This week

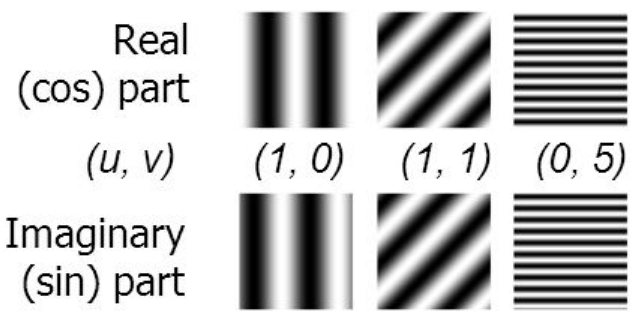
Continue on Fourier Analysis

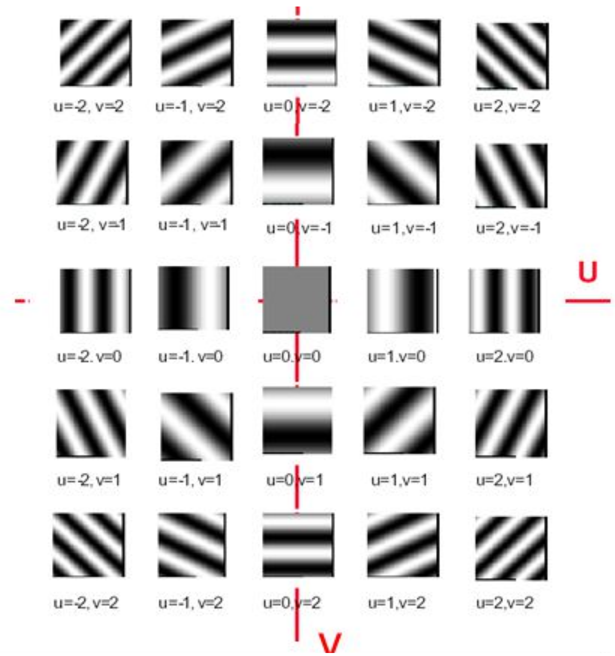
Review on PCA

Wavelet Transform

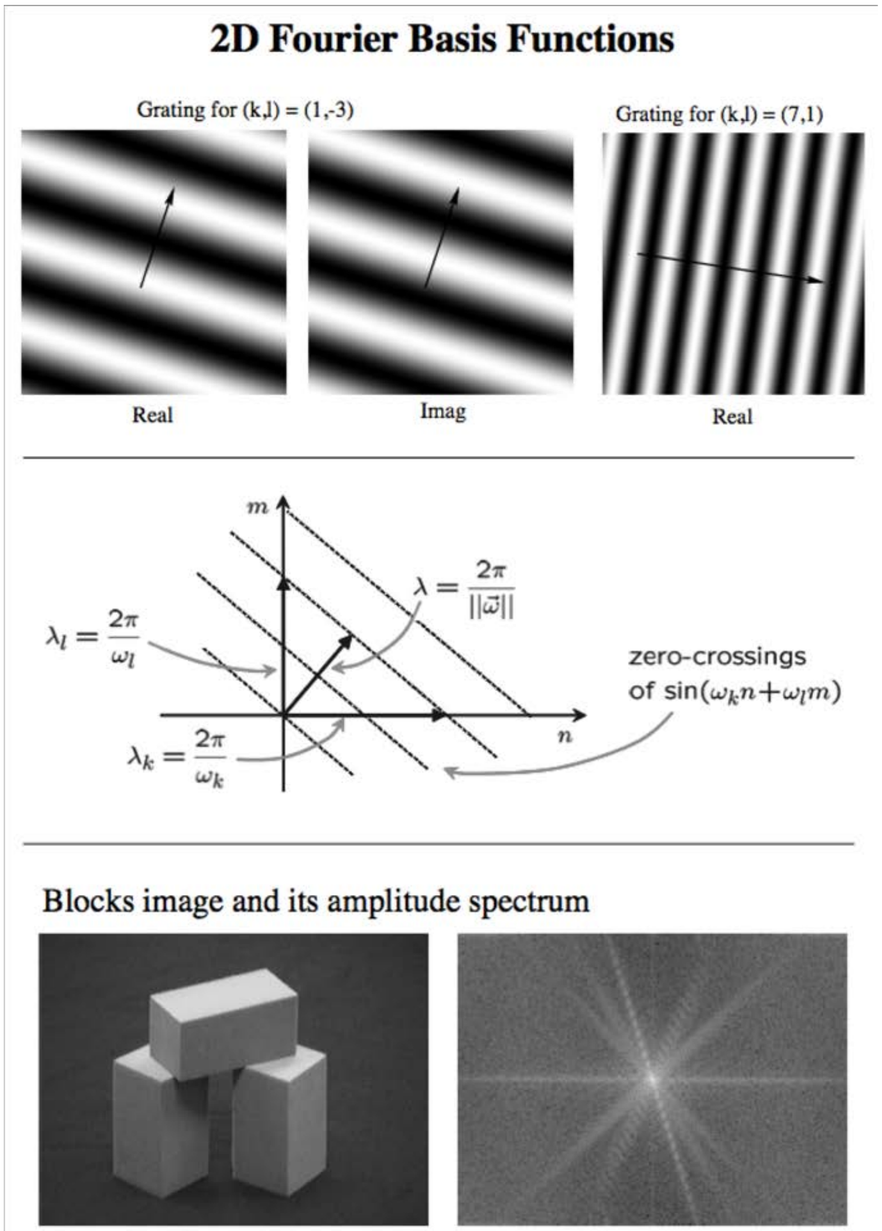
Fourier Transform as a Basis

$$\exp[j\, 2\pi(ux + vy)] = \cos[2\pi(ux + vy)] + j \sin[2\pi(ux + vy)]$$





Example 2D Fourier basis functions



Convolution Theorem

Let $\mathcal{F}\{f\}$ respresents the Fourier transform of f

If $\mathcal{F}\{\mathrm{f}\} = F$, then $\mathcal{F}^{-1}\{\mathrm{F}\} = f$

1. $\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$ or $f * g = \mathcal{F}^{-1}\left\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\right\}$

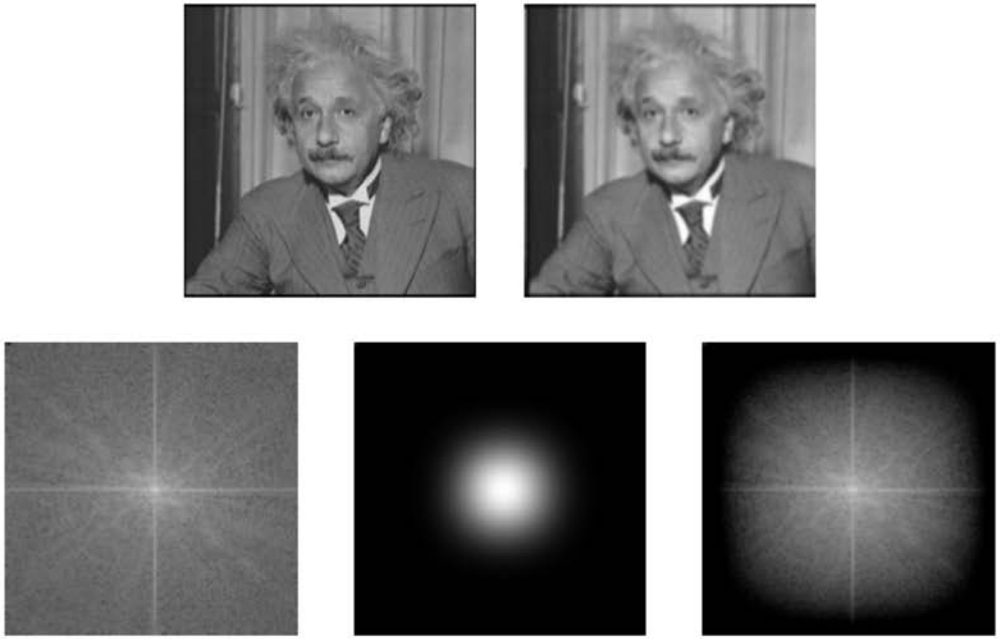
   F.T. of **Convolution** in the **spatial domain** is the same as **multiplication** in the **frequency domain**

2. $\mathcal{F}\{f \cdot g\} = \mathcal{F}\{f\} * \mathcal{F}\{g\}$ or $f \cdot g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} * \mathcal{F}\{g\}\}$

   F.T. of **Multiplication** in the **spatial domain** is the same as **convolution** in the **frequency domain**

Examples of 2D filters & their spectra

Common Filters and their Spectra



Top Row: Image of Al and a **low-pass** (blurred) version of it. The low-pass kernel was separable, composed of 5-tap 1D impulse responses $\frac{1}{16}(1, 4, 6, 4, 1)$ in the x and y directions.

Bottom Row: From left to right are the amplitude spectrum of Al, the amplitude spectrum of the impulse response, and the product of the two amplitude spectra, which is the amplitude spectrum of the blurred version of Al. (Brightness in the left and right images is proportional to log amplitude.)

Examples of 2D filters & their spectra

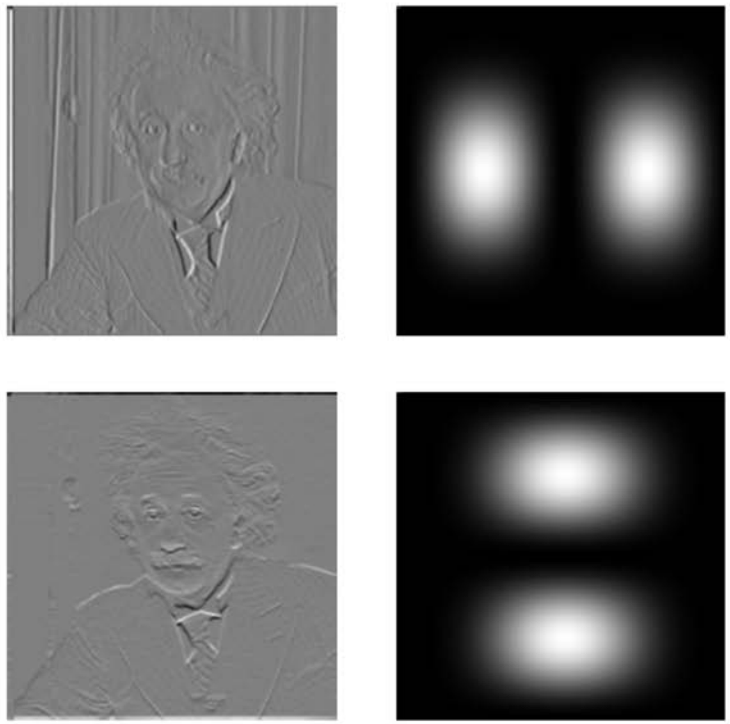
Common Filters and their Spectra (cont)

From left to right is the original Al, a **high-pass** filtered version of Al, and the amplitude spectrum of the filter. This impulse response is defined by $\delta(n) - h(n, m)$ where $h[n, m]$ is the separable blurring kernel used in the previous figure.

From left to right is the original Al, a **band-pass** filtered version of Al, and the amplitude spectrum of the filter. This impulse response is defined by the difference of two low-pass filters.

Examples of 2D filters & their spectra

Common Filters and their Spectra (cont)



Top Row: Convolution of Al with a horizontal derivative filter, along with the filter's Fourier spectrum. The 2D separable filter is composed of a vertical smoothing filter (i.e., $\frac{1}{4}(1, 2, 1)$) and a first-order central difference (i.e., $\frac{1}{2}(-1, 0, 1)$) horizontally.

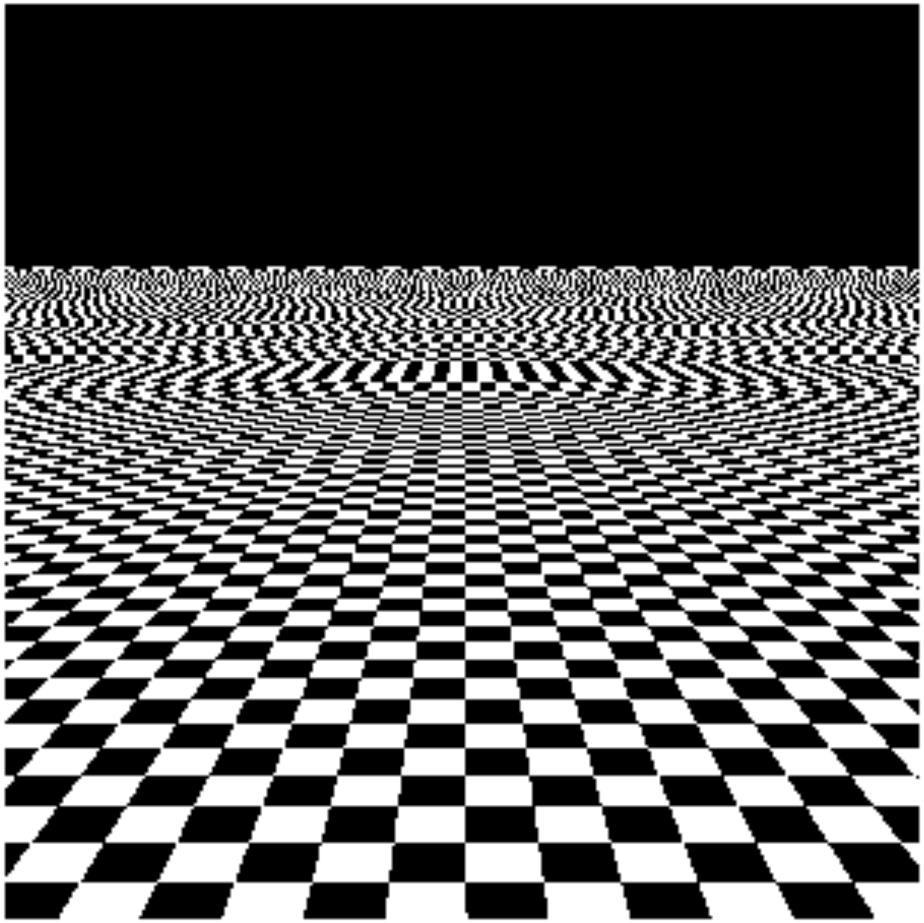
Bottom Row: Convolution of Al with a vertical derivative filter, and the filter's Fourier spectrum. The filter is composed of a horizontal smoothing filter and a vertical first-order central difference.

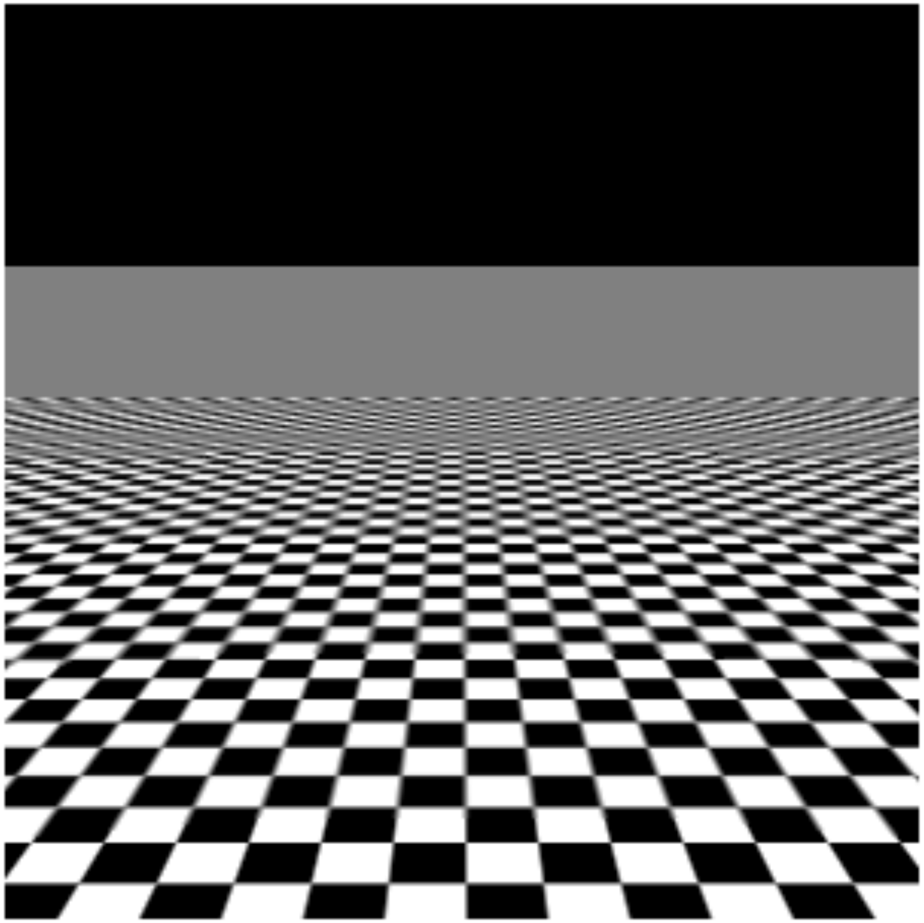
Linear Filters & Fourier Analysis

Tutorial
week 11

- Filter-based view of image formation
- Linear systems & 1D convolution
- Example 1D filters
- Filtering in 2D
- The Fourier series
- **Introduction to sampling & aliasing**
- Discrete-time filters & the DFT

Aliasing

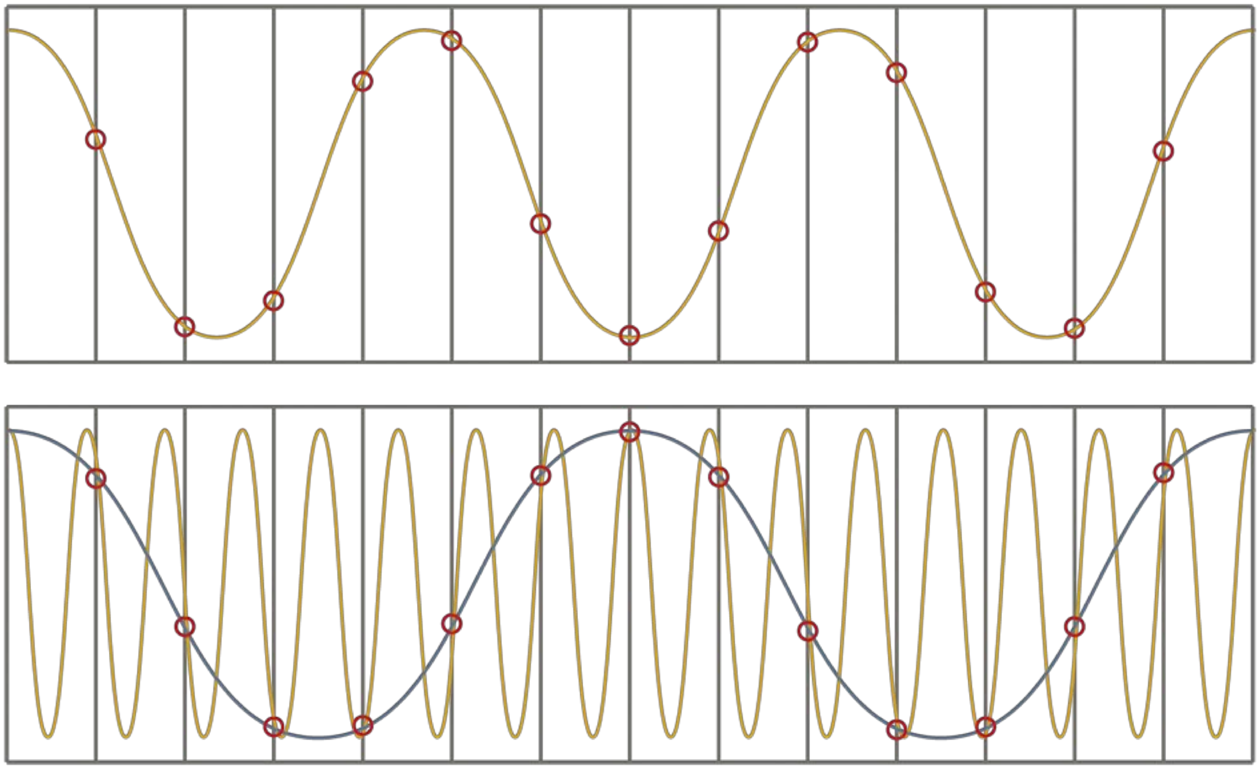




https://commons.wikimedia.org/wiki/File:Mipmapping_example.png

Problem with sampling

Nyquist–Shannon sampling theorem

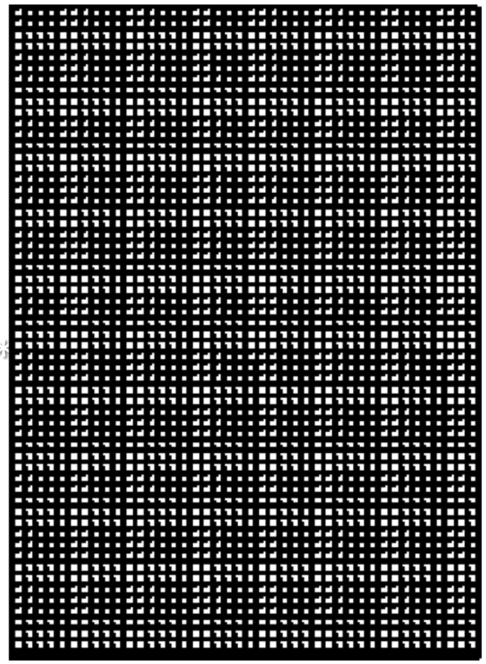


https://electronics.stackexchange.com/questions/235870/sampling-anti-aliasing-filter-bandwidth

