

제 11 회 학사학위 졸업논문
지도교수 하 동 환

Digital Watermarking by Detail Coefficient of Wavelet Transform

설 현 수

School of Integrative Engineering
in Chung-Ang University

2019 . 06

제 11 회 학사학위 졸업논문
지도교수 하 동 환

Digital Watermarking by Detail Coefficient of Wavelet Transform

이 논문을 학사학위 졸업논문으로 제출합니다.

중앙대학교 공과대학
융합공학부
설 현 수

2019년 06월

중앙대학교 공과대학 융합공학과 설 현 수 의 공학사 학위 취득을
위한 요구 사항 중 그 일부인 본 졸업논문을 인정함.

2019년 06월

지도교수 : 하 동 환 (인)

약 력



설현수는 1992년 대전시 대덕구 와동에서 설진호씨와 이영남씨의 장남으로 태어났다. 2011년 대전이문고등학교를 졸업하고, 2011년도에 중앙대학교 공과대학 융합공학부로 입학하여 현재 4학년에 재학 중이며 2019년 08월에 졸업할 예정이다.

감사의 글

먼저 바쁘신 와중에도 부족한 저의 논문을 지도해주시고 검토해주신 하동환 교수님 정말 감사드립니다. 논문 작성에 유용한 조언을 아끼지 않은 학과 친구들 일규, 현진에게도 고맙다는 말을 전하고 싶습니다.

또한 고등학생 혹은 그 이전부터 함께한 가장 친한 친구들인 윤석, 채영, 성현에게도 큰 감사를 전합니다. 낯선 서울생활에 힘들던 대학 초년시절 큰 도움을 주신 작은고모와 고모부, 직장생활 중 여러 가지 편의를 봐주신 김효순 원장님, 부족한 저의 수업을 열심히 따라와 준 학생 여러분들에게도 정말 감사드립니다.

마지막으로 정말 사랑하는 어머니 아버지 그리고 동생 혜수. 항상 고맙고 감사하고 사랑합니다.

2011년 처음 입학하여 2019년 현재 뒤늦게 졸업하기까지 많은 어려움이 있었습니다. 하지만 돌이켜보면 제 주위에는 언제나 여러분들이 있었고 그 덕에 제가 여기까지 올 수 있었습니다. 앞으로도 남은 인생이 많지만 여러분들이 있기에 어떤 일이 닥쳐오더라도 이겨낼 수 있을 것 같습니다. 위에 언급한 분들 그리고 언급하지 못한 분들 모두 다시 한번 감사드리고 앞으로 좋은 일이 있기를 바라겠습니다.

2019년 6월

설현수 올림

목 차

Digital Watermarking by Detail Coefficient of Wavelet Transform

1. Introduction.....	7
2. Materials and Methods.....	7
2-1. 비가시적 영역에서의 워터마킹.....	8
2-3. 웨이블릿 변환과 푸리에 변환.....	9
2-4. 웨이블릿 변환 수행.....	10
2-5. 세부값에 임계치 적용.....	11
3. Results and Discussion.....	13
3-1. 워터마킹된 이미지의 변형.....	14
3-2. 기존 방법과의 비교.....	14
3-2. 개선 방안.....	14
4. Conclusion.....	15
5. Matlab Code.....	16
6. Reference.....	23

1. Introduction

21세기에 접어들며 IT 기술의 발달로 인터넷이 보편화 되고 통신 속도가 나날이 증가함에 따라 미디어 콘텐츠들 역시 디지털화 되고 있다. 하지만 디지털 매체의 특성상 유통, 복제, 배포가 쉬워 불법 복제 등 창작물에 대한 보호가 문제가 되고 있다. 이에 대한 대안으로 대표적인 것이 디지털 워터마킹 기법으로 이는 동영상 혹은 사진 등 각종 디지털 데이터에 저작권을 나타내는 정보를 삽입한다. 이는 불법 복제 및 변형이 이루어진 후에도 남아있어 차후 문제가 발생 시 워터마킹 된 데이터를 추출하여 저작권에 대한 정보를 확인할 수 있다. 이에 워터마크에는 다양한 기능들이 구현되어 있는데 저작권자의 정보를 저장하는 기능, 불법 위조를 판별하기 위해 워터마크의 손상 여부를 확인하는 기능, 불법 복제자를 추적하기 위해 저작물을 받은 사용자의 정보를 저장하는 기능, 복사 횟수를 제한하여 무단 복사를 방지하는 기능, 제한된 사용자만 저작물을 이용할 수 있도록 사용자 명단을 저장하는 기능 등 다양한 역할을 지닌 워터마킹 기술들이 개발되어 있다.

