

CS 754 : Advanced Image Processing - Assignment 1

Karan Taneja - 15D070022

Sucheta Ravikanti- 1600401001

Contents

- [2 \(c\) Haar wavelet for barbara256 image](#)
- [2 \(c\) Haar wavelet for barbara256 image: loop over patches](#)
- [2 \(c\) Haar wavelet for barbara256 image: results](#)
- [2 \(c\) Haar wavelet for goldhill image](#)
- [2 \(c\) Haar wavelet for goldhill image: loop over patches](#)
- [2 \(d\) Reconstruction of x from y and h: preparing x, y, h](#)
- [2 \(d\) Reconstruction of x from y and h: A matrix and ISTA](#)
- [2 \(d\) Reconstruction of x from y and h: results](#)

2 (c) Haar wavelet for barbara256 image

```
clear;
clc;

sigma = sqrt(4);
image = double(imread('barbara256.png'));
noise = randn(size(image))*sigma;
image = image + noise;
[height, width] = size(image);

imshow(image/max(image, [], 'all'));
title("Noisy image")
hold off

sensing_matrix = randn(32, 64);
haar_matrix = get_haar_matrix(8);
A_matrix = sensing_matrix * haar_matrix';

patch_size = 8; patch_stride = patch_size;
padded_height = height+2*patch_stride;
padded_width = width+2*patch_stride;
alpha = eigs(A_matrix' * A_matrix, 1) * 2;
lambda = 1;
epsilon = 1e-2;

padded_image = zeros(padded_height, padded_width);
padded_image(patch_stride+1:end-patch_stride, ...
    patch_stride+1:end-patch_stride) = image;
```

Noisy image



2 (c) Haar wavelet for barbara256 image: loop over patches

```
padded_image_recon = zeros(padded_height, padded_width);

tic
for phid = 1:patch_stride:padded_height-patch_stride
    for pwid = 1:patch_stride:padded_width-patch_stride
        %         sprintf("x = %d, y = %d", phid, pwid)
        current_xi = padded_image(phid:phid+patch_size-1, ...
                                   pwid:pwid+patch_size-1);
        current_xi = current_xi(:);
        current_yi = sensing_matrix * current_xi;

        current_theta_i_recon = ista_reconstruction(current_yi, ...
            A_matrix, alpha, lambda, epsilon);
        current_xi_recon = (haar_matrix') * current_theta_i_recon;
        current_xi_recon = reshape(current_xi_recon, patch_size, ...
            patch_size);
        %         curr_rrmse = norm(current_xi(:)-current_xi_recon(:)) / ...
        %         norm(current_xi(:))
        padded_image_recon(phid:phid+patch_size-1, ...
            pwid:pwid+patch_size-1) = padded_image_recon(phid: ...
            phid+patch_size-1, pwid:pwid+patch_size-1) + current_xi_recon;
        fid = fopen('log.txt', 'a+');
        fprintf(fid, 'x = %d y = %d \n', phid, pwid);
        fclose(fid);
    end
end
toc
```

Elapsed time is 16.840579 seconds.

2 (c) Haar wavelet for barbara256 image: results

```
padded_image_recon = padded_image_recon ./ 4;
image_recon = padded_image_recon(patch_stride+1:end-patch_stride, ...
```

```

        patch_stride+1:end-patch_stride);

imshow(image_recon/max(image_recon, [], 'all'));
title("Reconstructed Image")
hold off

rrmse = norm(image(:)-image_recon(:)) / norm(image(:));
rrmse

```

```

rrmse =

    0.8874

```



2 (c) Haar wavelet for goldhill image

```

clear;
clc;

sigma = sqrt(4);
image = double(imread('goldhill.png'));
image = image(1:256, 1:256);
noise = randn(size(image))*sigma;
image = image + noise;
[height, width] = size(image);

imshow(image/max(image, [], 'all'));
title("Noisy image")
hold off

sensing_matrix = randn(32, 64);
haar_matrix = get_haar_matrix(8);
A_matrix = sensing_matrix * haar_matrix';

patch_size = 8; patch_stride = patch_size;
padded_height = height+2*patch_stride;

```

```

padded_width = width+2*patch_stride;
alpha = eigs(A_matrix' * A_matrix, 1) * 1.05;
lambda = 1;
epsilon = 1e-2;

padded_image = zeros(padded_height, padded_width);
padded_image(patch_stride+1:end-patch_stride, ...
    patch_stride+1:end-patch_stride) = image;

