Wavelets for image processing

group member:
Jingrui Zhang, Jiashun He, Jingyuan Meng, Zishen Jiang, Zian Mo

August 3, 2025

Outline

Image compression

Image compression

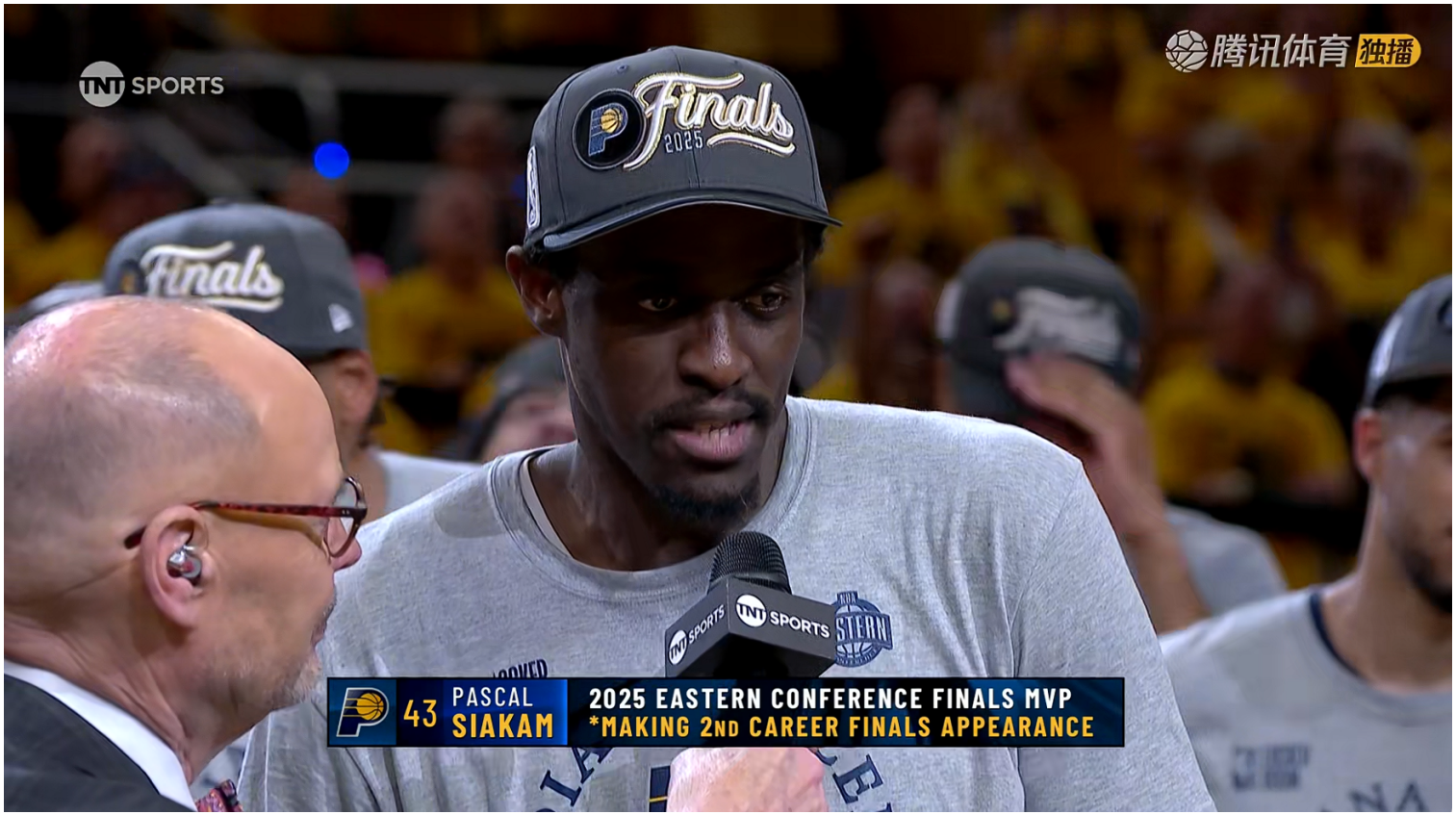


Figure: Siakam won 2025 NBA eastern conference finals MVP

Image compression

The picture is stored as RGB array in computer.

