

# Topic 7:

## 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 coeffs
- Wavelet-based image compression
- The 2D Haar wavelet transform

# Reminders

---

The tar command in the handout for A3 is incorrect, the correct command is available on the course's main website.

The screenshot shows a web browser window with the URL [www.cs.toronto.edu/~mangas/teaching/320/index.html](http://www.cs.toronto.edu/~mangas/teaching/320/index.html). The page title is "CSC320W: Introduction to Visual Computing". The content includes:

**Instructor:** Fernando Flores-Mangas  
**Email:** mangas320 'at' cs 'dot' toronto 'dot' edu  
**Lectures:** Wednesdays 6-8 pm, BA1190  
**Tutorials:** Wednesdays 8 pm, BA1190  
**TAs:** Micha Livne, Shenlong Wang, Kaustav Kundu and Wenjie Luo.  
**Office Hours:** Wednesdays 11 am, PT265C or by appointment.

Below this are four icons with links: **Calendar**, **Info Sheet**, **Discussion Forum**, and **Other resources**.

A green header bar at the bottom contains the text "Announcements (Newest on top)".

**Mar 13, 2014** **The tar command for A3 is incorrect** The correct command should be:  
`tar cvfz assign3.tar.gz Blend/CHECKLIST.txt  
Blend/partB/{README.txt,*.jpg} Blend/partA/bin/viscomp  
Blend/partA/src/{Makefile,ADDITIONS} Blend/partA/src/pyramid`

Please make sure you tar your files with this command. If you submitted already using the command from the handout we did not get all of your files! Please tar your files again and re-submit.

**Mar 13, 2014** **This announcement is regarding A3** The file part2/README.txt is missing from the handout of A3. Please find it [here](#). The submission deadline for A3 was extended by 2 days: you now have until the last second of Friday March 21 to submit without using grace days. The number of marks given to each section is indicated in the checklist. There is a total of 55 marks for the programming section and 5 marks for the experiments section.

## Last leg

---

- As the course comes to an end, we will start closing some loops.
  - This class is the first one
- This means we will combine some of the tools we have learned into bigger, better or more powerful methods.

## Last leg

---

- The wavelet-based representation of images collapses a few of the concepts covered so far.
- Think of the Laplacian Pyramid representation of an image.

## Last leg

---

- The wavelet-based representation of images collapses a few of the concepts covered so far.
- Think of the Laplacian Pyramid representation of an image.
  - What is needed to recover an image from a Pyramid?

## Last leg

---

- The wavelet-based representation of images collapses a few of the concepts covered so far.
- Think of the Laplacian Pyramid representation of an image.
  - What is needed to recover an image from a Pyramid?
  - The pyramid of “detail images” and...

## Last leg

---

- The wavelet-based representation of images collapses a few of the concepts covered so far.
- Think of the Laplacian Pyramid representation of an image.
  - What is needed to recover an image from a Pyramid?
  - The pyramid of “detail images” and...
  - The filter!

## Last leg

---

- The wavelet-based representation of images collapses a few of the concepts covered so far.
- Think of the Laplacian Pyramid representation of an image.
  - What is needed to recover an image from a Pyramid?
    - The pyramid of “detail images” and...
    - The filter! (it defines the whole pyramid)
  - Is this a data efficient representation?

# The Laplacian Pyramid Representation

How many pixels does a Laplacian Pyramid have?

$$(2^N+1) + (2^{N-1}+1) + \dots + (2^0+1)$$
$$L_0 \quad \uparrow \quad 1 + \frac{1}{4} - \frac{1}{16} + \dots - \frac{1}{4^3}$$
$$\uparrow g_n$$

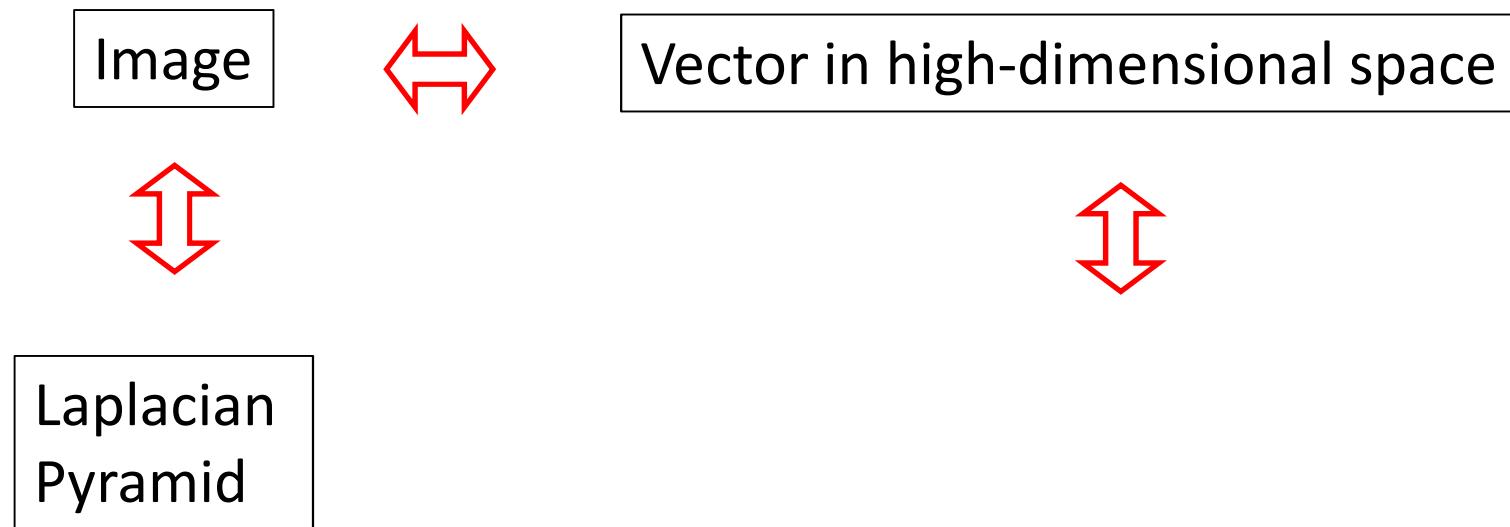
The representation is over-complete! (i.e. there are more pixels in the pyramid than in the image itself)



# Wavelet-Based Image Representations

---

We know we can represent images as:



# Wavelet-Based Image Representations

---

We know we can represent images as:

Image



Vector in high-dimensional space



Laplacian  
Pyramid



PCA: dimensionality reduction.  
An efficiently computable  
compact representation of  
images (from an image class)

# Remember?

$X_1$  ( M dimensions )

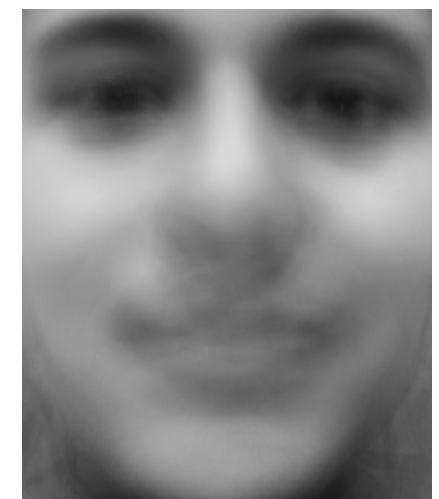


$\approx$

$x_i$  ( d-dimensional  
approx d=3 )

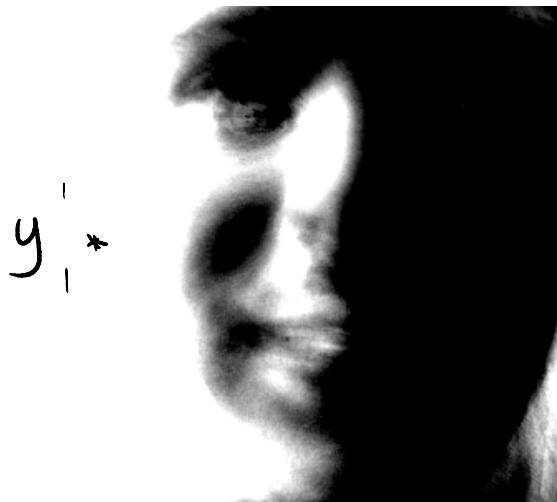


$\bar{X}$



$+$

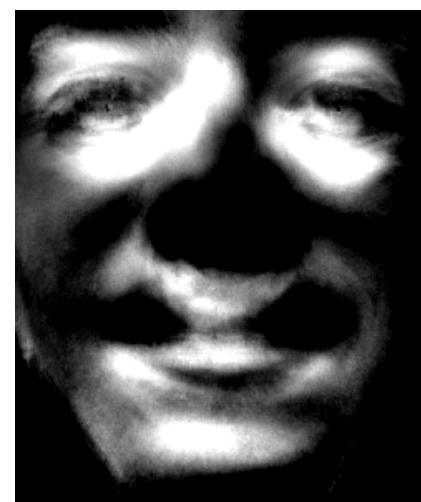
$B_1$



$y_1^1 *$

$+ y_1^2 *$

$B_2$



$y_1^3 *$

$B_3$



# Representing Images by their PCA Basis

---

Start by stacking all your images as in:

$$\begin{bmatrix} z_1^1 & z_2^1 & \dots & \xrightarrow{\text{all pixels of one image}} & z_N^1 \\ z_1^2 & z_2^2 & & & z_N^2 \\ \vdots & \vdots & & & \vdots \\ z_1^M & z_2^M & \dots & & z_N^M \end{bmatrix}$$

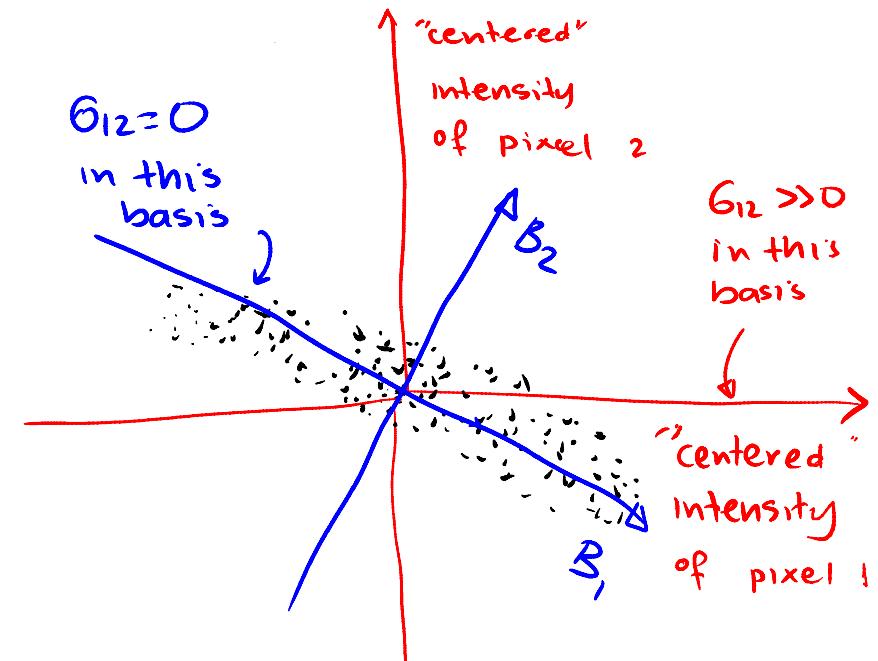
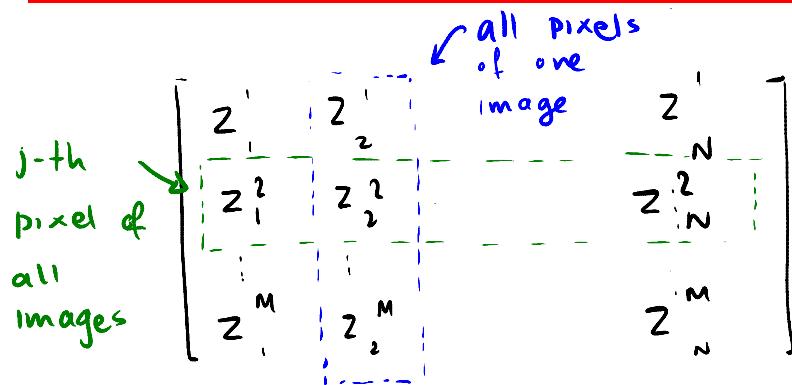
# Representing Images by their PCA Basis

---

Start by stacking all your images as in:

$$\begin{matrix} & & \text{all pixels} \\ & & \text{of one} \\ & & \text{image} \\ \text{j-th} & \text{pixel of} & \\ \text{all} & & \\ \text{images} & & \end{matrix} \left[ \begin{array}{c|c|c|c|c} z_1^1 & z_2^1 & \dots & & z_N^1 \\ \hline z_1^2 & z_2^2 & & & z_N^2 \\ \hline \vdots & \vdots & & & \vdots \\ \hline z_1^M & z_2^M & & & z_N^M \end{array} \right]$$

# Representing Images by their PCA Basis



You can then represent these pixel values using a new basis  $B$  along the directions of maximum variation.

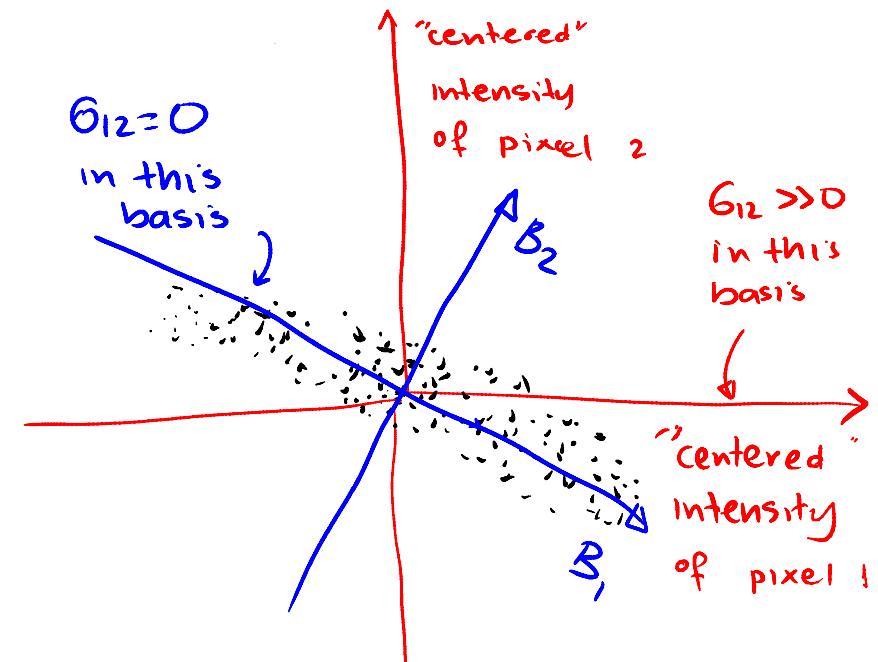
(How do we interpret a point in this new basis?)