Image formation from a filtering perspective

$B(x,y)$

Impulse train



Sampled image

$D(x,y) = B(x,y) \cdot S(x,y)$

$S(x,y) = 1$ iff x, y integer multiples of pixel pitch $\Delta x, \Delta y$ & zero otherwise

Blur due to lens & finite pixel footprint

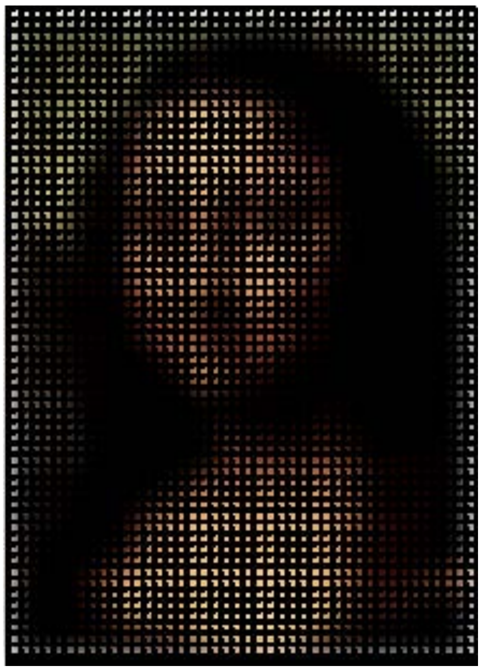
$B(x,y) = (I * f)(x,y)$

Ideal sensor irradiance

$I(x,y) \quad (x,y) \in [-\pi,\pi] \times [-\pi,\pi]$

Image formation from a filtering perspective

$D(x,y)$



Sampled image

sampling & aliasing

$$D(x,y) = B(x,y) \cdot S(x,y)$$

$S(x,y) = 1$ iff x, y integer multiples of pixel pitch $\Delta x, \Delta y$ & zero otherwise

filtering & convolution

Blur due to lens & finite pixel footprint

$$B(x,y) = (I * f)(x,y)$$

Ideal sensor irradiance

$$I(x,y) \quad (x,y) \in [-\pi,\pi] \times [-\pi,\pi]$$

goal #1: understanding image aliasing...

close-up photo of my office carpet (looks pretty good)

goal #1: understanding image aliasing...

why do fringes appear in this image??

Sampling in spatial vs. Fourier domain

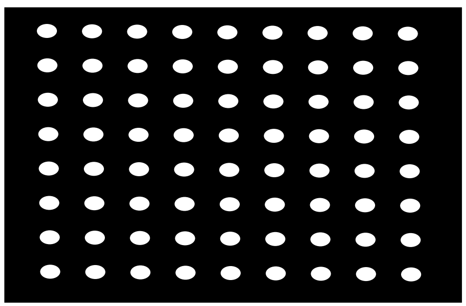
Let us consider the sampling operation in the Fourier Transform domain

$$\sum_{k=-N}^{N}\sum_{\ell=-M}^{M}\delta\left(x-\frac{\pi}{N}k\right)\delta\left(y-\frac{\pi}{M}\ell\right)$$

|| *

2N samples

2M samples

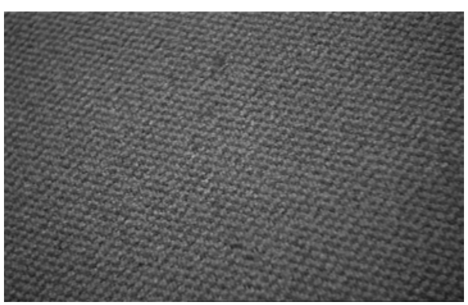


$S(x,y)$

$F\{S\}$

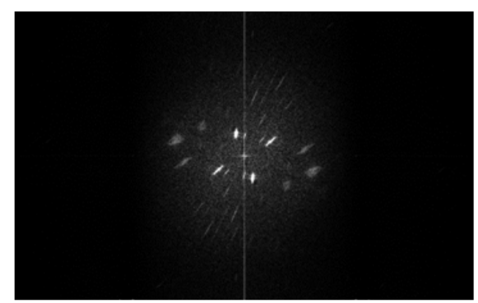
×

$*$



$B(x,y)$

$F\{B\}$



Sampling in spatial vs. Fourier domain

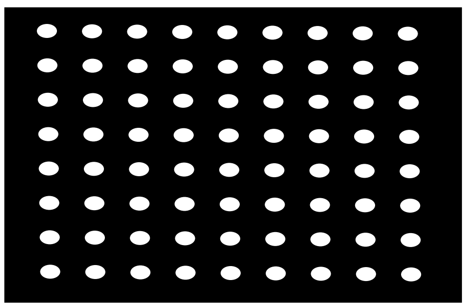
Let us consider the sampling operation in the Fourier Transform domain

$$\sum_{k=-N}^{N}\sum_{l=-M}^{M}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\delta\left(x-\frac{\pi}{N}k\right)\delta\left(y-\frac{\pi}{M}l\right)e^{-j(xn+ym)}\,dx\,dy$$

$*$ ||

2N samples

2M samples



$S(x,y)$

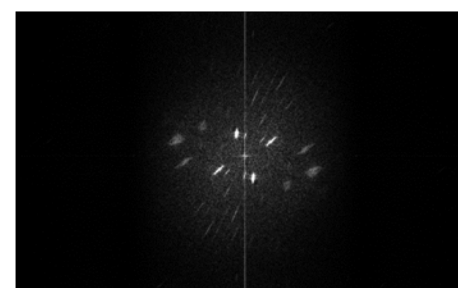
$F\{S\}$

$\times$

$*$

$B(x,y)$

$F\{B\}$



Sampling in spatial vs. Fourier domain

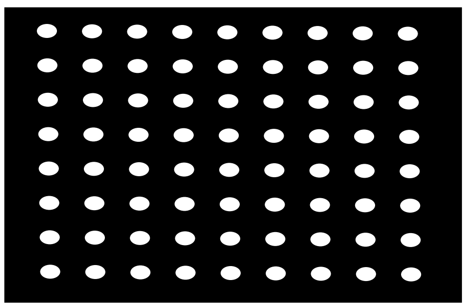
Sample distances in spatial domain inversely proportional to Fourier domain

$$\sum_{k=-N}^{N}\sum_{l=-M}^{M}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\delta\left(x-\frac{\pi}{N}k\right)\delta\left(y-\frac{\pi}{M}l\right)e^{-j(xn+ym)}\,dx\,dy$$

$=$

2N samples

2M samples

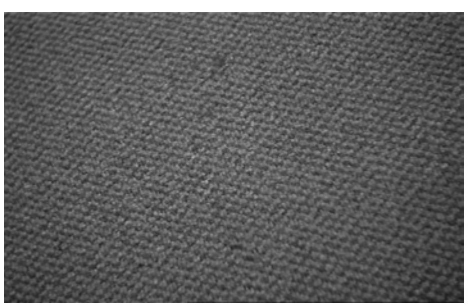


$S(x,y)$

$\frac{\pi}{M}$

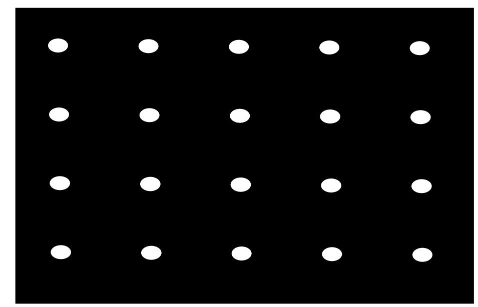
$\frac{\pi}{N}$

$\times$



$B(x,y)$

$F\{S\}$

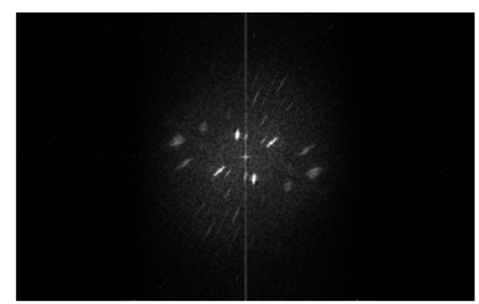


$\frac{M}{\pi}$

$\frac{N}{\pi}$

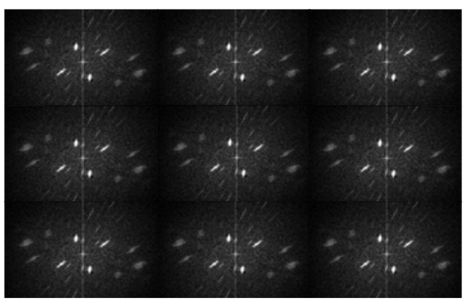
$*$

$F\{B\}$



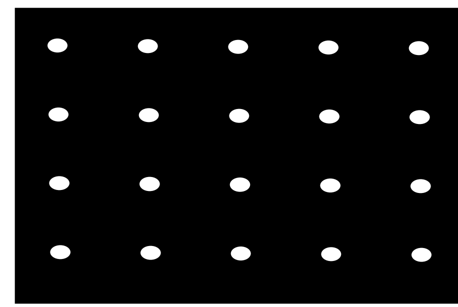
Convolution with an impulse train

a superposition of shifted copies of $\mathcal{F}\{B\}$



$\Longleftarrow$

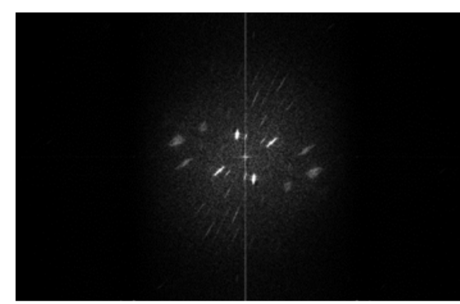
$\mathcal{F}\{S\}$



$\frac{M}{n}$ $\frac{N}{n}$

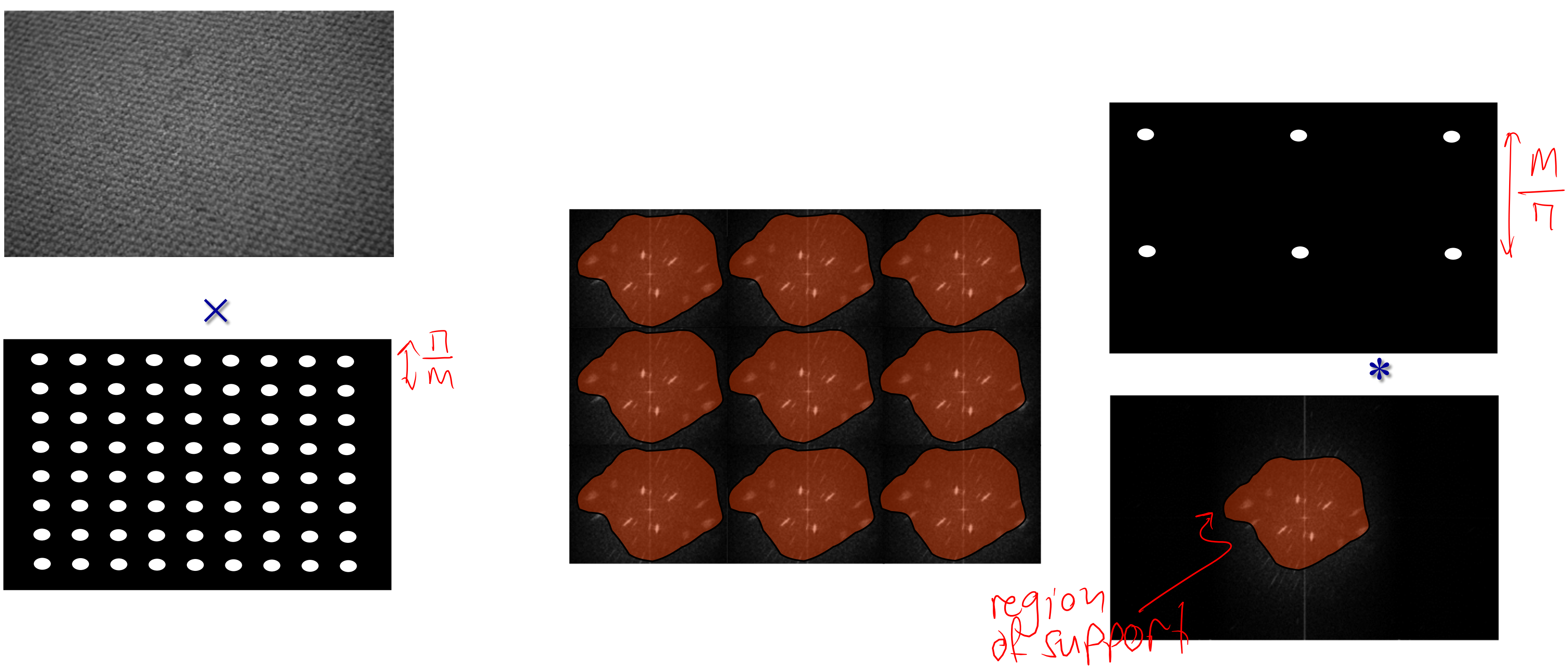
$*$

$\mathcal{F}\{B\}$



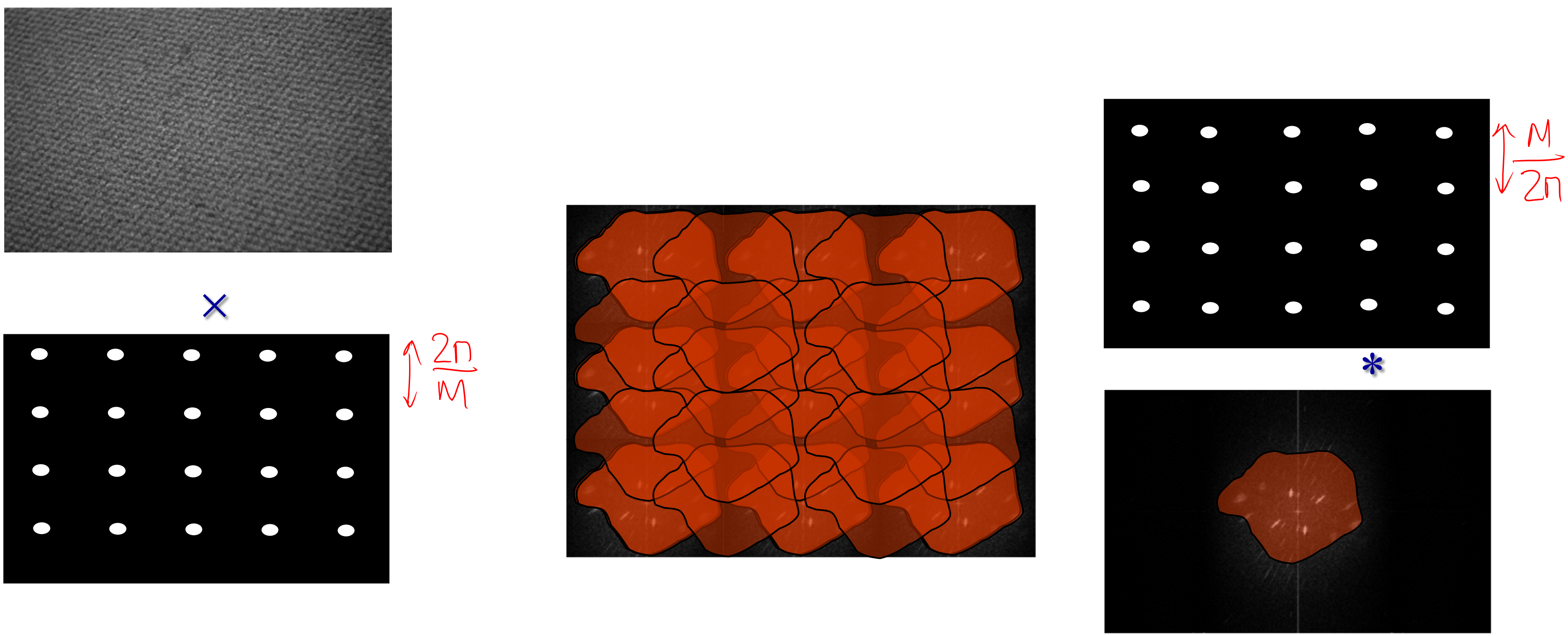
Sampling in spatial vs. Fourier domain

When region of support of $F\{B\}$ does not overlap with adjacent copies it is possible to reconstruct B from $F^{-1}\{F\{B \cdot S\}\}$



Aliasing

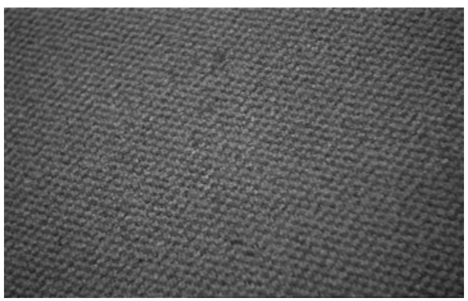
If overlap occurs in $F\{B \cdot S\}$ not possible to reconstruct $F\{B\}$ any longer



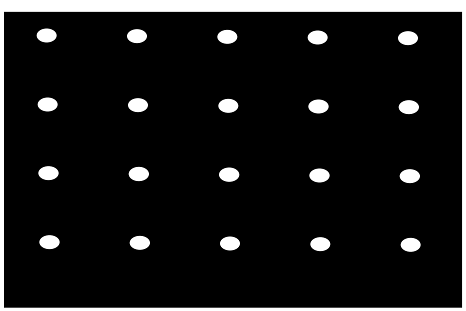
Nyquist Sampling Theorem

Let $f(x)$ be a band-limited signal such that $F_w\{f\}=0$ for $|w|>w_0$

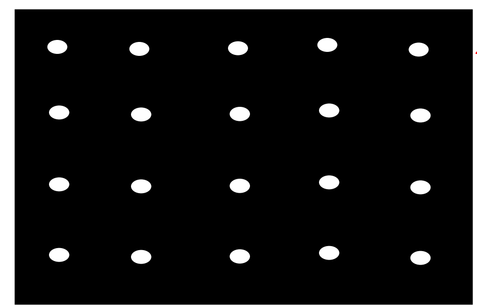
* Then f uniquely determined by its samples $g(n)=f(n\cdot\Delta x)$ when $\frac{2\pi}{\Delta x}>2w_0$



