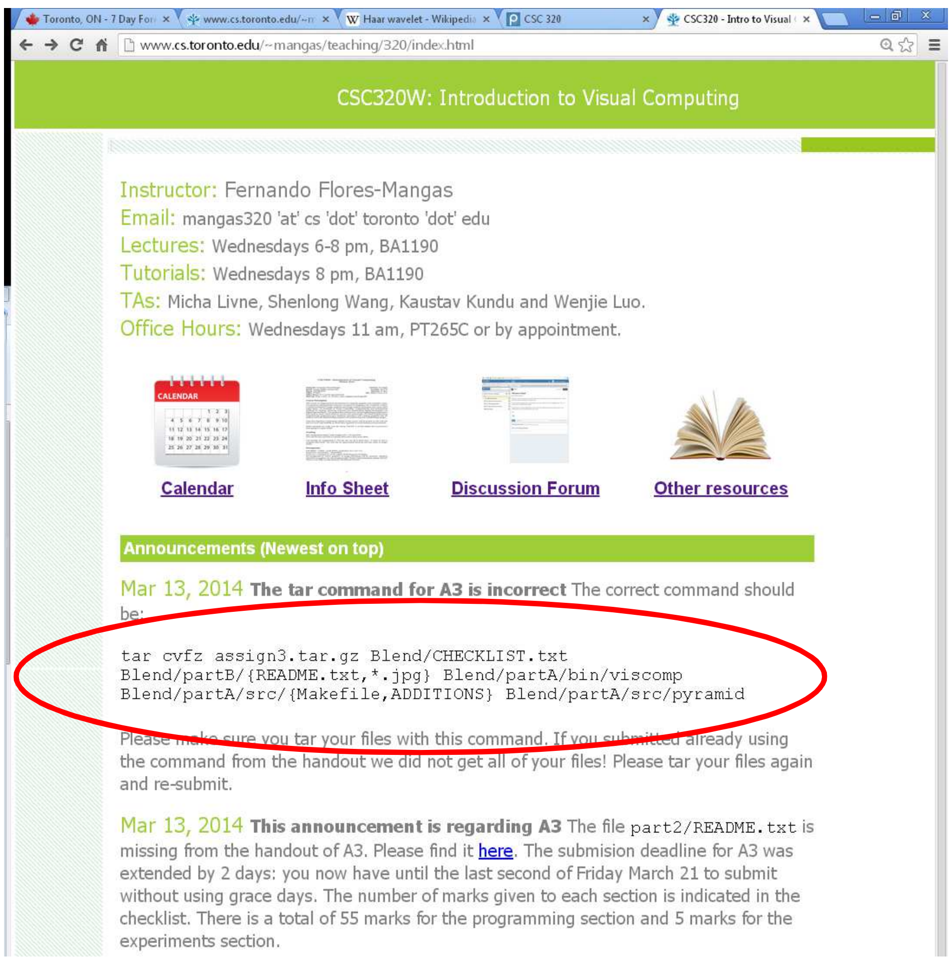
Topic 7:

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 coefs
- 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.



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?

$$(2^N+1) + (2^{N-1}+1) + \cdots (2^0+1)$$

L_0 g_N

$$1 + 1/4 + 1/16 + \cdots = 4/3$$

The representation is <u>over-complete</u>! (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:

Image

Vector in high-dimensional space

Laplacian Pyramid

Wavelet-Based Image Representations

We know we can represent images as:

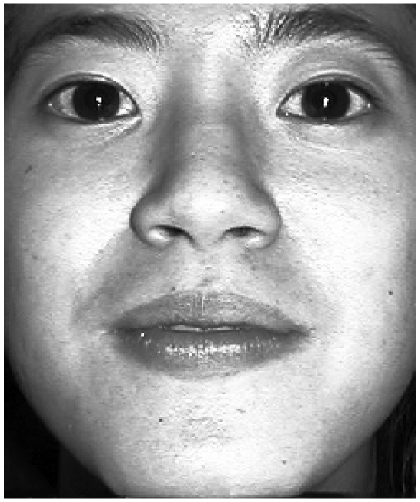
Image ⇔ Vector in high-dimensional space

⇕ Laplacian Pyramid

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

Remember?

X_1 (M dimensions)

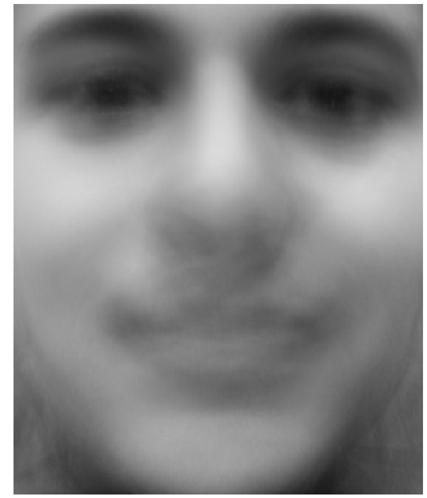


$\simeq$

X_i (d-dimensional approx d=3)

$=$

$\bar{X}$

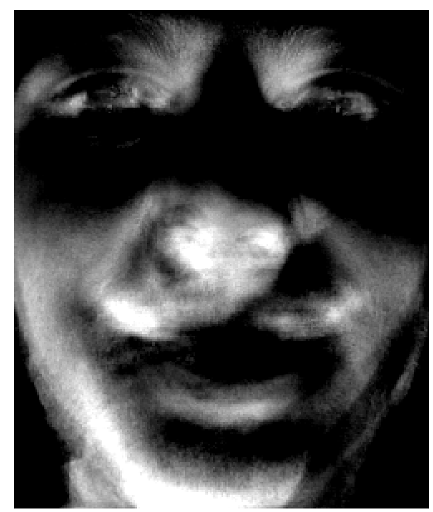


$+$

$y_1^1 *$ B_1

$+ \; y_1^2 *$ B_2

$+ \; y_1^3 *$ B_3



Representing Images by their PCA Basis

Start by stacking all your images as in:

all pixels of one image

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

Representing Images by their PCA Basis

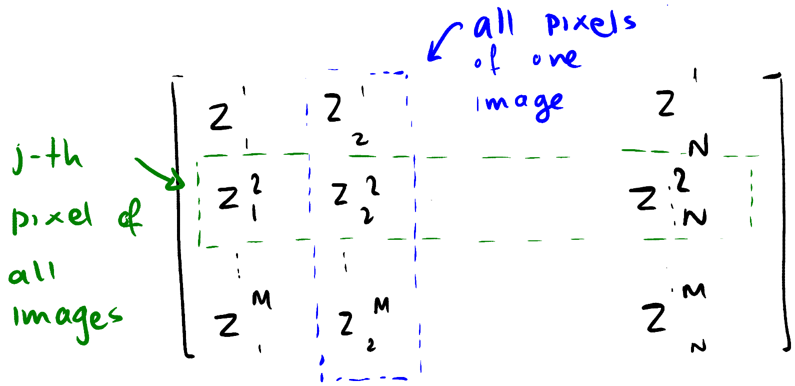
Start by stacking all your images as in:

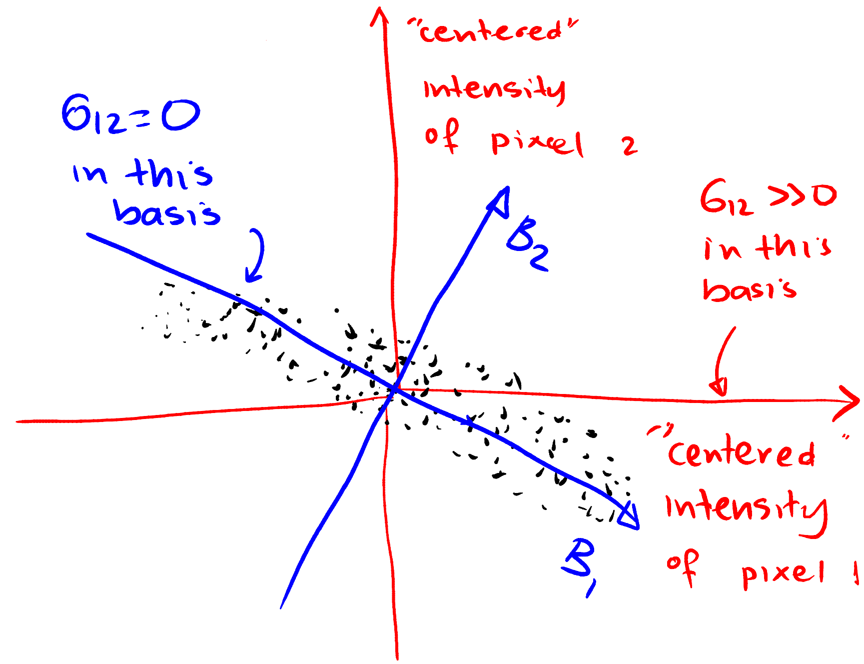
all pixels of one image

j-th pixel of all images

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

Representing Images by their PCA Basis





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

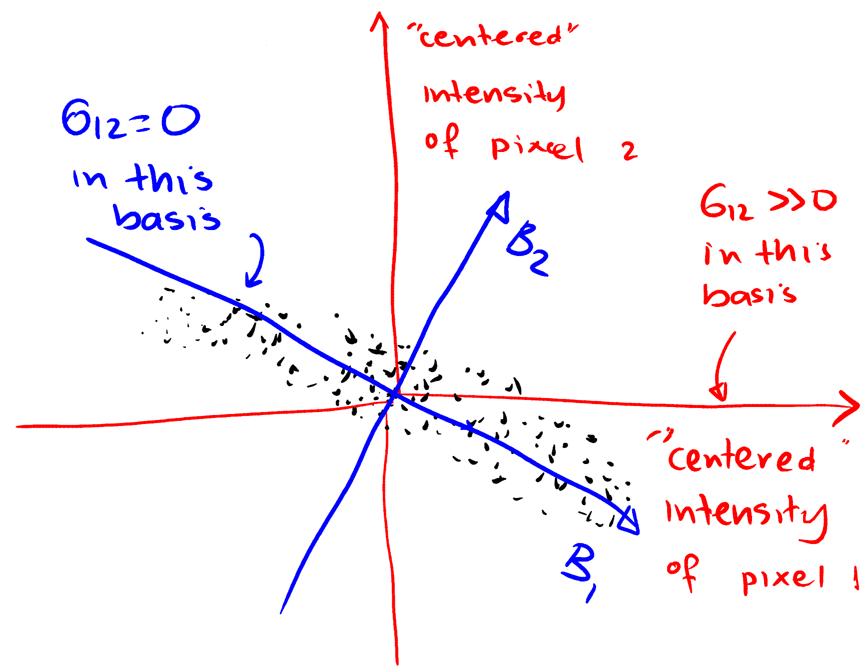
(How do we interpret a point in this new basis?)