# Representing Images by their PCA Basis

j-th pixel of all images

$$\begin{bmatrix} z_1^1 & z_1^2 & \dots & z_1^N \\ z_2^1 & z_2^2 & \dots & z_2^N \\ \vdots & \vdots & \ddots & \vdots \\ z_M^1 & z_M^2 & \dots & z_M^N \end{bmatrix}$$

all pixels of one image



$$[B_1 \ B_2 \ \dots \ B_M] \begin{bmatrix} y_1^1 & y_2^1 & y_N^1 \\ y_1^d & y_2^d & y_N^d \\ y_1^{d+1} & y_2^{d+1} & y_N^{d+1} \\ \vdots & \vdots & \vdots \\ y_1^M & y_2^M & y_N^M \end{bmatrix}$$

large

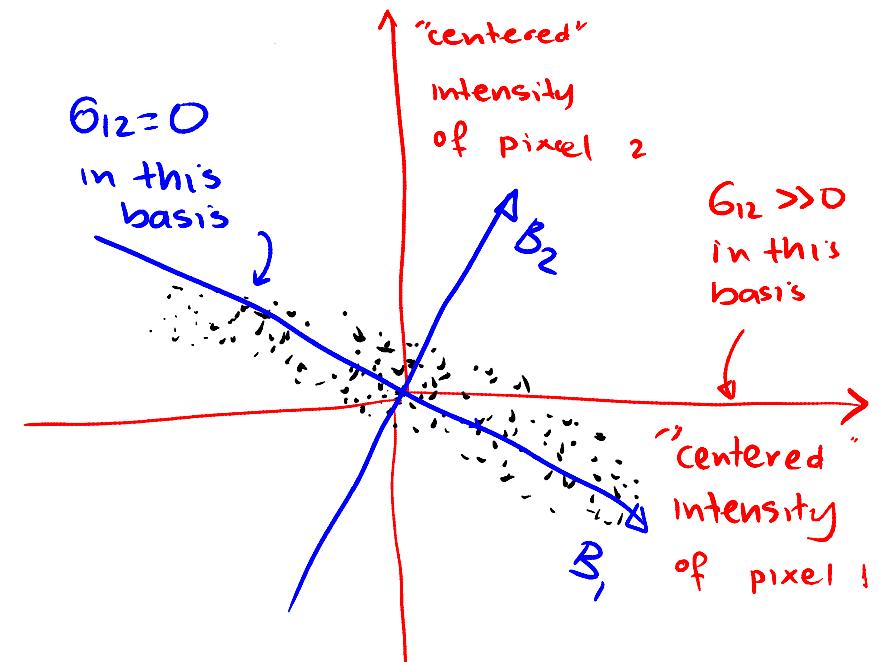
near zero

# Representing Images by their PCA Basis

j-th pixel of all images

all pixels of one image

$$\begin{bmatrix} z_1^1 & z_1^2 & \dots & z_1^N \\ z_2^1 & z_2^2 & \dots & z_2^N \\ \vdots & \vdots & \ddots & \vdots \\ z_M^1 & z_M^2 & \dots & z_M^N \end{bmatrix}$$



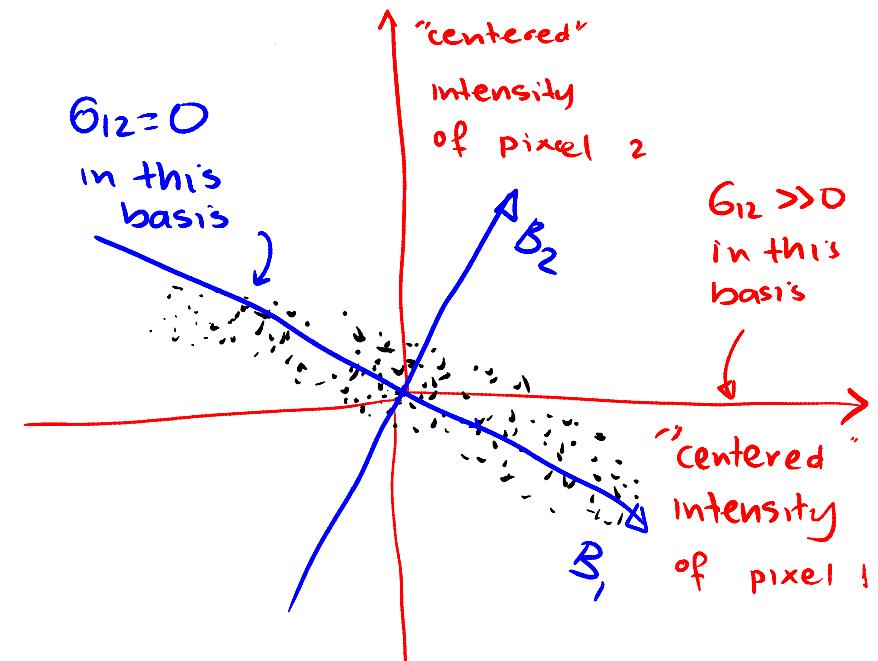
$$Z = B \cdot Y$$

# Representing Images by their PCA Basis

j-th pixel of all images

$$\begin{bmatrix} z_1^1 & z_1^2 & \dots & z_1^N \\ z_2^1 & z_2^2 & \dots & z_2^N \\ \vdots & \vdots & \ddots & \vdots \\ z_M^1 & z_M^2 & \dots & z_M^N \end{bmatrix}$$

all pixels of one image



$$Z = B \cdot Y$$

eigenfaces

mean-subtracted images

coordinates of each image in PCA/eigenface basis

# Representing Images by their PCA Basis

---

Image **reconstruction** (from basis coordinates to images):

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

Image **transform** (from images to basis coordinates):

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

$$Z = \mathcal{B} \cdot Y$$

*eigenfaces*  
*mean-subtracted images*      *coordinates of each image in PCA/eigenface basis*

---

I thought we were talking about wavelets today...

---

We are, but the similarities with PCA are huge...

# Representing Images by their PCA Basis

---

Replace the mean-subtracted images:

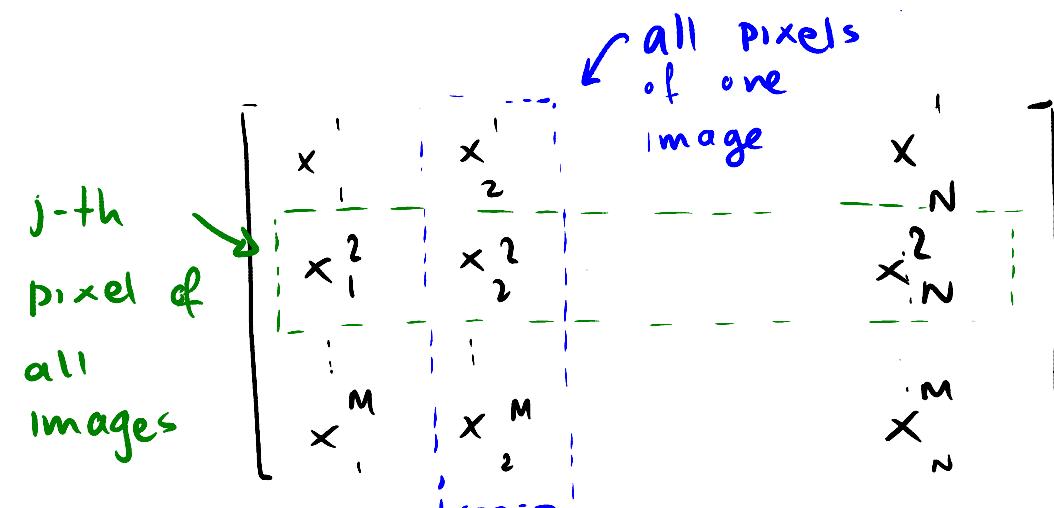
$$\begin{matrix} \text{j-th pixel of all images} & \left[ \begin{array}{c|c|c|c|c} z^1_1 & z^1_2 & \dots & z^1_N \\ z^2_1 & z^2_2 & & z^2_N \\ \vdots & \vdots & \ddots & \vdots \\ z^M_1 & z^M_2 & \dots & z^M_N \end{array} \right] \end{matrix}$$

all pixels of one image

# Representing Images by their PCA Basis

---

With the actual images:



# Representing Images by their PCA Basis

---

And apply the same base-representation machinery:

$$\left[ \mathbf{B}_1 \ \mathbf{B}_2 \ \dots \ \mathbf{B}_M \right] \begin{bmatrix} \mathbf{y}_1' & \mathbf{y}_2' & & \mathbf{y}_N' \\ \mathbf{y}_1^d & \mathbf{y}_2^d & & \mathbf{y}_N^d \\ \mathbf{y}_1^{d+1} & \mathbf{y}_2^{d+1} & & \mathbf{y}_N^{d+1} \\ \vdots & \vdots & & \vdots \\ \mathbf{y}_1^M & \mathbf{y}_2^M & & \mathbf{y}_N^M \end{bmatrix}$$

The diagram illustrates the PCA basis representation of an image. It shows a 2D coordinate system with two basis vectors  $\mathbf{B}_1$  and  $\mathbf{B}_2$  originating from the origin. A point on the  $\mathbf{B}_2$  vector is labeled "centered intensity of pixel 2". A point on the  $\mathbf{B}_1$  vector is labeled "centered intensity of pixel 1". A horizontal dashed line through the origin is labeled " $G_{12}=0$  in this basis". A vertical dashed line through the origin is labeled " $G_{12} \gg 0$  in this basis".

---

And you'll go from PCA

# Representing Images by their PCA Basis

j-th pixel of all images

$$\begin{bmatrix} z_1^1 & z_1^2 & \dots & z_1^N \\ z_2^1 & z_2^2 & \dots & z_2^N \\ \vdots & \vdots & \ddots & \vdots \\ z_M^1 & z_M^2 & \dots & z_M^N \end{bmatrix} = \begin{bmatrix} z_1^1 & z_1^2 & \dots & z_1^N \\ z_2^1 & z_2^2 & \dots & z_2^N \\ \vdots & \vdots & \ddots & \vdots \\ z_N^1 & z_N^2 & \dots & z_N^N \end{bmatrix}$$

all pixels of one image

large {

$$\begin{bmatrix} y_1^1 & y_2^1 & \dots & y_N^1 \\ y_1^d & y_2^d & \dots & y_N^d \\ y_1^{d+1} & y_2^{d+1} & \dots & y_N^{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^M & y_2^M & \dots & y_N^M \end{bmatrix}$$

near zero {

$$\begin{bmatrix} y_1^0 & y_2^0 & \dots & y_N^0 \\ y_1^{0+1} & y_2^{0+1} & \dots & y_N^{0+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{M+1} & y_2^{M+1} & \dots & y_N^{M+1} \end{bmatrix}$$

$\Leftrightarrow$  eigenfaces

$$Z = B \cdot Y$$

mean-subtracted images

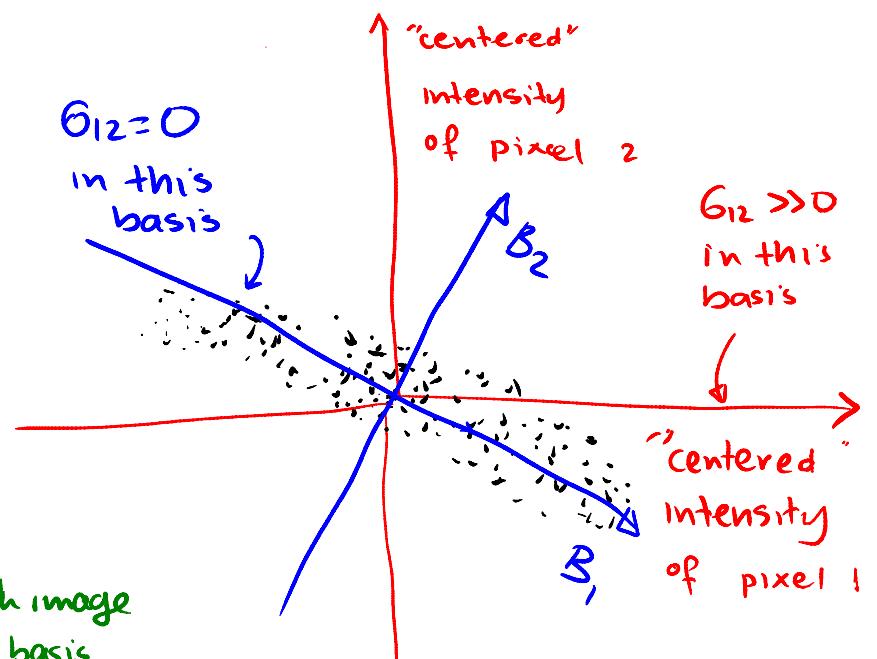
coordinates of each image in PCA/eigenface basis

- Image reconstruction:

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

- Image transform

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



# The Discrete Wavelet Transform

j-th pixel of all images

all pixels of one image

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

$$[B_1 \ B_2 \ \dots \ B_M] \cdot$$

large {

$$\begin{bmatrix} y_1^1 & y_2^1 & \dots & y_N^1 \\ y_1^d & y_2^d & \dots & y_N^d \\ y_1^{d+1} & y_2^{d+1} & \dots & y_N^{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^M & y_2^M & \dots & y_N^M \end{bmatrix}$$

near zero {

$\iff$

wavelet basis

wavelet coefficients

$$X = B \cdot Y$$

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 maps an image onto yet another basis, defined by a “special” matrix  $B$ .

# The Discrete Wavelet Transform

---

The (discrete) wavelet transform maps an image onto yet another basis, defined by a “special” matrix  $\mathbf{B}$ .

This transform:

- Captures scale,
- Is invertible, orthogonal and square
- Is image **independent** (not all my images have to be faces, or eyes).

# Topic 7:

## 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 coeffs
- Wavelet-based image compression
- The 2D Haar wavelet transform

# A Simple, Minimal 2-D Image Transform

---

More properties of the wavelet transform:

- No need for more pixels
- Explicit multi-scale representation
- Invertible
- Linear

Input image ( $2^N \times 2^N$ )



wavelet  
transform

Transformed image ( $2^N \times 2^N$ )



# A Simple, Minimal 2-D Image Transform

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

Input image ( $2^N \times 2^N$ )

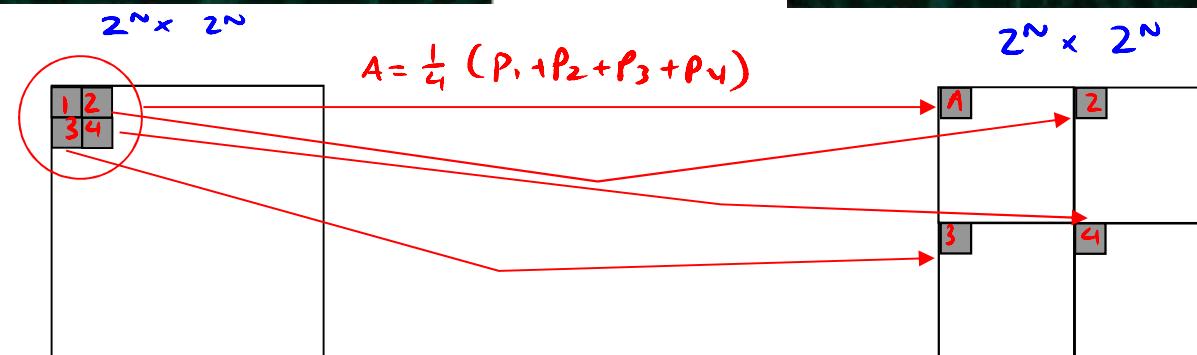


Transformed image ( $2^N \times 2^N$ )



W<sub>0</sub>:

Step 1



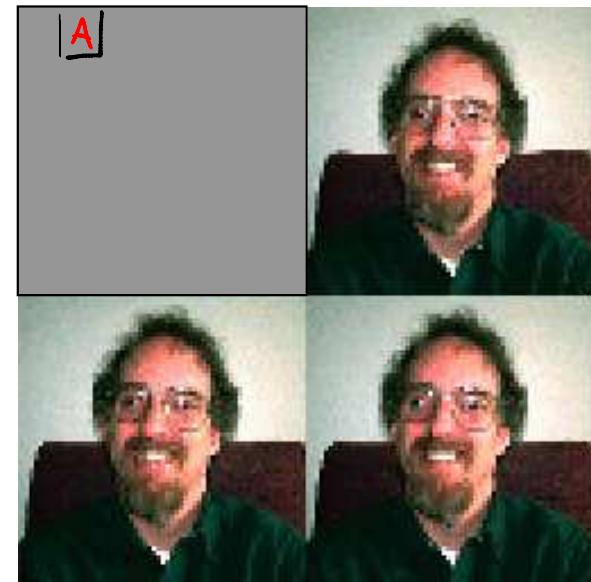
# A Simple, Minimal 2-D Image Transform

and repeat for the rest of the image!