본 논문에서는 저작권을 나타내는 정보를 담는 기능에 초점을 두어 저작물의 복제, 변형이 이루어지더라도 워터마킹 된 데이터의 손실을 최대한 줄이는 방법에 대해 논하고자 한다. 워터마킹의 삽입 영역에는 가시적 영역, 비가시적 영역이 있는데 이미지의 비가시적 영역인 주파수 영역에 데이터를 삽입하여 리사이징(Resizing), 회전, 압축 등의 변형을 한 후 이를 원본과 비교하여 결과를 고찰한다.

2. Materials and Methods

2-1. 비가시적 영역에서의 워터마킹

워터마크의 삽입 영역에는 삽입된 데이터가 사용자의 눈에 실제로 보이는 가시적 영역과 보이지 않는 비가시적 영역으로 나뉜다. 먼저 가시적 영역에서의 워터마킹은 원본 소스의 픽셀값들을 그대로 조작하여 워터마크를 삽입하는 방법으로 계산 과정이 단순하며 그 속도가 빠르다. 하지만 압축, 필터링등에 취약하며 원본 소스와의 단순 비교를 통해 워터마킹 된 데이터가 그대로 드러나기 때문에 보안에 취약하기에 삽입할 수 있는 데이터에 한계가 존재한다.

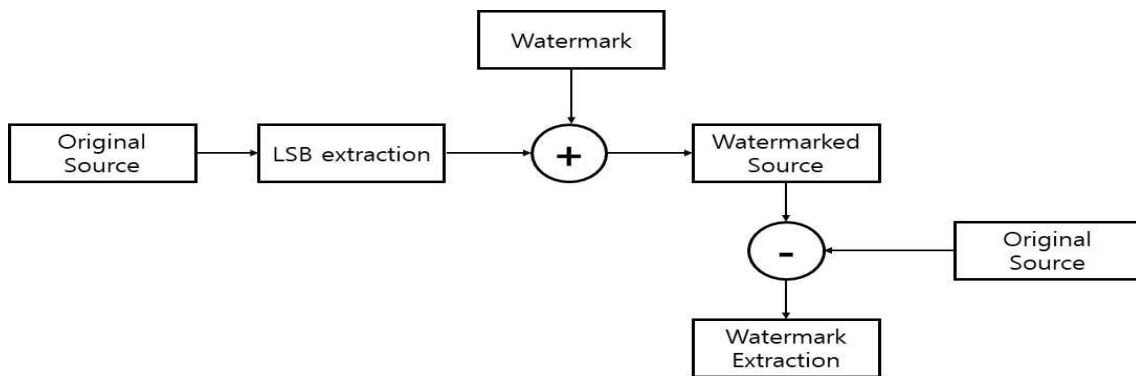


그림 2-1. 가시적 영역에서의 워터마킹 과정

반면 비가시적 영역의 워터마크 삽입은 원본 소스를 주파수대 영역으로 변환 후 계수 (Coefficient)를 변환하는 방법으로 변환, 압축에 강하지만 계산이 복잡하고 처리도 오래걸린다.

