Digital Imaging Processing
數字影像處理
Project Three

Lin Tung-wen

05, 2013

1 Step 1:Haar Wavelet transform

1.1 How to

從題目可以知道scaling function 為1/2,1/2 , 而wavelet coefficients function 為1,-1. 假設N 是X_n 的長度, 則有M = $[\frac{X_n}{2}]$, 可得到一下的式子，其中$1 \leq k \leq M$

$$L_k = \frac{1}{2}(x_{2k-1} + x_{2k}); \tag{1}$$

$$H_k = |(x_{2k-1} - x_{2k}|; \tag{2}$$

1.2 The result

從(1(a)) 可以看到做水平變換的結果。

(a) Horizontally Produced Lena　　(b) Verticallly Produced Lena

Figure 1: The result for the first problem

類似的，再對垂直方向做同樣的變化，結果如(1(b)). TIPS：兩次變化以後，由於白色不太清晰，放大會看的比較清楚。也可參見vertically produce.jpg和horizontally produce.jpg這兩個檔案。

The related code is in main.m: Step 1 and Haar Wavelet transform is in hwt.m

2 Step 2:Watermark 2 Binary

2.1 How to

第二步需要做的是把要植入的水印做變化使它變為Binary的Code，圖像在這裡會丟失一些內容。變為兩個明暗度。然後再對其做行列變化使其變為一列。比如說原來是M行N列的矩陣，在變化后，變為有$M \times N$個元素的行向量。

2.2 The result

詳細可看附的文件夾中文件名為“Myname.jpg”（變化前）的文件，和文件名為“bits.jpg”（變化后）的文件。分別是做變化前的水印和變化后的水印。相關代碼參見main.m的Step 2的部份。

3 Step 3:Embeding

第三步要做的是Watermark的Embed的過程。這一步是整個程式里最核心的部份。

3.1 Related Knowledge

加密算法主要通過Haar Wavelet Transform來實現，也叫做Interger Wavelet Transform或Difference Expansion Transform，The difference Expansion的原理如下：

假設需要展開兩個值$x = 206$, $y = 201$, 則有

$$l = [\frac{206 + 201}{2}] = [\frac{407}{2}], h = 206 - 201 = 5$$

我們想要在裏面植入bit=1的話，讓

$$h' = 2 \times h + b = 2 \times 5 + 1$$

這樣新的值就變成了

$$x = l + [\frac{h+1}{2}] = 209, y = l - [\frac{h}{2}] = 198$$

通過這樣的原理就可以植入bit在圖像裏面。

爲了區別植入和沒有植入的變量，我們又新引入了一個閾值T。根據T將圖像中的點分成不同的情況：

- 如果$|h| \leq [\frac{T-1}{2}]$ 則為集合M
- 如果$2T+1 \geq |h| > [\frac{T-1}{2}]$ 則為集合N
  - 如果$T \geq |h| > [\frac{T-1}{2}]$則為集合$N_e$
  - 如果$2T+1 \geq |h| > T$則為集合$N_{\bar{e}}$
- 如果$|h| > 2T+1$ 則為集合U

藏數據的時候，我們會把所有的數據都藏在集合N_e和M裏面。

關於T值確定的辦法：

在Decode的時候，由於集合N_e和$N_{\bar{e}}$的確定需要Map才能確定，這樣才能保證能夠恢復到原圖片。而在N裏面只有N_e 是可以藏數據的，所以一共只能藏$M - N_{\bar{e}}$個的數據，首先確定需要藏的水印的大小，然後根據這個大小來確定需要T的最小值。T越大，則圖像的失真會越大。具體的確定算法參見**caculate_T.m**

3.2 How to

Step0 首先計算T的值得大小。

Step1 通過引入一個同圖像像素行列相同的矩陣ID來表示圖像每一個屬於的集合類型

Step2 根據矩陣ID來得出Map表，用來表示矩陣N_e和$N_{\bar{e}}$

Step3 將Map與Payload連接起來，然後首先把數據藏在M矩陣之中，然後再把剩下的數據藏在N_e里。加密的時候利用加密算法。

3.3 The result

The result can be seen in fig (2), 從肉眼很難看出不同。

Figure 2: Watermarked Lena Image

4 Step 4:Compute the Histogram

4.1 Related Knowledge

在計算機圖像學領域中，常用一種灰度直方圖。灰度直方圖是灰度級的函數，描述的是圖像中具有該灰度級的像素的個數：橫坐標是灰度級，縱坐標是該灰度出現的頻率（像素個數）。

4.2 How to

使用Matlab的一個函數叫imhist即可繪出直方圖（需要注意的是先把圖片換成uint8的類型）

4.3 The result

The result can be seen in fig (3).