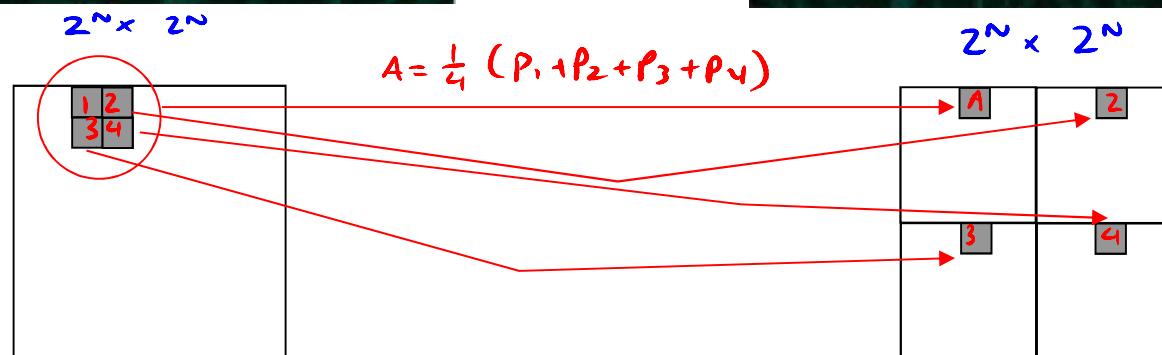
Input image ( $2^N \times 2^N$ )



Transformed image ( $2^N \times 2^N$ )

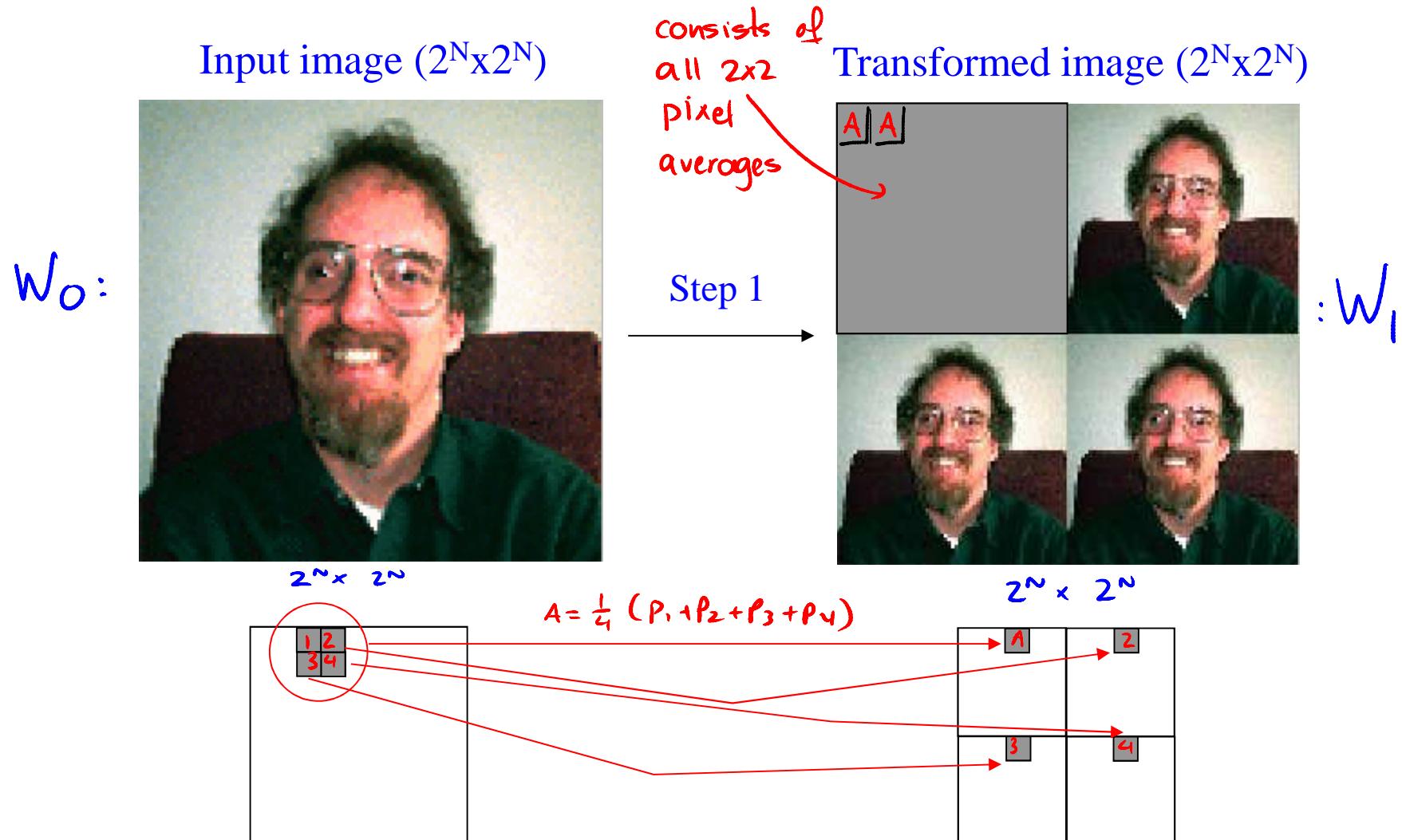


W<sub>O</sub>:



# A Simple, Minimal 2-D Image Transform

You end up with a half-size-per-side image of  $2 \times 2$  pixel averages.



# A Simple, Minimal 2-D Image Transform

---

Guesses for step 2?

# A Simple, Minimal 2-D Image Transform

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

Result of Step 1 ( $2^{N-1} \times 2^{N-1}$ )



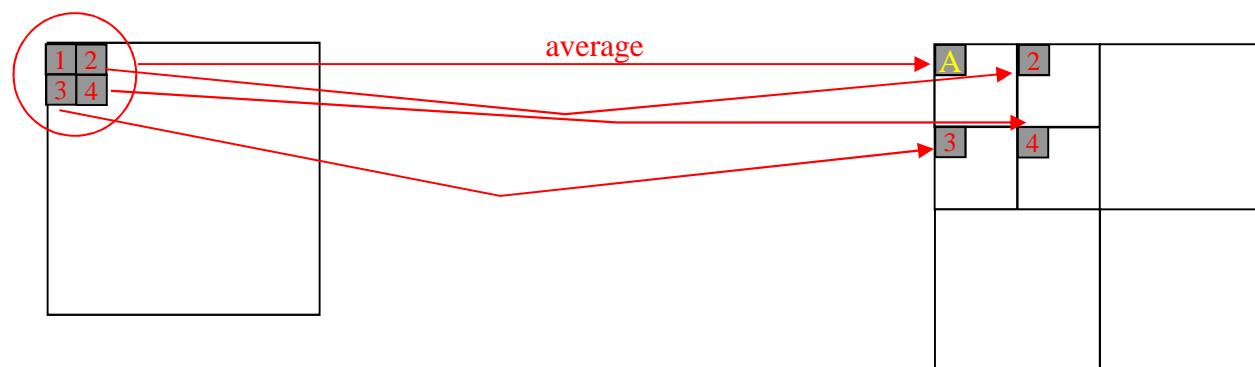
$2^{N-1} \times 2^{N-1}$

Transformed image ( $2^N \times 2^N$ )



:  $W_2$

Step 2



# A Simple, Minimal 2-D Image Transform

---

Step 3: Recursion stops when average image is 1 pixel

Transformed image ( $2^N \times 2^N$ )



:  $W_N$

# A Simple, Minimal 2-D Image Transform

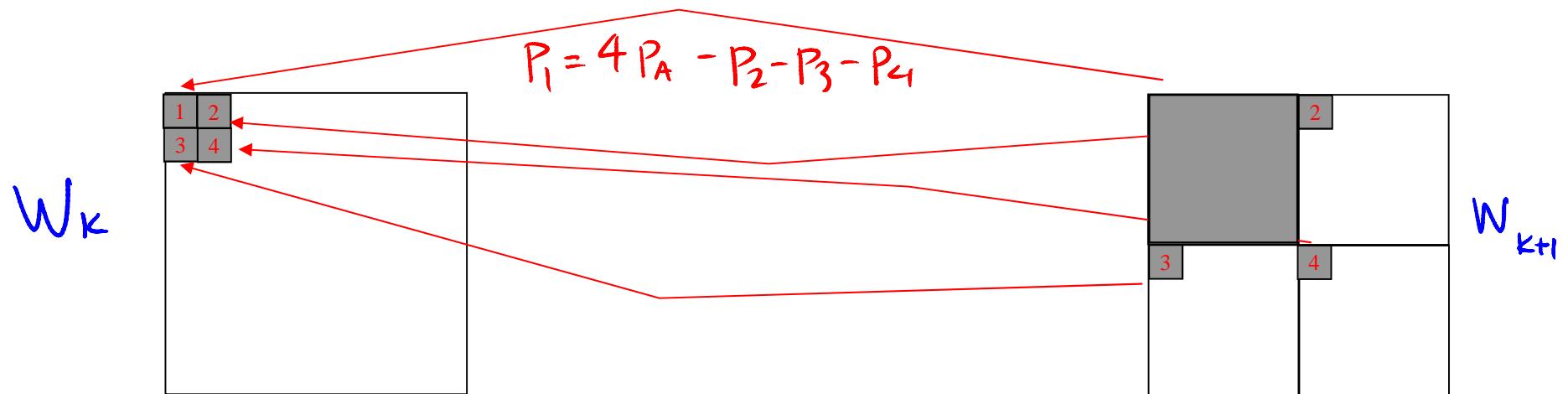
---

Is this invertible?

(i.e. Can we go from the wavelet transform  $W_0$ , to the original image?)

# Invertibility of the Transformation

Yes, because  $W_k$  can be reconstructed from  $W_{k+1}$



which means that  $W_0$  can be reconstructed from  $W_N$



# Topic 7:

## 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 coeffs
- Wavelet-based image compression
- The 2D Haar wavelet transform

# 1D Haar Wavelet Transform: Recursive Definition

---

The Haar Wavelet Basis:

- Simplest possible
- Discrete (non-continuous)
- 105 years old!

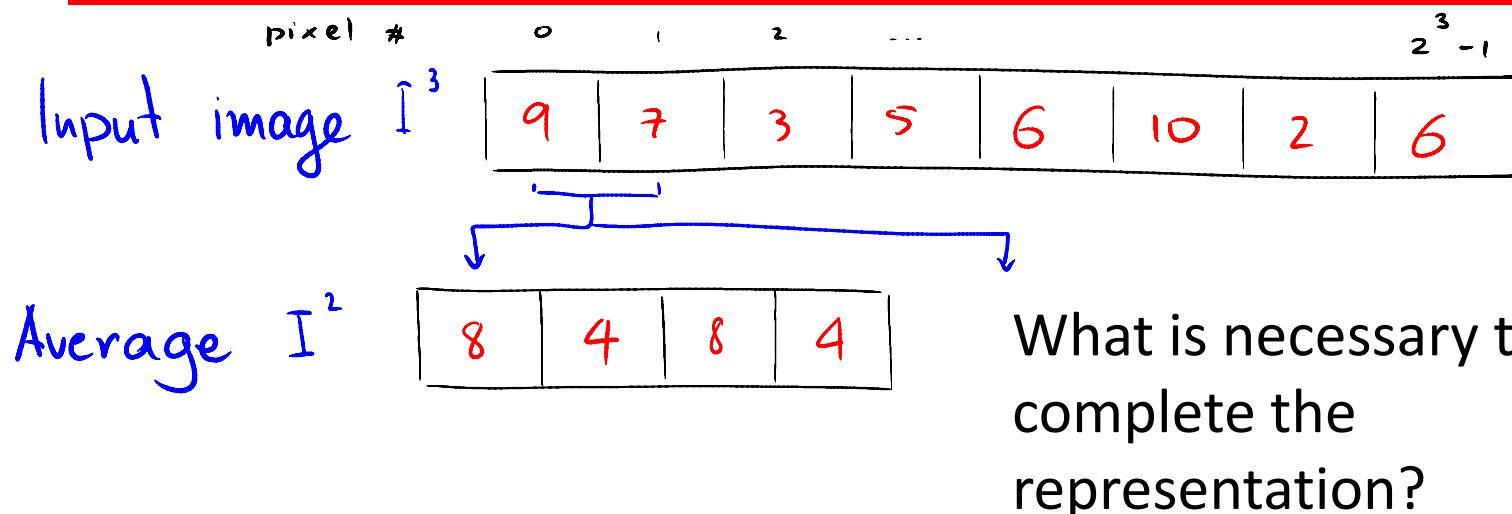
The discussion will start with an example.

# 1D Haar Wavelet Transform: Recursive Definition

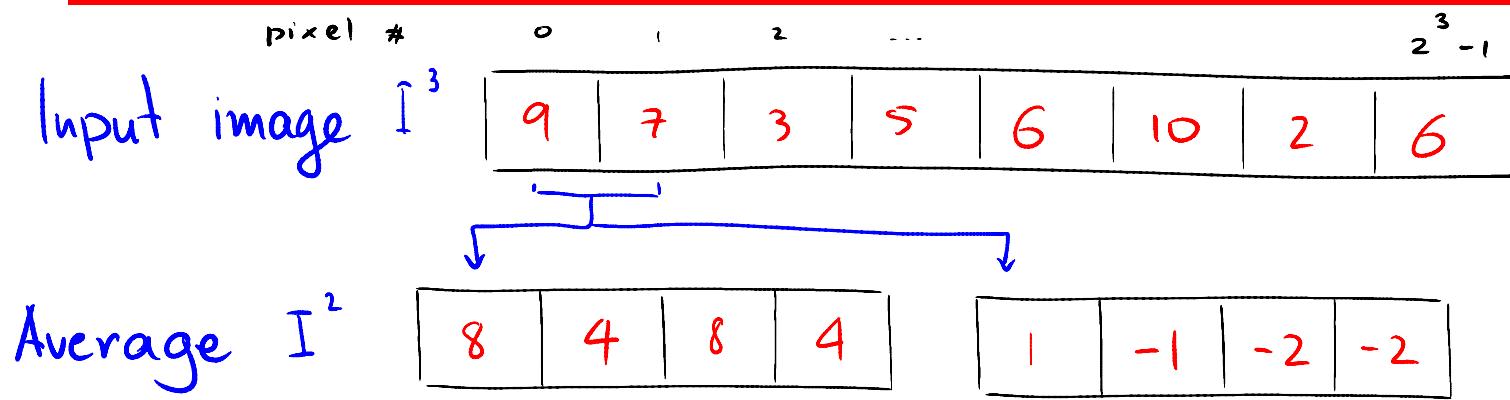
---

pixel #	0	1	2	...	$\frac{3}{2} - 1$			
Input image I	9	7	3	5	6	10	2	6

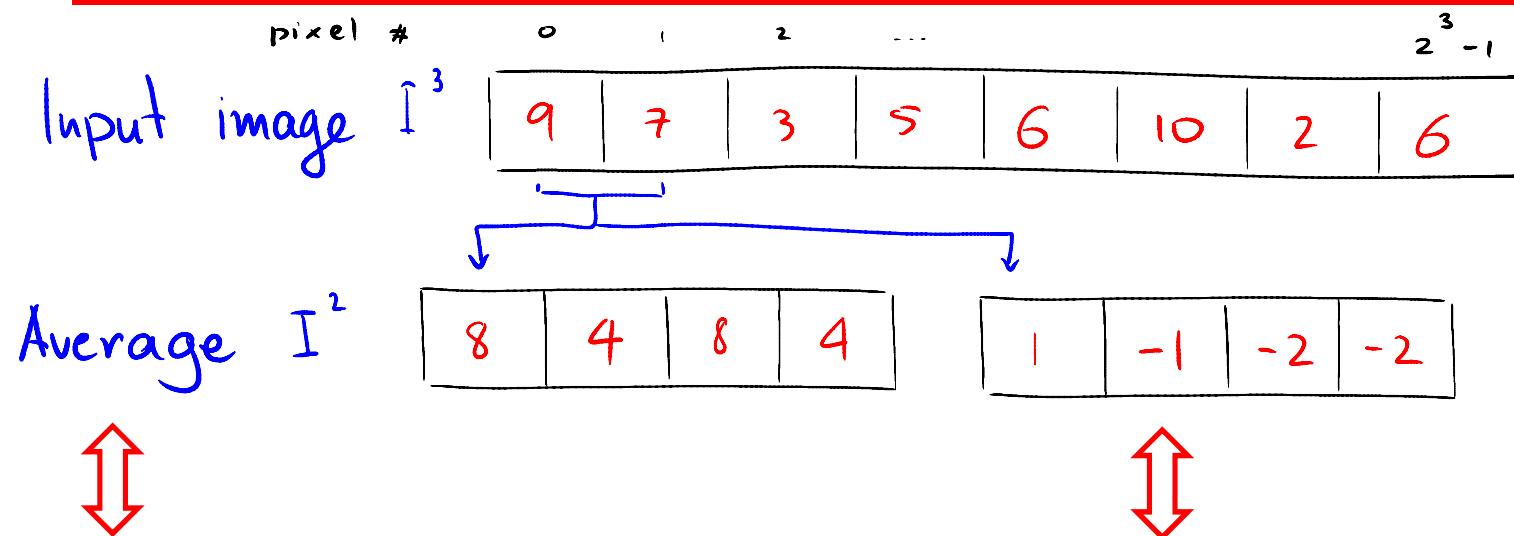
# 1D Haar Wavelet Transform: Recursive Definition



# 1D Haar Wavelet Transform: Recursive Definition



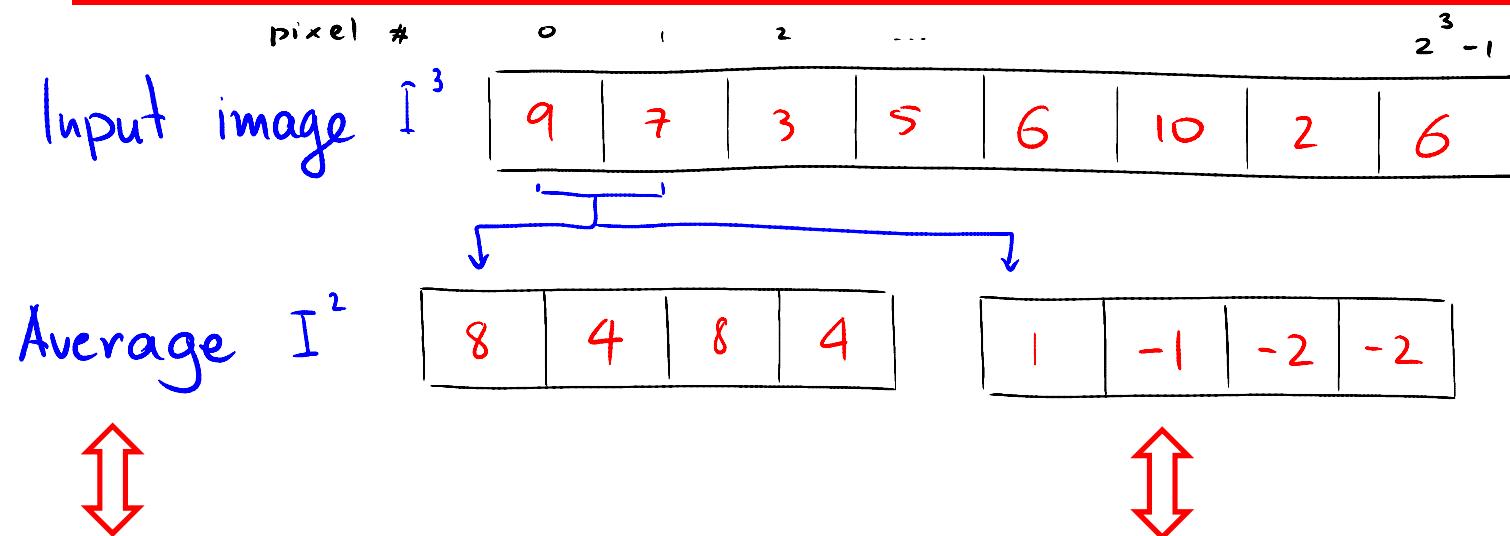
# 1D Haar Wavelet Transform: Recursive Definition



What is this  
analogous to?