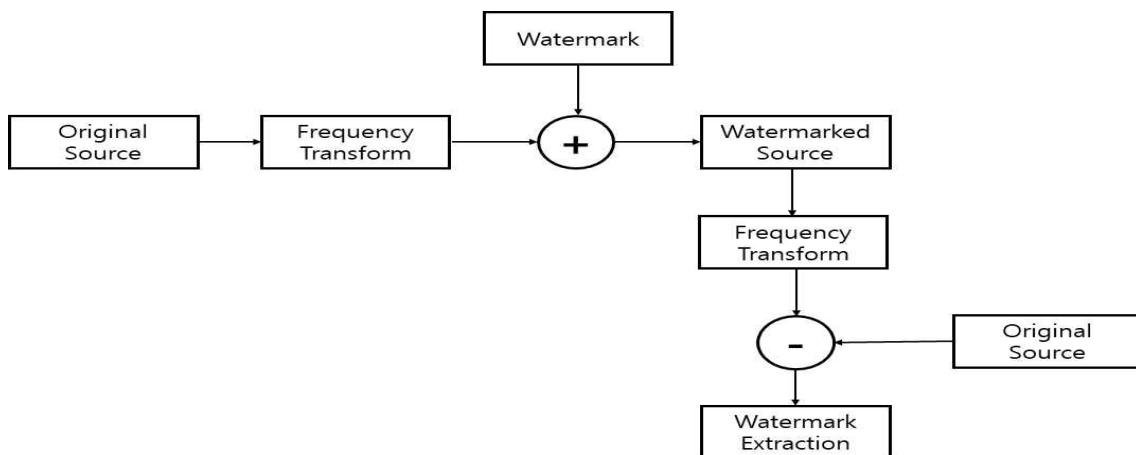


그림 2-2. 비가시적 영역에서의 워터마킹 과정

2-2. 웨이블릿 변환과 푸리에 변환

비가시적 영역에서의 워터마크 삽입을 위해서는 우선 원본 소스를 주파수 영역으로 변환하여야 한다. 이 변환에는 대표적으로 푸리에 변환(Discrete Fourier transform)과 웨이블릿 변환(Wavelet Transform)이 있는데 먼저 이산 푸리에 변환을 이용하여 시도해보았다.

원본 소스를 주파수 영역으로 변환하는 알고리즘으로 FFT(Fast Fourier transform)을 사용하였으며 삽입하고자 하는 데이터를 원형으로 이어붙여 삽입하는 방법을 생각해 보았으나 변환 평면의 계수값들이 복소수 성분을 갖게 되며 원본과 비교시 정합에 문제가 발생하였다.

이에 웨이블릿 변환을 사용하였는데 웨이블릿 변환은 푸리에 변환에서 시간 분해능과 주파수 분해능이 반비례 관계에 있어 이 둘 모두를 잡는 것은 불가능하다는 단점이 보완된 방법으로 고주파 대역에서는 주파수 해상도를 낮추며 시간 해상도를 높이고, 저주파 대역에서는 이를 역으로 실시한다.

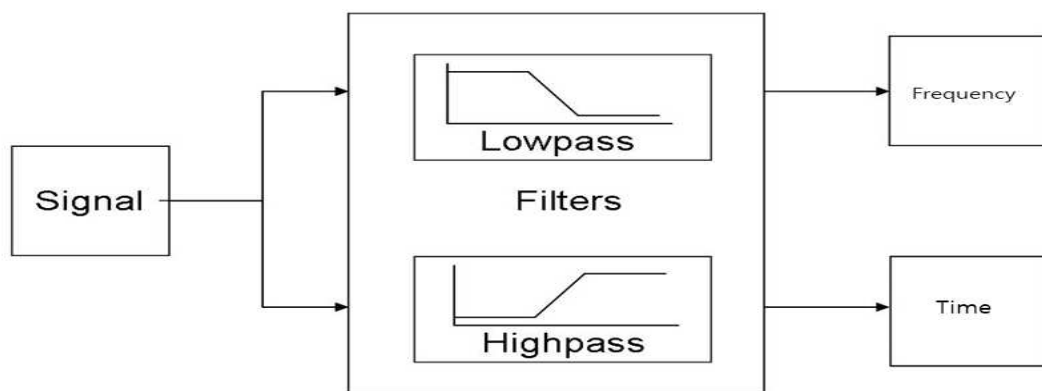


그림 3-1. 웨이블릿 변환의 개요도

웨이블릿 변환을 2D 이미지에 적용시 수직, 수평, 대각의 세 가지 세부값과 한 가지의 근사값으로 분해가 되며 이렇게 나온 근사값을 재차 웨이블릿 변환을 하여 단계(level)를 높여간다.