Representing Images by their PCA Basis

all pixels of one image

j-th pixel of all images

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



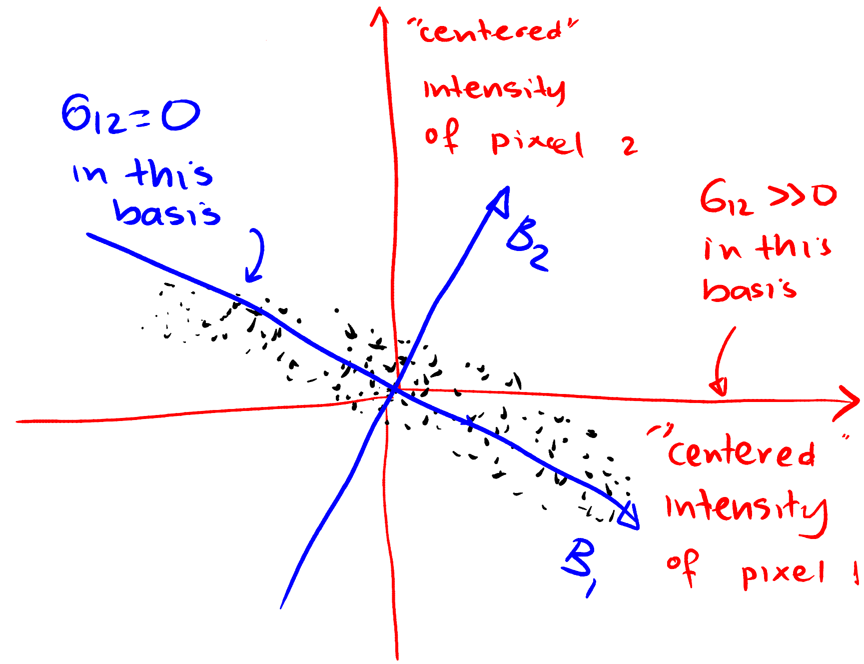
$$\begin{bmatrix} B_1 & B_2 & \cdots & B_M \end{bmatrix} \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_N^1 \\ y_1^d & y_2^d & \cdots & y_N^d \\ y_1^{d+1} & y_2^{d+1} & \cdots & y_N^{d+1} \\ \vdots & \vdots & & \vdots \\ y_1^M & y_2^M & \cdots & y_N^M \end{bmatrix} \begin{array}{l} \text{large} \\ \\ \\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

Representing Images by their PCA Basis

all pixels of one image

j-th pixel of all images

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



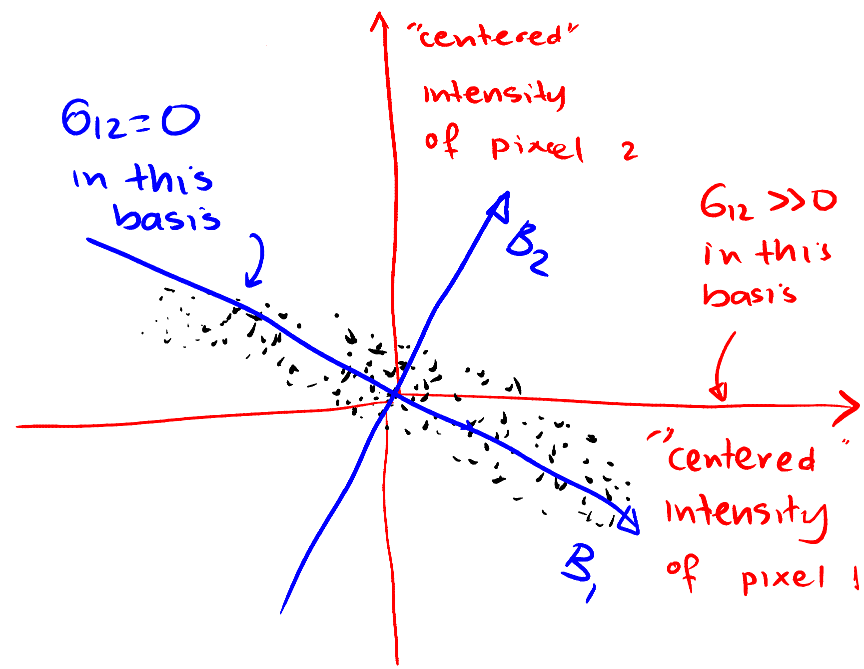
$$Z = B \cdot Y$$

Representing Images by their PCA Basis

all pixels of one image

j-th pixel of all images

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



eigenfaces

$$Z = B \cdot Y$$

mean-subtracted images

coordinates of each image in PCA/eigenface basis

Representing Images by their PCA Basis

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

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

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

$$\begin{bmatrix} y_i \end{bmatrix} = B^T \begin{bmatrix} x_i - \overline{x} \end{bmatrix}$$

eigenfaces

$$Z = B \cdot Y$$

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:

all pixels of one image

j-th pixel of all images

$$\begin{bmatrix} Z_1^1 & Z_2^1 & \cdots & Z_N^1 \\ Z_1^2 & Z_2^2 & \cdots & Z_N^2 \\ \vdots & \vdots & & \vdots \\ Z_1^M & Z_2^M & \cdots & Z_N^M \end{bmatrix}$$

Representing Images by their PCA Basis

With the actual images:

all pixels of one image

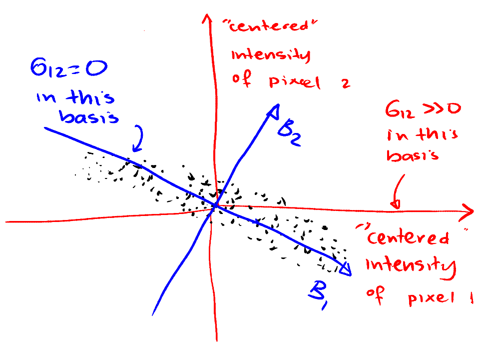
j-th pixel of all images

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

Representing Images by their PCA Basis

And apply the same base-representation machinery:

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



And you'll go from PCA

Representing Images by their PCA Basis

all pixels of one image

$$\begin{array}{l} \text{j-th pixel of all images} \\ \\ \end{array} \begin{bmatrix} z_1^1 & z_2^1 & & z_N^1 \\ z_1^2 & z_2^2 & & z_N^2 \\ \vdots & \vdots & & \\ z_1^M & z_2^M & & z_N^M \end{bmatrix} =$$