What is this  
analogous to?

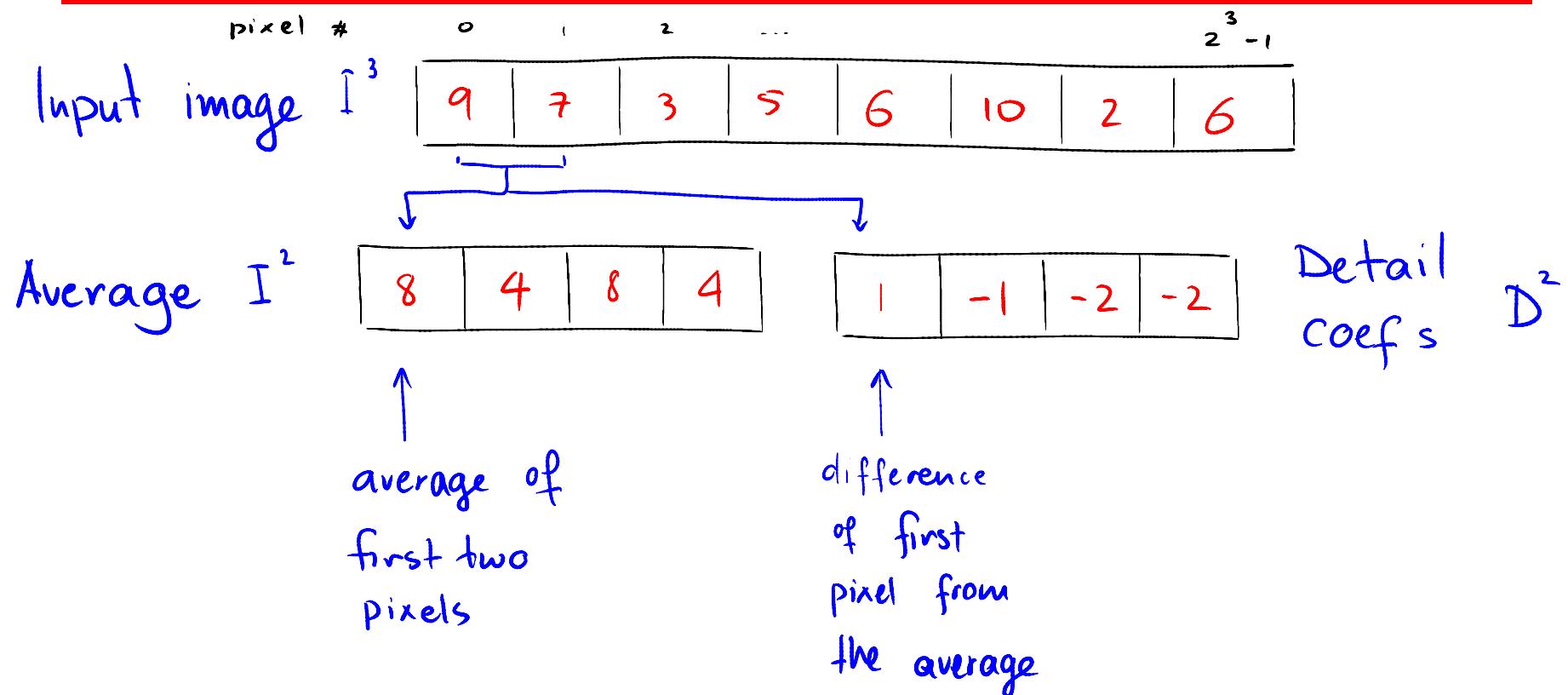
# 1D Haar Wavelet Transform: Recursive Definition



Analogous to  
 $G_{N+1}$  of the  
Gaussian Pyramid  
(blur and down-  
sample)

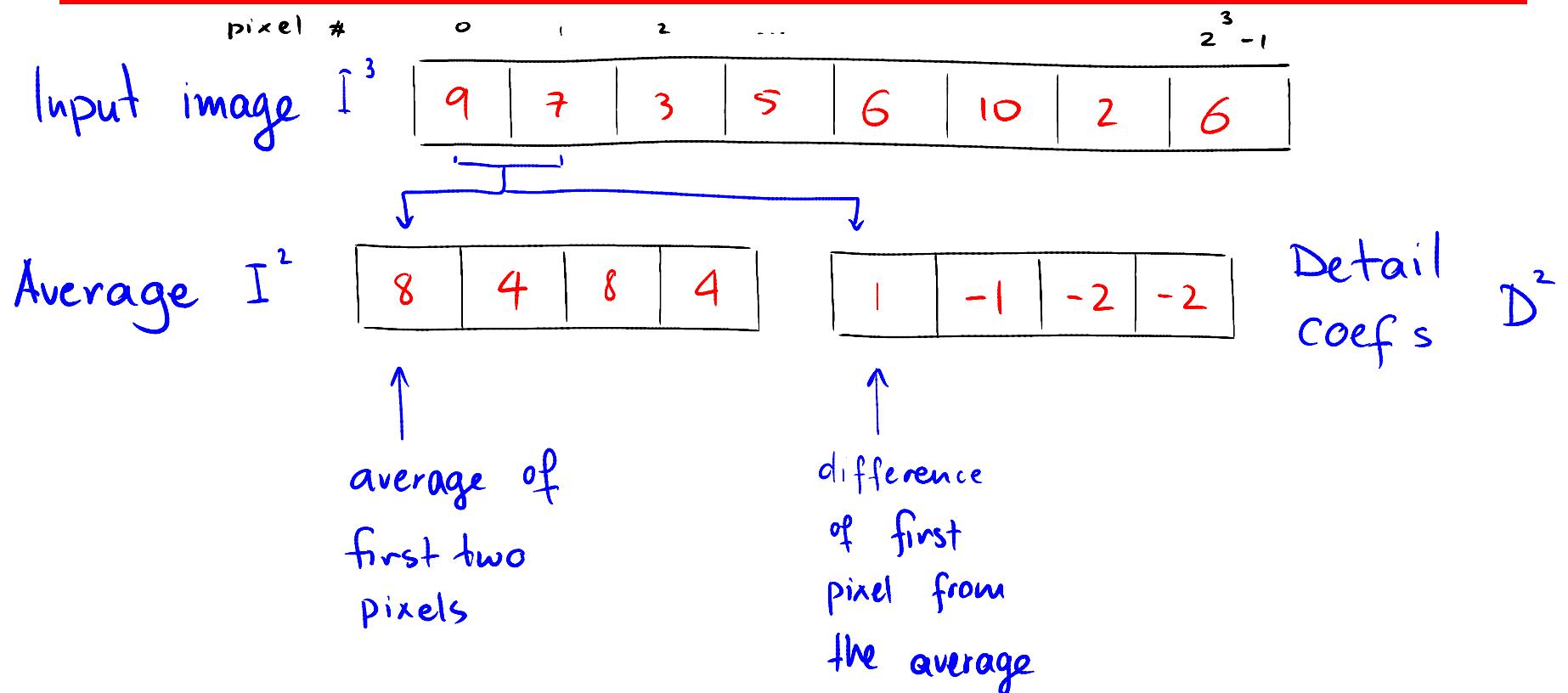
Analogous to  
the Laplacian  
(the detail)

# 1D Haar Wavelet Transform: Recursive Definition



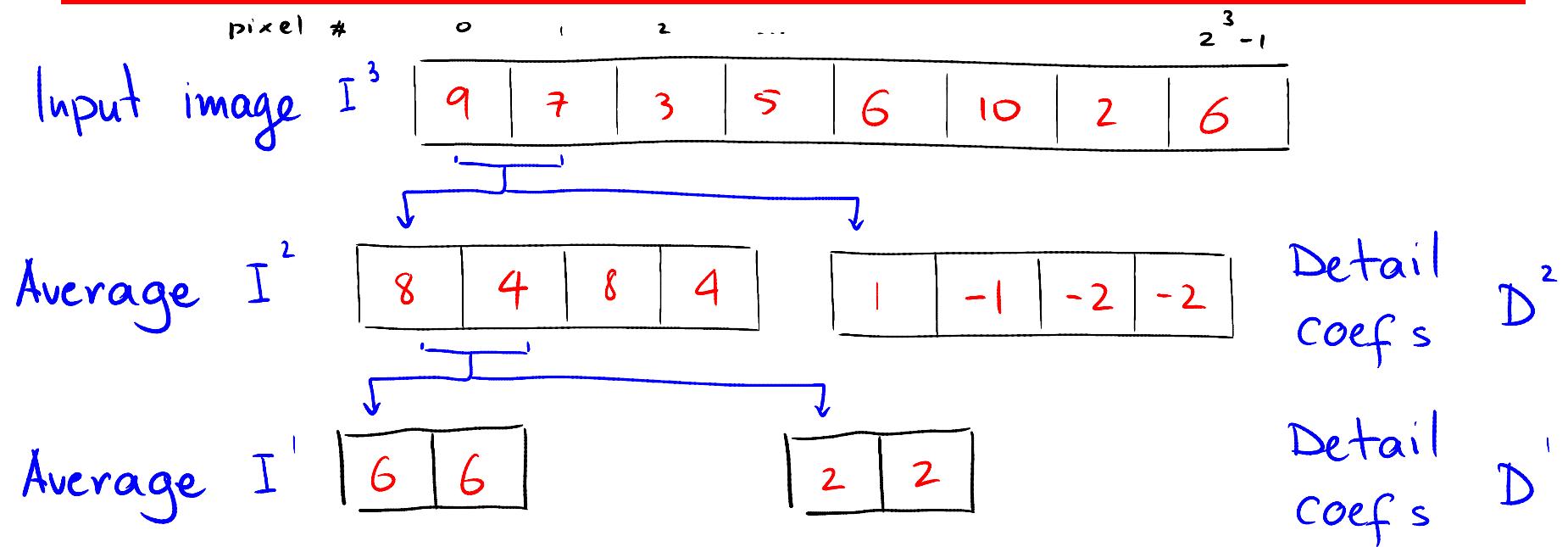
Do we need to store the difference of the 2<sup>nd</sup> pixel from the average?

# 1D Haar Wavelet Transform: Recursive Definition

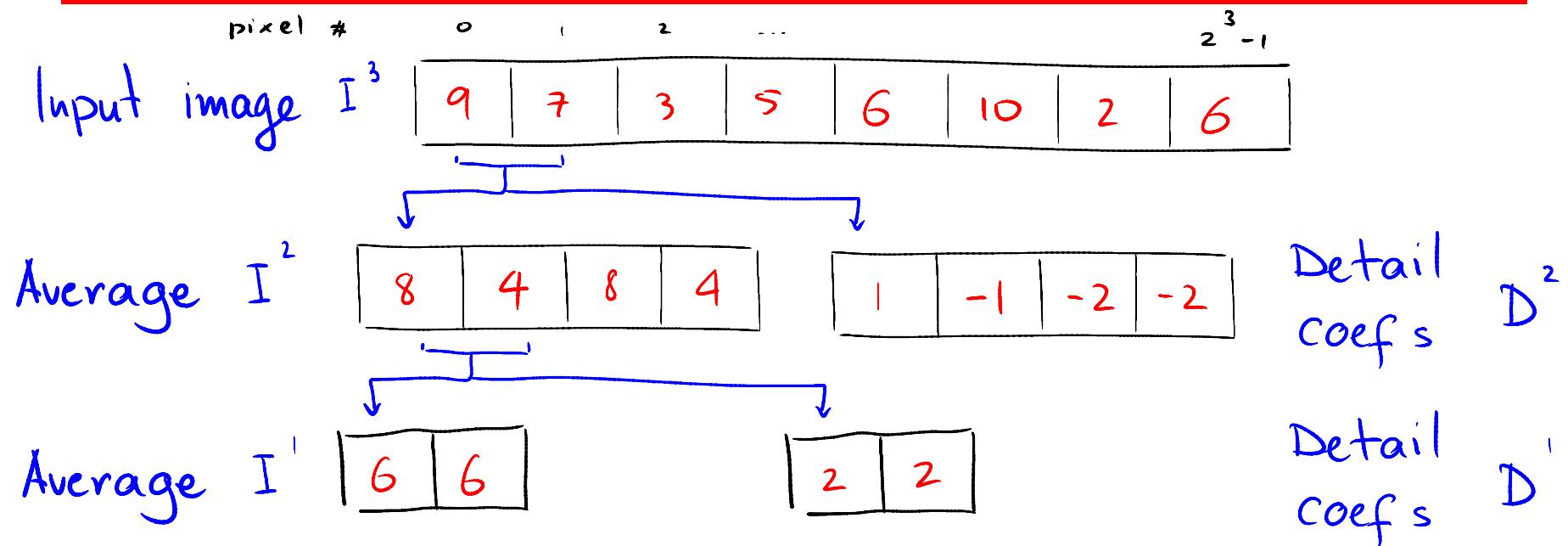


No need to store the difference of the 2<sup>nd</sup> pixel from the average!  
D<sup>0</sup> has  $\frac{1}{2}$  the size of the corresponding Laplacian L<sub>0</sub>

