# 媒体信号处理基础-实验报告6

学号：3170105728 姓名：林昭炜

## 实验内容及要求

实验工具：MATLAB 2017b

实验内容：

1. 编码实现1D离散信号序列的DWT变换及其反变换IDWT；
2. 基于1的实现完成如下应用：
3. 图像分解
4. 图像压缩

## 关键代码及注释

% Major Functions

% ----------------------------------------

function [r] = dwt\_once(arr)

% perform dwt once

sz = size(arr);

cnt = floor(max(sz) / 2);

next = cnt;

r = zeros(sz);

for i=1:cnt

x1 = arr((i-1) \* 2 + 1);

x2 = arr(2\*i);

r(i) = (x1 + x2) / 2; % average

r(next + i) = (x1 - x2) / 2; % detail

end

end

function [arr] = dwt(arr, varargin)

% perform dwt on arr,

% the times is dtermined by varargin{1}

% by default will perform until one average left

% . . .

while cnt >= 2 && times ~= 0

% perform dwt at average part of dwt

% the average part ranges from 1 to cnt

arr(1:cnt) = dwt\_once(arr(1:cnt));

cnt = floor(cnt / 2);

if times > 0

times = times - 1;

end

end

% . . .

function [img] = dwt2(img, varargin)

% dwt2(img, [times, channels]);

% perform a 2d dwt2 on img

% img: the orginal image

% times: how many times dwt will be applied to img

% channel: will apply dwt each channel, default is the first channel.

% . . .

% apply on each row

for i=1:sz(1)

img(i,:, chnl) = dwt(img(i,:, chnl), times);

end

% apply on each column

for i=1:sz(2)

img(:,i, chnl) = dwt(img(:,i, chnl), times);

end

% . . .

function [arr] = idwt(arr, varargin)

% perform idwt on image

% idwt(arr [, times])

% times: how many times required to restore

% . . .

% perform idwt until reaches whole length of array

% done means how many are already retored to be average

while done < len

% x stores all the avarage signals

x = arr(1:cnt);

for i=1:cnt

d = arr(done + i);

xx = x(i);

arr(2\*i - 1) = xx + d;

arr(2 \* i) = xx - d;

end

done = done \* 2;

cnt = cnt \* 2;

end

% . . .

% idwt2 is the same as dwt2 except it calls idwt instead of dwt.

% Main Script

% ----------------------------------------

%% Task1: do dwt

B = dwt(A);

C = idwt(B);

%% Task 2: img compression

% . . .

% pad image

imsize = size(im);

padding = mod(cell\_cnt - mod(imsize, cell\_cnt), cell\_cnt);

padding(3) = 0;

img = zeros(imsize + padding);

img(1:imsize(1), 1:imsize(2), :) = im;

% perform dwt. . .

% Calculate filter according to imge

filter = (abs(img\_dwt) > cutoff);

% make sure the most important part stays untouched.

filter(1:width,1:height,:) = 1;

img\_dwt = img\_dwt .\* filter; % filter image

% perform idwt and unpad . . .

## 实验结果及分析

这是经过DWT后的图像



为了把黑色区域看的更清楚，把图片对比度和亮度增强一下，可以看到下面一张:



这是原图，



这是把DWT后的图像所有小于等于5的值设为0后的结果，可以发现有比较明显的小方格。