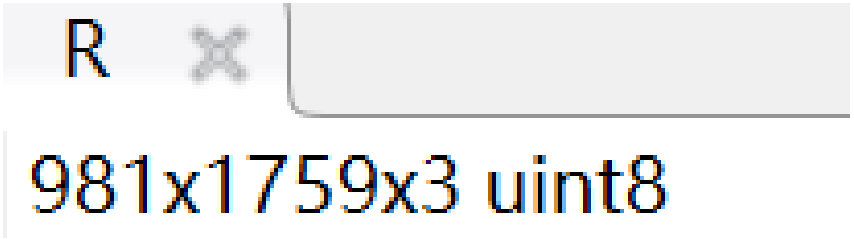


Figure: Using MATLAB function "imread" to get its corresponding RGB array

The corresponding RGB array is quite Huge!

Image compression

Our Purpose: Compress the image to reduce storage space

For convenience, we convert the color image into a grayscale image.

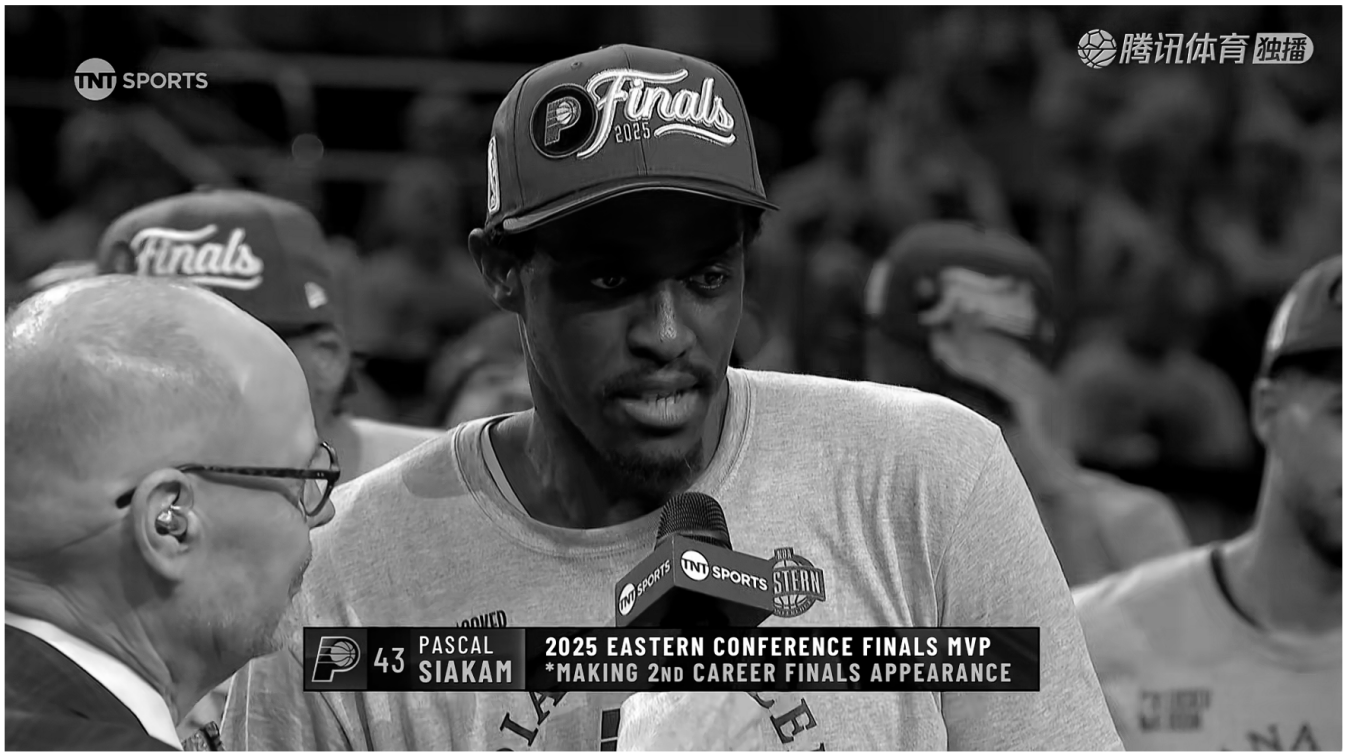


Figure: Using MATLAB function "rgb2gray" to convert the image

Image compression

The RGB array can be viewed as a combination of three matrices, while a grayscale image is only decided by a Grayscale matrix.

⇓⇓⇓⇓

We can compress a color image
as long as we can compress a grayscale image.

Image compression

The RGB array can be viewed as a combination of three matrices, while a grayscale image is only decided by a Grayscale matrix.

We can compress a color image
as long as we can compress a grayscale image.

Then, how to do that?

Image compression

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 $\mathbb{R}^{m \times n}$.

Meanwhile, for an an integer *num*, if

$$\exists\ q \in \mathbb{N}_+,\ num = 2^q, \tag{2.1}$$

we can get a set of orthonormal bases of $\mathbb{R}^{num}$ using following matlab code.

Image compression

Get a set of orthonormal bases of $\mathbb{R}^{num}$(matlab)

```
function AnsM = basevecM_D(num) % Unitization
    AnsM = orthvecM_D(num);
    len = size(AnsM, 1);
    for i = 1:len
        AnsM(i,:) = AnsM(i,:)./sqrt(AnsM(i,:)*(AnsM(i,:)'));
    end
end

function AnsM = orthvecM_D(num) % Get an orthogonal basis