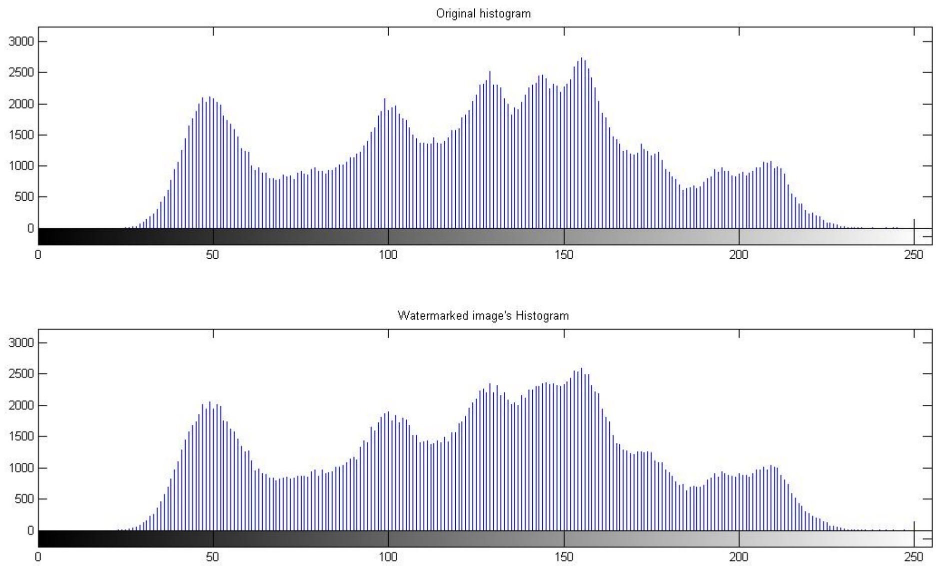

Figure 3: watermarked

5 Step 5:Decoding

5.1 How to

Decoding的過程與encoding的過程恰好相反，可以將水印從圖片里提取出來，同時也會將圖像還原到原來的情況。Decoding的相關算法如下:

Step1 根據T的值與H的值把不同的點對按照集合劃分為三類，M,N和U，分別標記其ID，$ID = 1$為U $ID = 4$為M，$ID = 5$為N。

Step2 根據N的大小計算出Map的位數，并從M中取出Map。

Step3 依據Map把N分為N_e和$N_{\bar{e}}$。標記ID，$ID = 2$為$N_{\bar{e}}$，$ID = 3$為N_e，

Step4 根據ID對$M(ID = 4$，先恢復)以及N_e（$ID = 3$，后恢復）進行復原，並提取出Bit。

5.2 The result

The result can be seen in fig (4)

(a) The water mark (b) Verticallly Produced Lena

Figure 4: See the watermark original one and the decoding one

6 Step 6:Compare the extracted binary signature and the recovered image

6.1 How to

使用Matlab的函數imhist繪出直方圖即可，同理需要先把圖像變換成uint8的類型。注意一個圖的值是0到255，而恢復的圖只有0到1兩個值。

6.2 The result

從fig (5)可以看出，恢復的圖像只有兩個值，0和1，所以一定會有一些成份是丟掉了的，但如果直接植入的圖像就是二值的圖像或數據的話，則可以認為是不會丟失任何的成份的。

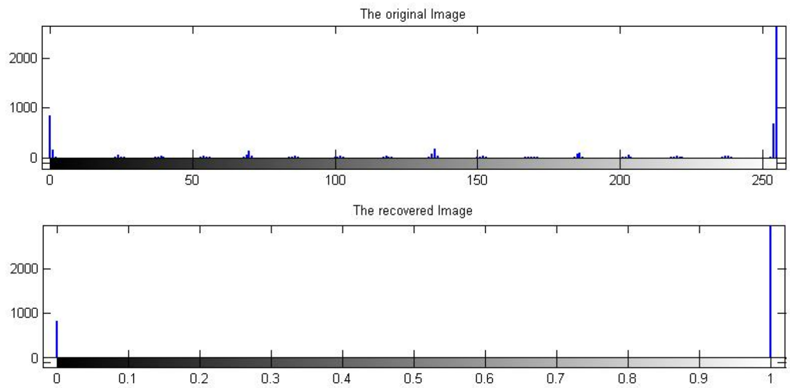

Figure 5: The histogram of the watermark(payload)