$\times$

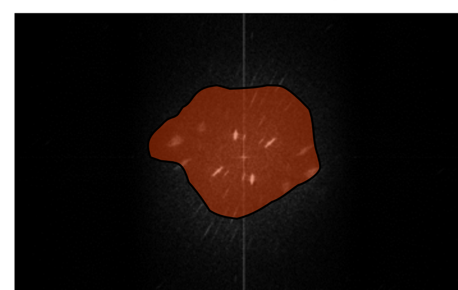


$\frac{2\pi}{M}$



$\frac{M}{2\pi}$

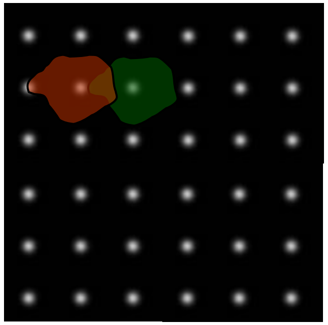
$*$



The Source of Aliasing…

The end result

- High frequencies are “masked” (aliased) as lower frequencies



Topic 9:

Discrete Wavelet Transform

- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

Review

Fourier Analysis

- Frequency domain
- Predefined basis

Principle Component Analysis

- Reduce dimension
- Calculate basis from data set

Laplacian Pyramid Representation (next week)

- Transformation stores the pixel difference
- Overcomplete representation

Wavelet vs. other transformations

Predefined basis

– similar to Fourier transform

Reduce dimension (compression algorithm)

– similar to PCA

Transformation stores the pixel difference

– similar to Laplacian pyramid representation

Find the principle components

Given N image patches of M dimensions:

1) Calculate mean of image patch vectors

$$\bar{\boldsymbol{X}} = \frac{1}{N}\sum \boldsymbol{X}_i$$

2) Subtract the mean from all patches (centre)

$$\boldsymbol{Z}_i = \boldsymbol{X}_i - \bar{\boldsymbol{X}}$$

3) Create an $M \times N$ matrix of all centred patch vectors (arranged as columns of matrix)

$$Z = [\boldsymbol{Z}_1 \quad \boldsymbol{Z}_2 \cdots \quad \boldsymbol{Z}_N]$$

4) Find ***eigenvectors*** $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d$ corresponding to the d (where $d \ll M$) largest ***eigenvalues*** $\lambda_1, \ldots, \lambda_d$ of the **covariance matrix**

$$\Sigma = ZZ^T$$

Reminder: The Eigenface/PCA Image Basis

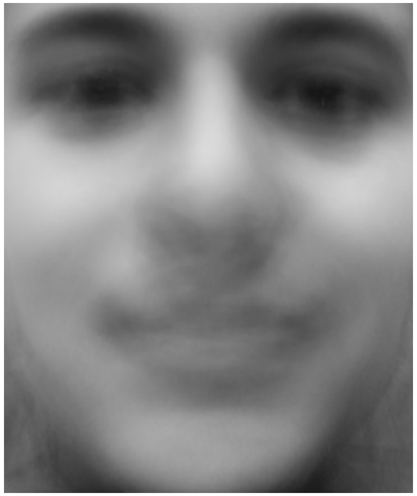
x_1 (M dimensions)

$\simeq$

x_i (d-dimensional approx d=3)

$=$

$\bar{x}$



$+$

$y_1^1 *$

B_1

$+ \; y_1^2 *$

B_2

$+ \; y_1^3 *$

B_3

Representing Images by their PCA Basis

all pixels of one image

j-th pixel of all images

$$\begin{bmatrix} z_1^1 & z_2^1 & \cdots & z_N^1 \\ z_1^2 & z_2^2 & \cdots & z_N^2 \\ \vdots & \vdots & & \vdots \\ z_1^M & z_2^M & \cdots & z_N^M \end{bmatrix} =$$

$$[B_1\ B_2 \cdots B_M] \cdot \begin{array}{l} \\ \text{large} \\ \\ \text{near zero} \\ \\ \end{array} \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_N^1 \\ \vdots & \vdots & & \vdots \\ y_1^d & y_2^d & \cdots & y_N^d \\ y_1^{d+1} & y_2^{d+1} & \cdots & y_N^{d+1} \\ \vdots & \vdots & & \vdots \\ y_1^M & y_2^M & \cdots & y_N^M \end{bmatrix}$$

$\Longleftrightarrow$

eigenfaces

$$Z = B \cdot Y$$

mean-subtracted images

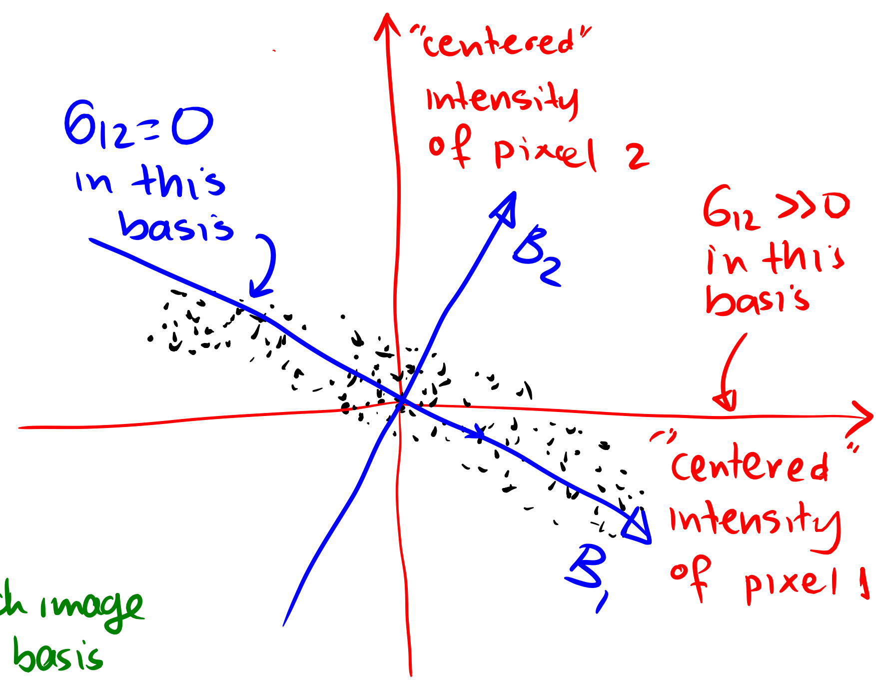
coordinates of each image in PCA/eigenface basis

- Image reconstruction:

$$\begin{bmatrix} x_i^1 \\ \vdots \\ x_i^M \end{bmatrix} = B \cdot \begin{bmatrix} y_i^1 \\ \vdots \\ y_i^M \end{bmatrix} + \bar{x}$$

- Image transform

$$[y_i] = B^T [x_i - \bar{x}]$$



The Discrete Wavelet Transform

all pixels of one image

j-th pixel of all images

$$\begin{bmatrix} x_1^1 & x_2^1 & & x_N^1 \\ x_1^2 & x_2^2 & & x_N^2 \\ \vdots & \vdots & & \vdots \\ x_1^M & x_2^M & & x_N^M \end{bmatrix} =$$

$$[B_1 \; B_2 \cdots B_M] \cdot$$

$$\begin{array}{c} \text{large} \left\{ \begin{array}{c} \\ \\ \end{array} \right. \\ \text{near zero} \left\{ \begin{array}{c} \\ \\ \end{array} \right. \end{array} \begin{bmatrix} y_1^1 & y_2^1 & & y_N^1 \\ y_1^d & y_2^d & & y_N^d \\ y_1^{d+1} & y_2^{d+1} & & y_N^{d+1} \\ \vdots & \vdots & & \vdots \\ y_1^M & y_2^M & & y_N^M \end{bmatrix}$$

$$\Longleftrightarrow$$

wavelet basis

$$X = B \cdot Y$$

wavelet coefficients

Image reconstruction:

$$(1) \quad \begin{bmatrix} x_i^1 \\ \vdots \\ x_i^M \end{bmatrix} = B \cdot \begin{bmatrix} y_i^1 \\ \vdots \\ y_i^M \end{bmatrix} + \bar{x}$$

Image transform

$$(2) \quad [y_i] = B^T [x_i - \bar{x}]$$

The (discrete) wavelet transform maps an image into yet another basis defined by a "special" matrix B:

- captures scale
- invertible, orthogonal, square
- image independent

The Discrete Wavelet Transform

all pixels of one image

j-th pixel

$$\begin{bmatrix} x_2^1 \\ x_2^2 \\ \vdots \\ x_2^M \end{bmatrix} = \begin{bmatrix} B_1 & B_2 & \cdots & B_M \end{bmatrix} \cdot \begin{bmatrix} y^1 \\ y^2 \\ y \\ \vdots \\ y^M \end{bmatrix}$$

coarse-scale details

fine-scale details

$$\Longleftrightarrow$$

wavelet basis

$$X = B \cdot Y$$

wavelet coefficients

Image reconstruction:

(1) $\begin{bmatrix} x_i^1 \\ \vdots \\ x_i^M \end{bmatrix} = B \cdot \begin{bmatrix} y_i^1 \\ \vdots \\ y_i^M \end{bmatrix} + \bar{x}$

Image transform

(2) $\left[y_i \right] = B^T \left[x_i - \bar{x} \right]$

The (discrete) wavelet transform maps an image into yet another basis defined by a "special" matrix B

- captures scale
- invertible, orthogonal, square
- image independent

Topic 9:

Discrete Wavelet Transform

- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

A Simple, Minimal 2-D Image Transform

Properties of transformation

- Minimal (no "wasted" pixels)
- Multiple scales represented simultaneously
- Invertible, linear

Input image ($2^N x 2^N$)

wavelet transform

Transformed image ($2^N x 2^N$)

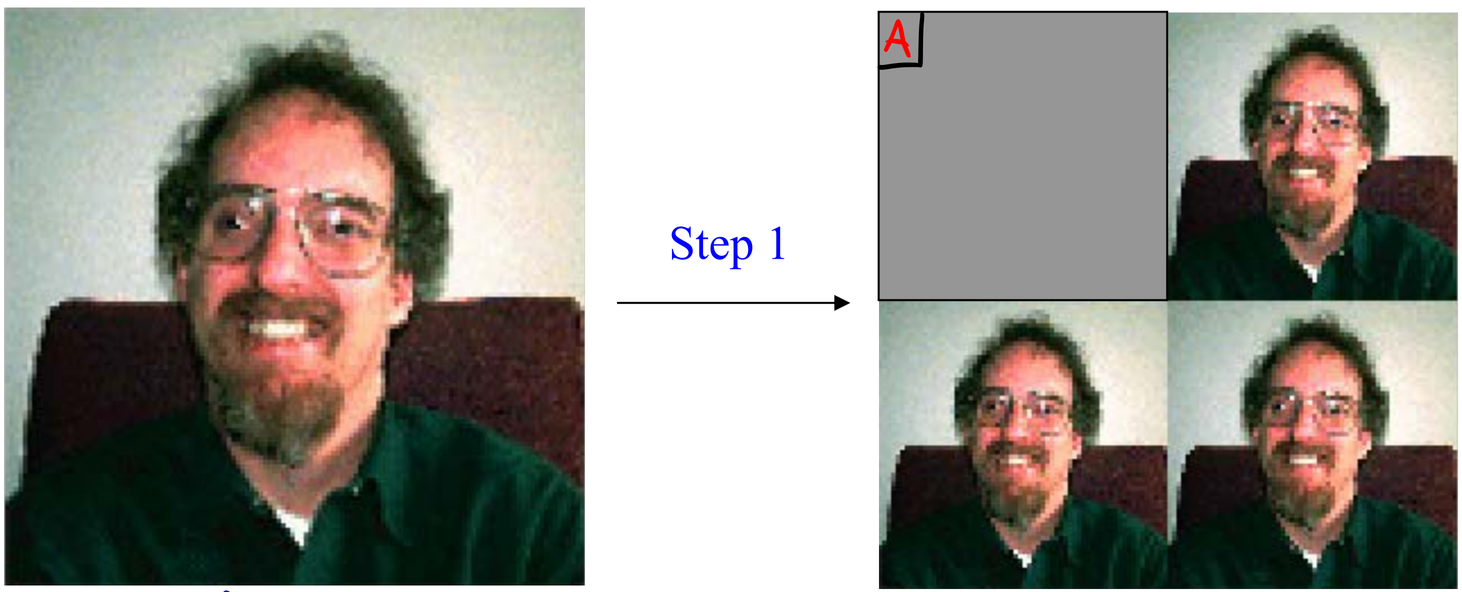
A Simple, Minimal 2-D Image Transform

Step 1: Create 4 new images of size $2^{N-1}x2^{N-1}$ as shown in figure

Input image (2^Nx2^N)

Transformed image (2^Nx2^N)

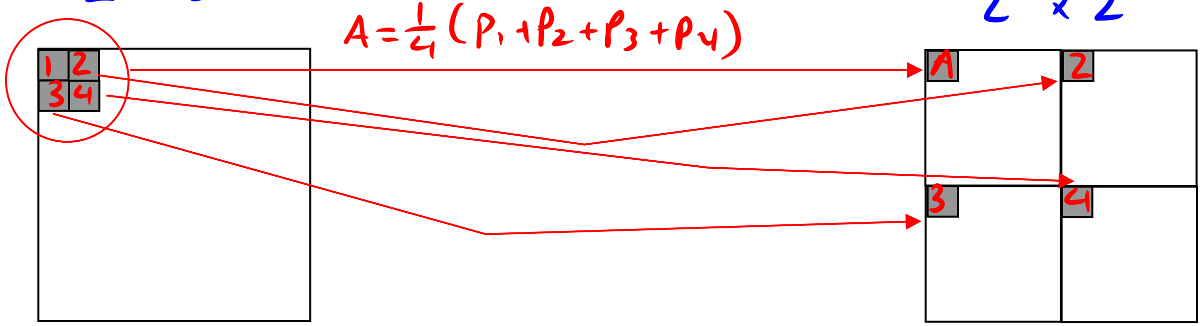
W_0:



$2^N \times 2^N$

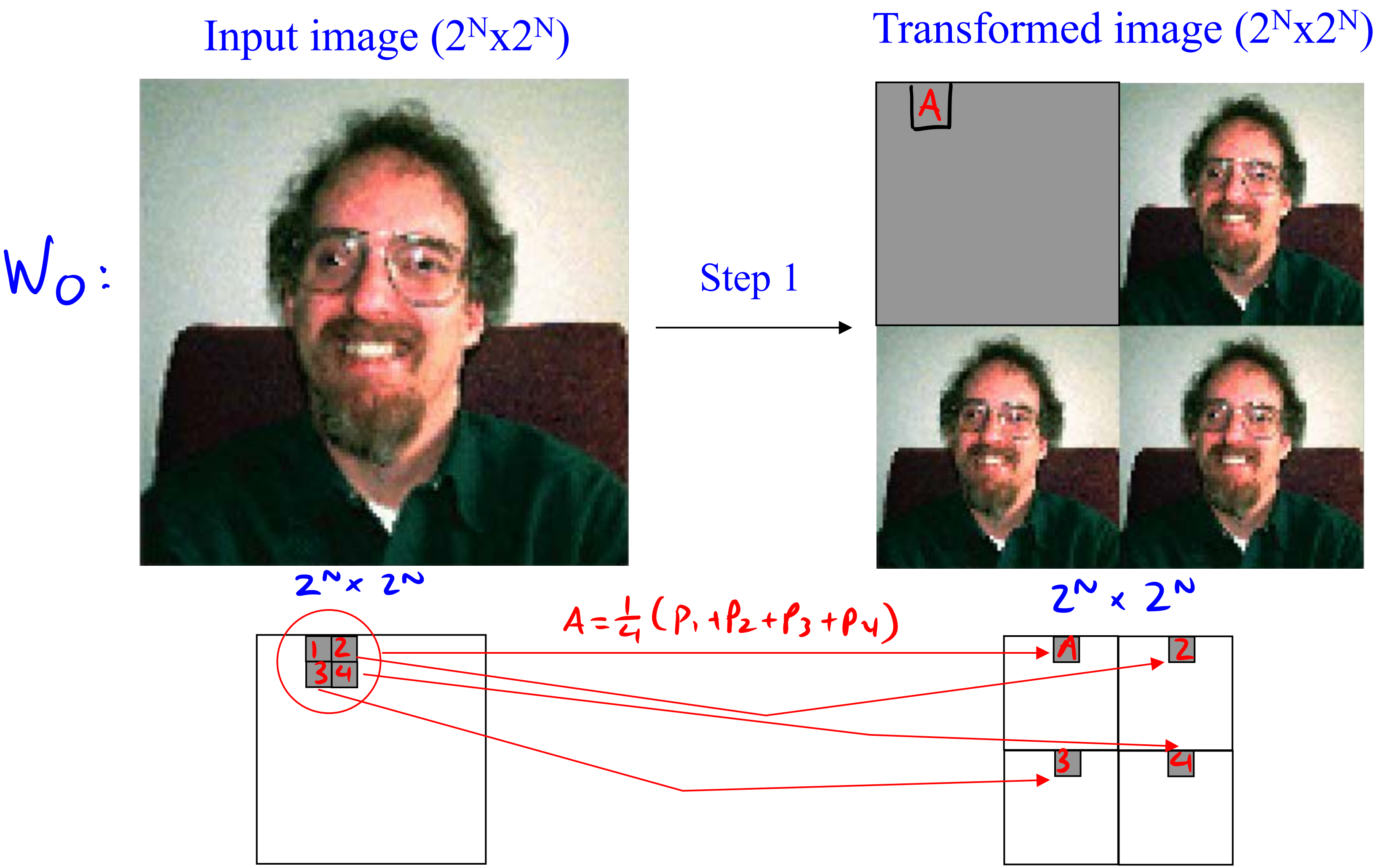
$2^N \times 2^N$

$A = \frac{1}{4}(p_1 + p_2 + p_3 + p_4)$



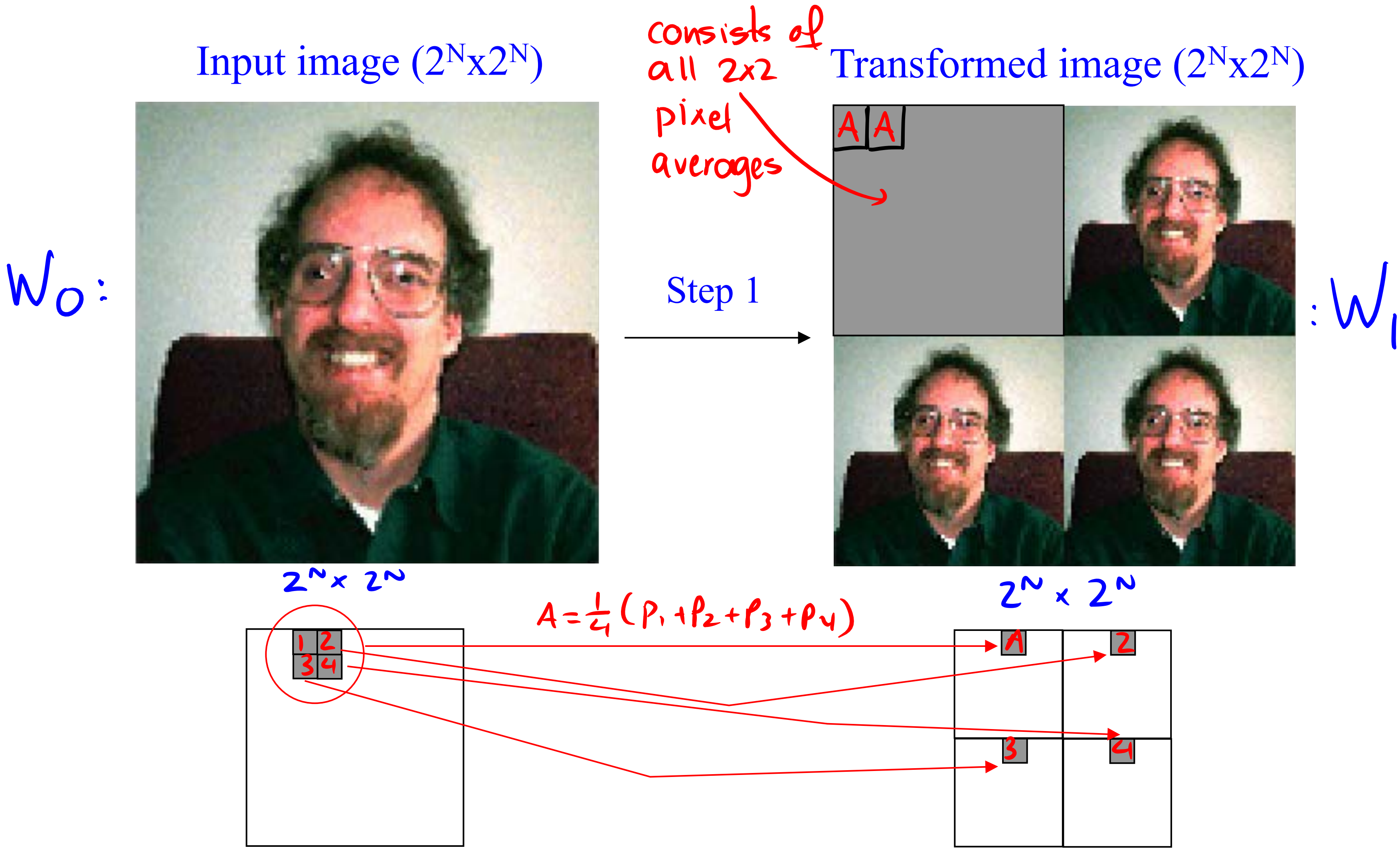
A Simple, Minimal 2-D Image Transform

Step 1: Create 4 new images of size $2^{N-1}x2^{N-1}$ as shown in figure



A Simple, Minimal 2-D Image Transform

Step 1: Create 4 new images of size $2^{N-1}x2^{N-1}$ as shown in figure



A Simple, Minimal 2-D Image Transform

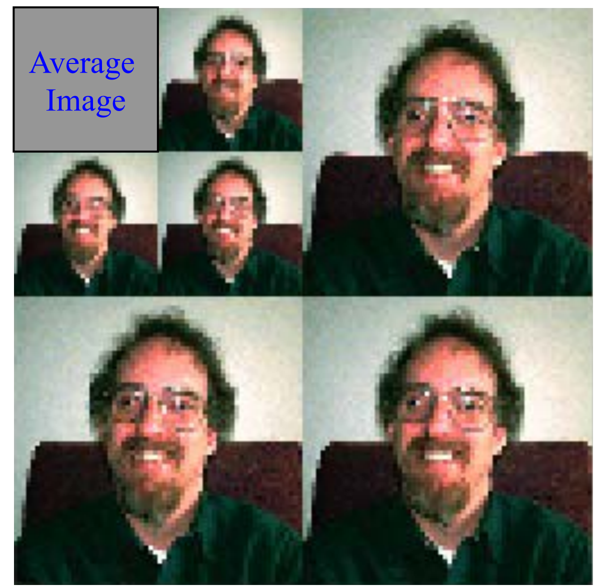
Step 2: Recursively perform Step 1 for top-left quadrant of result