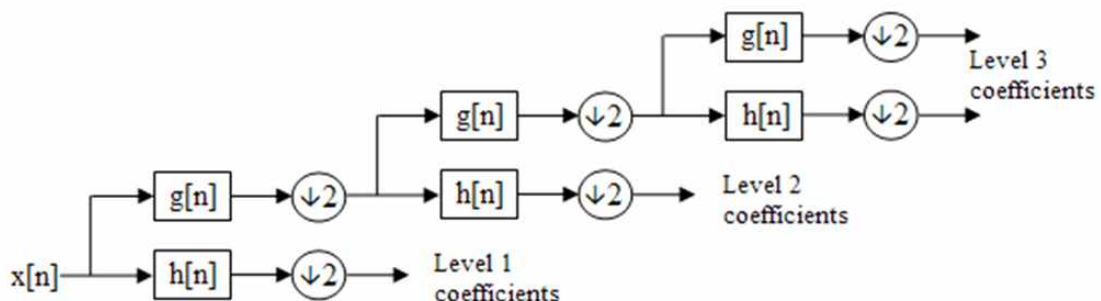


그림 3-2. 웨이블릿 변환의 단계

2-3. 웨이블릿 변환 수행

위터마크 삽입을 위해 먼저 원본 이미지에 웨이블릿 수행한다. 보통의 영상처리에서는 3 단계 혹은 4단계에서 작업을 하지만 본 논문에서는 2단계까지 실시하였다.

하단의 이미지를 보면 웨이블릿 변환을 통해 나온 세부값(Detail Coefficient)들이 이미지의 전체적인 형태를 나타내는 윤곽과 유사함을 알 수가 있다. 이런 영상의 특징점들에 데이터를 삽입하면 원본 소스의 변형시 위터마킹된 데이터의 손실을 줄일 수가 있다.

일반적으로 인간의 시각은 고주파수 대역과 대각성분의 잡음에 덜 민감하기에 대각성분의 세부값에 필터를 적용하여 임계치를 적용해 데이터를 삽입하기에 적합한 영역을 추출한다.



그림 4-1. 원본 이미지

Approximation Coef. of Level 1



Horizontal detail Coef. of Level 1



Vertical detail Coef. of Level 1



Diagonal detail Coef. of Level 1

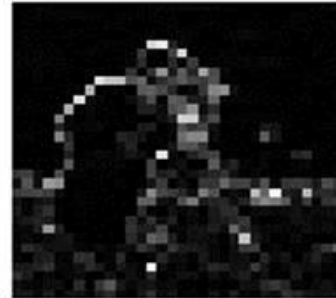


그림 4-2. 1단계 웨이블릿 변환

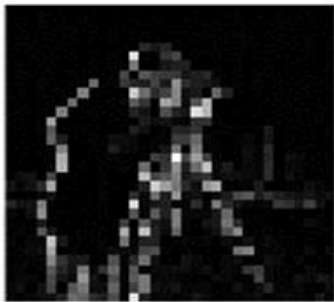
Approximation Coef. of Level 2



Horizontal detail Coef. of Level 2



Vertical detail Coef. of Level 2



Diagonal detail Coef. of Level 2

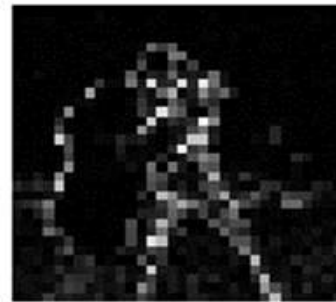


그림 4-3. 2단계 웨이블릿 변환

2-4. 세부값에 임계치 적용

2단계 대각성분의 세부값에 임계치를 적용하기 위해 소벨 필터(Sobel Filter)와 가우시안 스무딩 필터(Gaussian Smoothing Filter)를 순차적으로 적용하였다. 소벨 필터는 3x3의 행렬을 이용하여 중심값을 기준으로 상하, 앞뒤의 값을 비교해 그 변화량을 계산하는 방식으로 수직과 수평 필터 둘 모두를 적용하였다.