# 1D Haar Wavelet Transform: Recursive Definition



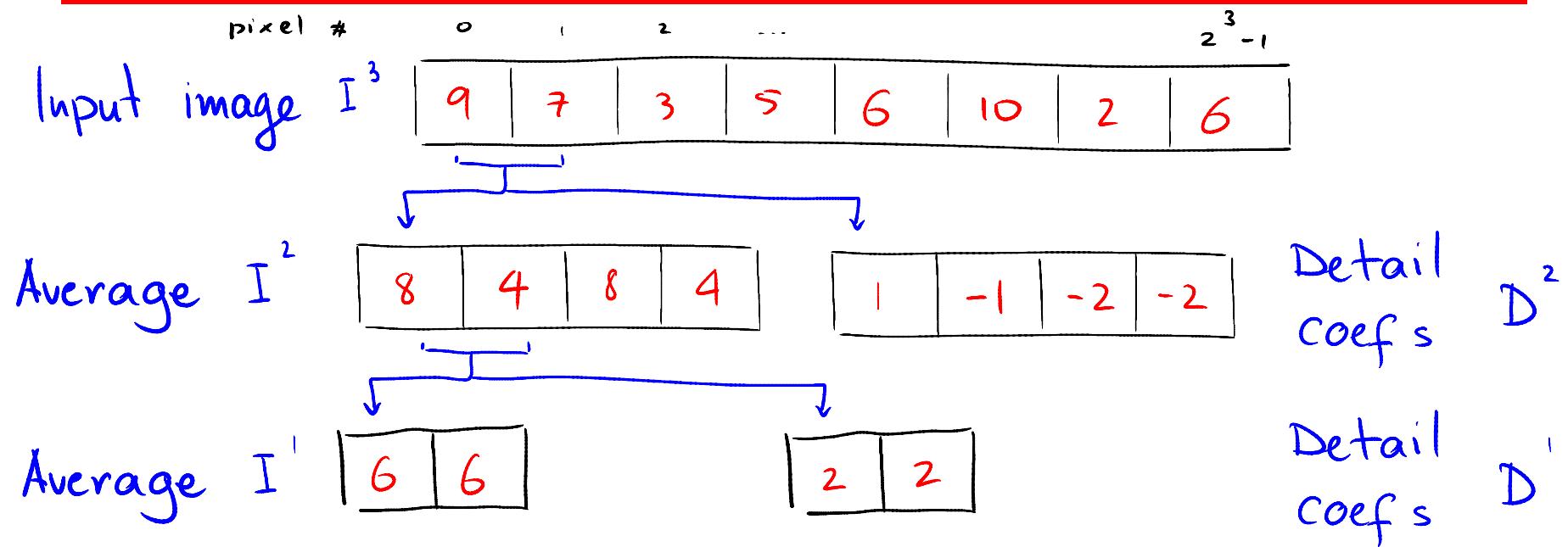
# 1D Haar Wavelet Transform: Recursive Definition



In general:

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

# 1D Haar Wavelet Transform: Recursive Definition

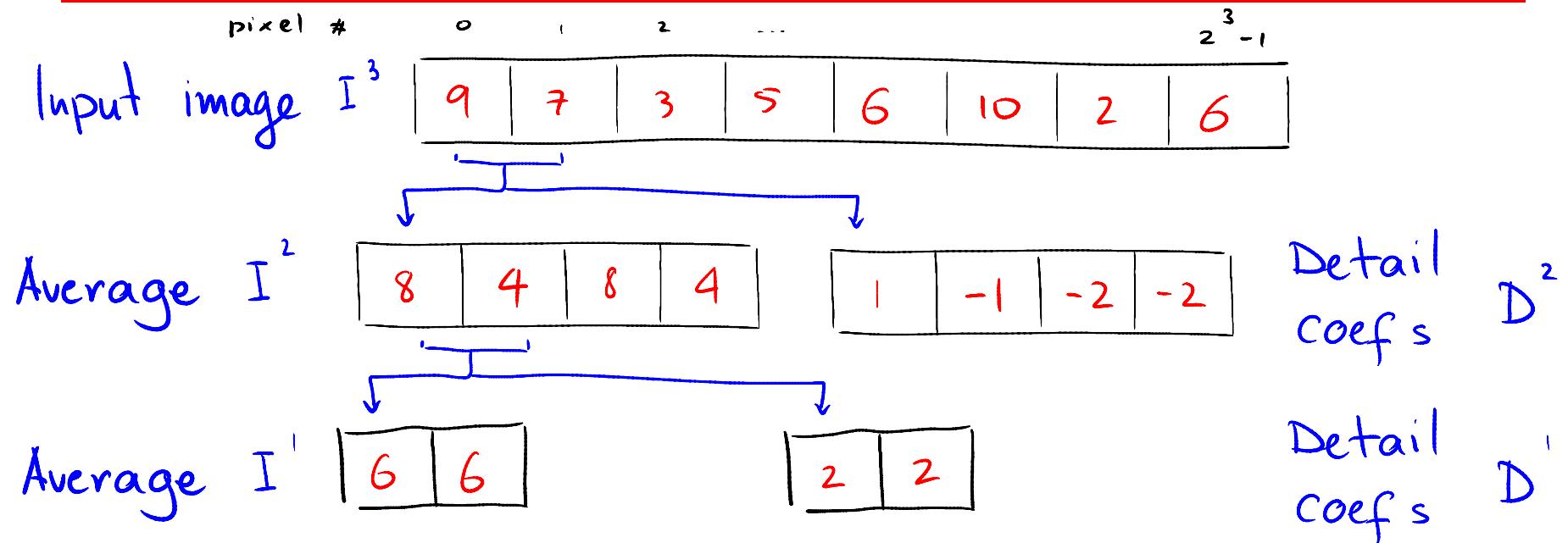


In general:

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

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

# 1D Haar Wavelet Transform: Recursive Definition



In general:

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

j-th level of "pyramid" contains  
2<sup>j</sup> pixels

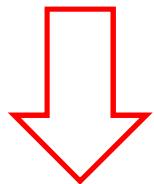
$$D_i^j = I_{2i}^{j+1} - \frac{1}{2} ( I_{2i}^{j+1} - I_{2i+1}^{j+1} )$$

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

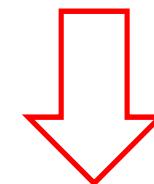
# 1D Haar Wavelet Transform: Recursive Definition

---

Can these two operations be written as convolutions?



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

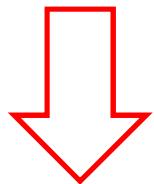


$$\begin{aligned} D_i^j &= I_{2i}^{j+1} - \frac{1}{2} \left( I_{2i}^{j+1} - I_{2i+1}^{j+1} \right) \\ &= \frac{1}{2} \left( I_{2i}^{j+1} - I_{2i+1}^{j+1} \right) \end{aligned}$$

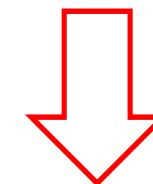
# 1D Haar Wavelet Transform: Recursive Definition

---

Can these two operations be written as convolutions?



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

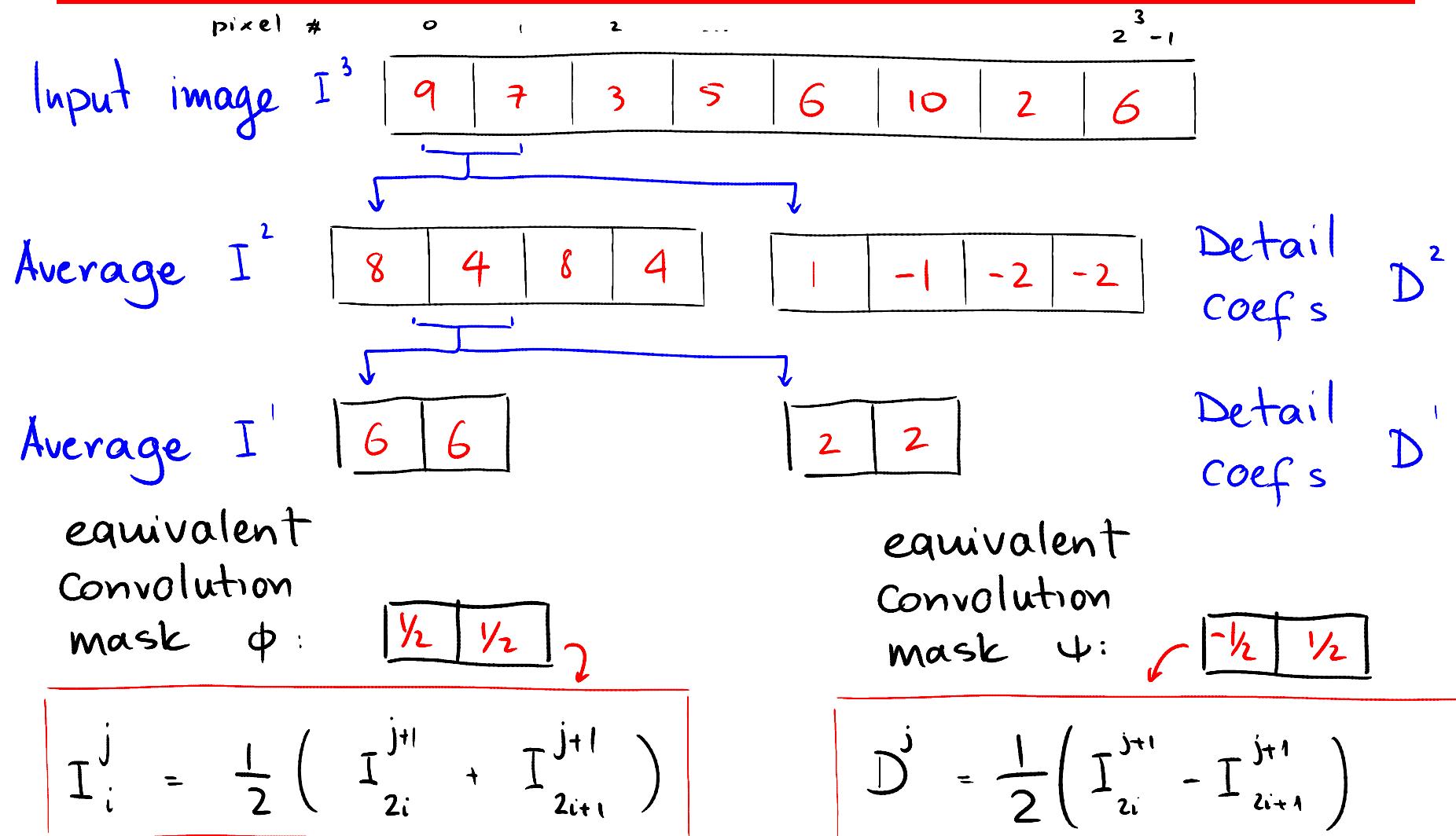


$$D_i^j = I_{2i}^{j+1} - \frac{1}{2} \left( I_{2i}^{j+1} - I_{2i+1}^{j+1} \right)$$

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

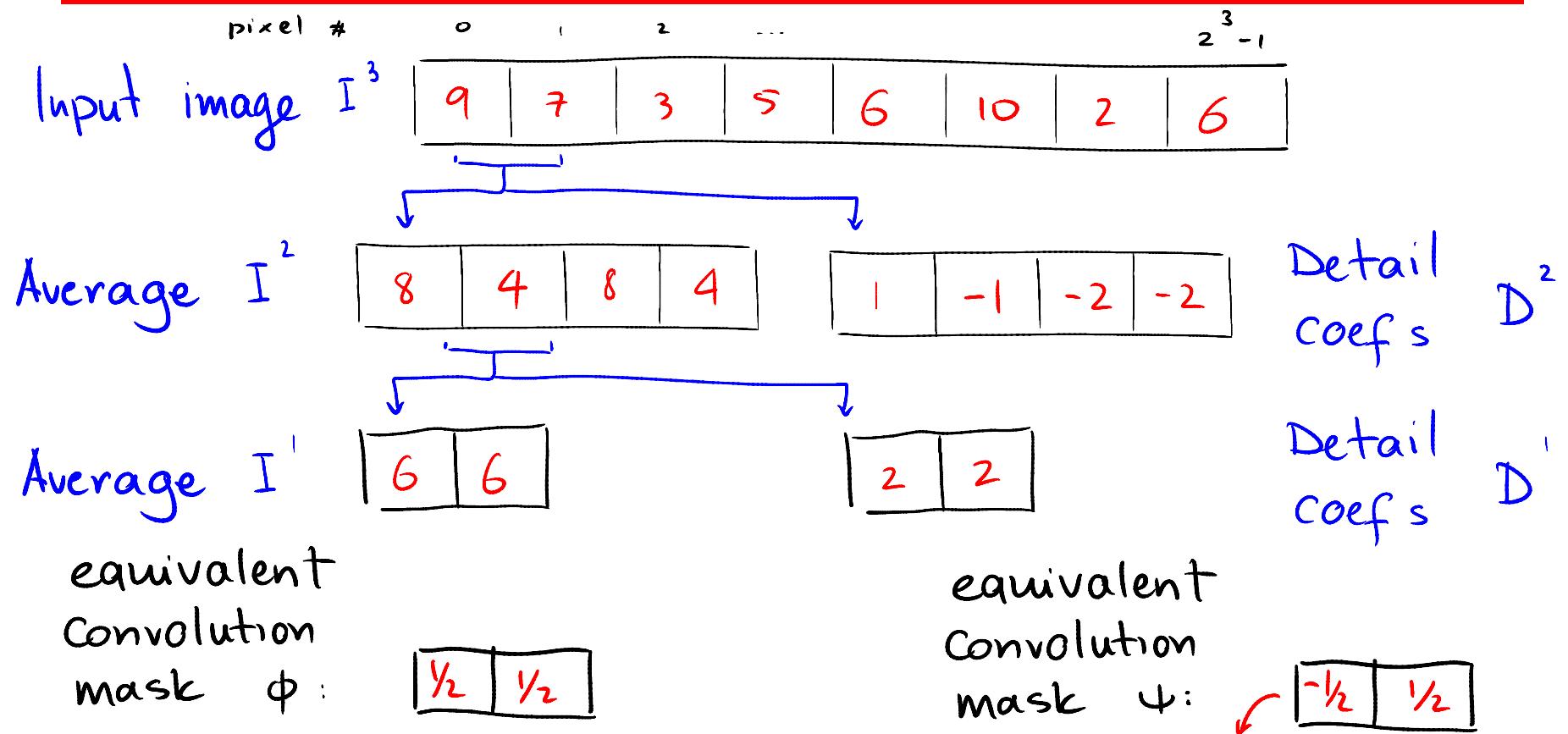
and what are the masks?

# 1D Haar Wavelet Transform: Recursive Definition



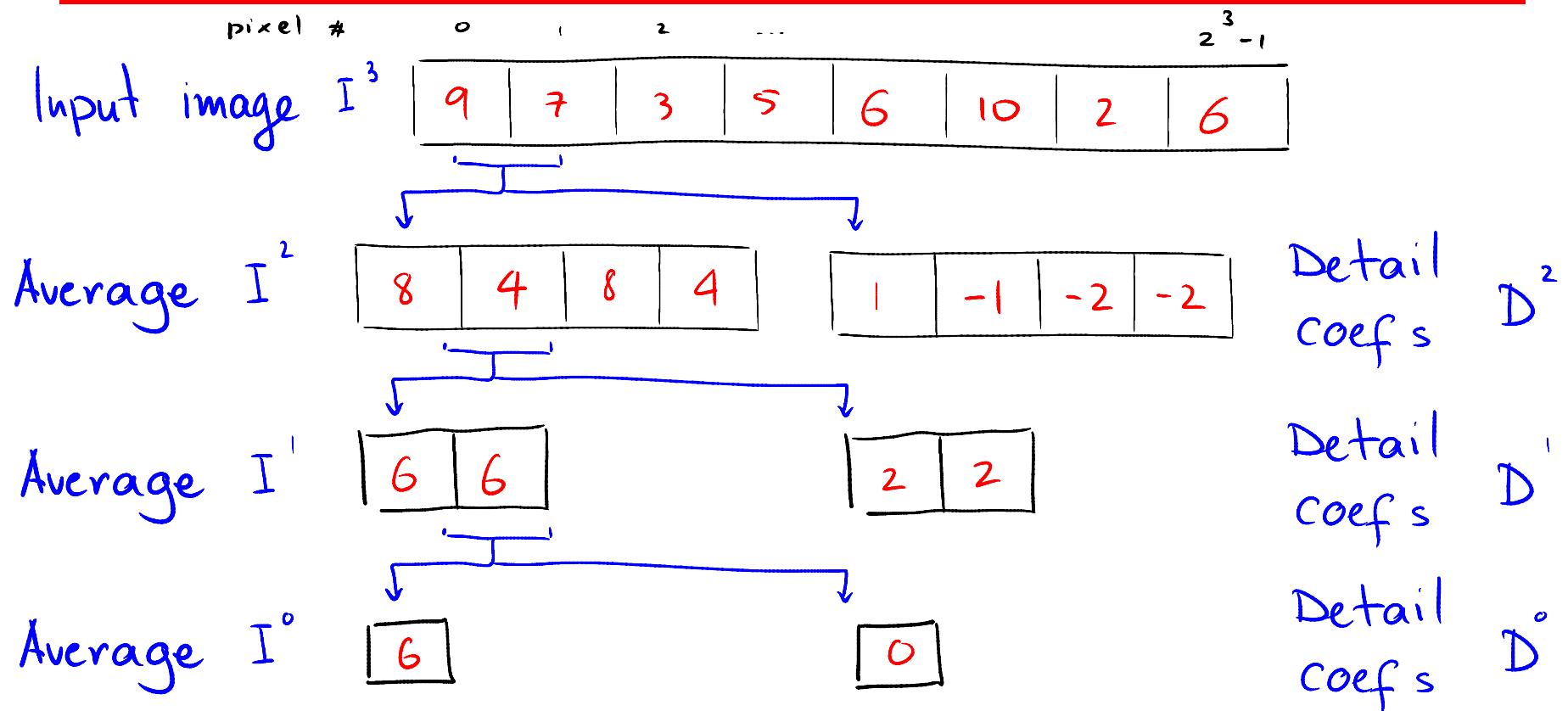
j-th level of "pyramid" contains  $2^j$  pixels