Result of Step 1 (2^{N-1}x2^{N-1})

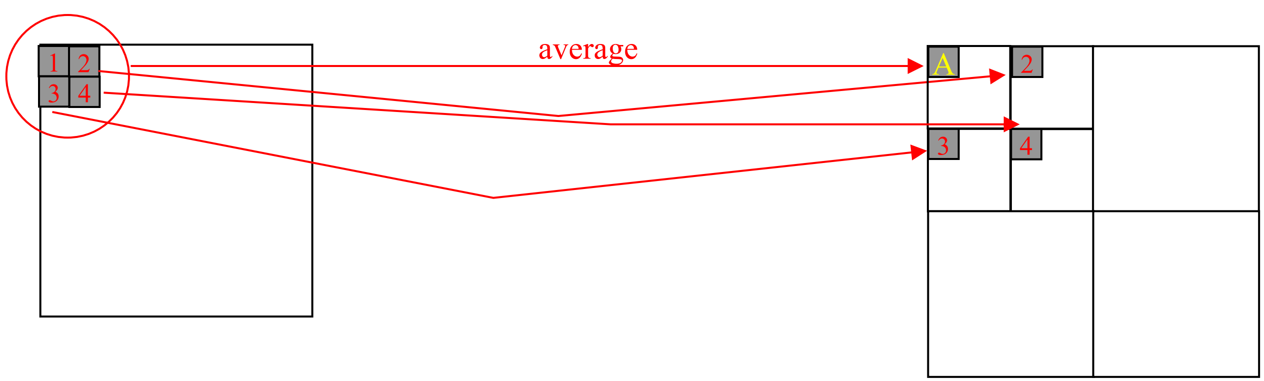
Transformed image (2^Nx2^N)

$2^{N-1} \times 2^{N-1}$

Step 2



: W_2



A Simple, Minimal 2-D Image Transform

Step 3: Recursion stops when average image is 1 pixel

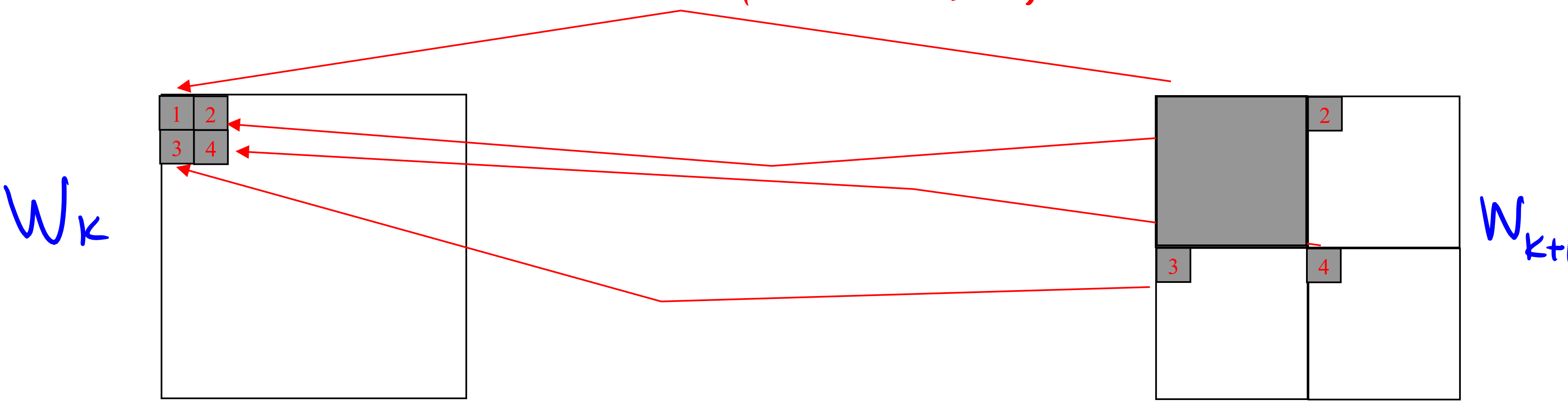
Transformed image (2^Nx2^N)

: W_N

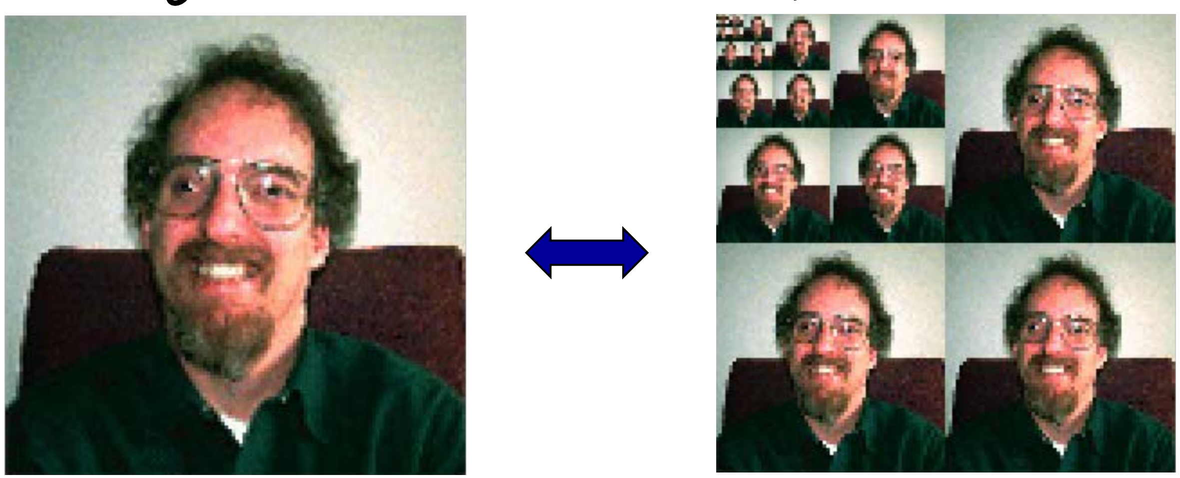
Invertibility of the Transformation

Property: W_k can be reconstructed from W_{k+1}

$P_1 = 4P_A - P_2 - P_3 - P_4$



$\Longleftrightarrow$ W_0 reconstructible from W_N



Topic 9:

Discrete Wavelet Transform

- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

1D Haar Wavelet Transform: Recursive Definition

Input image I^3

pixel #	0	1	2	...				2^3-1
	9	7	3	5	6	10	2	6

Average I^2

(analogous to level g_0 of Gaussian pyramid)

8	4	8	4

average of first two pixels

Detail coefs D^2

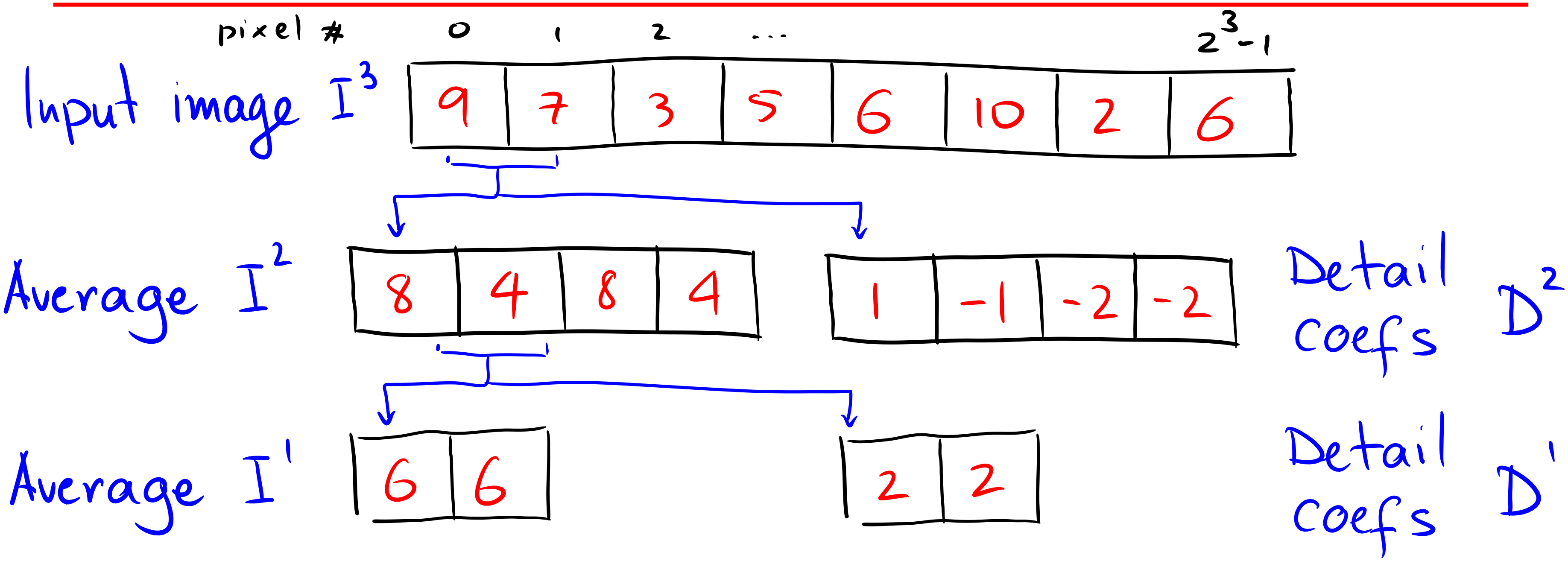
(analogous to level L_0 of Laplacian pyramid)

1	-1	-2	-2

difference of first pixel from the average

(note: we don't need to store difference of 2nd pixel from average $\Rightarrow$ D^0 has ½ the size of the corresponding Laplacian level L_0!)

1D Haar Wavelet Transform: Recursive Definition



$$I_i^j = \frac{1}{2}\left(I_{2i}^{j+1} + I_{2i+1}^{j+1} \right)$$

j-th level of "pyramid" contains 2^j pixels

$$D_i^j = I_{2i}^{j+1} - \frac{1}{2}\left(I_{2i}^{j+1} + I_{2i+1}^{j+1} \right)$$

$$= \frac{1}{2}\left(I_{2i}^{j+1} - I_{2i+1}^{j+1} \right)$$

1D Haar Wavelet Transform: Recursive Definition

pixel #	0	1	2	...				2^3-1
Input image I^3	9	7	3	5	6	10	2	6

Average I^2

8	4	8	4

Detail coefs D^2

1	-1	-2	-2

Average I^1

6	6

Detail coefs D^1

2	2

equivalent convolution mask ϕ:

$1/2$	$1/2$

$$I_i^j = \frac{1}{2}\left(I_{2i}^{j+1} + I_{2i+1}^{j+1} \right)$$

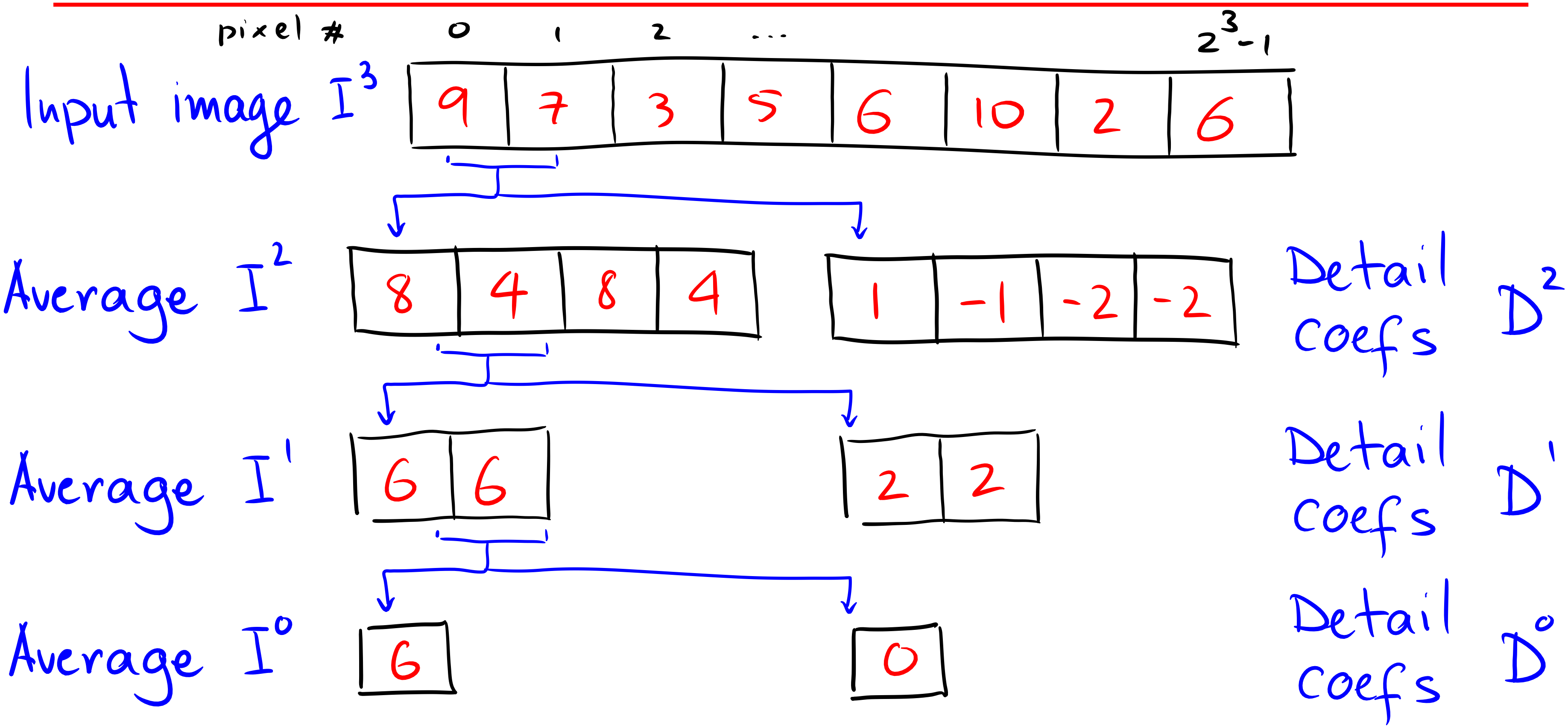
equivalent convolution mask ψ:

$-1/2$	$1/2$

$$D^j = \frac{1}{2}\left(I_{2i}^{j+1} - I_{2i+1}^{j+1} \right)$$

j-th level of "pyramid" contains 2^j pixels

1D Haar Wavelet Transform: Recursive Definition

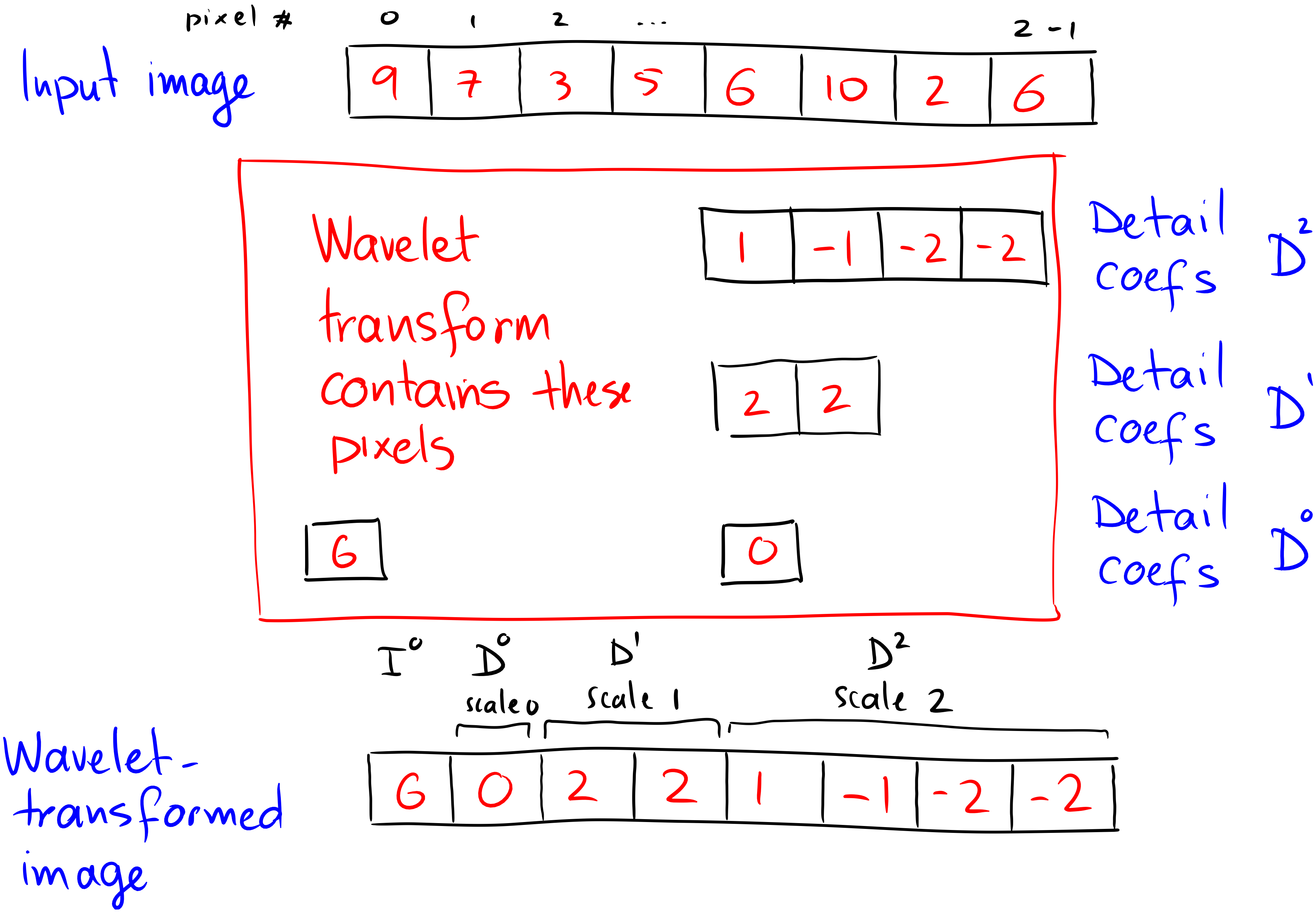


$$I^j_i = \frac{1}{2}\left(I^{j+1}_{2i} + I^{j+1}_{2i+1} \right)$$

$$D^j = \frac{1}{2}\left(I^{j+1}_{2i} - I^{j+1}_{2i+1} \right)$$

j-th level of "pyramid" contains 2^j pixels

1D Haar Wavelet Transform: Recursive Definition

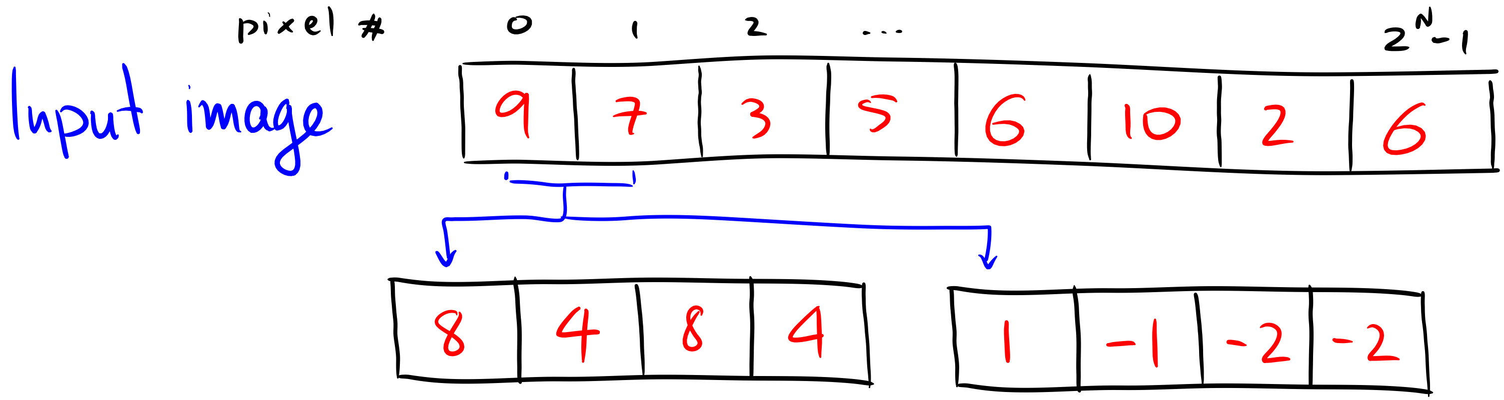


Topic 9:

Discrete Wavelet Transform

- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

1D Haar Wavelet Transform as a Matrix Product

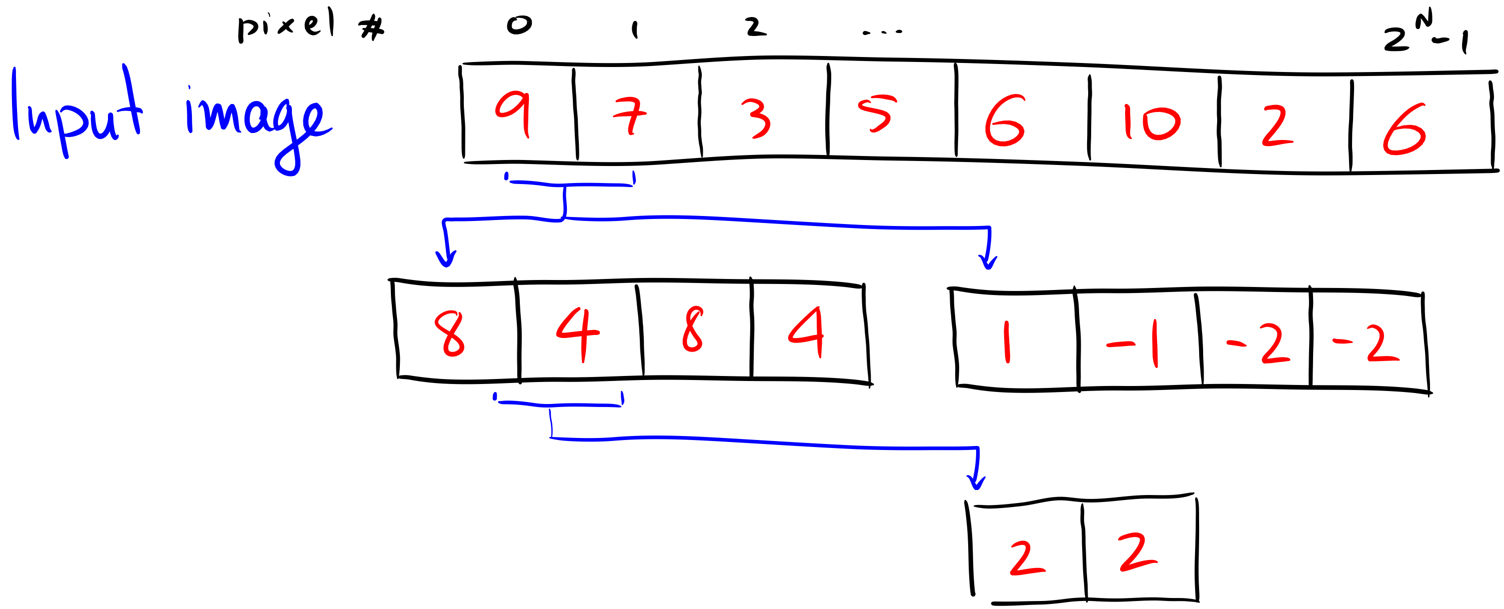


Wavelet transformed image

Original image

$$\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array} \begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

1D Haar Wavelet Transform as a Matrix Product

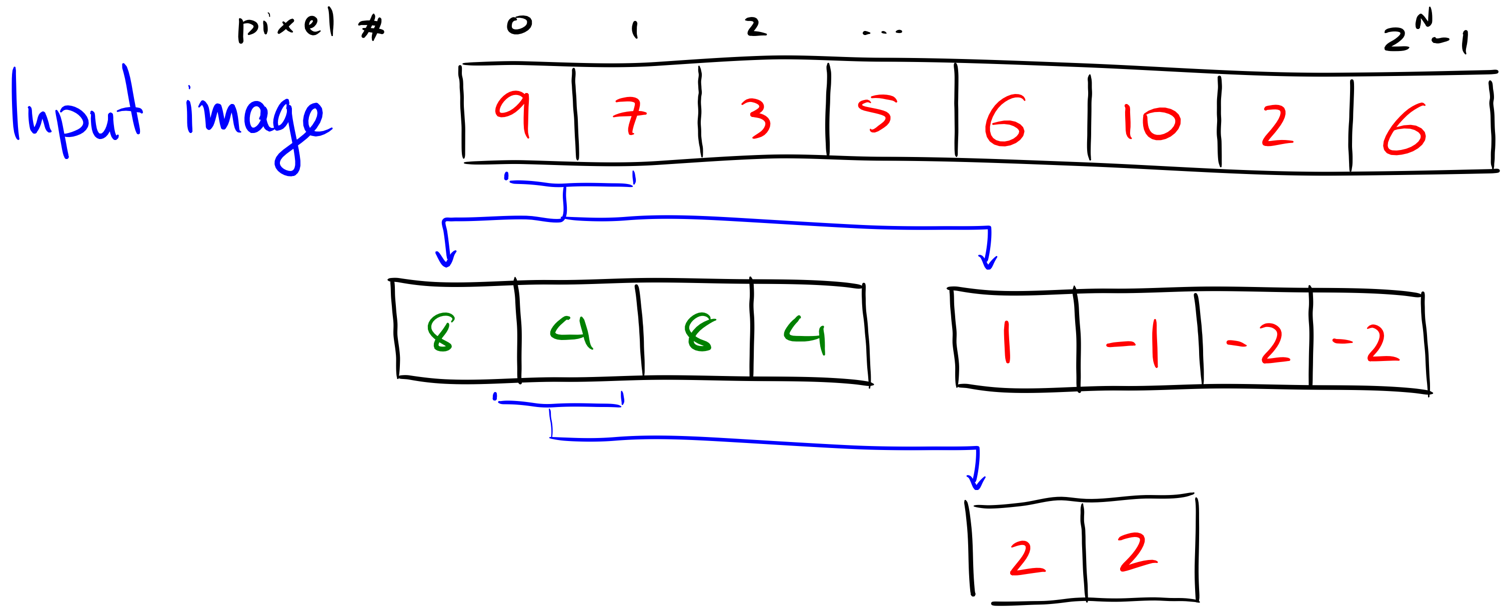


Wavelet transformed image

$$\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array}\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} ? \\ \frac{1}{2}\cdot\begin{matrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{matrix} \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

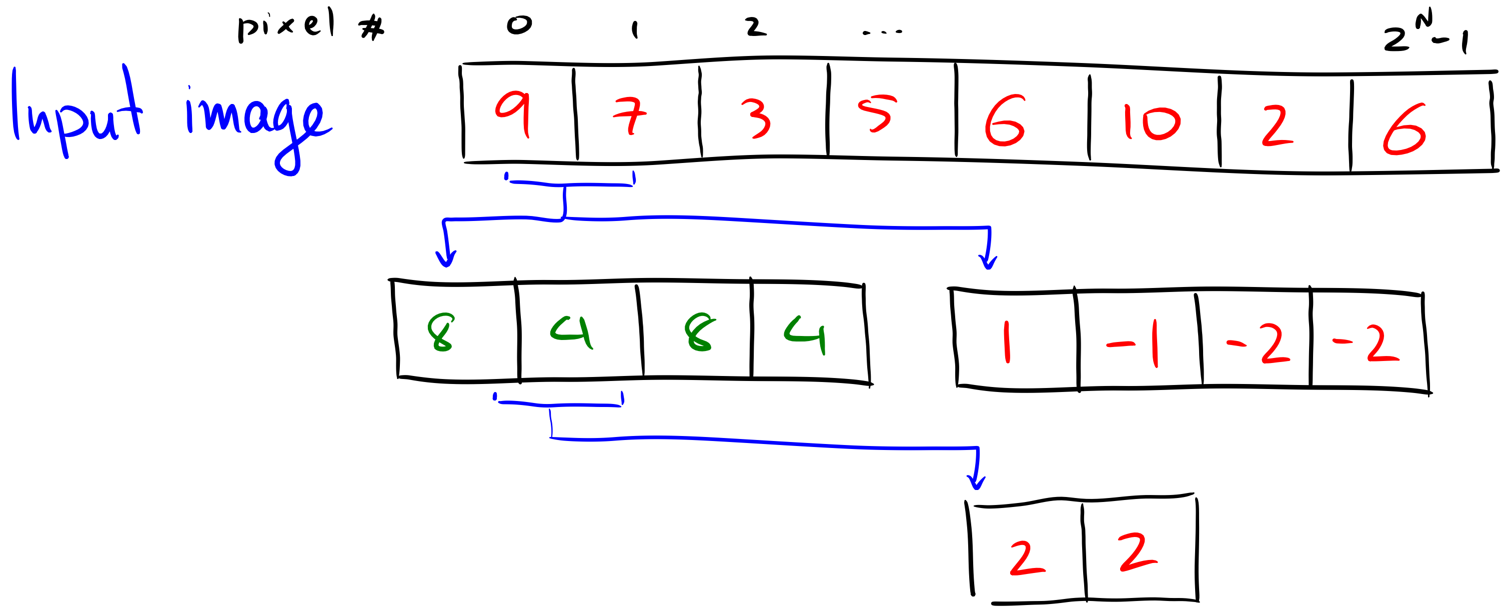
Original image

1D Haar Wavelet Transform as a Matrix Product



$$
\begin{array}{c}
I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{bmatrix}
 & & & & & & & \\
 & & & & & & & \\
\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} 8 \\ 4 \\ 8 \\ 4 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
$$

1D Haar Wavelet Transform as a Matrix Product



$$
\begin{array}{c}
I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{bmatrix}
 & & & & & & & \\
 & & & & & & & \\
1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2 & -1/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\cdot
\frac{1}{2} \cdot
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

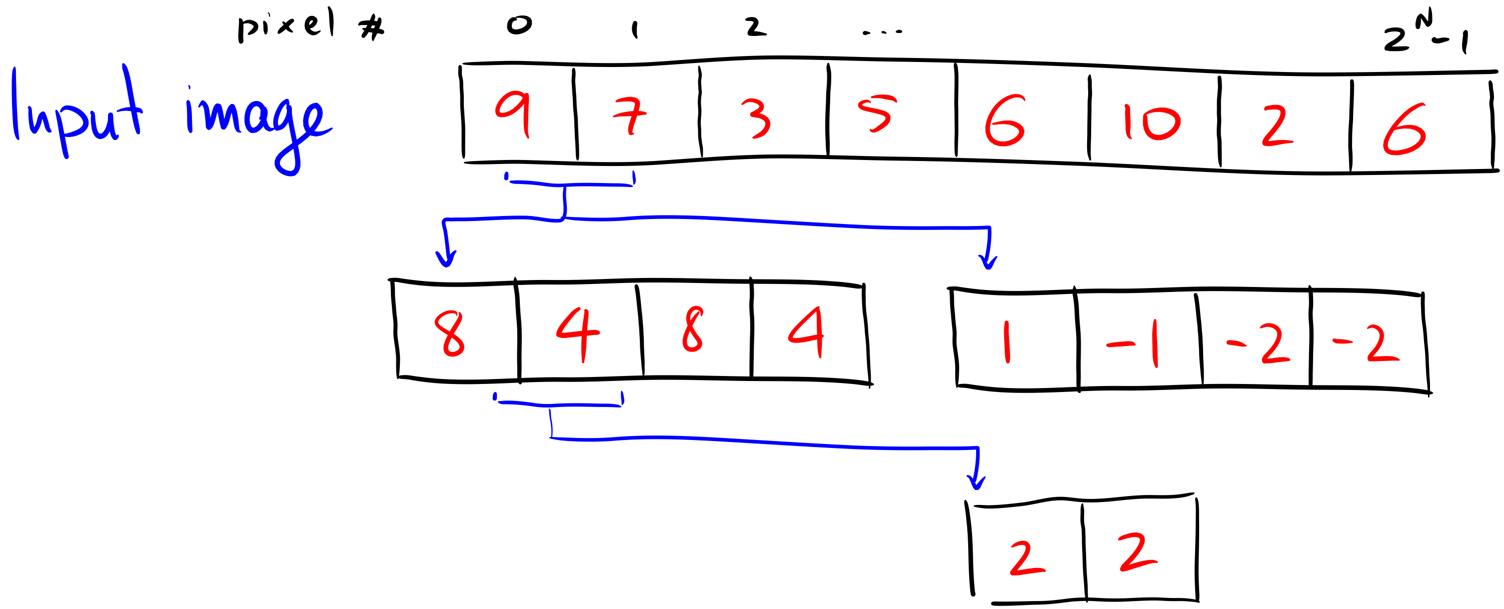
1D Haar Wavelet Transform as a Matrix Product

3rd & 4th rows of product

$$
\left[\begin{array}{cccccccc}
 & & & & & & & \\
 & & & & & & & \\
\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & &
\end{array}\right]
$$

$$
\begin{array}{c}
I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\
\end{array}
\left[\begin{array}{c}
6 \\ \hline 0 \\ \hline 2 \\ 2 \\ \hline 1 \\ -1 \\ -2 \\ -2
\end{array}\right]
=
\left[\begin{array}{cccccccc}
 & & & & & & & \\
\hline
\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]
\cdot
\frac{1}{2} \cdot
\left[\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\hline
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{array}\right]
\left[\begin{array}{c}
9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6
\end{array}\right]
$$

1D Haar Wavelet Transform as a Matrix Product

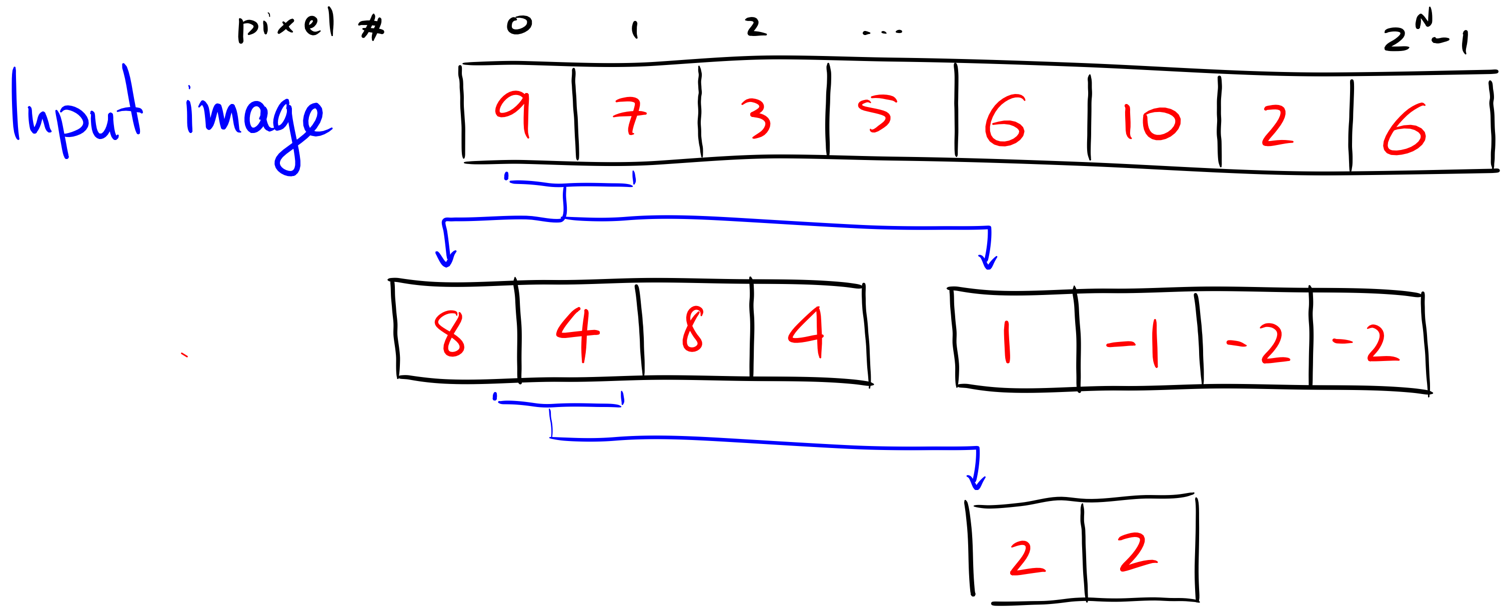


Wavelet transformed image

$$
\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ \\ D^2 \\ \\ \\ \end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{array}{c} \\ \\ \\ 1/4 \\ \\ \\ 1/2 \\ \\ \end{array}
\begin{bmatrix}
 & & & & & & & \\
 & & & & & & & \\
1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

Original image

1D Haar Wavelet Transform as a Matrix Product



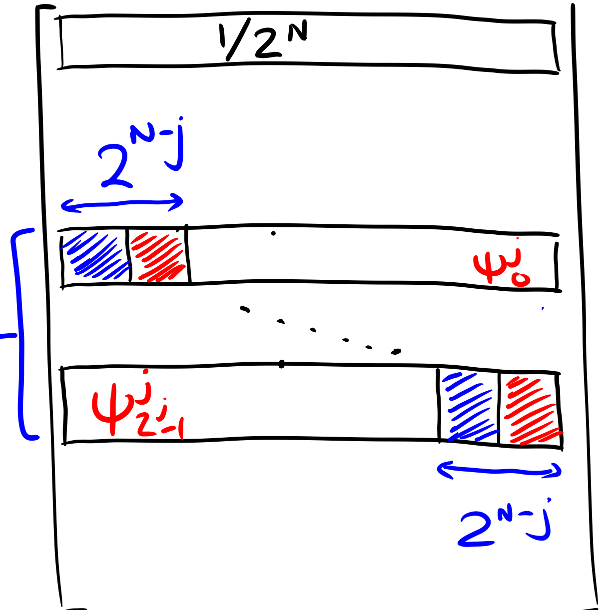
Wavelet transformed image

$$\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array} \begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} = \begin{array}{c} 1/8 \\ 1/8 \\ 1/4 \\ \\ \\ \\ 1/2 \\ \\ \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