Approximation Coef. of Level 2



-1	-2	-1
0	0	0
-1	-2	-1

Vertical

-1	0	1
-2	0	2
-1	0	1

Horizontal

그림 5-1. 소벨 필터 적용

가우시안 스무딩 필터는 가우시안 분포를 이용하여 저주파수 대역은 증폭시키며 고주파수 대역은 감폭시키는 LPW(Low Pass Filter)로 잡음을 제거하기 위해 적용하였다. 본 논문에서는 5x5 의 마스크를 이용해 필터를 적용하였다.

Approximation Coef. of Level 2



$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

그림 5-2. 가우시안 필터 적용



그림 5-3. 필터가 적용된 데이터



그림 5-4. 워터마크가 삽입된 이미지

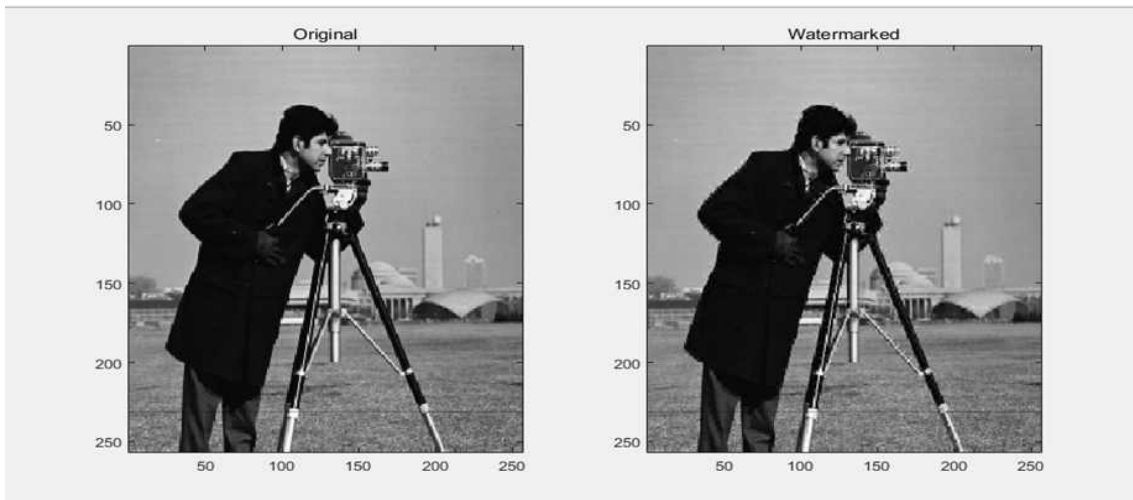


그림 5-5. 원본 이미지와 워터마킹된 이미지의 비교

3. Result and Discussion

3-1. 워터마킹 된 이미지의 변형

워터마크를 적용한 이미지에 대해서 리사이징(resizing), 회전, 압축, 잘라내기의 변형을 가해보았다. 리사이징과 회전은 동시에 진행하였으며 SURF(Speeded-Up Robust Features) 알고리즘을 이용하여 관심점을 찾아 복원하였다. SURF 알고리즘은 계산속도 향상을 위해 적분영상을 이용하여 고속 헤시안 검출(Fast Hessian Detector)을 수행한다.