    AnsM = orthvecM_D1(num);
    AnsM = [ones(1, num); AnsM];
end

function AnsM = orthvecM_D1(num)

if num == 2
    AnsM = [1, -1];
else
    tempM1 = orthvecM_D1(num/2);
    tempM2 = zeros(size(tempM1, 1), num/2);
    tempM3 = ones(1, num/2);
    AnsM = [tempM3, -tempM3; tempM1, tempM2; tempM2, tempM1];
end
end
```

Image compression

Then if both m and n satisty the condition (2.1), we can get sets of **orthonormal bases** of $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively. $(\{u_i\}, \{v_j\})$

Therefore, $\forall\ A \in \mathbb{R}^{m\times n},\ \exists\ \omega_{i,j} \in \mathbb{R}, i = 1, \cdots m, j = 1, \cdots, n$

$$A = \sum_{i,j} \omega_{i,j}\ \cdot\ u_i \cdot v_j^T. \tag{2.2}$$

Let $W = (\omega_{i,j})$. We can covert W to A by (2.2), which means we only need to store W.

Image compression

How can we get W?

As $\{u_i\}, \{v_j\}$ are sets of **orthonormal bases** of $\mathbb{R}^m$ and $\mathbb{R}^n$,

$$u_a^T A v_b = \sum_{i,j} \omega_{i,j} \ \cdot \ (u_a^T u_i) \cdot (v_j^T v_b) = \omega_{a,b}.$$

Let $M = (u_1, u_2, \cdots, u_m),\ N = (v_1, v_2, \cdots, v_n)$. Then

$$M^T A N = W.$$

Meanwhile, since M, N are **orthogonal matrices**,

$$MWN^T = (MM^T) \cdot A \cdot (N^T N) = A.$$

Image compression

How can we compress the image?

Formula (2.2) indicates the basis linear representation of A:

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

The coefficient $\omega_{i,j}$ reflects **the influence of corresponding basis** $u_i \cdot v_j^T$ on the final result A.