$$\begin{bmatrix} B_1 & B_2 & \cdots & B_M \end{bmatrix} \cdot$$

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

$\Longleftrightarrow$

eigenfaces

$$Z = B \cdot Y$$

mean-subtracted images

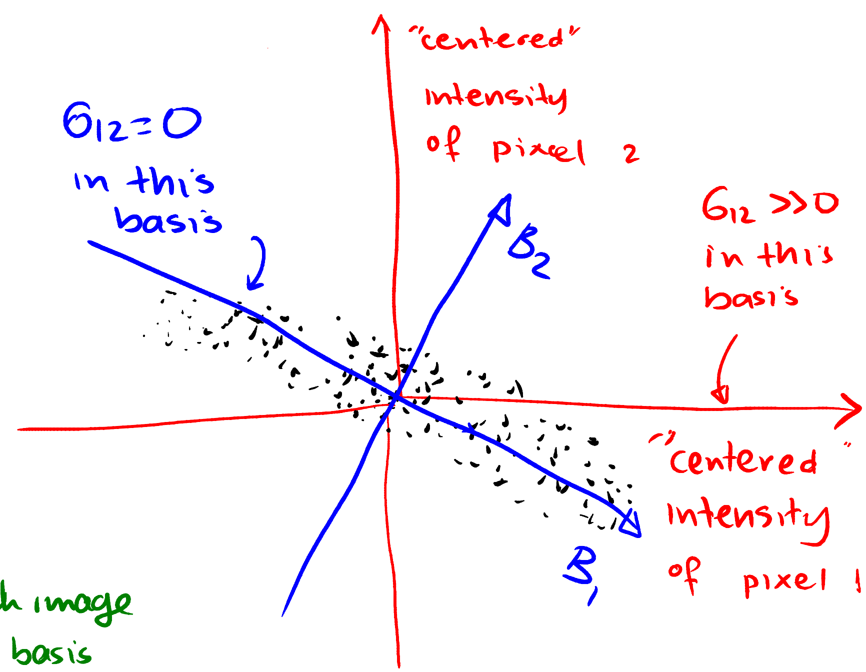
coordinates of each image in PCA/eigenface basis

- 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

$$\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & & \vdots \\ x_1^M & x_2^M & \cdots & x_N^M \end{bmatrix} = \begin{bmatrix} B_1 & B_2 & \cdots & B_M \end{bmatrix} \cdot \begin{array}{l} \\ \text{large}\left\{ \vphantom{\begin{matrix} y \\ \vdots \\ y \end{matrix}} \right. \\ \text{near zero}\left\{ \vphantom{\begin{matrix} y \\ \vdots \\ y \end{matrix}} \right. \end{array} \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_N^1 \\ \vdots & \vdots & & \vdots \\ y_1^d & y_2^d & \cdots & y_N^d \\ y_1^{d+1} & y_2^{d+1} & \cdots & y_N^{d+1} \\ \vdots & \vdots & & \vdots \\ y_1^M & y_2^M & \cdots & y_N^M \end{bmatrix}$$

$\Longleftrightarrow$

wavelet basis

$$X = B \cdot Y$$

wavelet coefficients

Image reconstruction:

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

Image transform

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

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 **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 coefs
- 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 ($2^N x 2^N$)

wavelet transform

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

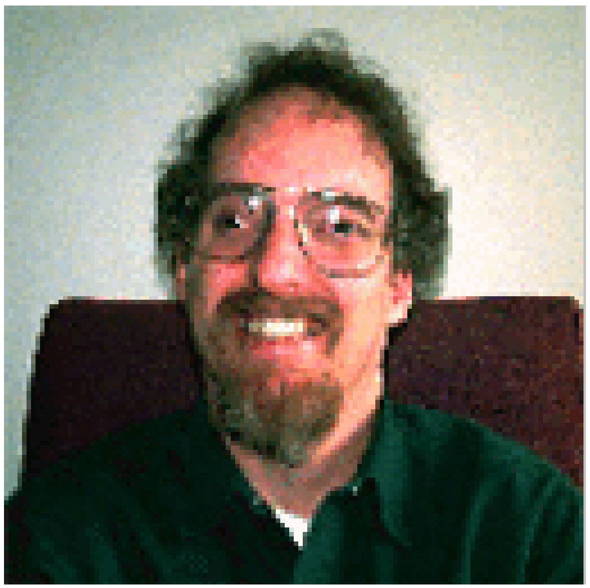
A Simple, Minimal 2-D Image Transform

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

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

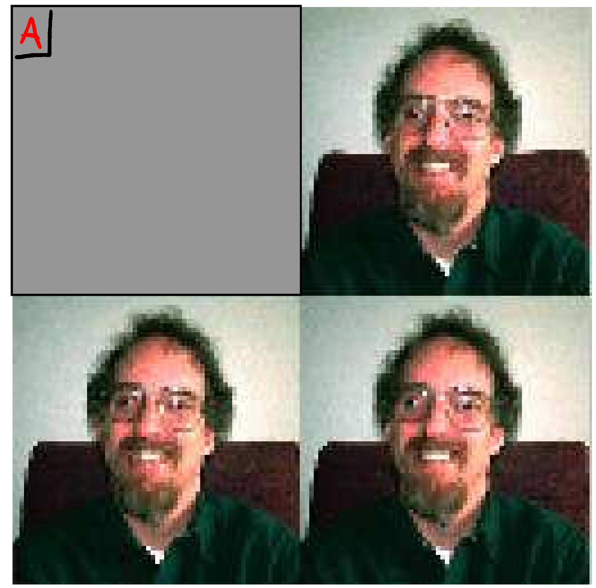
Transformed image ($2^N x 2^N$)

W_0:



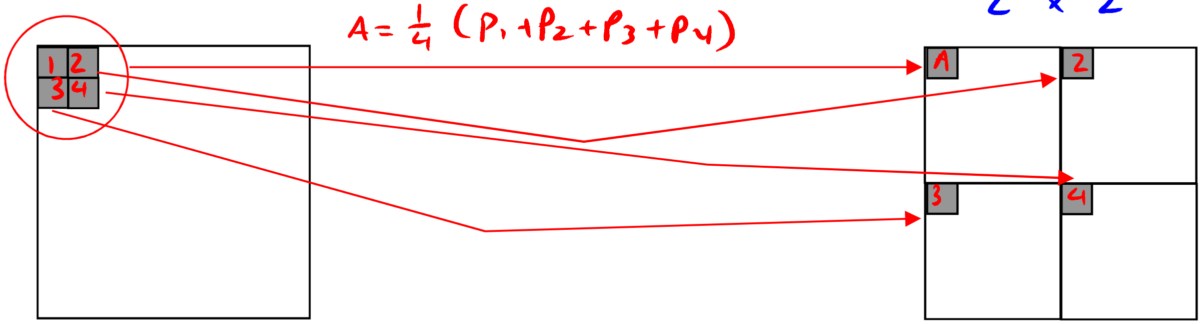
$2^N x \ 2^N$

Step 1



$2^N \times \ 2^N$

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



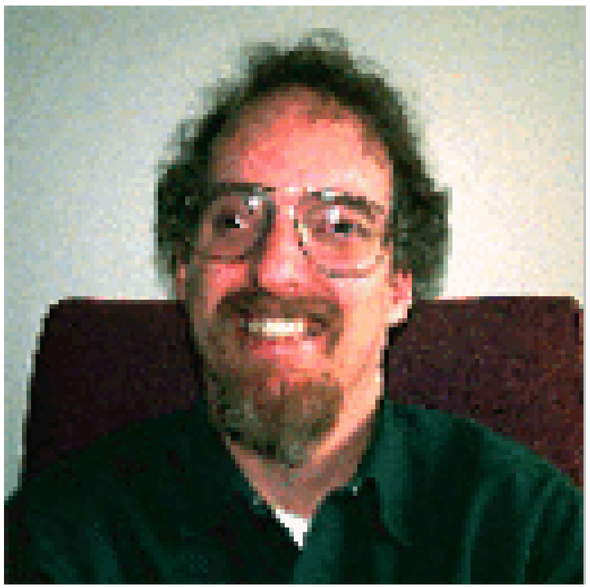
A Simple, Minimal 2-D Image Transform

and repeat for the rest of the image!

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

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

W_0:

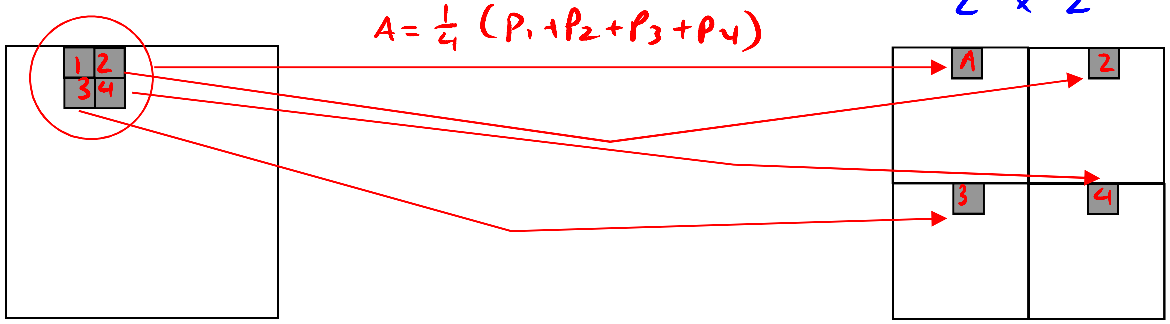


Step 1 →

$2^N \times 2^N$

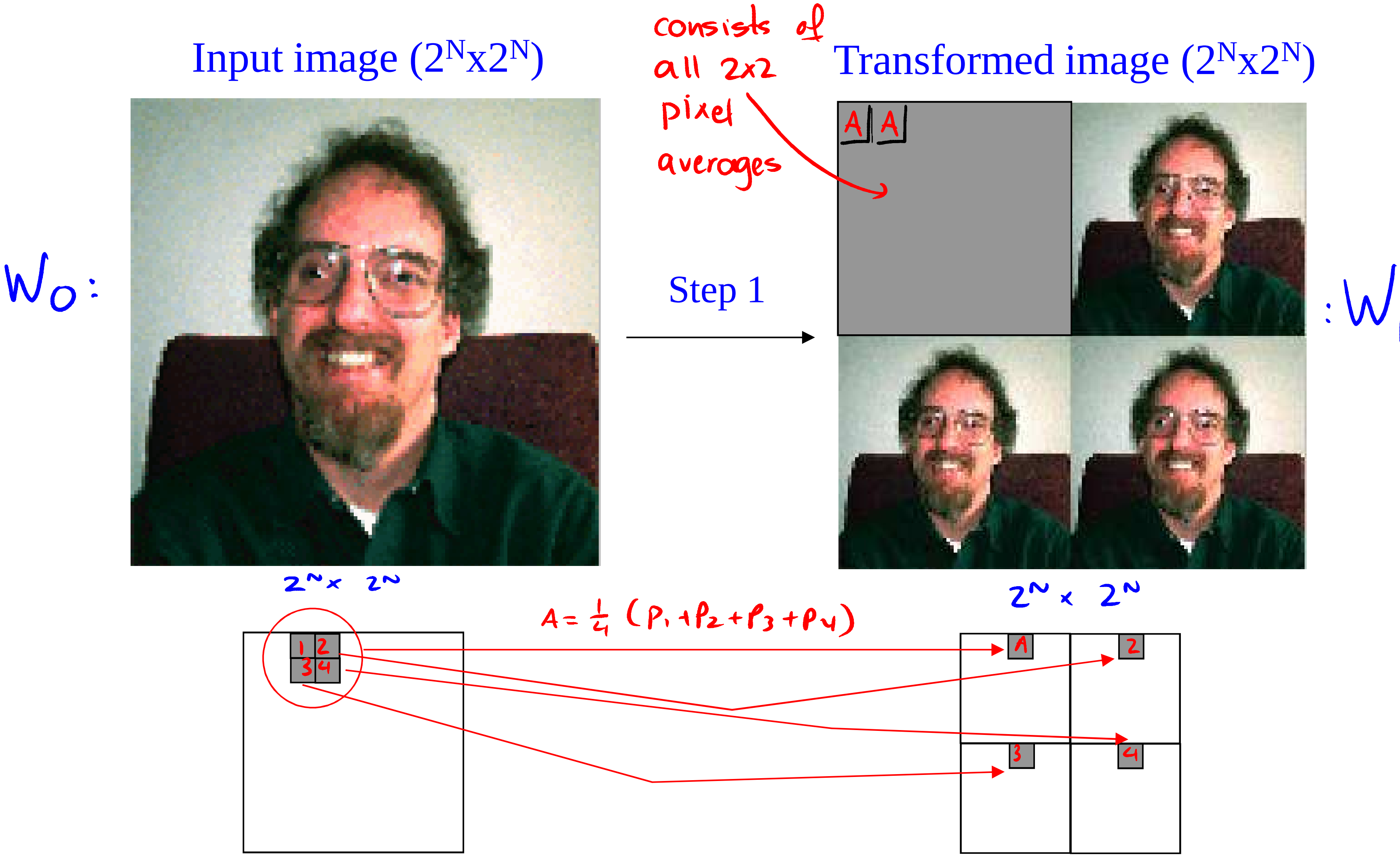
$2^N \times 2^N$

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



A Simple, Minimal 2-D Image Transform

You end up with a half-size-per-side image of 2x2 pixel averages.



A Simple, Minimal 2-D Image Transform

Guesses for step 2?

A Simple, Minimal 2-D Image Transform

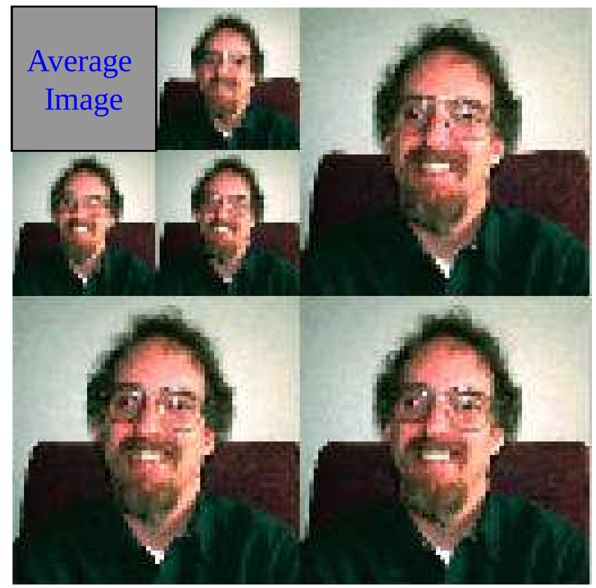
Step 2: Recursively perform Step 1 for top-left quadrant of result

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

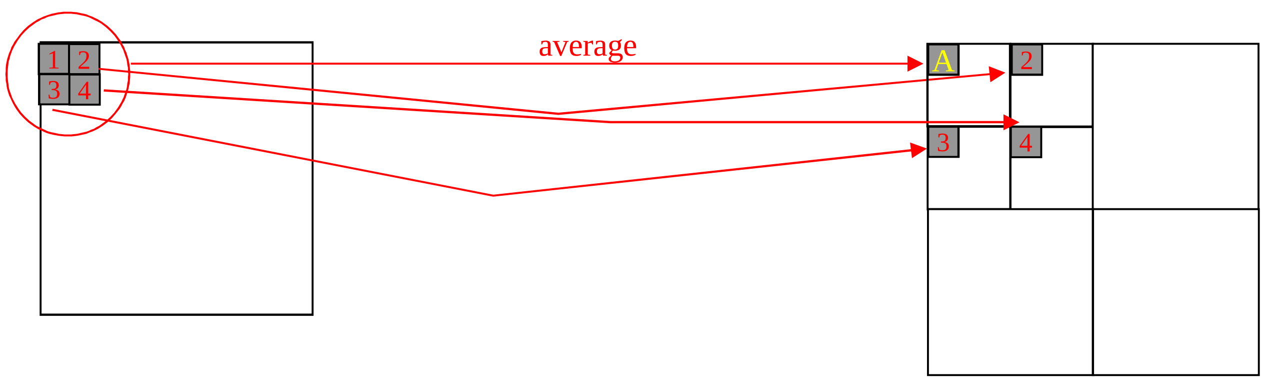
$2^{N-1} \times 2^{N-1}$

Step 2

Transformed image (2^Nx2^N)



: W_2



A Simple, Minimal 2-D Image Transform

Step 3: Recursion stops when average image is 1 pixel

Transformed image (2^Nx2^N)

$: W_N$

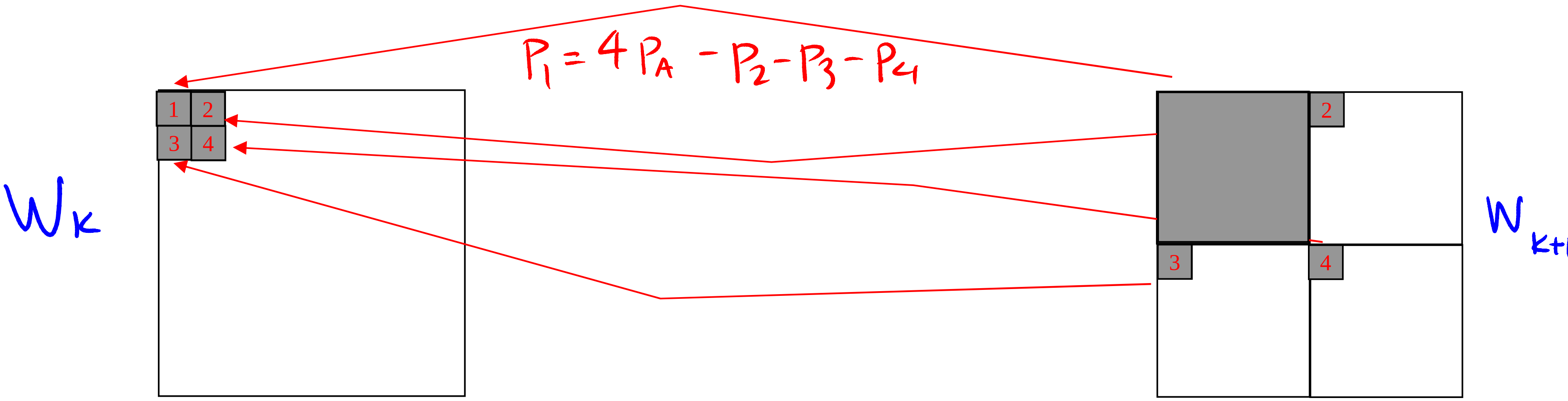
A Simple, Minimal 2-D Image Transform

Is this invertible?