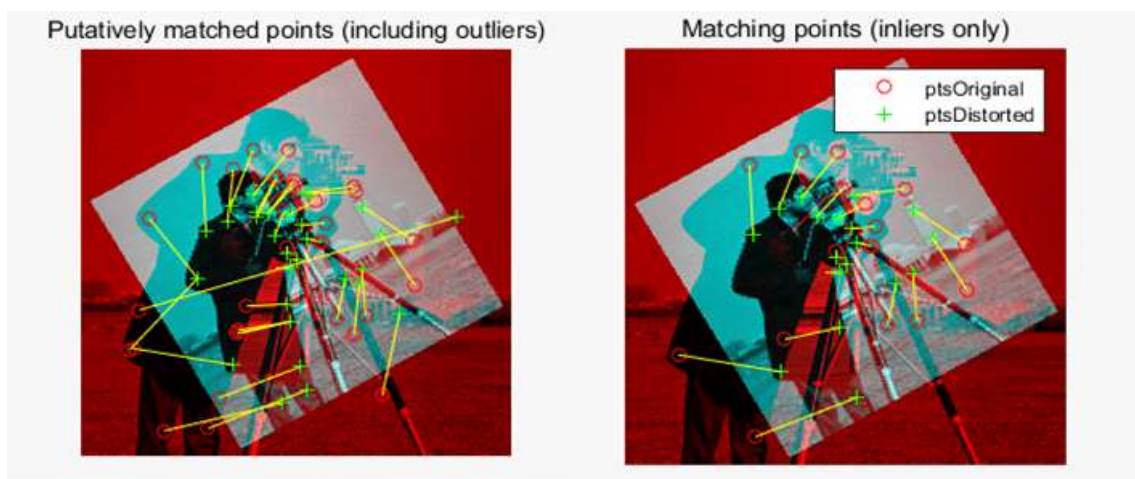


그림 6-1. SURF 알고리즘을 통한 리사이징& 회전된 이미지의 복원

압축에는 널리 사용되는 기법인 JPEG-2000 방식을 적용하였다. JPEG(Joint Photographic Experts Group)은 정지된 영상 혹은 이미지에 적용하기 위한 손실 압축방법 표준이다. JPEG 역시 웨이블릿 변환을 이용한 압축 방법이며 손실 압축방법 이기에 압축률과 데이터 손실율이 정비례한다. 잘라내기의 경우에는 임의의 부분을 잘라내었다.

3-2. 기존 방법과의 비교

비가시적 워터마킹시의 일반적인 방법에서는 원본 소스를 웨이블릿 변환한 후 전 주파수 대역에 워터마킹을 하고자 하는 데이터를 고르게 분포시킨다. 이러한 기존의 방법인 분산 삽입과 본 논문의 방법인 특징점 삽입을 비교해본 결과 특징점 삽입 방식이 압축, 회전, 리사이징의 변형에는 우수한 모습을 보였으나 잘라내기의 변형에서는 데이터가 특정 부분에 집중되어 있기에 손실률의 변동폭이 커 불안정하였다. 또한 소벨필터와 가우시안 필터를 적용하므로 계산량이 많아져 시간복잡도가 크게 증가하였다.

3-3. 개선방안

결론으로부터 도출된 문제점중 잘라내기 변형시 데이터 손실에는 근본적 특성으로 인해 개선이 어렵지만 필터의 적용으로 인한 시간복잡도의 증가 문제는 먼저 적분영상을 구해 이미지에 대한 분석 실시 후 차등 필터 적용 등으로 개선할 수 있을 것 같다. 또한 기존 방법과의 비교 시 샘플 데이터의 절대량을 늘리고 워터마킹 된 데이터의 변형에도 단계성을 가미한다면 더 정확한 비교치를 얻을 수 있을 것이다.

4. Conclusion

디지털 매체는 물리적 매체에 비해 쉽게 접하고 유통되기 때문에 불법적으로 이용될 가능성이 높다. 따라서 워터마킹 된 데이터는 원본 소스의 조작에도 데이터가 소실되지 않는 강인성을 갖추어야 한다.

본 논문에서는 웨이블릿 변환을 이용하여 2D 원본 소스를 주파수 영역으로 변환한 뒤 계수들의 특성과 인간의 시각적 특성을 이용하여 비가시적이고 외부 조작에 강한 디지털 워터마크 삽입법에 대해 다루어 보았다. 기존 널리 사용되고 있던 웨이블릿 변환 후 데이터를 전 영역에 걸쳐 고르게 분포하는 방식과 비교하여 회전, 리사이징, 압축에 대해 더 우수한 효과를 보이나 워터마크 데이터가 특정 부분에 집중되어 있어 잘라내기에 대해서는 취약한 모습을 보인다. 또한 필터를 적용하기에 계산량이 늘어 대용량의 영상 파일이나 실시간 영상처리에 사용하기엔 어려움이 있을 것으로 보이지만 삽입하고자 하는 데이터 소스의 특성이나 원본 매체의 활용을 잘 고려하여 사용한다면 이점을 거둘 수 있으리라 생각한다.