Image compression

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

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

$\widetilde{A}$ 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.

Meanwhile, we only need to save $\widetilde{W}$ to store $\widetilde{A}$.

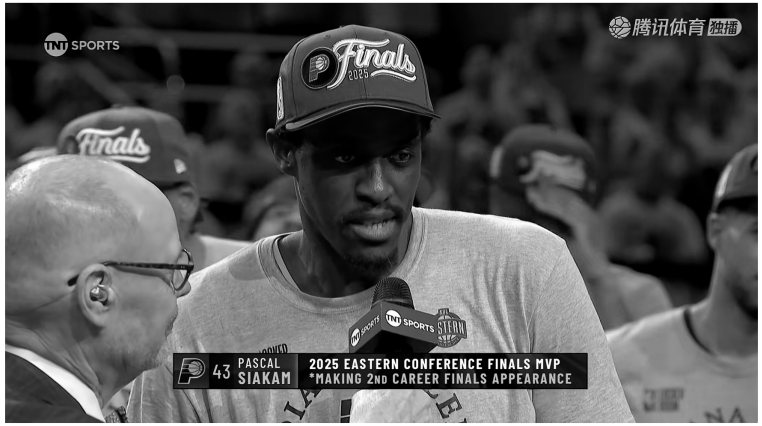
Main Process

1. Get basis matrixs of $\mathbb{R}^m$ *and* $\mathbb{R}^n$. (M,N)
2. Use $M^T AN = W$ to get coefficient matrix W.
3. Compress matrix W to sparse matrix $\widetilde{W}$.

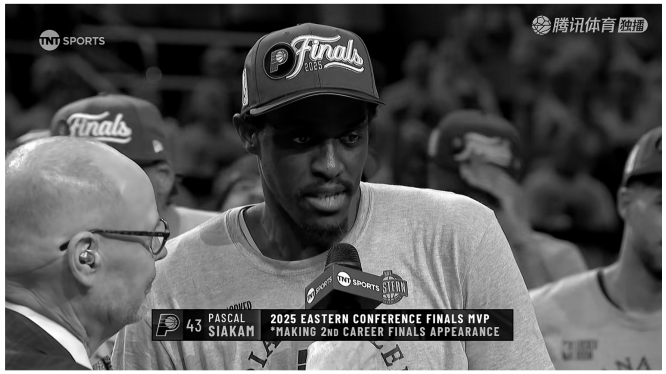
However, the method to get basis matrixs can be used **only if** m,n satisfy the condition (2.1). Commonly, this is not true. We usually **expand** the original image so that the final expanded image meets this condition and the expanded part is all white.

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

Image compression

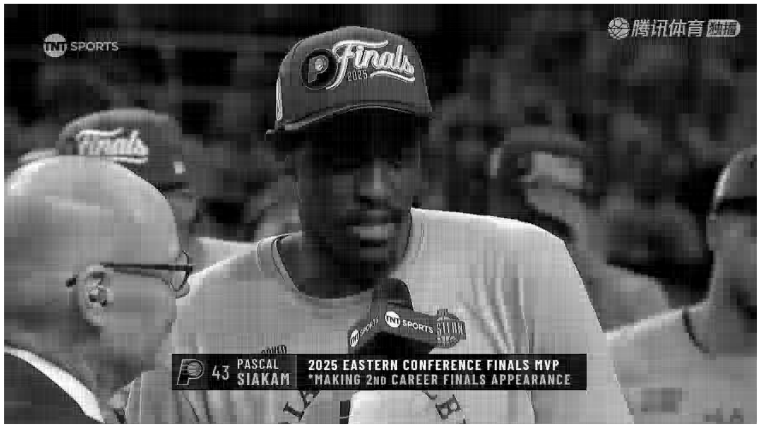


(a) Original image



(b) Expanded image

(c) Compressed expanded image



(d) 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.