Table of Contents

JPEG2000-Style Compression (Manual Haar DWT - SIMPLIFIED)

Author : Sandesh Jat MATLAB Online compatible – NO Wavelet Toolbox Simplified 1-level DWT version

```
clc; clear; close all;
```

1. Read Image

Use a test image if 'cat.png' is not available img = imread('peppers.png'); % Alternative test image

```
img = imread("Img.png");
figure; imshow(img); title("Original RGB Image");
```

2. RGB to YCbCr

```
ycbcr = rgb2ycbcr(img);
Y  = double(ycbcr(:,:,1));
Cb = ycbcr(:,:,2);
Cr = ycbcr(:,:,3);
```

3. Ensure Even Dimensions

```
[H,W] = size(Y);
H = floor(H/2)*2;   % needed for 1-level DWT
W = floor(W/2)*2;

Y  = Y(1:H,1:W);
Cb = Cb(1:H,1:W);
Cr = Cr(1:H,1:W);
```

4. Chroma Subsampling (4:2:0)

```
Cb_ds = Cb(1:2:end,1:2:end);
Cr_ds = Cr(1:2:end,1:2:end);
```

==================================

5. Haar Transform Matrices

==================================

Haar low-pass filter coefficients

```
h = [1 1]/sqrt(2);  % Normalized
% Haar high-pass filter coefficients
g = [1 -1]/sqrt(2);

% Create transformation matrices for rows (vertical transform)
P = H/2;
Haar_L_rows = kron(eye(P), h);
Haar_H_rows = kron(eye(P), g);

% Create transformation matrices for columns (horizontal transform)
Q = W/2;
Haar_L_cols = kron(eye(Q), h);
Haar_H_cols = kron(eye(Q), g);
```

==================================

6. 1-Level Haar DWT (2D Transform)

==================================

Apply vertical transform (rows)

```
Y_vertical_low = Haar_L_rows * Y;
Y_vertical_high = Haar_H_rows * Y;

% Apply horizontal transform (columns)
LL = Y_vertical_low * Haar_L_cols';
LH = Y_vertical_low * Haar_H_cols';
HL = Y_vertical_high * Haar_L_cols';
HH = Y_vertical_high * Haar_H_cols';
```

7. Thresholding (Compression Simulation)

```
T = 20;  % Threshold value

% Apply thresholding to detail coefficients
LH_thresh = LH;
HL_thresh = HL;
HH_thresh = HH;

LH_thresh(abs(LH) < T) = 0;
```

```
HL_thresh(abs(HL) < T) = 0;
HH_thresh(abs(HH) < T) = 0;
```

====================================

8. Inverse Haar Transform

====================================

Inverse horizontal transform

```
vertical_low_rec = LL * Haar_L_cols + LH_thresh * Haar_H_cols;
vertical_high_rec = HL_thresh * Haar_L_cols + HH_thresh * Haar_H_cols;

% Inverse vertical transform
Y_rec = Haar_L_rows' * vertical_low_rec + Haar_H_rows' * vertical_high_rec;

% Clip and convert to uint8
Y_rec = uint8(min(max(Y_rec, 0), 255));
```