同時，從(6) 可以看出來，lena這個圖像被完全的恢復了。

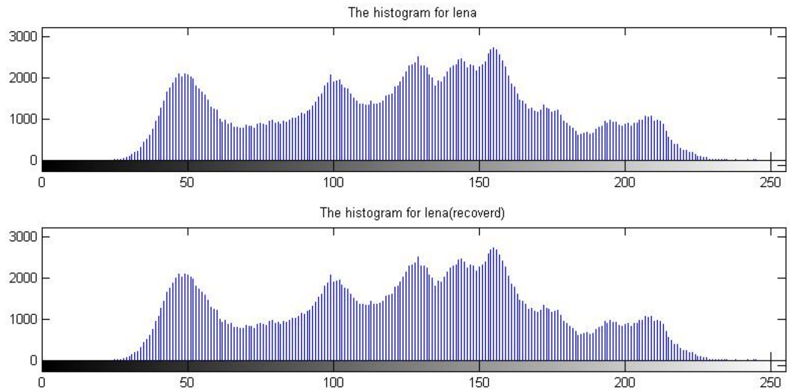

Figure 6: The histogram of lena, The original one and the recovered one

7 Source Code

Here are the source code for this project.

Input matlab source:main.m

```
clear all

%% ------------STEP1: Haar Wavelet Transform-------------- %%

disp('------------STEP1: Haar Wavelet Transform--------------')
%1.Read The Image
x=imread('lena_gray_512.tif');
[row,col]=size(x);
x_original=x;
x=double(x);
%myimshow(x,'Original Image');

%2Horizontal
for j=1:row
    tmp1=x(j,:);
    [L,H]=hwt(tmp1);
    x1(j,:)=[L,H];
end
%figure;myimshow(x1,'horizontally produce');
imwrite(x1,'horizontally produce.jpg');
```

```
%3Vertically
for i=1:col
    tmp1=x(:,i)';
    [ra1,rd1]=hwt(tmp1);
    x2(:,i)=[ra1,rd1]';
end
x2 = uint8(x2)
%figure;myimshow(x1,'vertically produce');
imwrite(x2,'vertically produce.jpg');

%% ----------STEP2: Watermark 2 Binary------------ %%
disp('----------STEP2: Watermark 2 Binary------------')
%1.To get the Binary Code
orginal_watermark = imread('myname.JPG');
watermark = imtobinary('myname.JPG');
figure;myimshow(watermark,'Befroe');

%2.Change it shape to a line of binary code.
bits = reshape(watermark,1,[]);
%figure;myimshow(bits,'after');
imwrite(bits,'bits.jpg')

%% ----------STEP3Embeding------------ %%
disp('----------STEP3Embeding------------ ')
%To demonstrate the principle, we simply let T=1; other wise T could
%be another variable.

M=floor(col/2);
T=caculate_T(length(bits),x);

%Mark ID
%Step 3.1
%USE ID to mark the node to different set. ID=1:U ID=2:Ne_bar ID=3:Ne,
%ID=4:M ID=0:Error or Unknown.
count_ne_bar=0;count_u=0;count_m=0;count_ne=0;count_n=0;count_total=0;ID=[];
for i=1:row
    for k=1:M
        count_total = count_total + 1;
        h(i,k) = abs(x(i,2*k-1) - x(i,2*k));
        if h(i,k) > 2 * T + 1 || ((x(i,2*k-1) + x(i,2*k)) * 1/2) >=
```

```
255 - T || ((x(i,2*k-1) + x(i,2*k)) * 1/2) < T
        count_u =count_u + 1;
        ID(i,k) = 1; %u
        elseif h(i,k) > floor((T-1)/2)
        count_n=count_n + 1;
            if h(i,k) > floor(T)
            count_ne_bar=count_ne_bar + 1;
            ID(i,k) = 2; %ne_bar
            else
            count_ne=count_ne + 1;
            ID(i,k) = 3; %ne
            end
        else
        count_m=count_m + 1;
        ID(i,k) = 4; %_m
        end
    end
end

%Step 3.2 Draw the Map
map = zeros(1,count_n);
count = 1; %to indicate where I'm writing
for i=1:row
    for k=1:M
        if ID(i,k) == 2
            count=count+1;
        elseif ID(i,k) == 3
            map(count) = 1;
            count=count+1;
        end
    end
end

%Thus payload = [Map, watermark]
payload = [map, bits];
payload = [payload, zeros(1,count_m+count_ne-length(payload))]; %match the size of th
%myimshow(payload,'payload')


%Step 3.3 Watermark_Embeding
```

```
%%% ATTENTION
%%% In this case M is for more larger than the data that we will
%%% embed, which means that we did not use Ne to embed payload.
%%% Otherwise the code would be diffenrent

count = 1; %to indicate where I'm writin
for i=1:row
    for k=1:M
        if (ID(i,k) == 4) % If that is in SET M
            [x(i,2*k-1),x(i,2*k)] = embed(x(i,2*k-1),x(i,2*k),payload(count));
            count = count + 1;
        end
    end
end
str=sprintf('We have embeded %d bits in M',count-1);
disp(str)

count;%to indicate where I'm writin
for i=1:row
    for k=1:M
        if (ID(i,k) == 3) % If that is in SET N_e
            [x(i,2*k-1),x(i,2*k)] = embed(x(i,2*k-1),x(i,2*k),payload(count));
            count = count + 1;
        end
    end
end
str=sprintf('We have embeded %d bits in M & Ne ',count-1);
disp(str);

%Step 3.4.Image Demonstration or Writing
figure;myimshow(x,'Watermarked Image');
%imwrite(x,'marked.tif')