Original image

The 1D Haar Wavelet Transform Matrix W

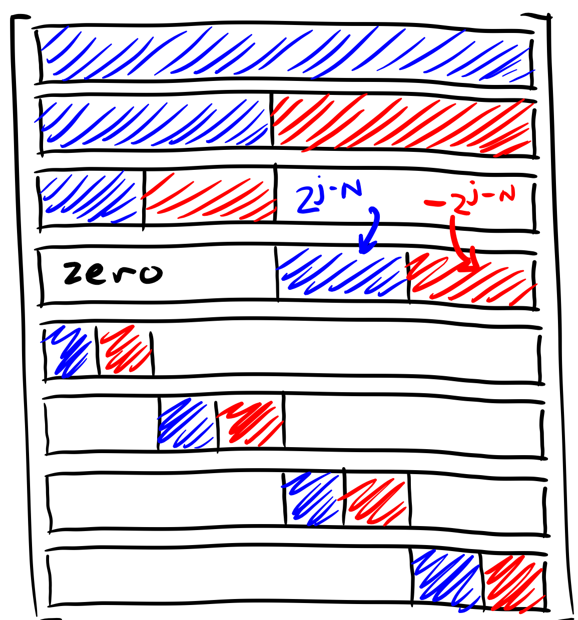
- Matrix contains N-1 scales
- Scale j represented by 2^j rows $\psi^j_0, \ldots, \psi^j_{2^j-1}$



- Row ψ^i_j has $\frac{2^N}{2^j} = 2^{N-j}$ non-zero pixels
- They are pixels $x = i\,2^{N-j}, \ldots, (i+1)2^{N-j}-1$ with $|\psi^i_j(x)| = \frac{1}{2^{N-j}}$

Wavelet transformed image

$$
\begin{array}{c}
I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
$$



$$
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

Original image

Topic 9:

Discrete Wavelet Transform

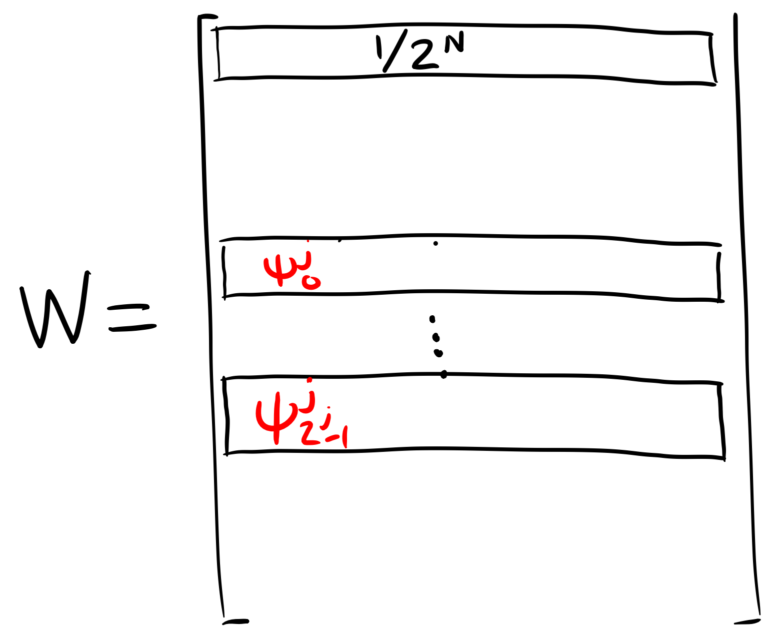
- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

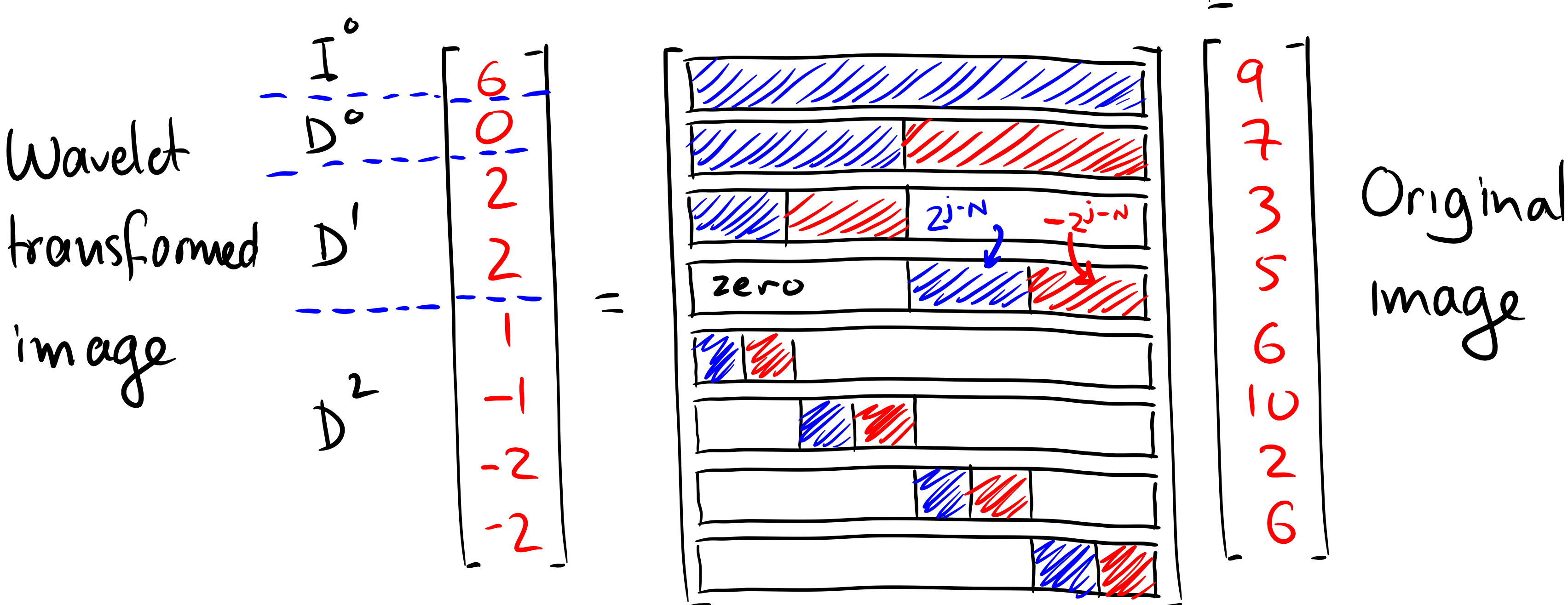
Reconstructing an Image from its Wavelet Coefs

Question: What is the dot product

$$\psi_i^j \cdot \psi_{i'}^{j'}$$

of two distinct rows of W?





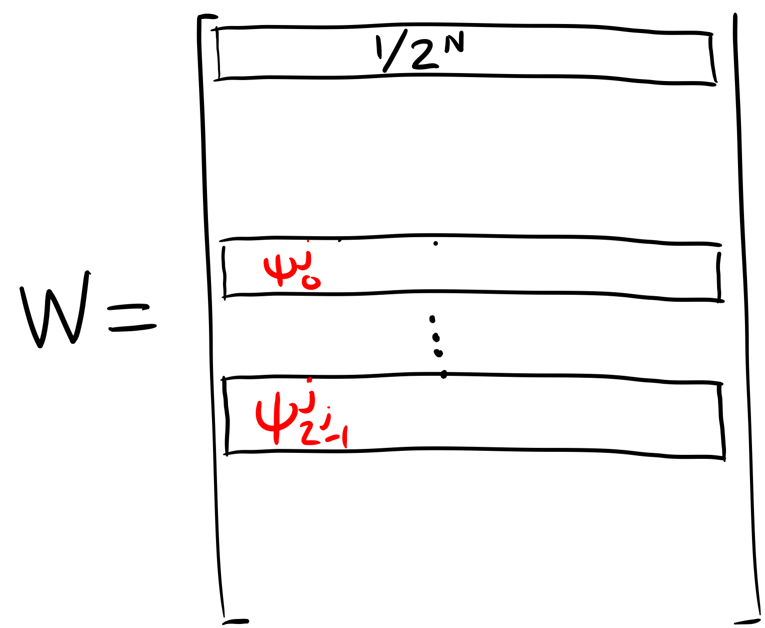
Reconstructing an Image from its Wavelet Coefs

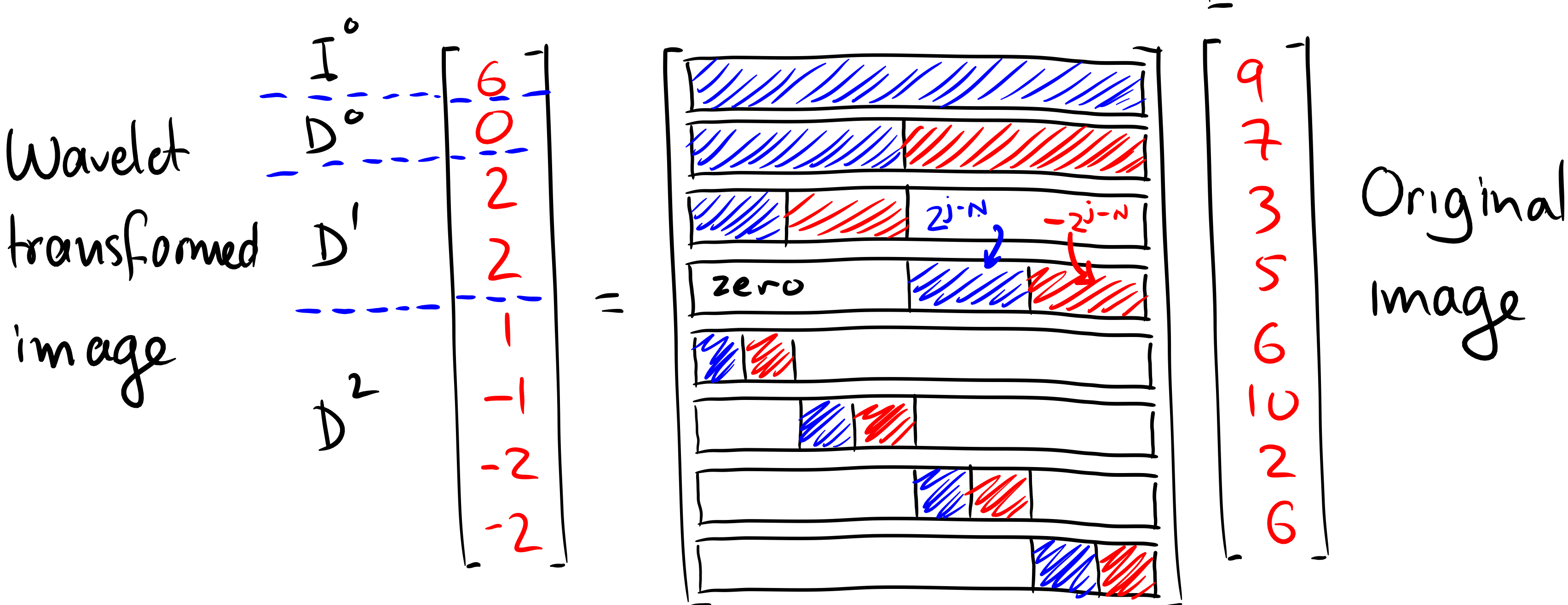
Answer:

$$\psi_i^j \cdot \psi_{i'}^{j'} = 0$$

for two distinct rows of W $\Longleftrightarrow$

$$WW^T =$$





Reconstructing an Image from its Wavelet Coefs

$$WW^T = \text{diagonal} \overset{\text{def}}{=} \Lambda$$

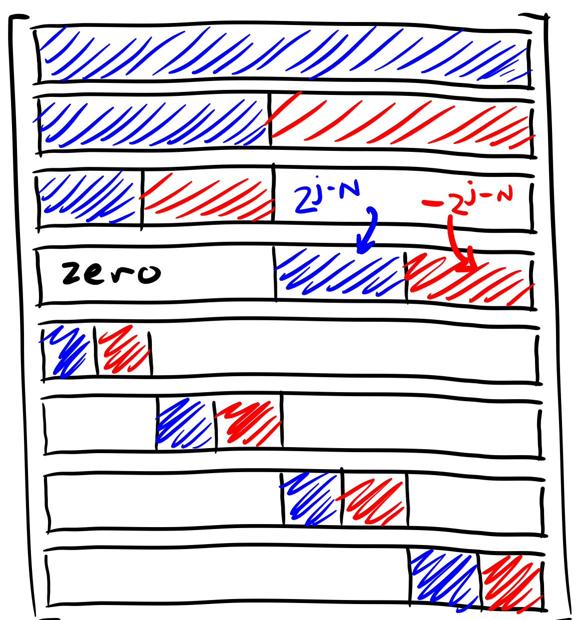
$$= \begin{bmatrix} \ddots & & \\ & \frac{1}{2^{N-j}} & \\ & & \ddots \end{bmatrix}$$

$$(\psi_i^j)\cdot(\psi_i^j)^T \rightarrow \frac{1}{2^{N-j}}$$

$$W = \begin{bmatrix} 1/2^N \\ \\ \psi_0^j \\ \vdots \\ \psi_{2^j-1}^j \\ \\ \end{bmatrix}$$

Wavelet transformed image

$$\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array}\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} =$$



$$\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

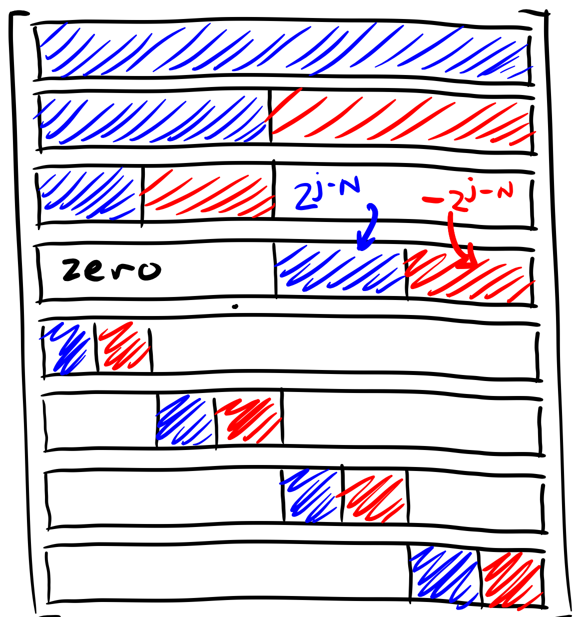
Original image

Reconstructing an Image from its Wavelet Coefs

elet

med

$$
\begin{array}{c}
I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
$$



$$
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

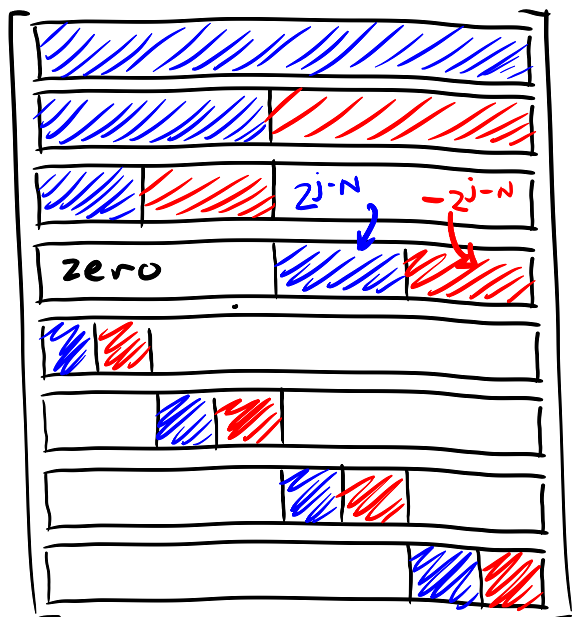
$$WW^T\Lambda^{-1} = I$$

Wavelet transformed image

$$
\begin{array}{c}
I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
= I
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
$$

Reconstructing an Image from its Wavelet Coefs

$$
\begin{array}{ll}
I^0 & \\
\text{elt} \; D^0 & \\
\text{med} \; D^1 & \\
D^2 &
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} =
$$



$$
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

$$W W^T \Lambda^{-1} = I$$

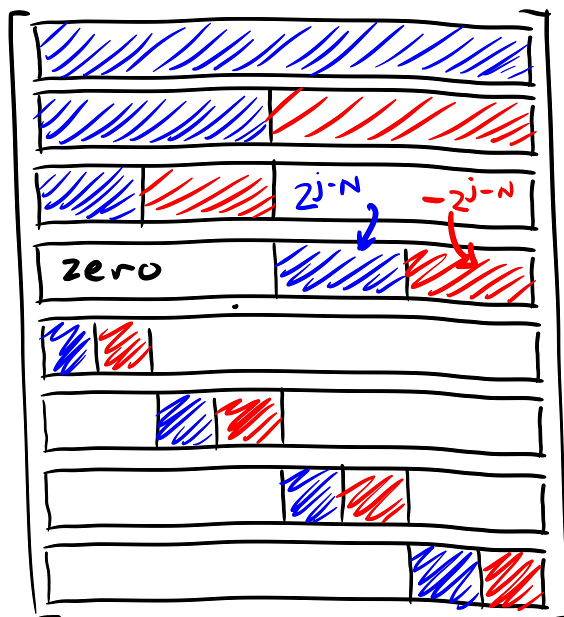
$$
\begin{array}{ll}
I^0 & \\
\text{Wavelet} \; D^0 & \\
\text{transformed} \; D^1 & \\
\text{image} \; D^2 &
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} = W W^T \Lambda^{-1} \begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
$$

Reconstructing an Image from its Wavelet Coefs

elct

med

$$\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array} \begin{bmatrix}6\\0\\2\\2\\1\\-1\\-2\\-2\end{bmatrix} =$$



$$\begin{bmatrix}9\\7\\3\\5\\6\\10\\2\\6\end{bmatrix}$$

these are equal

Wavelet transformed image

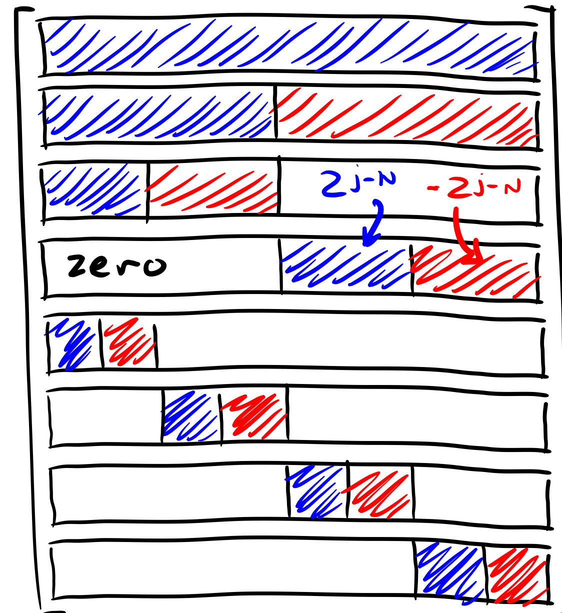
$$\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array} \begin{bmatrix}6\\0\\2\\2\\1\\-1\\-2\\-2\end{bmatrix} = W W^T \Lambda^{-1} \begin{bmatrix}6\\0\\2\\2\\1\\-1\\-2\\-2\end{bmatrix}$$

these are equal

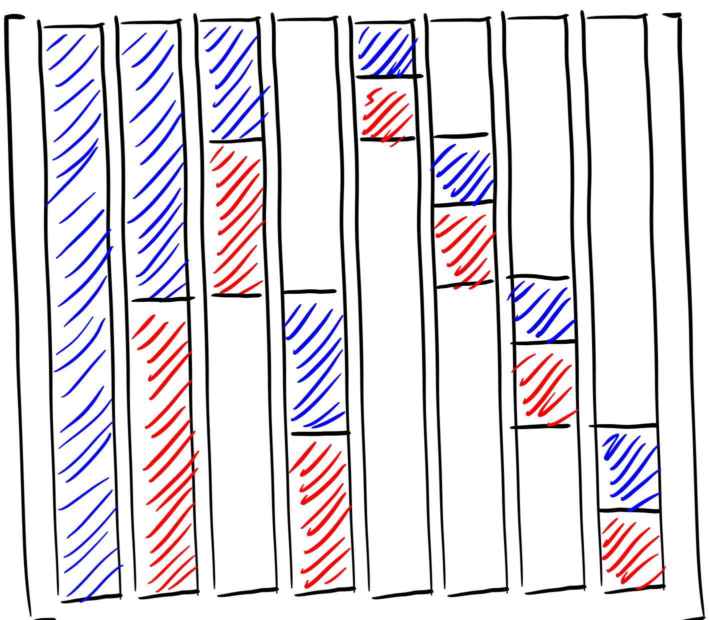
Reconstructing an Image from its Wavelet Coefs

$$\Lambda^{-1} = \begin{bmatrix} \lambda_p^{-1} & & 0 \\ & \ddots & \\ 0 & & \lambda_{2^N-1}^{-1} \end{bmatrix}$$