9. Chroma Upsampling

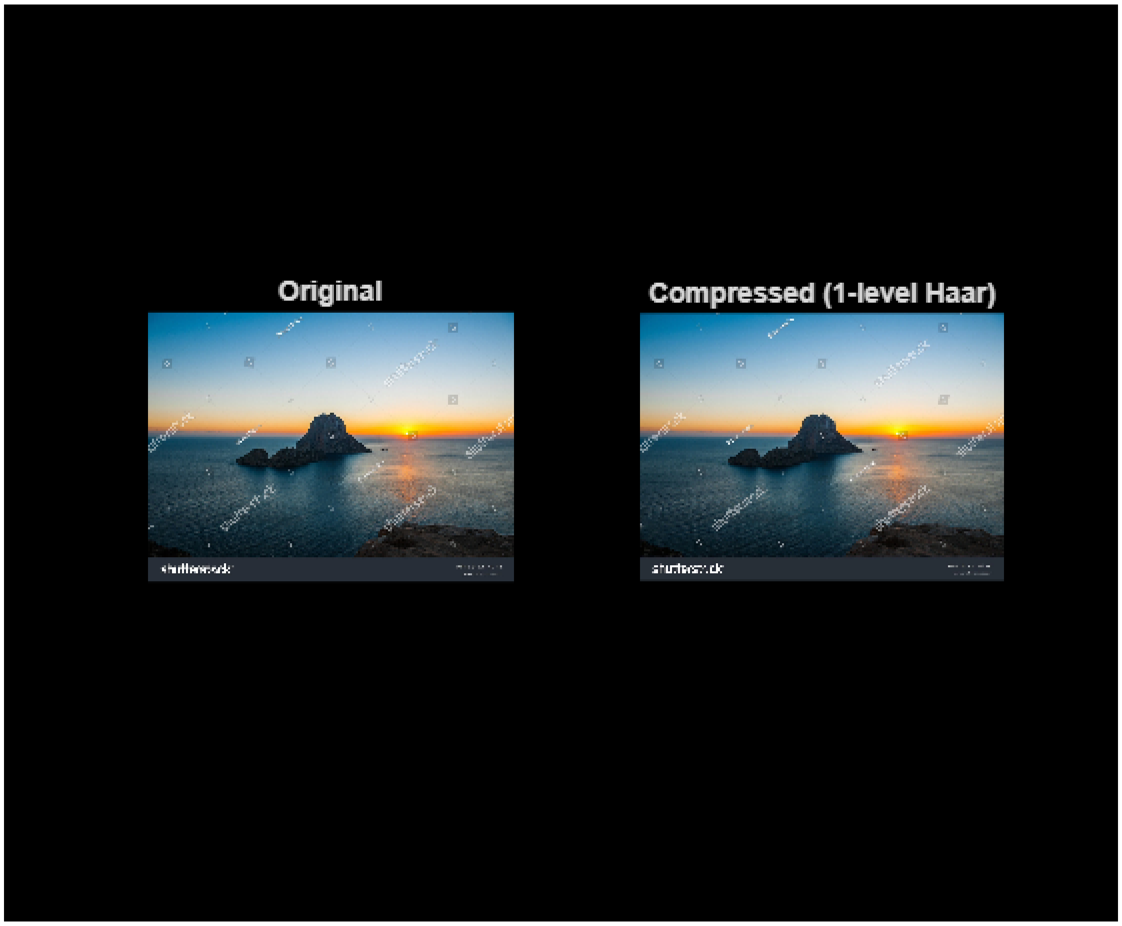
```
Cb_rec = uint8(imresize(Cb_ds, [H W], 'bilinear'));
Cr_rec = uint8(imresize(Cr_ds, [H W], 'bilinear'));
```

10. Reconstruct RGB Image

```
ycbcr_rec = cat(3, Y_rec, Cb_rec, Cr_rec);
img_rec = ycbcr2rgb(ycbcr_rec);
```

11. Display Results

```
figure;
subplot(1,2,1), imshow(img(1:H,1:W,:)), title("Original");
subplot(1,2,2), imshow(img_rec), title("Compressed (1-level Haar)");
```