%% ----------STEP4 Compute the Histgram------------ %%%
% x2 = uint8(x)
disp('----------STEP4 Compute the Histgram------------')
%figure;subplot(2,1,1);imhist(x_original);title('Original histogram');
%subplot(2,1,2);imhist(x2);title('Watermarked image''s Histogram');
```

```
%% ———————STEP5 Decoding———————— %%
disp(' ———————STEP5 Decoding———————— ')
%watermark=decoder(x,length(bits))

%function [wartermark]=decoder(watermarked_image,length_watermark)
%
%This function is used to decode the image. In case to decode the watermark
%we need to know the length of map and the length of the watermarked image
%in this case.
%
%TIPS: A more universal way is to save those details in the head of the payload,
%but since this project is just for demonstration use, there is no need to
%do these things.
%
% David Lin
% NATIONAL CHIAO TUNG UNIVERSITY
%
%x=watermarked_image;

[d_row,d_col]=size(x);
M=floor(d_col/2);

%length_map=count_n
%length_watermark=200*100=20000

% STEP5.1: Mark the ID————— %%
% This is used to ruturn a martrix to mark each of pixel that is we are here to mark

%ID=1:U
%ID=2:Ne_bar
%ID=3:Ne
%ID=4:M
%ID=5;N
%ID=0:Error or Unknown. (NOT USED IN THIS CASE)

d_count_ne_bar=0;d_count_u=0;d_count_m=0;d_count_ne=0;d_count_n=0;d_count_total=0;d_I
for i=1:d_row
    for k=1:M
        d_count_total = d_count_total + 1;
        d_h(i,k)=x(i,2*k-1) - x(i,2*k);
```

```
        %1. This case is U.
        if abs(d_h(i,k)) > 2 * T + 1 || ((x(i,2*k-1) + x(i,2*k)) * 1/2) >=
255 - T || ((x(i,2*k-1) + x(i,2*k)) * 1/2) < T
        d_count_u =d_count_u + 1;
        d_ID(i,k) = 1;
        %3.This case is M
        elseif d_h(i,k) >= 2 * fix(-(T-1)/2) && d_h(i,k) <= 2 * fix((T-1)/2) + 1 % d_
        d_count_m=d_count_m + 1;
        d_ID(i,k) = 4; %m
        %2. This case is N
        else
            d_count_n=d_count_n + 1;
            d_ID(i,k) = 5;
        end
    end
end

% STEP5.2:Draw the location map from M
d_count = 1; d_map=[]
for i=1:d_row
    for k=1:M
        if(mod(d_h(i,k),2)==0) && d_count <= d_count_n
            d_map(d_count) = 0;
            d_count=d_count+1;
        end
        if (mod(d_h(i,k),2)==1) && d_count <= d_count_n
            d_map(d_count) = 1;
            d_count=d_count+1;
        end
    end
end

% STEP5.3: Use map to recover Ne and Ne_bar
    %2_1 This case is N_e_bar
d_count=1;
for i=1:d_row
    for k=1:M
        if d_ID(i,k) == 5;
            if d_map(d_count) == 0
            d_ID(i,k) = 2; %ne_bar
```

```
            d_count_ne_bar = d_count_ne_bar + 1;
            d_count=d_count+1;
            elseif d_map(d_count) == 1
            d_ID(i,k) = 3; %ne
            d_count=d_count+1;
            d_count_ne = d_count_ne + 1;
            end
        end
    end
end

% STEP5.4: Recorver the data.
count = 1; d_payload = []
for i=1:row
    for k=1:M
        if ID(i,k) == 4
            [x(i,2*k-1),x(i,2*k),d_payload(count)] = recover(x(i,2*k-1),x(i,2*k));
            count=count+1;
        end
    end
end

count;
for i=1:row
    for k=1:M
        if ID(i,k) == 3
            [x(i,2*k-1),x(i,2*k),d_payload(count)] = recover(x(i,2*k-1),x(i,2*k));
            count=count+1;
        end
    end
end

d_watermark = reshape(d_payload((d_count_n+1):(d_count_n+20000)),100,200);
figure;
figure;myimshow(d_watermark,'After')
imwrite(d_watermark,'d_watermark.jpg');


%% ----------STEP6 Comparing------------ %%
%figure;
```

```
%subplot(2,1,1);imhist(orginal_watermark)

for i = 1:100
    for j = 1:200
        if d_watermark(i,j) == 1
        d_watermark(i,j) = 255;
        end
    end
end

%subplot(2,1,2);imhist(d_watermark)
%title('compare the hist of the image')

figure;
subplot(2,1,1)
imhist(x_original)
title('The histogram for lena')
subplot(2,1,2)
x = uint8(x)
imhist(x)
title('The histogram for lena')
```