# 1D Haar Wavelet Transform: Recursive Definition

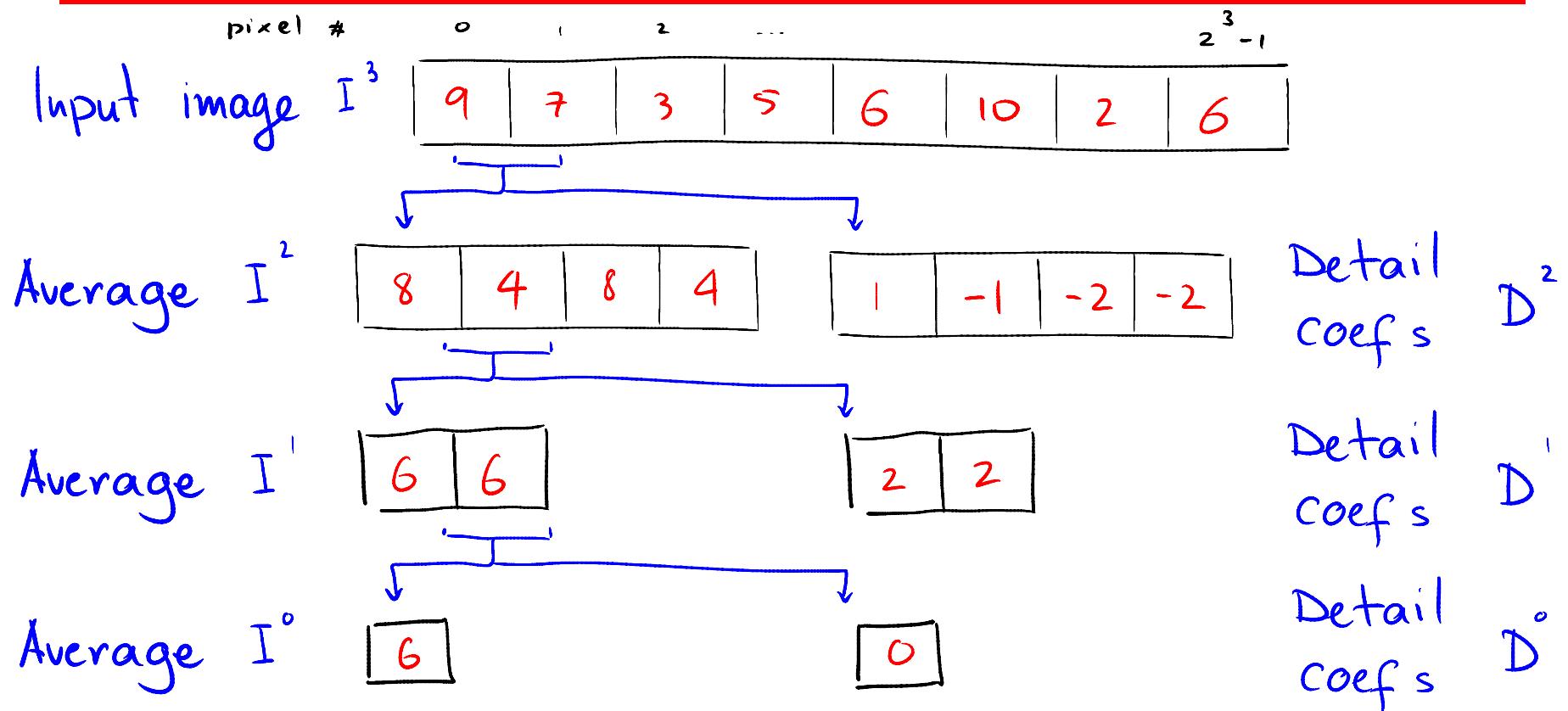


Let's use these masks to estimate the first level

# 1D Haar Wavelet Transform: Recursive Definition



# 1D Haar Wavelet Transform: Recursive Definition



What is the least amount of information that we need to store to recover  $I^3$  fully?

# 1D Haar Wavelet Transform: Recursive Definition

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

Wavelet  
transform  
contains these  
pixels

1	-1	-2	-2
---	----	----	----

6

2	2
---	---

0

Detail  
coefs

$D^2$

Detail  
coefs

$D^1$

Detail  
coefs

$D^0$

$I^0$     $D^0$     $D^1$   
 $\underbrace{\hspace{1cm}}$     $\underbrace{\hspace{1cm}}$     $\underbrace{\hspace{1cm}}$   
scale 0   scale 1   scale 2

Wavelet-  
transformed  
image

6	0	2	2	1	-1	-2	-2
---	---	---	---	---	----	----	----

---

Were we not using a basis representation?

# Topic 7:

## 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 coeffs
- Wavelet-based image compression
- The 2D Haar wavelet transform

# 1D Haar Wavelet Transform as a Matrix Product

pixel #	0	1	2	...	$2^{N-1}$			
Input image	9	7	3	5	6	10	2	6

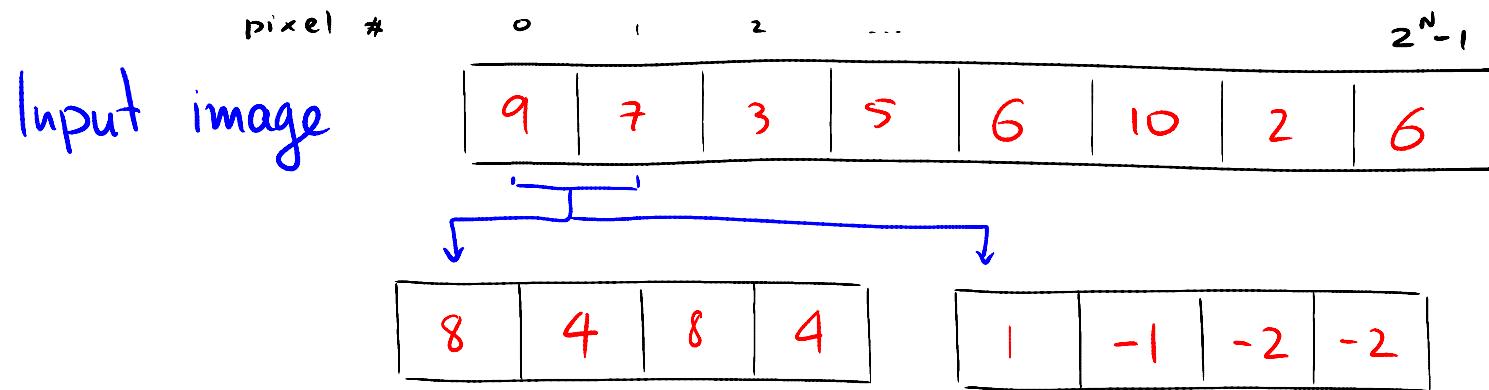
We know the result (because we just computed it), so let's start from the finest detail coefficients  $D^2$ . (Remember that the convolution mask was:  $\begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$ )

Wavelt transformed image

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

Original image

# 1D Haar Wavelet Transform as a Matrix Product

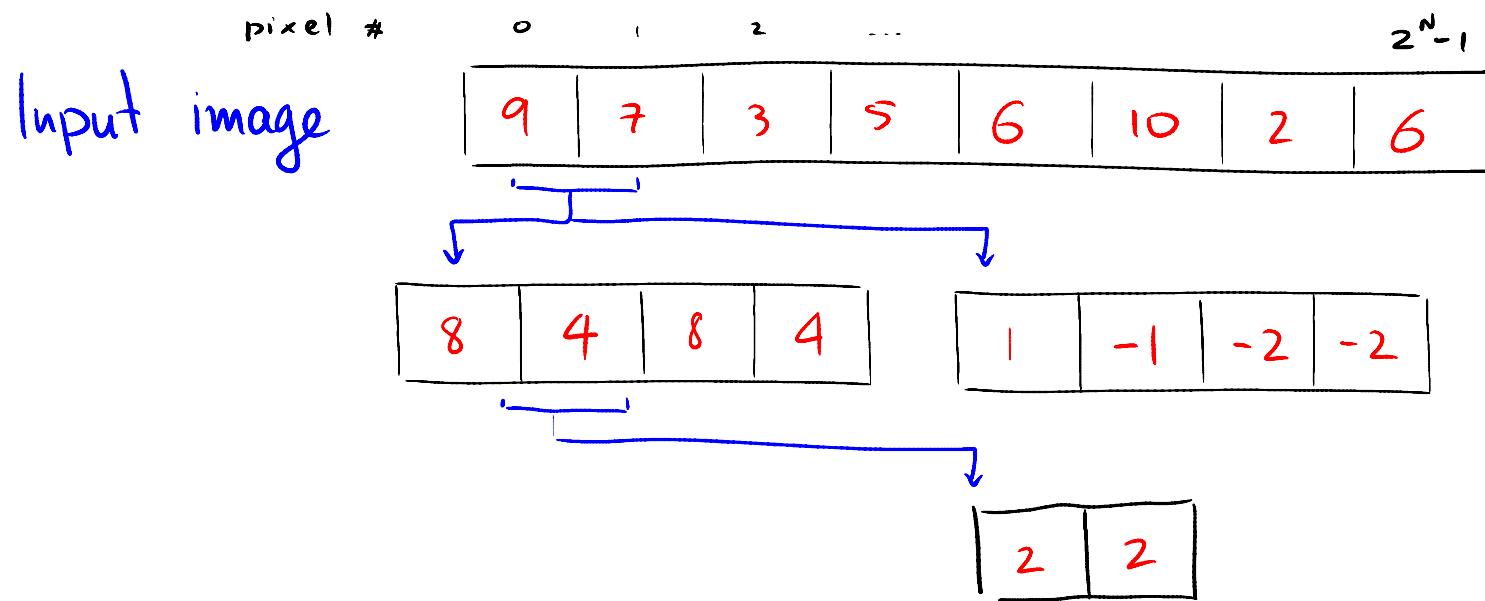


Wavelt transformed image

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

Original image

# 1D Haar Wavelet Transform as a Matrix Product



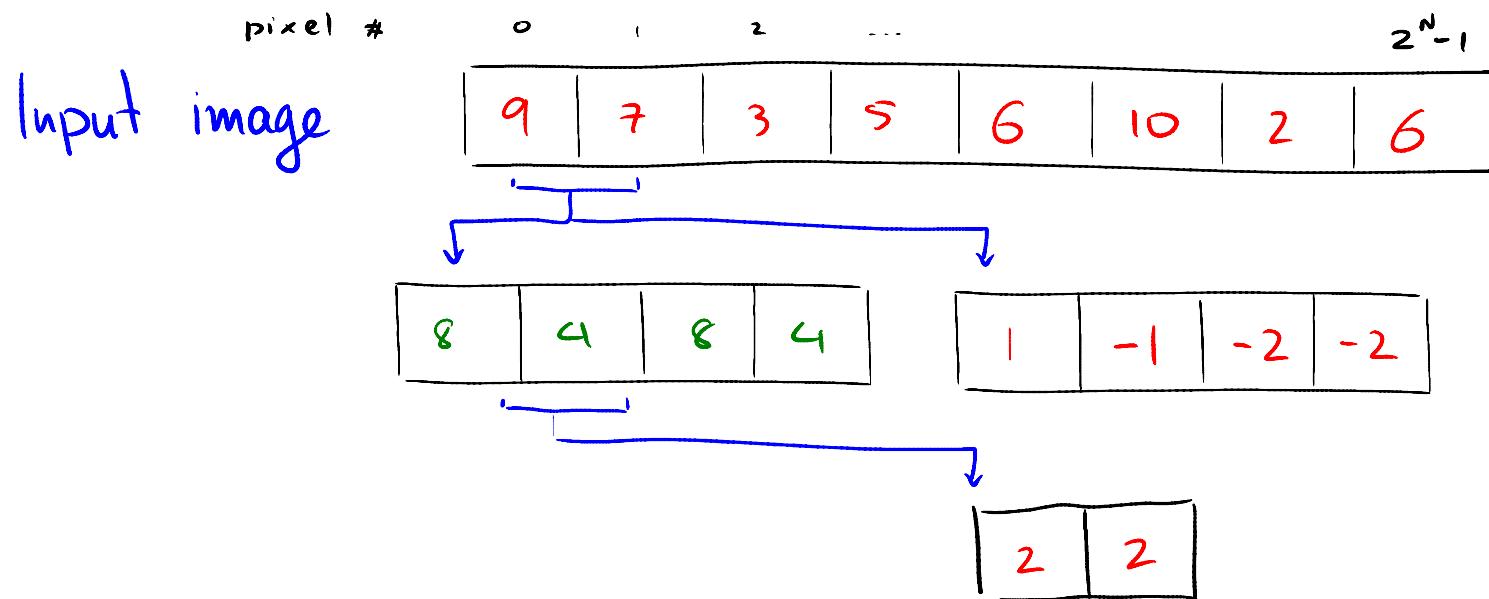
Wavelet transformed image

$$\begin{matrix}
 \overset{I}{\text{---}} & \overset{D}{\text{---}} & = & \overset{\text{---}}{\text{---}} & \overset{?}{\text{---}} & \overset{\text{---}}{\text{---}} & \overset{D^2}{\text{---}} \\
 \overset{D'}{\text{---}} & \overset{D'}{\text{---}} & & \overset{\text{---}}{\text{---}} & \overset{\bullet}{\text{---}} & \overset{\text{---}}{\text{---}} & \overset{D''}{\text{---}}
 \end{matrix}$$

Original image

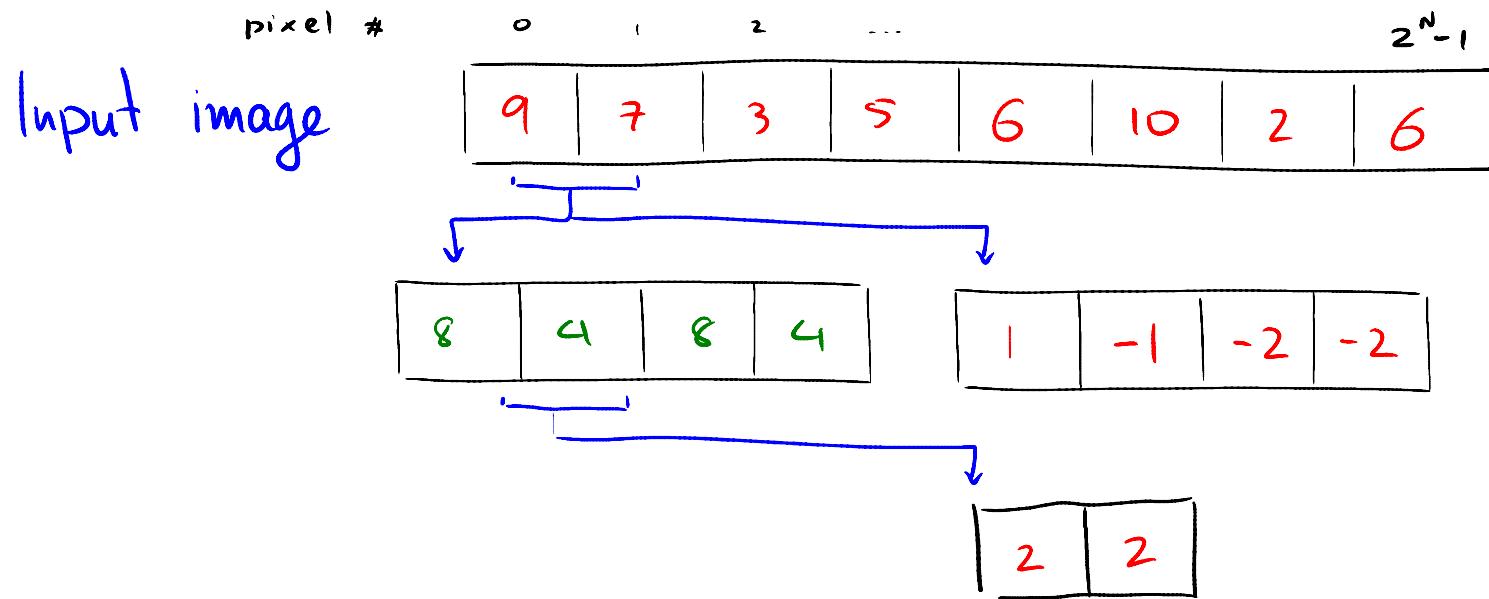
$$\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 4 \\ 8 \\ 4 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}$$