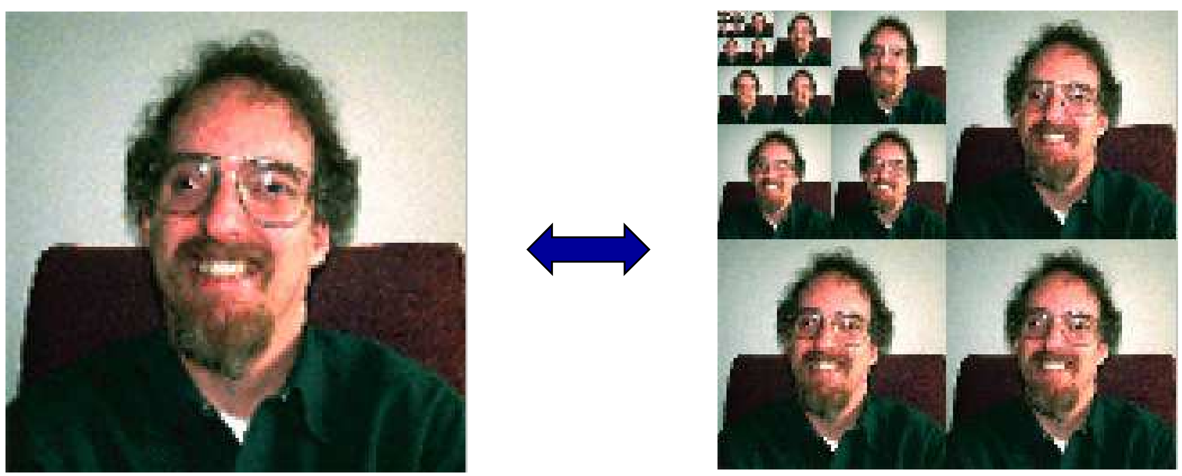
(i.e. Can we go from the wavelet transform W_0, to the original image?)

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 coefs
- 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	...				2^3-1
Input image I	9	7	3	5	6	10	2	6

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

8	4	8	4

What is necessary to complete the representation?

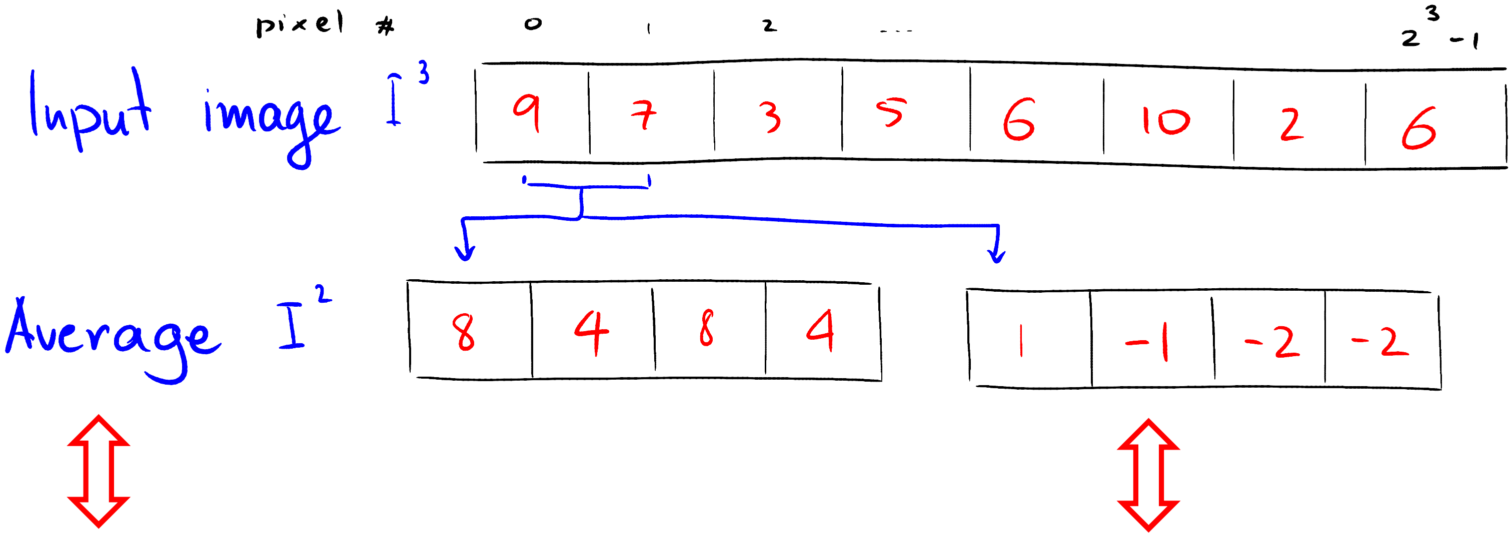
1D Haar Wavelet Transform: Recursive Definition