Input matlab source:lib/hwt.m

```
function [cA,cD]=hwt(x)

%This function is used to do Haar Wavelet Transform

sfc=[1/2,1/2];wfc=[1,-1];

N=length(x);          % The length of the sequence
M=floor(N/2);
k=1:M;
cA(k)=x(2*k-1) * 1/2 + x(2*k) * 1/2;
cD(k)=abs(x(2*k-1) - x(2*k));
```

Input matlab source:lib/myimshow.m

```
function myimshow(x,str)
%%%%
%function myimshow(x,str)
```

```
%eg. myimshow(x,'Original Image');
%To show the image as well as the size of the image
%%%%
[row,col]=size(x);
imshow(x);title(str);
xlabel(['Size : ',num2str(row),'*',num2str(col)]);
```

Input matlab source:lib/caculate_T.m

```
function T = caculate_T(data_size,x)

[row,col]=size(x)
M=floor(col/2);
T=0;
while 1
    count_ne_bar=0;count_u=0;count_m=0;count_ne=0;count_n=0;count_total=0;ID=[];
    T=T+1;
for i=1:row
    for k=1:M
        count_total = count_total + 1;
        h(i,k) = abs(x(i,2*k-1) - x(i,2*k));
        if h(i,k) > 2 * T + 1 || ((x(i,2*k-1) + x(i,2*k)) * 1/2) >=
255 - T || ((x(i,2*k-1) + x(i,2*k)) * 1/2) < T
        count_u =count_u + 1;
        ID(i,k) = 1; %u
        elseif h(i,k) > floor((T-1)/2)
        count_n=count_n + 1;
            if h(i,k) > floor(T)
            count_ne_bar=count_ne_bar + 1;
            ID(i,k) = 2; %ne_bar
            else
            count_ne=count_ne + 1;
            ID(i,k) = 3; %ne
            end
        else
        count_m=count_m + 1;
        ID(i,k) = 4; %_m
        end
    end
end
if count_m >= count_n && count_m - count_ne > data_size
```

```
    break;
end
end
str=sprintf('T=%d;count_ne_bar=%d;count_u=%d;count_m=%d;count_ne=%d;count_n=%d;count_
disp(str)
```

Input matlab source:lib/recover.m

```
function [x,y,bit] = recover(x,y)

%This is used to recover the image

x = double(x);
y = double(y);
l = floor(x/2 + y/2);
h = x - y;
bit = mod(h,2);
h = (h - bit) / 2;
x = l + floor((h + 1)/2);
y = l - floor(h/2);
```

Input matlab source:lib/imtobinary.m

```
function watermark = imtobinary(str)

watermark = imread(str);
watermark = boolean(watermark);
watermark = double(watermark);
```

References

[1] Jun Tian, Reversible Data Embedding Using a Difference Expansion, IEEE, 2003

[2] wikipedia.http://en.wikipedia.org/wiki/Haar_wavelet

[3] Hyoung Joong Kim, Yun Qing Shi, Jeho Nam, Hyon-Gon Choo, Vasiliy Sachnev, A Novel Difference Expansion Transform for Reversible Data Embedding, IEEE, 2008