```

Noisy image



2 (c) Haar wavelet for goldhill image: loop over patches

```

padded_image_recon = zeros(padded_height, padded_width);

tic
for phid = 1:patch_stride:padded_height-patch_stride
    for pwid = 1:patch_stride:padded_width-patch_stride
        % sprintf("x = %d, y = %d", phid, pwid)
        current_xi = padded_image(phid:phid+patch_size-1, ...
            pwid:pwid+patch_size-1);
        current_xi = current_xi(:);
        current_yi = sensing_matrix * current_xi;

        current_theta_i_recon = ista_reconstruction(current_yi, ...
            A_matrix, alpha, lambda, epsilon);
        current_xi_recon = get_idwt2(current_theta_i_recon);
        current_xi_recon = reshape(current_xi_recon, patch_size, ...
            patch_size);
        % curr_rrmse = norm(current_xi(:)-current_xi_recon(:)) / ...
        % norm(current_xi(:))
        padded_image_recon(phid:phid+patch_size-1, ...
            pwid:pwid+patch_size-1) = padded_image_recon(phid: ...
            phid+patch_size-1, pwid:pwid+patch_size-1) + current_xi_recon;
        fid = fopen('log.txt', 'a+');
        fprintf(fid, 'x = %d y = %d \n', phid, pwid);
        fclose(fid);
    end
end

```

```
toc
```

```
padded_image_recon = padded_image_recon ./ 4;  
image_recon = padded_image_recon(patch_stride+1:end-patch_stride, ...  
    patch_stride+1:end-patch_stride);
```

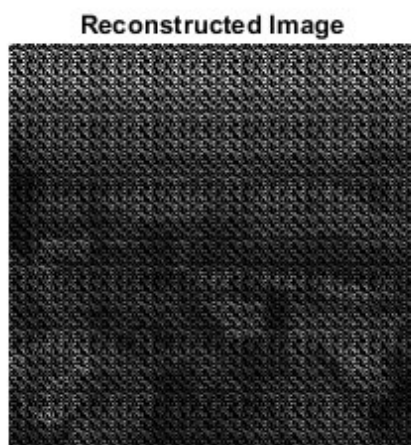
```
imshow(image_recon/max(image_recon, [], 'all'));  
title("Reconstructed Image")  
hold off
```

```
rrmse = norm(image(:)-image_recon(:)) / norm(image(:));  
rrmse
```

Elapsed time is 21.142274 seconds.

```
rrmse =
```

```
0.8863
```



2 (d) Reconstruction of x from y and h: preparing x, y, h

```
clear;  
clc;  
  
x = zeros(100, 1);  
non_zero_values = rand(10, 1)*2;  
non_zero_indices = randsample(100, 10);  
x(non_zero_indices, 1) = non_zero_values;  
  
h = [1 2 3 4 3 2 1]';  
h = h ./ 16;  
h_len = size(h, 1);  
  
noise = randn(100, 1) .* norm(x)/10 .* 0.05;
```

```
y = conv(h, x);  
y = y(1:100);  
y = y + noise;
```

2 (d) Reconstruction of x from y and h: A matrix and ISTA

```
sensing_matrix = zeros(100, 100);  
for i = 1:100  
    max_index = min(i+h_len-1, 100);  
    sensing_matrix(i:max_index,i) = h(1:max_index-i+1,1);  
end  
sparsity_transform = eye(100); % just an identity matrix  
  
A_matrix = sensing_matrix*sparsity_transform';  
alpha = eigs(A_matrix' * A_matrix, 1)*2;  
lambda = 1e-2;  
  
theta_recon = ista_reconstruction(y, A_matrix, alpha, lambda, 1e-5);  
x_recon = sparsity_transform' * theta_recon;
```

2 (d) Reconstruction of x from y and h: results

```
rrmse = norm(x - x_recon) / norm(x)  
  
figure  
  
subplot(3, 1, 1)  
stem(x)  
title("X")  
  
subplot(3, 1, 2)  
stem(y)  
title("Y")  
  
subplot(3, 1, 3)  
stem(x_recon)  
title("X Reconstructed")
```

```
rrmse =  
  
    0.1628
```