5. Matlab Code

```
close all; clear; clc;
[FileName,PathName] = uigetfile('*.','Select the Image file');
img = rgb2gray(imread(strcat(PathName, FileName)));
[c,s]=wavedec2(img,3,'haar');
Mc = size(img,1);
Nc = size(img,2);
[A1img,H1img,V1img,D1img] = dwt2(img,'haar');
[A2img,H2img,V2img,D2img] = dwt2(A1img,'haar');

temp = imresize(A1img,[1, size(A1img,1)*size(A1img,2)]);
var_(1) = var(temp);
temp = imresize(H1img,[1, size(H1img,1)*size(H1img,2)]);
var_(2) = var(temp);
temp = imresize(V1img,[1, size(V1img,1)*size(V1img,2)]);
var_(3) = var(temp);
temp = imresize(D1img,[1, size(D1img,1)*size(D1img,2)]);
var_(4) = var(temp);
temp = imresize(A2img,[1, size(A2img,1)*size(A2img,2)]);
var_(5) = var(temp);
temp = imresize(H2img,[1, size(H2img,1)*size(H2img,2)]);
var_(6) = var(temp);
temp = imresize(V2img,[1, size(V2img,1)*size(V2img,2)]);
var_(7) = var(temp);
temp = imresize(D2img,[1, size(D2img,1)*size(D2img,2)]);
var_(8) = var(temp);
%disp(var_);
```



```

figure;
subplot(2,2,1);
%imagesc(A1img);
imshow((A1img)/255);
%colormap ([1,1,1]);
title('Approximation Coef. of Level 1');
subplot(2,2,2);
imshow(H1img/255);
title('Horizontal detail Coef. of Level 1');
subplot(2,2,3);
imshow(V1img/255);
title('Vertical detail Coef. of Level 1');
subplot(2,2,4);
imshow(D1img/255);
title('Diagonal detail Coef. of Level 1');

```

```

figure;
subplot(2,2,1);
imshow(A2img/255);
%colormap pink(255);
title('Approximation Coef. of Level 2');
subplot(2,2,2);
imshow(H2img/255);
title('Horizontal detail Coef. of Level 2');
subplot(2,2,3);
imshow(V2img/255);
title('Vertical detail Coef. of Level 2');
subplot(2,2,4);
imshow(D2img/255);
title('Diagonal detail Coef. of Level 2');

```

```

new_A2img = A2img;
W = [0.025 0.025 0.025 0.025 0.025;
0.025 0.08 0.08 0.08 0.025;
0.025 0.08 0.2 0.08 0.025;
0.025 0.08 0.08 0.08 0.025;
0.025 0.025 0.025 0.025 0.025];
BW = edge(new_A2img,'Sobel');
new_A2img = new_A2img.*BW;
temp = zeros(size(new_A2img));
new_temp = 0;
count = 0;
for i = 3:size(A2img,1)-2
    for j = 3:size(A2img,2)-2
        temp(i,j) = sum(sum(new_A2img(i-2:i+2,j-2:j+2).*W));
        if(temp(i,j) ~= 0)
            count = count + 1;
            new_temp(count) = temp(i,j);
        end
    end
end
level = graythresh(new_temp);
threshold = max(new_temp) * level;
%disp(level)

new_temp = 0;
count = 0;
for i = 3:size(new_A2img,1)-2
    for j = 3:size(new_A2img,2)-2
        if(temp(i,j) < threshold)
            temp(i,j) = 0;

```

```

        else
            temp(i,j) = 1;
            count = count + 1;
            new_temp(count,1) = new_A2img(i,j);
            new_temp(count,2) = i;
            new_temp(count,3) = j;
        end
    end
end
figure;
new_A2img = new_A2img.*temp;
imshow(new_A2img/255);
figure;
new_temp = sortrows(new_temp,1);
value = 0;
while true
    value = 256 + round(rand()*256);
    total = 0;
    temp = dec2bin(value);
    stop = 0;
    for i = 1:length(temp)
        total = total + (temp(i)-'0');
    end

    %total
    stop = 0;
    if total == 3 || total == 4
        %disp(total)
        break;
    end
end

