

Theorem

Define

$$\psi(n) = \begin{cases} \frac{1}{2} & 1 \leq n \leq 2 \\ -\frac{1}{2} & 3 \leq n \leq 4 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and $\psi_{j,k}(x) = \psi(2^j x - k)$.

For a matrix $A \in M_8$,

$$H_1 = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

$$H_2 = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H = H_1 H_2 H_3, B = H^T A H$$

Choose a ϵ and turn every element in the matrix smaller than it into 0.

Denote this new matrix as B' , $A' = (H^T)^{-1}B'(H)^{-1}$

In this way, we successfully compress matrix A .

If H_1, H_2, H_3 is turned into orthogonal matrixes,
the compression will be of better quality.

Experiment Target

Utilize wavelet encoding method the compress image.

Simplification: In the experiment, we first convert the sample images into grayscale before compressing them. This approach reduces the amount of data and simplifies the processing, thereby enhancing the efficiency of compression.

Metrics for measuring the degree of image compression:

- 1 Comparison of image size before and after compression
- 2 Comparison of image file size before and after compression
- 3 Peak signal-to-noise ratio (PSNR)

- 1 **Initialization:** The original image is loaded into the program from a file. If the image is in color, it may be converted to grayscale to simplify the compression process.
- 2 **Preprocessing:** The image is processed row by row and column by column, applying the Haar wavelet transform. This step decomposes the image into a set of wavelet coefficients that represent the image in the wavelet domain.
- 3 **Quantization and Thresholding:** Wavelet coefficients are modified to compress the image data. Coefficients below a certain threshold are set to zero, effectively reducing the amount of data.
- 4 **Inverse Wavelet Transform:** An inverse wavelet transform is applied to the modified coefficients to reconstruct a compressed version of the original image.
- 5 **Performance Analysis:** After compression, metrics can be calculated to evaluate the quality and efficiency of the compression.

breaklines

```
function [approx, detail] = haar_wavelet_transform(signal)
    % Haar wavelet transformation function for one-dimensional signals

    N = length(signal);
    if mod(N, 2) ~= 0
        error('The length of the signal must be even.');
```

end

% Initialize coefficients

```
approx = zeros(1, N/2);
detail = zeros(1, N/2);

% Haar filters
for i = 1:2:N-1
    approx((i+1)/2) = (signal(i) + signal(i+1)) / sqrt(2);
    detail((i+1)/2) = (signal(i) - signal(i+1)) / sqrt(2);
end
end
```

```
breaklines
```

```
function signal = ihaar_wavelet_transform(approx, detail)
```

```
% Inverse Haar wavelet transformation for one-dimensional signals
```

```
% Approx and Detail must be the same length
```

```
if length(approx) ~= length(detail)
```

```
    error('Approximation and detail coefficients must be of the same length.
```

```
end
```

```
N = length(approx) * 2;
```

```
signal = zeros(1, N);
```

```
% Reconstruct the signal from approximation and detail coefficients
```

```
for i = 1:length(approx)
```

```
    signal(2*i-1) = (approx(i) + detail(i)) / sqrt(2);
```

```
    signal(2*i) = (approx(i) - detail(i)) / sqrt(2);
```

```
end
```

```
end
```



Original image

Image Size: 65536 bytes

Image File Size: 43632 bytes



Threshold: 20

PSNR value: 33.4615 dB

Image Memory Size: 65536 bytes

Image File Size: 25706 bytes

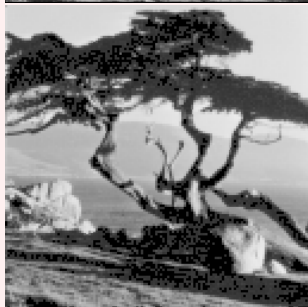


Threshold: 40

PSNR value: 29.6200 dB

Image Memory Size: 65536 bytes

Image File Size: 20288 bytes



Threshold: 100

PSNR value: 22.1378 dB

Image Memory Size: 65536 bytes

Image File Size: 15667 bytes

Reference:

- 1 [Wavelat Transformation, Wikipedia](#)
- 2 [Haar Wavelat, Wikipedia](#)

Complete code can be found in the following [Github repository](#)

The image displays a collection of 12 word clouds, each representing a different area of mathematics. The word clouds are arranged in a 3x4 grid. The top row contains 'vector', 'transpose', 'dimension', and 'basis'. The second row contains 'solution', 'matrix', 'set', and 'kernel'. The third row contains 'isomorphism', 'rank', 'linear map', 'matrix', 'basis', 'vector', 'group', 'row', 'transpose', and 'algebraic structures'.

Thank you!