size(datatest);
[C, S]=wavedec2(datatest, level, wavelet);
CA_test=appcoef2(C, S, wavelet, level);
%提取原始图像的小波低频系数
[C, S]=wavedec2(dataoriginal, level, wavelet);
realCA=appcoef2(C, S, wavelet, level);
%生成两种水印
realwatermark=waterCA-realCA;
testwatermark=CA_test-realCA;
%-----
[S_U, S_sigma2, S_V] = svd(testwatermark);
S_CA2 = S_U*S_sigma*S_V';
S_CA2=reshape(S_CA2, 1, S_S(1, 1)*S_S(1, 2));
S_C(1, 1:S_S(1, 1)*S_S(1, 2))=S_CA2;
spread_mark2 = waverec2(S_C, S_S, wavelet);
spread_mark2 = round(abs(spread_mark2));
goldcode2 = reshape(spread_mark2, [1, row*list]);

[S_U3, S_sigma3, S_V3] = svd(realwatermark);
S_CA3 = S_U3*S_sigma*S_V3';
S_CA3=reshape(S_CA3, 1, S_S(1, 1)*S_S(1, 2));
S_C(1, 1:S_S(1, 1)*S_S(1, 2))=S_CA3;
spread_mark3 = waverec2(S_C, S_S, wavelet);
spread_mark3 = round(abs(spread_mark3));
goldcode3 = reshape(spread_mark3, [1, row*list]);

%error = double(goldcode) - goldcode2;
error = goldcode3 - goldcode2;
correct_rate = length(find(error==0))/(384*384)
```

【实验分析】:

- 1、 请尽量使用曲线图、表等反映你的实验数据及性能
- 2、 对照实验数据从理论上解释原因
- 3、 如无明显必要，请不要大量粘贴实验效果图

■一、实现基于 DWT 的扩频水印嵌入和提取算法

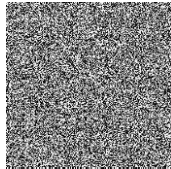
水印生成和嵌入:

命令行输入


```
fx >> [watermarkimagergb, watermarkimage, waterCA, watermark, S_sigma, S_C]=wavemarksvd('girl.jpg', 'embed.png', 4, 'db6', 2, 0.1)
```

即把 9999.jpg 的 R 层当作水印加到载体图片 girl.jpg 的 R 层生成图片 embed.png。

水印形态图



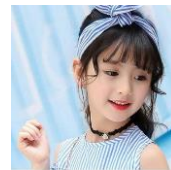
水印图像



原始图像



嵌入水印后的rgb图像



R层图像



嵌入水印后的R层图像



水印提取（嵌入的是原始 rgb 图像的 R 层）：

命令行输入：

```
fx >> [corr_coef, corr_DCTcoef]=detect('embed.png', 'girl.jpg', 4, 'db6', 2, 0.1)
```

```
corr_coef =
```

```
0.9871
```

```
corr_DCTcoef =
```

```
0.9847
```

可以看到两种相关系数还是比较大的。。

提取的水印图像：



这里水印图像为原始水印图像的 R 层，基本正确提取出来！

计算误码率：

```
correct_rate =  
  
    0.7311  
  
>>
```

上面为正确率，误码率=1-0.73=0.27.