$W =$



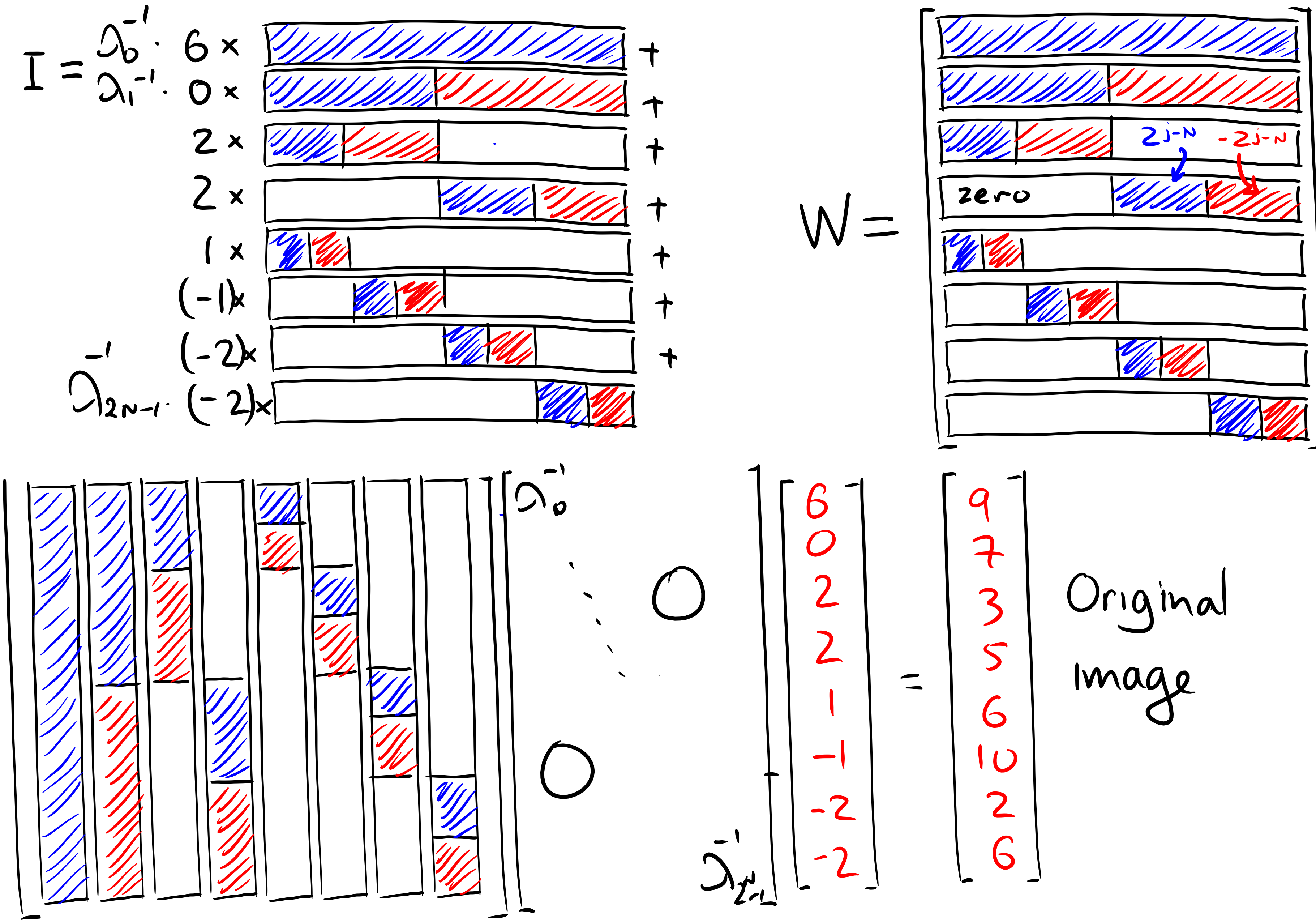
So we have



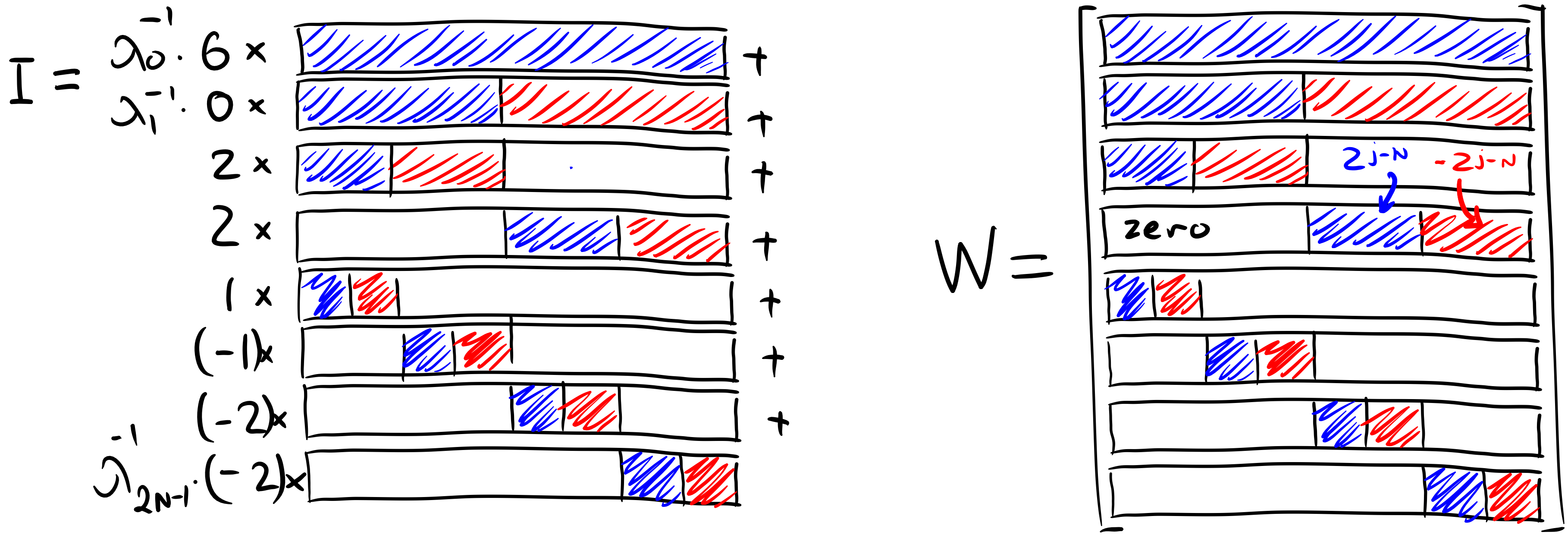
$$\Lambda^{-1} \begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

Original image

Interpreting the Wavelet Coefficients

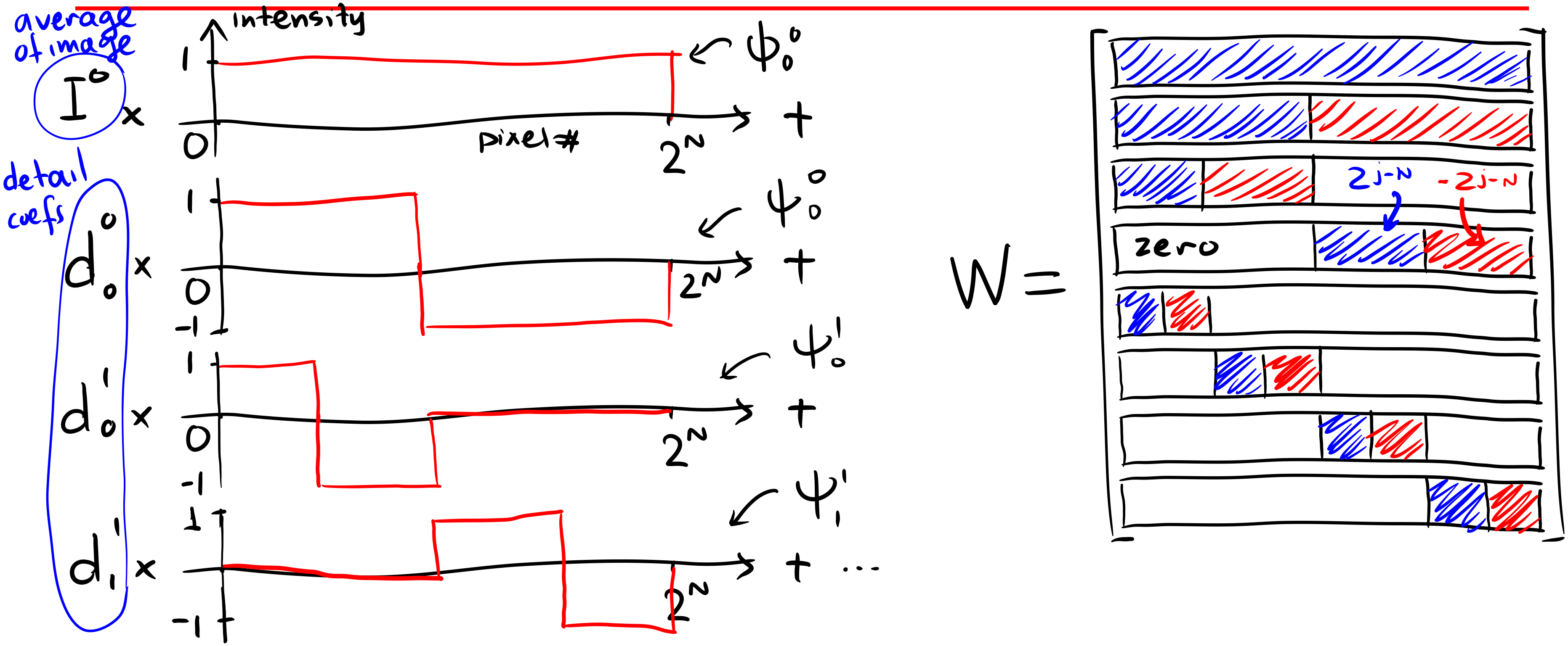


Interpreting the Wavelet Coefficients



$\Rightarrow$ By multiplying I with W we obtain a decomposition of the image into a sequence of basis images $\psi_0^0, \psi_0^1, \ldots \psi_j^i, \ldots$ that form an orthogonal basis of $\mathbb{R}^{2^N}$

Interpreting the Wavelet Coefficients



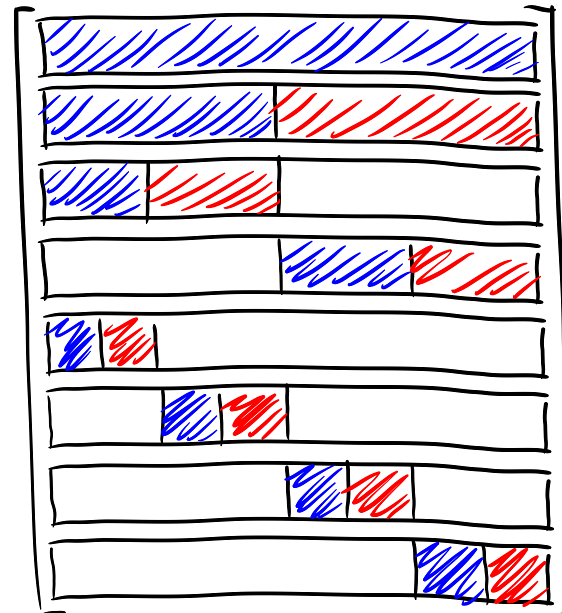
⇒ The wavelet coefficients are the coordinates of the image, considered as a vector in $\mathbb{R}^{2^N}$, in the basis defined by images $\phi_0^0, \psi_0^0, \psi_0^1, \dots$

The Normalized Haar Wavelet Matrix

We can normalize the wavelet transform matrix by multiplying

$$\tilde{W} = \begin{bmatrix} \sqrt{\lambda_1^{-1}} & & \\ & \ddots & \\ & & \sqrt{\lambda_{2^N-1}^{-1}} \end{bmatrix} \cdot W$$

$\tilde{W} =$

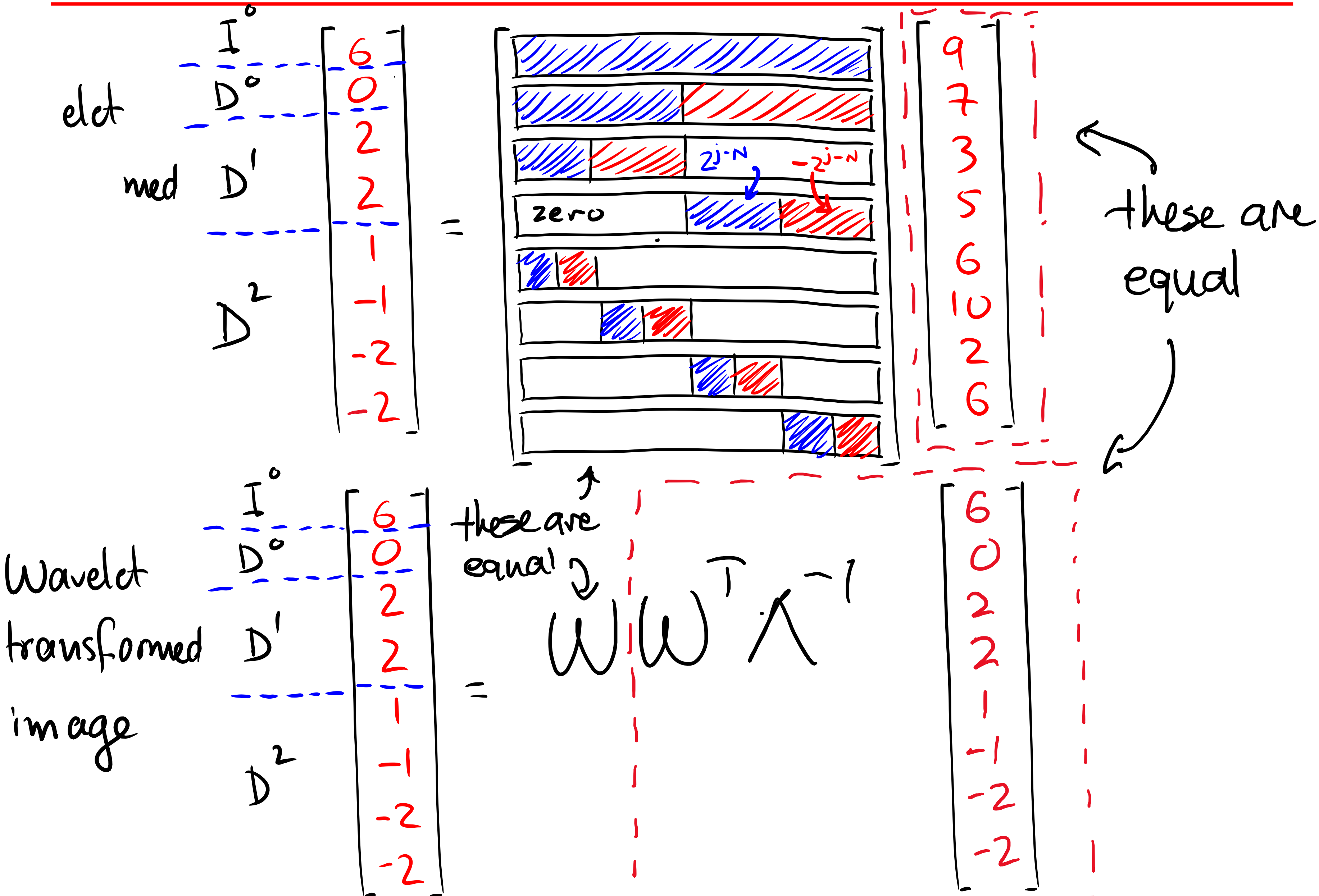


normalized wavelet coefficients

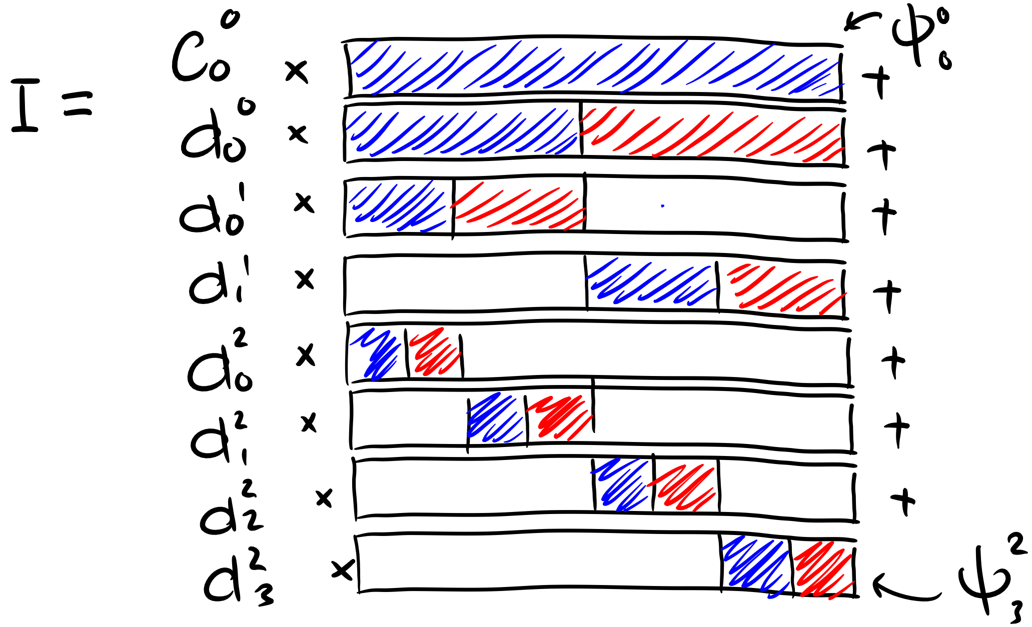
$$\begin{bmatrix} c_0^0 \\ d_0^1 \\ d_1^1 \\ \vdots \end{bmatrix} = \tilde{W} \cdot \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