```

```

end
value = dec2bin(value);
top = length(value);
max_value = new_temp(length(new_temp),1);
now_value = max_value;
for i = length(new_temp):-1:length(new_temp) - top+1
    A2img(new_temp(i,2),new_temp(i,3)) = now_value;
    if value(i-(length(new_temp)-top)) == '1'
        now_value = now_value - 10;
    end
end
end
data = 0;
for i = length(new_temp):-1:length(new_temp) - top+1
    data(top + (i - length(new_temp))) = A2img(new_temp(i,2),new_temp(i,3));
end
disp(data);
disp(value(2:length(value)));
figure; imshow(A2img/255);

%% idwt
relmg2 = idwt2(A2img,H2img,V2img,D2img,'haar');
relmg = idwt2(relmg2,H1img,V1img,D1img,'haar');

watermarked_image_uint8=uint8(relmg);

%% img rotation
% Step 1: Read Image
original = img;
imshow(original);
text(size(original,2),size(original,1)+15, ...

```

```

'Image courtesy of Massachusetts Institute of Technology', ...
'FontSize',7,'HorizontalAlignment','right');

% Step 2: Resize and Rotate the Image
scale = 0.7;
J = imresize(watermarked_image_uint8, scale); % Try varying the scale factor.

theta = 30;
distorted = imrotate(J,theta); % Try varying the angle, theta.
figure, imshow(distorted)

% Step 3: Find Matching Features Between Images
ptsOriginal = detectSURFFeatures(original);
ptsDistorted = detectSURFFeatures(distorted);
[featuresOriginal, validPtsOriginal] = extractFeatures(original, ptsOriginal);
[featuresDistorted, validPtsDistorted] = extractFeatures(distorted, ptsDistorted);
indexPairs = matchFeatures(featuresOriginal, featuresDistorted);
matchedOriginal = validPtsOriginal(indexPairs(:,1));
matchedDistorted = validPtsDistorted(indexPairs(:,2));
figure;
showMatchedFeatures(original,distorted,matchedOriginal,matchedDistorted);
title('Putatively matched points (including outliers)');

% Step 4: Estimate Transformation
[tform, inlierDistorted, inlierOriginal] = estimateGeometricTransform(...
    matchedDistorted, matchedOriginal, 'similarity');
figure;
showMatchedFeatures(original,distorted,inlierOriginal,inlierDistorted);
title('Matching points (inliers only)');
legend('ptsOriginal','ptsDistorted');

```

```

% Step 5: Solve for Scale and Angle
Tinv = tform.invert.T;
ss = Tinv(2,1);
sc = Tinv(1,1);
scaleRecovered = sqrt(ss*ss + sc*sc);
thetaRecovered = atan2(ss,sc)*180/pi;

% Step 6: Recover the Original Image
outputView = imref2d(size(original));
recovered = imwarp(distorted,tform,'OutputView',outputView);
figure, imshowpair(original,recovered,'montage')

[a1img,h1img,v1img,d1img] = dwt2(watermarked_image_uint8,'haar');
[a2img,h2img,v2img,d2img] = dwt2(a1img,'haar');

%% matching
for i = length(new_temp):-1:length(new_temp) - top+1
    data(top + (i - length(new_temp))) = a2img(new_temp(i,2),new_temp(i,3));
end
disp(data)
disp(value(2:length(value)))
%% display
imshow(watermarked_image_uint8)
image(watermarked_image_uint8);
subplot(1,2,1)
image(img);
title('Original')
subplot(1,2,2)
image(watermarked_image_uint8);
title('Watermarked')

```

6. References

- 1) 임용순, 강은영 (2012) Watermarking and QR code in Digital Image Coding: 102-104
- 2) 정영빈; (2008) Invisible Digital Watermarking Scheme Based on JPEG-2000: 4-12
- 3) 성광진, 김민태, 이종혁, 지인호, 안홍영; (2008) Digital Watermarking using Bit Concealment in Wavelet Transform Domain: 2-4
- 4) JPEG-2000 : <https://jpeg.org/jpeg2000/index.html>