# 1D Haar Wavelet Transform as a Matrix Product



$$\begin{matrix}
 I^0 \\
 D^0 \\
 D^1 \\
 D^2 \\
 D^3 \\
 D^4
 \end{matrix} = \frac{1}{\sqrt{2}} \cdot \begin{matrix}
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
 \end{matrix} \cdot \begin{matrix}
 9 \\
 7 \\
 3 \\
 5 \\
 6 \\
 10 \\
 2 \\
 6
 \end{matrix}$$

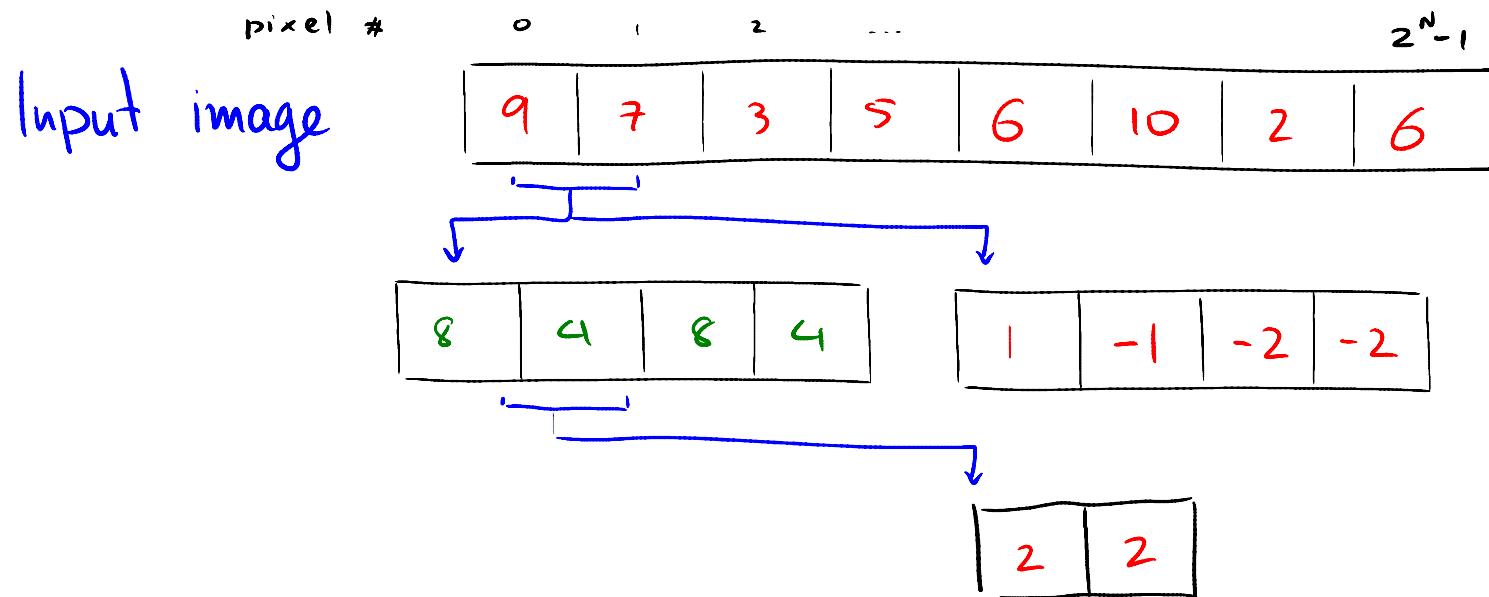
# 1D Haar Wavelet Transform as a Matrix Product



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

$$= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 8 \\ 4 \\ 8 \\ 4 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

# 1D Haar Wavelet Transform as a Matrix Product



$$\begin{matrix}
 I^0 \\
 D^0 \\
 D^1 \\
 D^2 \\
 D^3 \\
 D^4
 \end{matrix} = \begin{matrix}
 6 \\
 0 \\
 2 \\
 2 \\
 1 \\
 -1 \\
 -2 \\
 -2
 \end{matrix} \cdot \frac{1}{\sqrt{2}} \begin{matrix}
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
 \end{matrix} \cdot \begin{matrix}
 9 \\
 7 \\
 3 \\
 5 \\
 6 \\
 10 \\
 2 \\
 6
 \end{matrix}$$

# 1D Haar Wavelet Transform as a Matrix Product

---

3rd & 4th  
rows of  
product

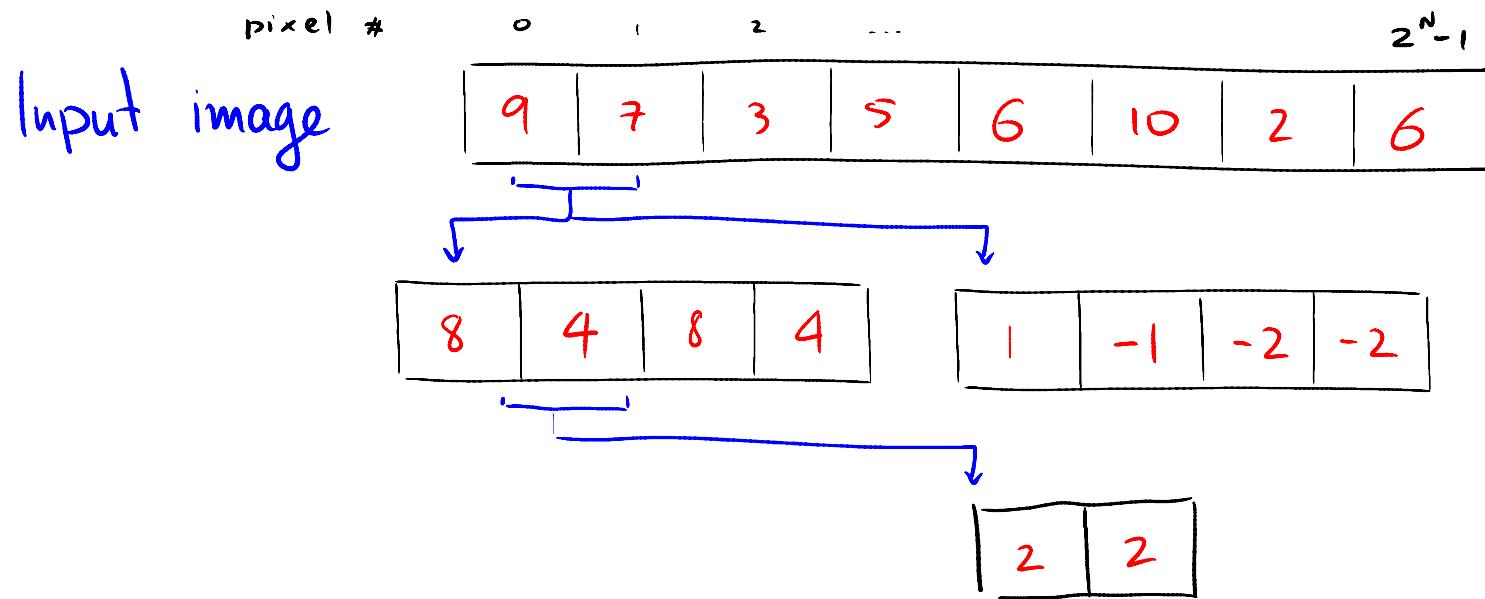
$$\left\{ \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \end{bmatrix} \right.$$

$$\begin{bmatrix} I^0 \\ D^0 \\ D^1 \\ D^2 \\ D^3 \end{bmatrix} =$$

$$\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

# 1D Haar Wavelet Transform as a Matrix Product

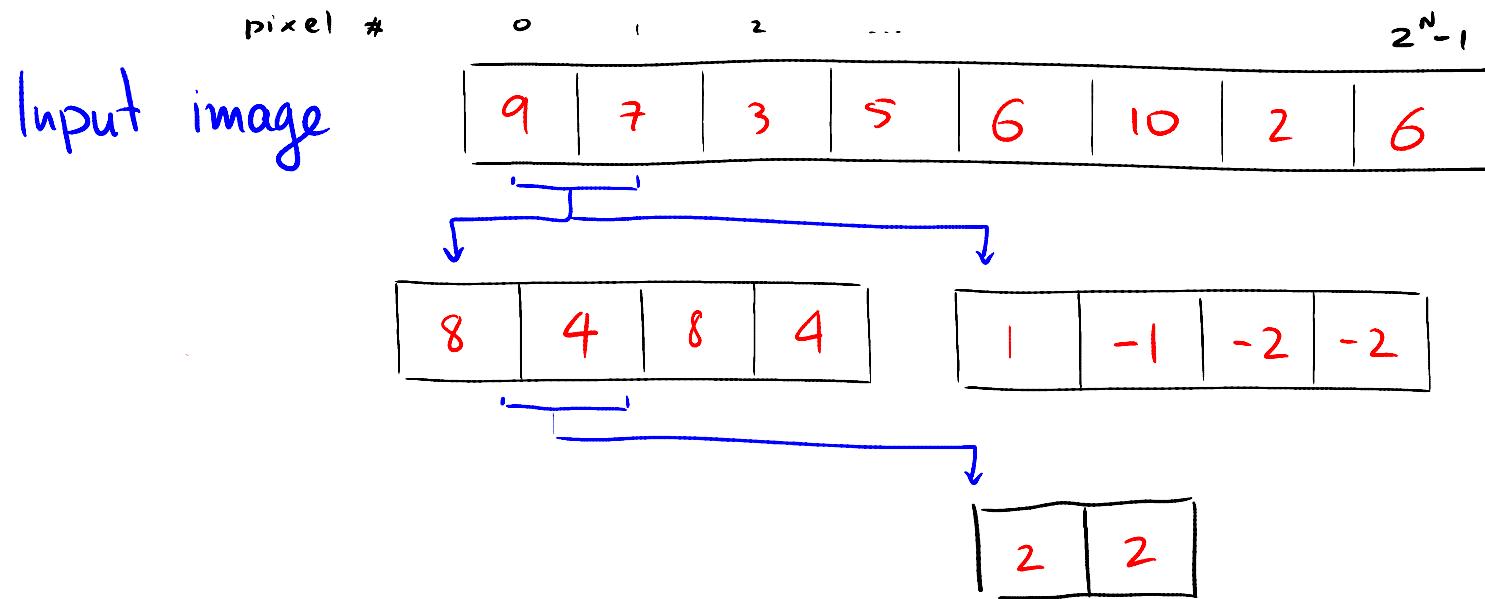


Wavelet transformed image

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

Original image

# 1D Haar Wavelet Transform as a Matrix Product



Wavelt transformed image

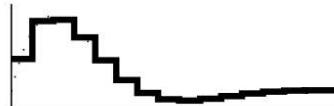
$$\begin{matrix}
 \frac{I}{D^0} & \left[ \begin{array}{c} 6 \\ 0 \end{array} \right] \\
 \frac{D^1}{D^1} & \left[ \begin{array}{c} 2 \\ 2 \end{array} \right] \\
 \frac{D^2}{D^2} & \left[ \begin{array}{c} 1 \\ -1 \\ -2 \\ -2 \end{array} \right]
 \end{matrix}
 = \frac{Y_8}{Y_4} \left[ \begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{array} \right] \left[ \begin{array}{c} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{array} \right]$$

Original image

# 1D Haar Wavelet Transform as a Matrix Product

---

Definitely, and unlike PCA it doesn't know anything about the underlying structure.



$V^4$  approximation



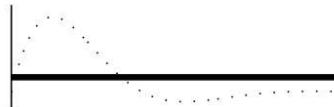
$V^3$  approximation



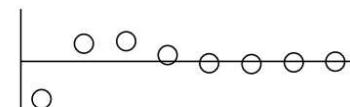
$V^2$  approximation



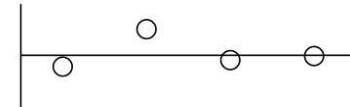
$V^1$  approximation



$V^0$  approximation



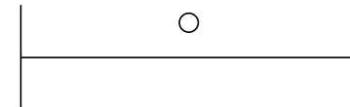
$W^3$  detail coefficients



$W^2$  detail coefficients



$W^1$  detail coefficients



$W^0$  detail coefficient

# The 1D Haar Wavelet Transform Matrix W

And the matrix W has interesting properties.

Wavelet transformed image

$$\begin{matrix} I^0 \\ D^0 \\ D^1 \\ D^2 \end{matrix} = \begin{matrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} & -\frac{1}{\sqrt{4}} & -\frac{1}{\sqrt{4}} & -\frac{1}{\sqrt{4}} \\ \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} & -\frac{1}{\sqrt{4}} & -\frac{1}{\sqrt{4}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{16}} & \frac{1}{\sqrt{16}} & -\frac{1}{\sqrt{16}} & -\frac{1}{\sqrt{16}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{16}} & \frac{1}{\sqrt{16}} & -\frac{1}{\sqrt{16}} & -\frac{1}{\sqrt{16}} & 0 & 0 & 0 \end{matrix} \begin{matrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{matrix}$$

Original image

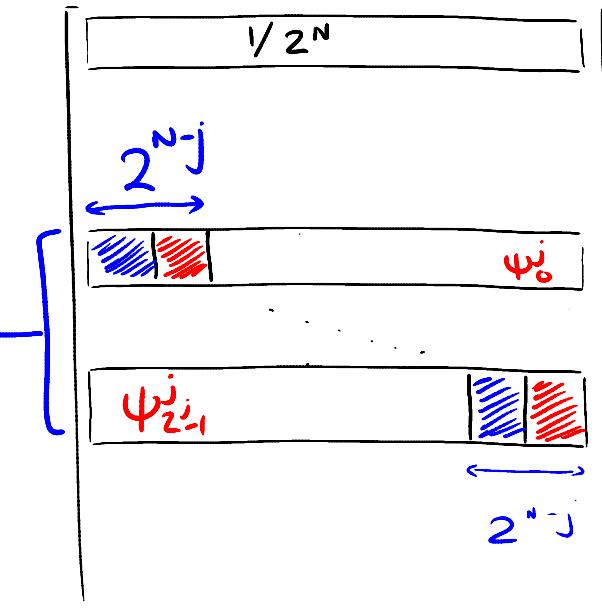
# The 1D Haar Wavelet Transform Matrix W

---

$$\begin{array}{c} I \\ D \\ D' \\ D^2 \end{array} = \begin{array}{c} \text{Diagram showing the Haar Wavelet Transform process:} \\ \text{Original Image: } \begin{bmatrix} 9 & 7 & 3 & 5 & 6 & 10 & 2 & 6 \end{bmatrix} \\ \text{Transformed Data: } \begin{bmatrix} 6 & 0 & 2 & 2 & -1 & -1 & -2 & -2 \end{bmatrix} \end{array}$$

# The 1D Haar Wavelet Transform Matrix W

- Matrix contains N scales
- Scale j represented by  $2^j$  rows  
 $\psi_0^j, \dots, \psi_{2^j-1}^j$



Wavelt transformed image

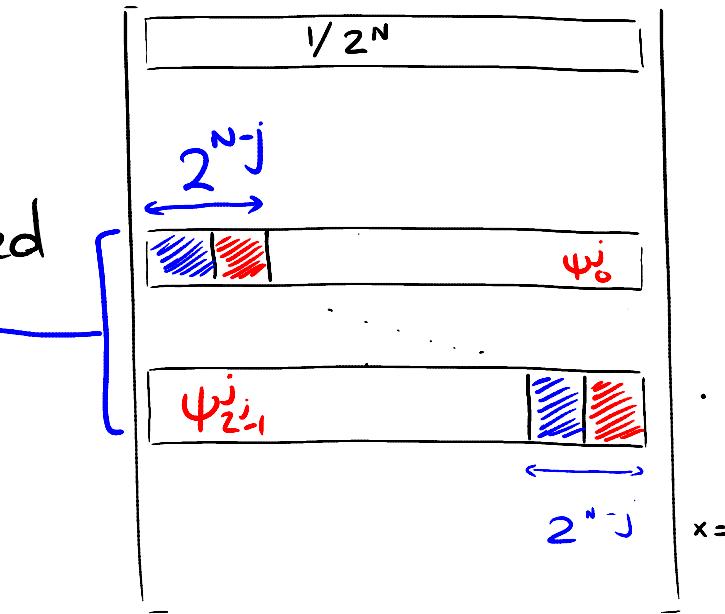
$$\begin{array}{c}
 \text{Wavelt} \\
 \text{transformed} \\
 \text{image}
 \end{array}
 \begin{array}{c}
 I \\
 D \\
 D' \\
 D \\
 D^2
 \end{array}
 \left[ \begin{array}{c} 6 \\ 0 \\ 2 \\ 2 \\ -1 \\ -1 \\ -2 \\ -2 \end{array} \right] = \left[ \begin{array}{c} \text{blue hatched} \\ \text{blue hatched} | \text{red hatched} \\ \text{blue} | \text{red} | 2^{j-N} | -2^{j-N} \\ \text{zero} | \text{blue hatched} | \text{red hatched} \\ \text{blue} | \text{red} \\ \text{white} | \text{blue} | \text{red} \\ \text{white} | \text{blue} | \text{red} | \text{blue} | \text{red} \end{array} \right] \left[ \begin{array}{c} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{array} \right]$$

Original image

# The 1D Haar Wavelet Transform Matrix W

- Matrix contains N scales

- Scale j represented by  $2^j$  rows  
 $\psi_0^j, \dots, \psi_{2^j-1}^j$



- Row  $\psi_j^i$  has  $\frac{2^n}{2^j} = 2^{n-j}$  non-zero pixels

They are pixels  $i 2^{n-j}, \dots, (i+1) 2^{n-j}-1$  with  $|\psi_j^i(x)| = \frac{1}{2^{n-j}}$

# Topic 7:

## 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 coeffs**
- Wavelet-based image compression
- The 2D Haar wavelet transform

# Reconstructing an Image from its Wavelet Coefs

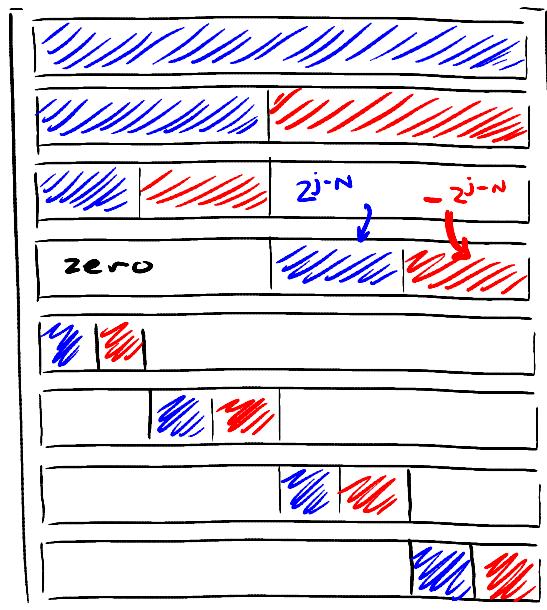
---

What is the dot product

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

of two distinct rows of  $W$ ?

