Wavelet Transform for Image Processing

Zhang Jinrui[1]
Mo Zian

October 21, 2025

[1] alternative email:zhangjr1022@mails.jlu.edu.cn

For a 2^N points sequence, the mother wavelet is

$$\psi(n) = \begin{cases} 0, & \text{if } n < 0 \\ \frac{1}{\sqrt{2^N}}, & \text{if } 0 \leq n < 2^{N-1} \\ \frac{-1}{\sqrt{2^N}}, & \text{if } 2^{N-1} \leq n < 2^N \\ 0, & \text{if } n >= 2^N \end{cases}$$

We have

$$\langle \psi, \psi \rangle = \sum_{k=0}^{2^N-1} \frac{1}{2^N} = 1$$

For other derived basis we use this discretized formula

$$\psi_{j,k}(n) = \sqrt{2^j}\psi(2^j(n - 2^{N-j}k)), 0 \le j \le N-1, k <= 2^j - 1$$

$$\begin{bmatrix} \psi_{0,0} & 0 & \cdots & 0 \\ \psi_{1,0} & \psi_{1,1} & \cdots & 0 \\ \psi_{2,0} & \cdots & \psi_{2,3} & 0 \\ \vdots & \vdots & \ddots & 0 \\ \psi_{N-1,0} & \psi_{N-1,1} & \cdots & \psi_{N-1,2^{N-1}-1} \end{bmatrix}$$

And a average basis as following

$$\phi(n) = \begin{cases} 0, & \text{if } n < 0 \\ \frac{1}{\sqrt{2^N}}, & \text{if } 0 \le n < 2^{N-1} \\ \frac{1}{\sqrt{2^N}}, & \text{if } 2^{N-1} \le n < 2^N \\ 0, & \text{if } n >= 2^N \end{cases}$$

there are

$$1 + \sum_{k=0}^{N-1} 2^k = 2^N$$

basis. which will be 2^N coefficients then every 2^N sequence $f(n)$ can be written as

$$f(n) = a\phi(n) + \sum_{j=0}^{N-1} \sum_{k=0}^{2^j-1} c_{j,k}\psi_{j,k}$$

Then each coefficients can be obtain by inner product.

$$a = \langle f(n), \phi(n) \rangle$$

$$c_{j,k} = \langle f(n), \psi_{j,k}(n) \rangle$$

The inner product of two sequence are define as follow.

$$\langle f(n), g(n) \rangle = \sum_{k=0}^{2^N-1} f(k)g(k)$$

the transformed sequence $\hat{f}(n)$ satisfies

$$c_{j,k} = \hat{f}(2^j + k)$$

$$a = \hat{f}(0)$$

Image compression

Let's see an application with Haar wavelet first.

By our research, $\forall\ m \in \mathbb{N}_+$, we can get **an set of orthonormal bases** of $\mathbb{R}^m$ quickly.

Theorem 2.1

Let $\{u_i\}_{i=1}^m$ is a set of **orthonormal bases** of $\mathbb{R}^m$, $\{v_i\}_{i=1}^n$ is a set of **orthonormal bases** of $\mathbb{R}^n$, then

$$\{u_i \times v_j \mid i = 1, \cdots m, j = 1, \cdots n\}$$

is a set of **orthonormal bases** of $\mathbb{R}^{m \times n}$.

Image compression

A grayscale image is stored as a grayscale matrix in the computer. Then we only need to know how to "compress" a matrix.

For $A \in \mathbb{R}^{m \times n}$, if $\{u_i\}, \{v_j\}$ are sets of **orthonormal bases** of $\mathbb{R}^m$ and $\mathbb{R}^n$, according to (*Theorem 2.1*), $\exists\ \omega_{i,j} \in \mathbb{R}, i = 1, \cdots m, j = 1, \cdots, n$,

$$A = \sum_{i,j} \omega_{i,j} \ \cdot \ u_i \cdot v_j^T.$$

Let $M = (u_1, u_2, \cdots, u_m),\ N = (v_1, v_2, \cdots, v_n), W = (\omega_{i,j})$. Then

$$M^T A N = W,\ MWN^T = A.$$

Image compression

If $\omega_{i,j}$ is too small, the influence of corresponding basis can be ignored. Then

$$\widetilde{A} = \sum_{i,j} \widetilde{\omega_{i,j}} \cdot u_i \cdot v_j^T, \ \widetilde{\omega_{i,j}} = \begin{cases} \omega_{i,j}, & |\omega_{i,j}| > \lambda \\ 0, & |\omega_{i,j}| \le \lambda \end{cases}.$$

$\widetilde{A}$ is an approximation of A, while λ is image compression strength.

Let $\widetilde{W} = (\widetilde{\omega_{i,j}})$. As long as λ is large enough, we can transform $\widetilde{W}$ into a **sparse matrix** which needs less storage space.

Then we set λ equal to 0.5 and see the result.

Image compression

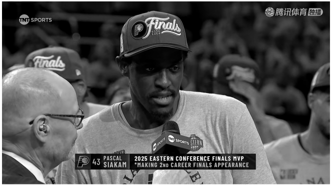

Figure: Original image

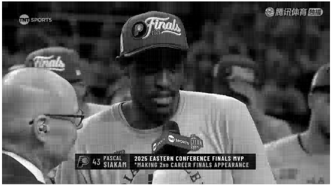

Figure: Final image

Image compression

For original image , the size of stored information is
$981 \times 1759 = 1725579$.

However, after compression, the size of stored information is only
$15364 \times 2 = 30728$.

Indeed, we reduce storage space by compression. However, we also lose much detailed information of original picture. That's why we should change λ for different purposes to reach the balance.