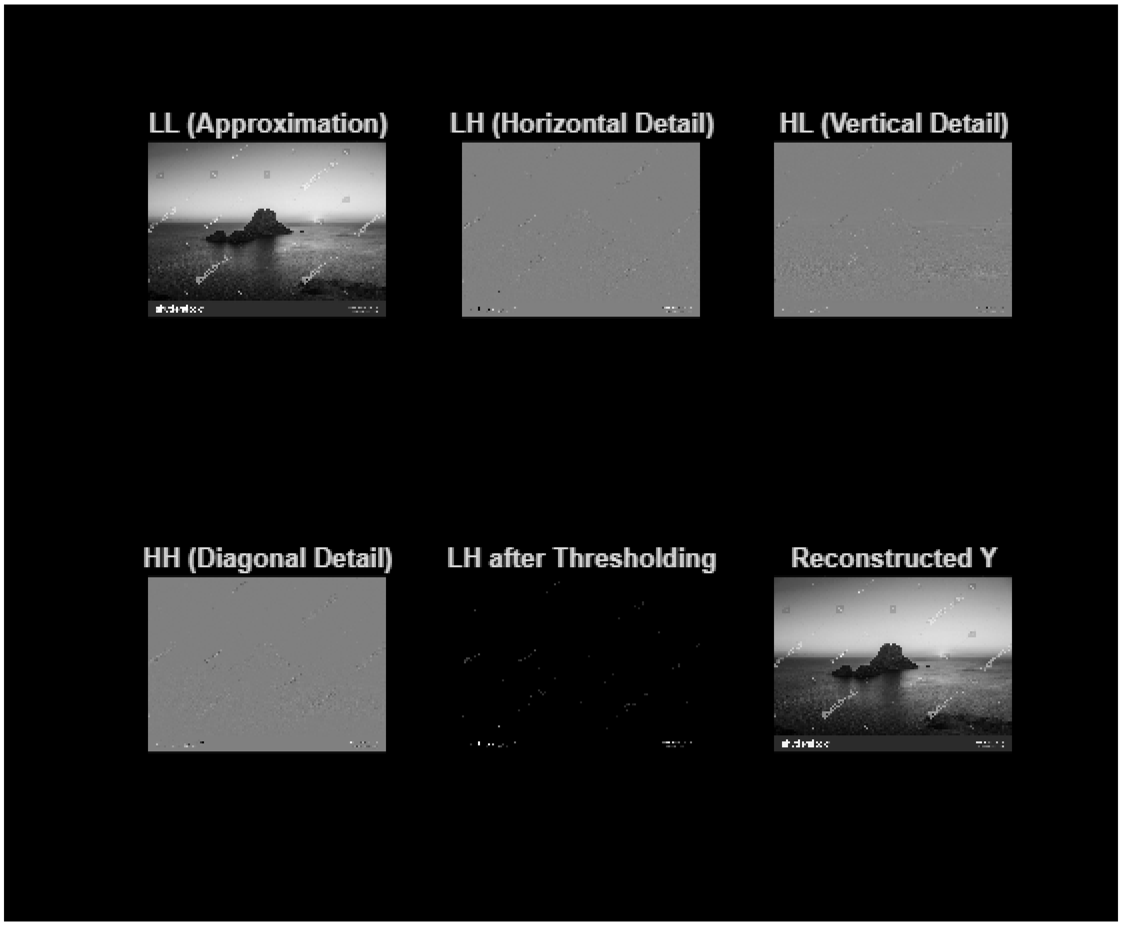
12. Quality Metrics

```
mseVal  = mean((double(img(1:H,1:W,:)) - double(img_rec)).^2, 'all');
psnrVal = 10*log10(255^2/mseVal);

fprintf("Compression Parameters:\n");
fprintf("  Threshold = %d\n", T);
fprintf("  Image Size = %d x %d\n", H, W);
fprintf("\nQuality Metrics:\n");
fprintf("  MSE  = %.4f\n", mseVal);
fprintf("  PSNR = %.2f dB\n", psnrVal);
```

```
Compression Parameters:
  Threshold = 20
  Image Size = 1100 x 1500

Quality Metrics:
  MSE  = 9.5230
  PSNR = 38.34 dB
```

13. Display DWT Subbands (Optional)

```
figure;
subplot(2,3,1), imshow(LL, []), title('LL (Approximation)');
subplot(2,3,2), imshow(LH, []), title('LH (Horizontal Detail)');
subplot(2,3,3), imshow(HL, []), title('HL (Vertical Detail)');
subplot(2,3,4), imshow(HH, []), title('HH (Diagonal Detail)');
subplot(2,3,5), imshow(abs(LH_thresh), []), title('LH after Thresholding');
subplot(2,3,6), imshow(Y_rec, []), title('Reconstructed Y');
```



14. Compression Ratio Estimation

Calculate how many coefficients were zeroed out

```
total_coeffs = numel(LH) + numel(HL) + numel(HH);
zeroed_coeffs = sum(LH_thresh(:) == 0) + sum(HL_thresh(:) == 0) +
sum(HH_thresh(:) == 0);
compression_ratio = total_coeffs / (total_coeffs - zeroed_coeffs);

fprintf("\nCompression Statistics:\n");
fprintf("  Total detail coefficients: %d\n", total_coeffs);
fprintf("  Zeroed coefficients: %d\n", zeroed_coeffs);
fprintf("  Compression ratio: %.2f:1\n", compression_ratio);
```

```
Compression Statistics:
  Total detail coefficients: 1237500
  Zeroed coefficients: 1226800
  Compression ratio: 115.65:1
```

Published with MATLAB® R2025b