$W =$



# Reconstructing an Image from its Wavelet Coefs

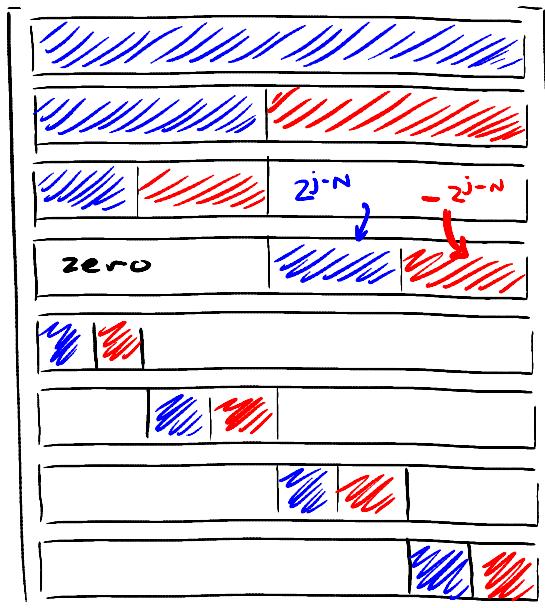
---

The dot product

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

for any two distinct rows of  $W$ .

$W =$



# Reconstructing an Image from its Wavelet Coefs

---

The dot product

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

for any two distinct rows of  $W$ .

This implies that  $WW^T$  is diagonal.

$$WW^T = \text{diagonal}$$
$$(\psi_i^j) \cdot (\psi_i^j)^T = \begin{bmatrix} & & 1 \\ & \ddots & \\ & & \frac{1}{2^{n-j}} \end{bmatrix}$$

## Reconstructing an Image from its Wavelet Coefs

---

Define  $\Lambda = WW^T$  with

$$\Lambda = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{2^{N_1}} \end{bmatrix}$$

This estimates the square magnitude ( $\lambda_j = \frac{1}{2^{N-j}}$ ) at each scale.

## Reconstructing an Image from its Wavelet Coefs

---

And because  $W$  is orthogonal, the inverse of  $W$  is its transpose:

$$W^{-1} = W^T$$

## Reconstructing an Image from its Wavelet Coefs

---

So we can assemble the matrix:

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

and use it to compute the image as

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

where  $C$  are the Haar wavelet coefficients.

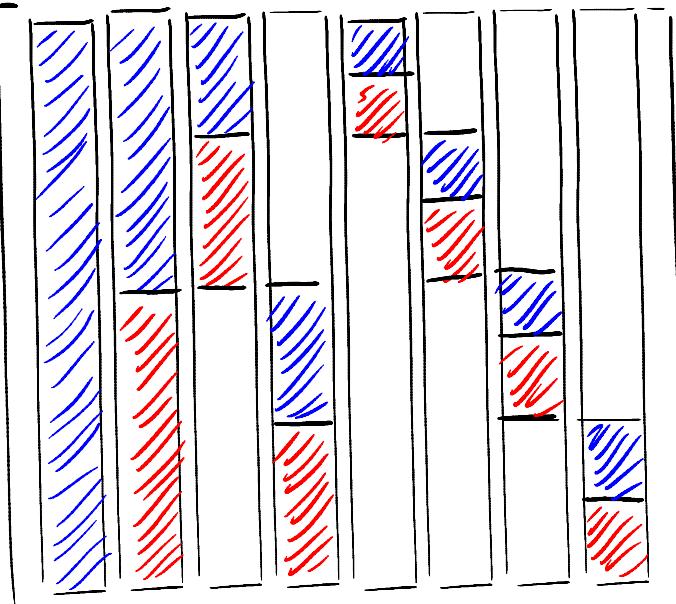
# Reconstructing an Image from its Wavelet Coefs

$$\hat{\Lambda}^{-1} = \begin{bmatrix} \hat{\sigma}_1^{-1} & 0 \\ 0 & \ddots \\ 0 & 0 & \hat{\sigma}_{2^{N_1}}^{-1} \end{bmatrix}$$

So we have

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

Original  
Image



# Interpreting the Wavelet Coefficients

$$I = \begin{matrix} \mathcal{D}_0^{-2} \cdot 6 \times & \text{[blue hatched]} \\ \mathcal{D}_1^{-2} \cdot 0 \times & \text{[blue hatched] / red hatched} \\ 2 \times & \text{[blue hatched] / red hatched / } \\ 2 \times & \text{[ ] / blue hatched / red hatched} \\ 1 \times & \text{[blue hatched] / red hatched} \\ (-1) \times & \text{[ ] / blue hatched / red hatched} \\ (-2) \times & \text{[ ] / blue hatched / red hatched} \\ \mathcal{D}_{2^{N-1}}^{-2} \cdot (-2) \times & \text{[ ] / blue hatched / red hatched} \end{matrix} + + + + + + +$$

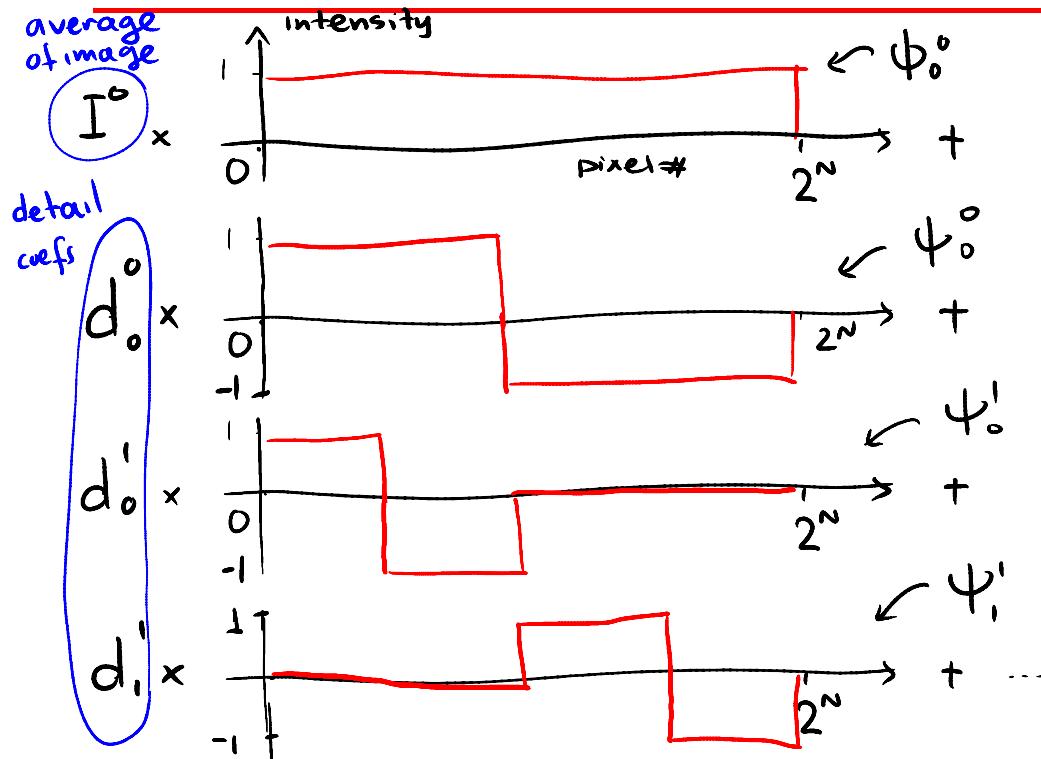
$$W =$$

$$\begin{matrix} \text{[blue hatched]} \\ \text{[blue hatched] / red hatched} \\ \text{[blue hatched] / red hatched / } \\ \text{[blue hatched] / red hatched / } \\ \text{zero} \\ \text{[blue hatched] / red hatched} \\ \text{[ ] / blue hatched / red hatched} \\ \text{[ ] / blue hatched / red hatched} \\ \text{[ ] / blue hatched / red hatched} \end{matrix}$$

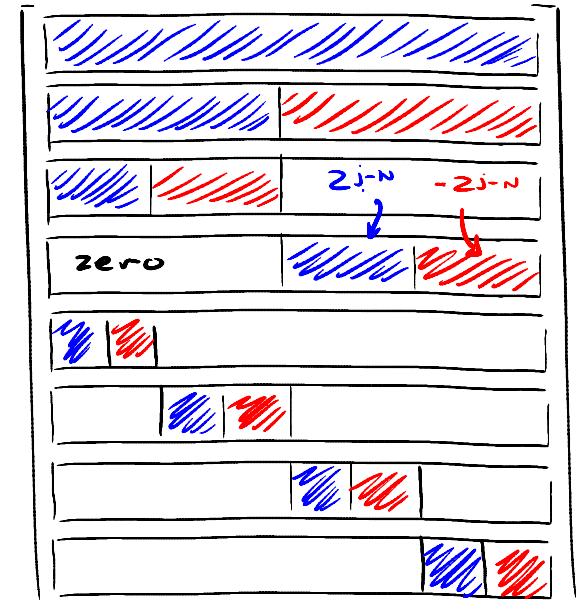
$2^{j-N}$        $-2^{j-N}$

⇒ By multiplying  $I$  with  $W$  we obtain a decomposition of the image into a sequence of basis images  $\psi_0^{\circ}, \psi_0^{'}, \dots, \psi_j^{'}, \dots$  that form an orthogonal basis of  $\mathbb{R}_{2^N}$

# Interpreting the Wavelet Coefficients



$W =$



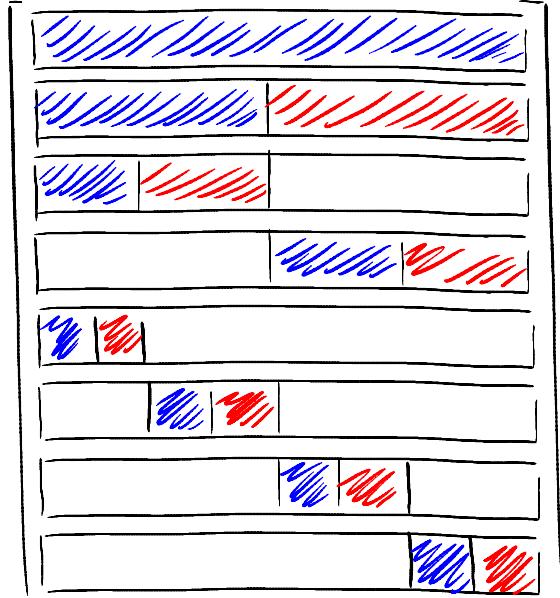
⇒ The wavelet coefficients are the coordinates of the image, considered as a vector in  $\mathbb{R}^{2^N}$ , in the basis defined by images  $\phi_0^0, \phi_0^1, \phi_1^1, \dots$

# The Normalized Haar Wavelet Matrix

We can normalize the wavelet transform matrix by multiplying

$$\tilde{W} = \begin{bmatrix} \sqrt{\alpha_1} & & \\ & \ddots & \\ & & \sqrt{\alpha_{2^N-1}} \end{bmatrix} \cdot W$$

$$\tilde{W} =$$



normalized  
wavelet coefficients

$$\begin{bmatrix} c_0 \\ d_0 \\ d_1 \\ \vdots \end{bmatrix}$$

$$=$$

$$\tilde{W} \cdot$$

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

Original  
Image

# The Normalized Haar Wavelet Coefficients

$$I = \begin{matrix} c_0^0 & \times & \text{[Blue]} \\ d_0^0 & \times & \text{[Blue]} | \text{Red} \\ d_0^1 & \times & \text{[Blue]} | \text{Red} | \text{[ ]} \\ d_1^1 & \times & \text{[ ]} | \text{[Blue]} | \text{Red} \\ d_0^2 & \times & \text{[Blue]} | \text{Red} | \text{[ ]} \\ d_1^2 & \times & \text{[ ]} | \text{[Blue]} | \text{Red} \\ d_2^2 & \times & \text{[ ]} | \text{[Blue]} | \text{Red} | \text{[ ]} \\ d_3^2 & \times & \text{[ ]} | \text{[Blue]} | \text{Red} | \text{[ ]} \end{matrix} \xrightarrow{\psi_0^0}$$

$$\tilde{W} = \begin{matrix} \text{[Blue]} \\ \text{[Blue]} | \text{Red} \\ \text{[Blue]} | \text{Red} | \text{[ ]} \\ \text{[ ]} | \text{[Blue]} | \text{Red} \\ \text{[Blue]} | \text{Red} | \text{[ ]} \\ \text{[ ]} | \text{[Blue]} | \text{Red} \\ \text{[ ]} | \text{[Blue]} | \text{Red} | \text{[ ]} \\ \text{[ ]} | \text{[Blue]} | \text{Red} | \text{[ ]} \end{matrix}$$

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

# Topic 7:

## 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 coeffs
- **Wavelet-based image compression**
- The 2D Haar wavelet transform

# Wavelet Compression Algorithm #1

---

Input: 1D Image  $I$ , desired compression  $k$

Output:  $k2^N$  coefficients.

What will the algorithm be?

# Wavelet Compression Algorithm #1

---

Input: 1D Image  $I$ , desired compression  $k$

Output:  $k2^N$  coefficients.

1. Compute  $\tilde{W} \leftarrow$
2. Sort the coefficients  $c_0^0, d_0^0, d_1^0, \dots$  in order of decreasing absolute value
3. Keep the top  $k2^N$  coefficients (we know the basis)

# Wavelet Compression Algorithm #1

---

Input: 1D Image  $I$ , desired error  $\epsilon$

Output:  $k2^N$  coefficients.

1. Compute  $\tilde{W} \cdot I$
2. Sort the coefficients  $c_0^0, d_0^0, d_1^0, \dots$  in order of decreasing absolute value
3. Keep the enough coefficients to satisfy  $|\tilde{I} - I| < \epsilon$

# Topic 7:

## 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 coeffs
- Wavelet-based image compression
- The 2D Haar wavelet transform

# The 2D Haar Wavelet Transform

---

To compute a 2D Haar Wavelet do:

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

# The 2D Haar Wavelet Transform

---

Show that every 2D wavelet coefficient can be expressed as the dot product of the Image I and an image defined by

$$(\Psi_i^j)^T \cdot (\Psi_i^{j'})$$

where  $\Psi_i^j$  are the 1D Haar basis images.

# The 2-D Haar Wavelet Basis

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