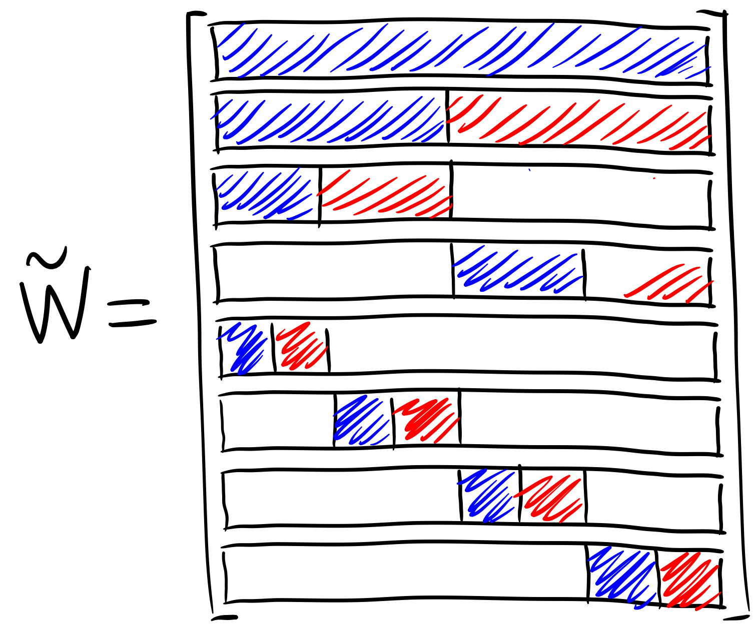
Original image

Reconstructing an Image from its Wavelet Coefs



The Normalized Haar Wavelet Coefficients





$\Rightarrow$ By multiplying I with $\tilde{W}$ we obtain a set of wavelet coefficients $c_0^0, d_0^0, \ldots$ that express I as a linear combination of the basis images $\phi_0^0, \psi_0^0, \psi_0^1, \psi_1^1, \ldots$

Topic 9:

Discrete Wavelet Transform

- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

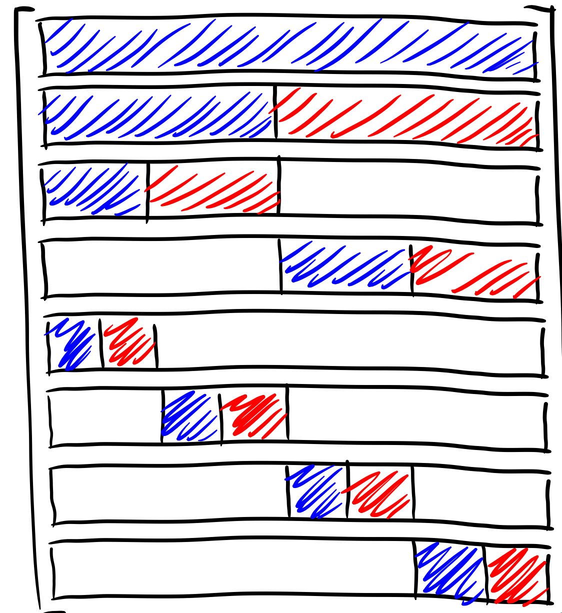
Wavelet Compression Algorithm #1

Input: 1D image I, desired compression k

Output: $k \cdot 2^N$ coefficients

1. Compute $\tilde{W} I$
2. Sort the coefficients $c_0^0, d_0^1, d_1^1, \ldots$ in order of decreasing absolute value
3. Keep the top $k \cdot 2^N$ coefs

$\tilde{W} =$



* Readings show that the algorithm gives the best least-squares approx of the image for the given compression level

normalized wavelet coefficients

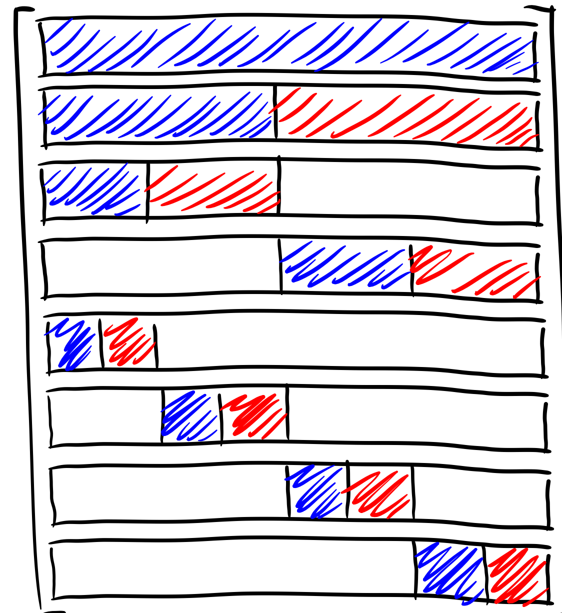
$$\begin{bmatrix} c_0^0 \\ d_0^1 \\ d_1^1 \\ \vdots \\ \\ \\ \\ \\ \end{bmatrix} = \tilde{W} \cdot \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

Original image I

Wavelet Compression Algorithm #2

Input: 1D image I, max error ϵ
Output: $k \cdot 2^N$ coefficients

1. Compute $\tilde{W} I$
2. Sort the coefficients $c_0^0, d_0^1, d_1^1, \ldots$ in order of decreasing absolute value
3. Keep the top $k \cdot 2^N$ coefs with k such that $|\tilde{I} - I| < \epsilon$, where $\tilde{I}$ is the image reconstructed from the top $k 2^N$ coefs

$\tilde{W} =$ 

normalized wavelet coefficients

$$\begin{bmatrix} c_0^0 \\ d_0^1 \\ d_1^1 \\ \vdots \\ \\ \\ \\ \\ \end{bmatrix} = \tilde{W} \cdot \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

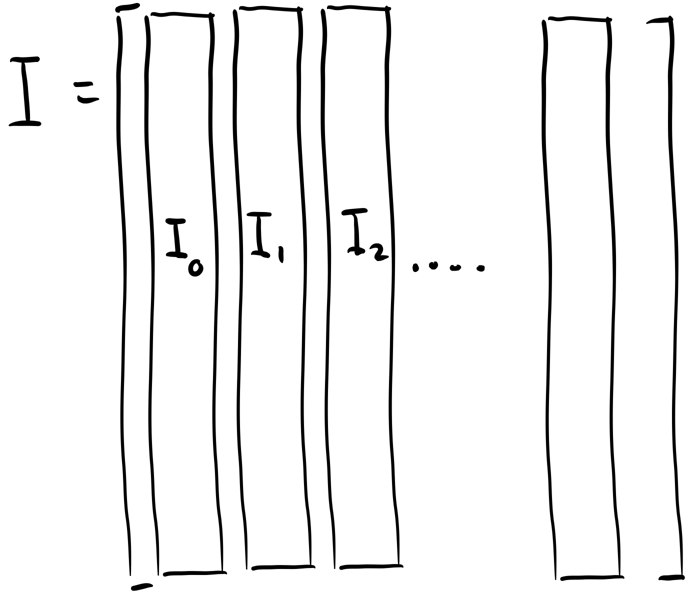
Original image I

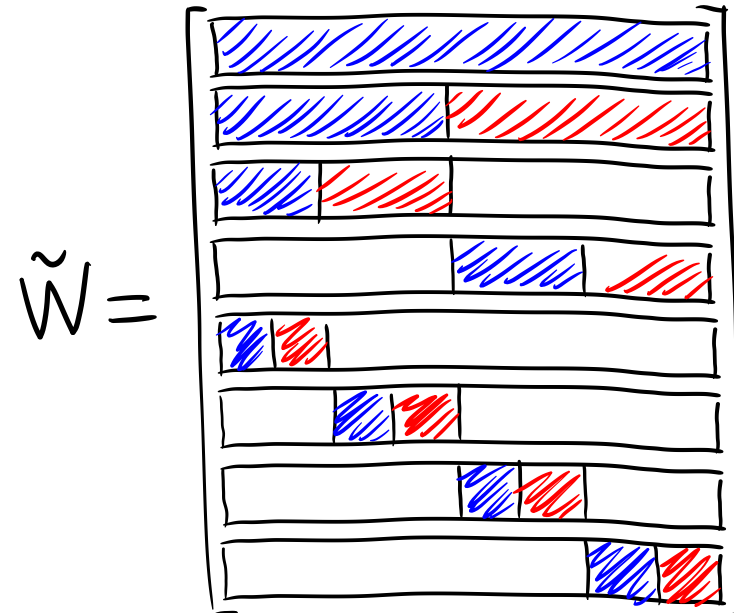
Topic 9:

Discrete Wavelet Transform

- Wavelet transform vs. Laplacian pyramid
- Basic intuition: a simple wavelet-like 2D transform
- The 1D Haar wavelet transform
- 1D Haar wavelet transform as a matrix product
- Reconstructing a 1D image from its wavelet coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

The 2D Haar Wavelet Transform





- To compute the wavelet transform of a 2D image:

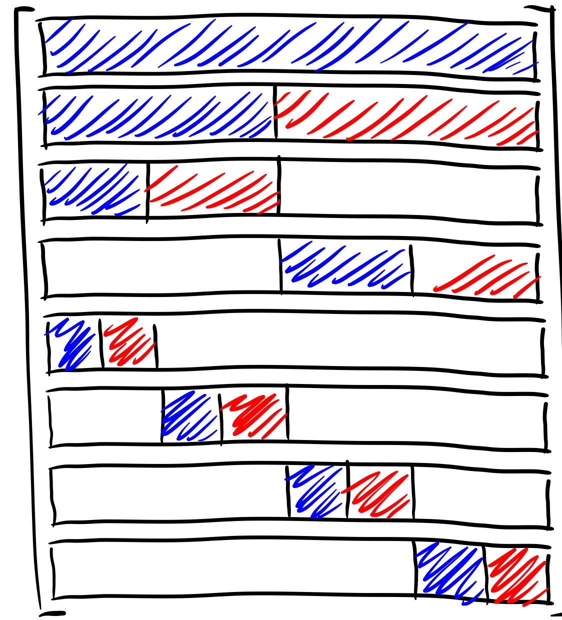
  ① Compute the 1D transform for each column and place the vectors $\tilde{W} I_i$ in a new image I'

  ② Compute the 1D transform of each row of I'

The 2D Haar Wavelet Transform

Exercise: Show that every 2D wavelet coefficient can be expressed as the result of a dot product of the image I and an image defined by $(\psi_i^j)^T \cdot (\psi_{i'}^{j'})$ where ψ_i^j are 1D Haar basis images

$\tilde{W} =$

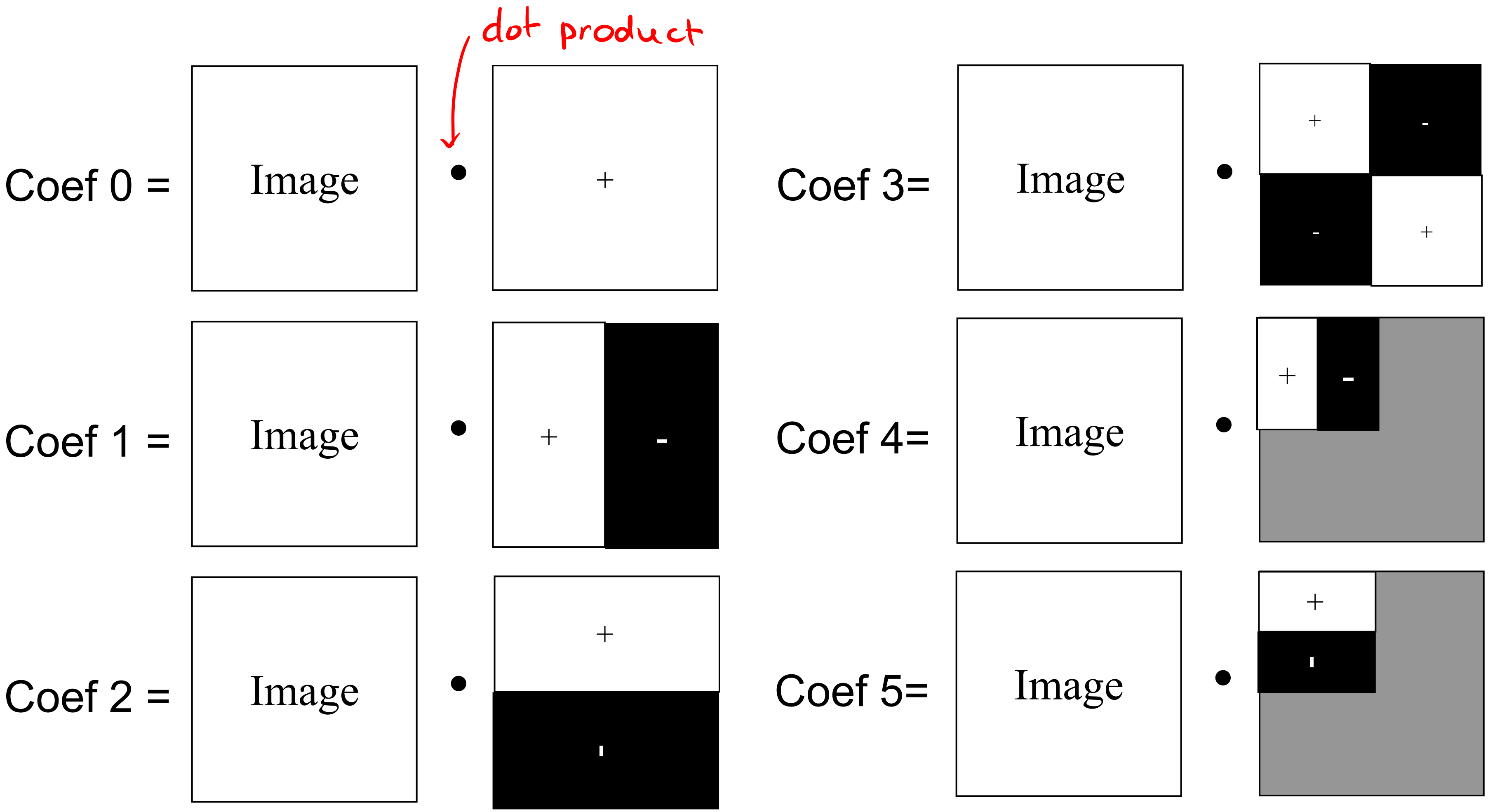


- To compute the wavelet transform of a 2D image:

  1. Compute the 1D transform for each column and place the vectors $\tilde{W} I_i$ in a new image I'
  2. Compute the 1D transform of each row of I'

The 2-D Haar Wavelet Basis

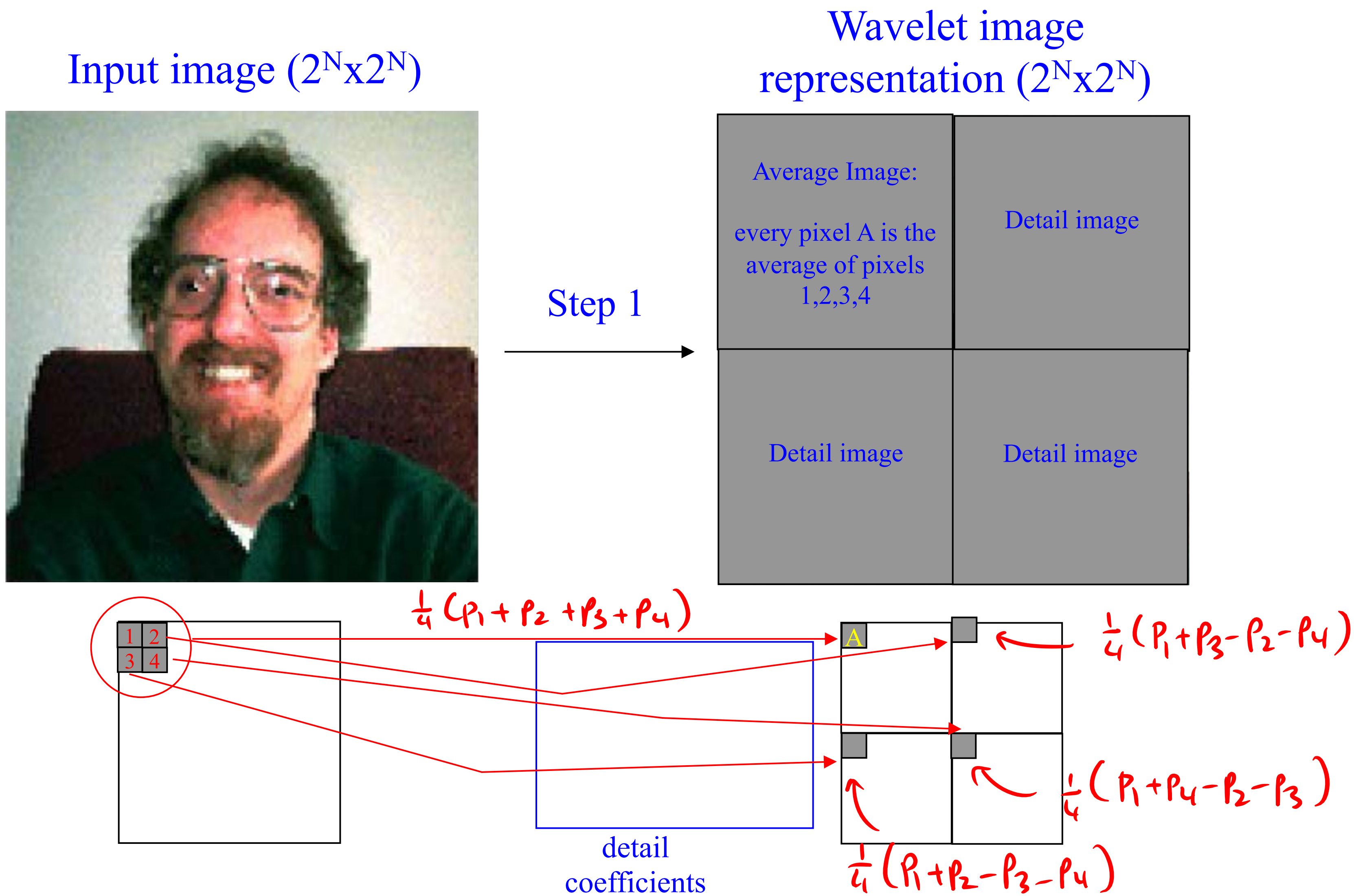
Definition of the first few (coarsest scale) wavelet coefficients of an image of dimensions of $2^N \times 2^N$



A Simple, Minimal 2-D Image Transform

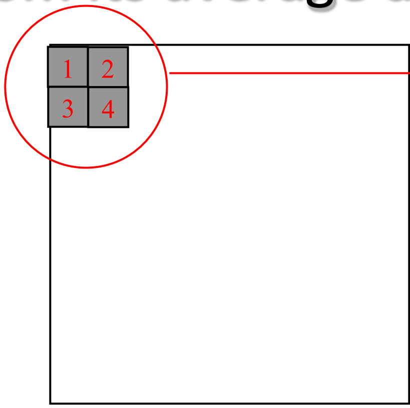
The Haar 2-D Wavelet Transform

The 2-D Haar Wavelet Transform corresponds to a modification of this minimal recursive transform

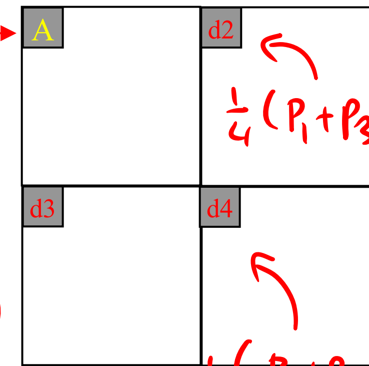


Invertibility of the 2D Haar Transform

We can recursively reconstruct the intensities of every 2x2 window from its average and detail coefficients



2x2 pixel average



$\frac{1}{4}(p_1+p_3-p_2-p_4)$

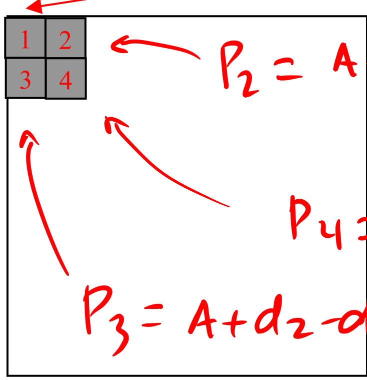
$\frac{1}{4}(p_1+p_2-p_3-p_4)$

$\frac{1}{4}(p_1+p_4-p_2-p_3)$

2 basic operations:

- sum of 4 pixels
- difference of pairwise sums of pixels

$p_1 = A+d_2+d_3+d_4$



$p_2 = A+d_3-d_2-d_4$

$p_4 = A+d_4-d_2-d_3$

$p_3 = A+d_2-d_3-d_4$