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

1	-1	-2	-2

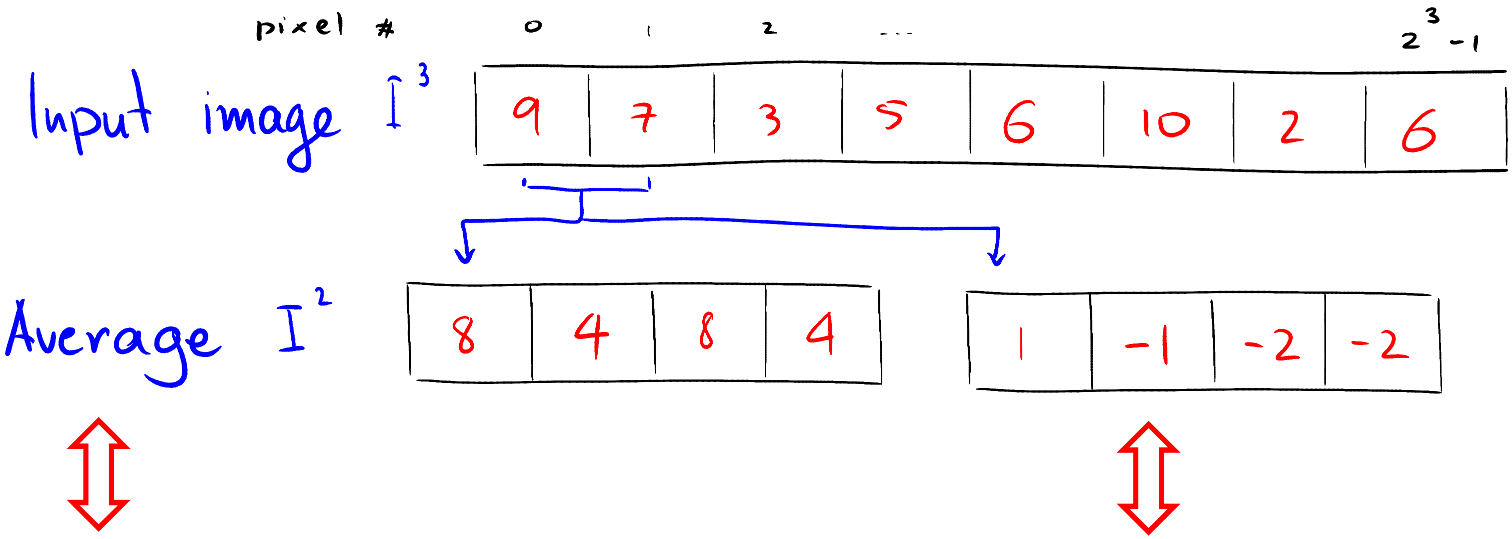
1D Haar Wavelet Transform: Recursive Definition



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

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 coef s D^2

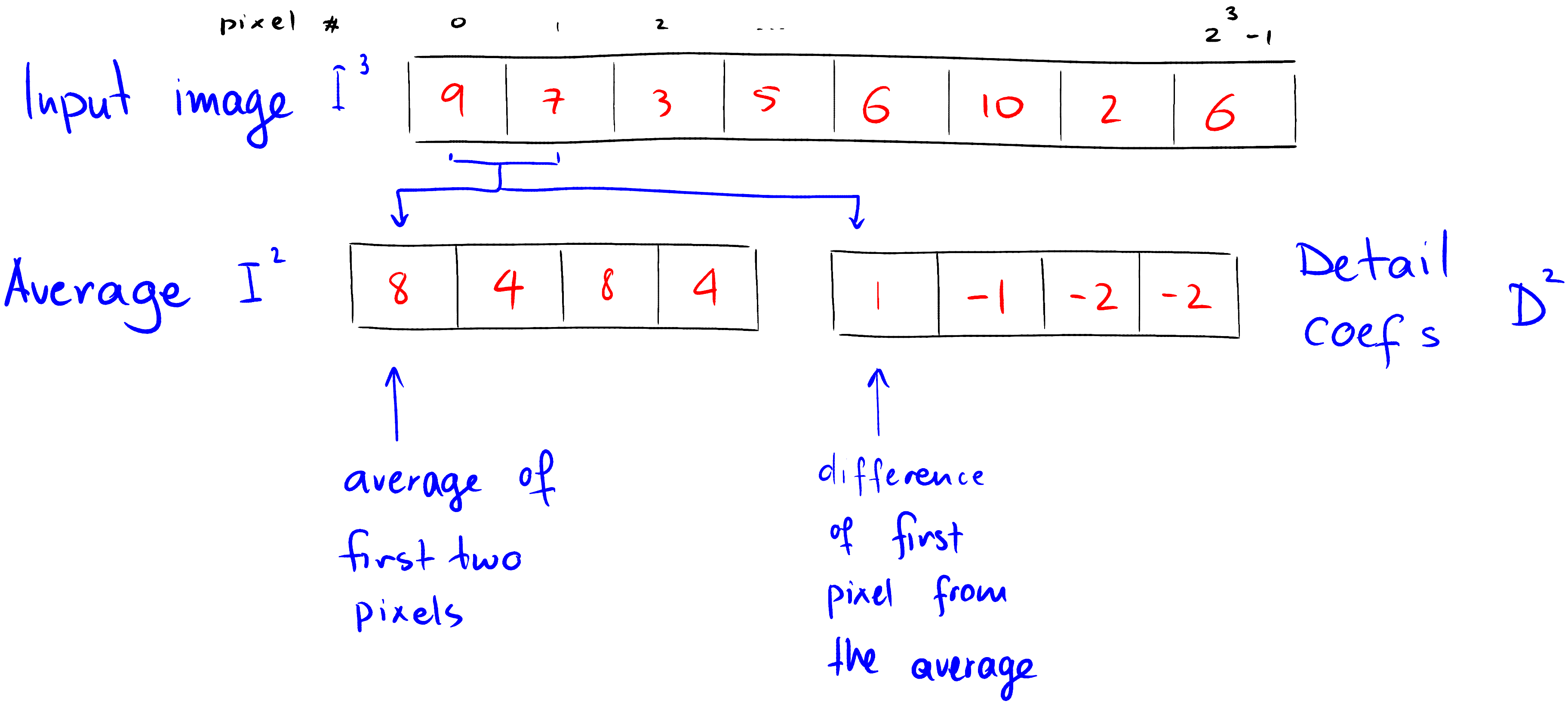
1	-1	-2	-2

average of first two pixels

difference of first pixel from the average

Do we need to store the difference of the 2nd pixel from the average?

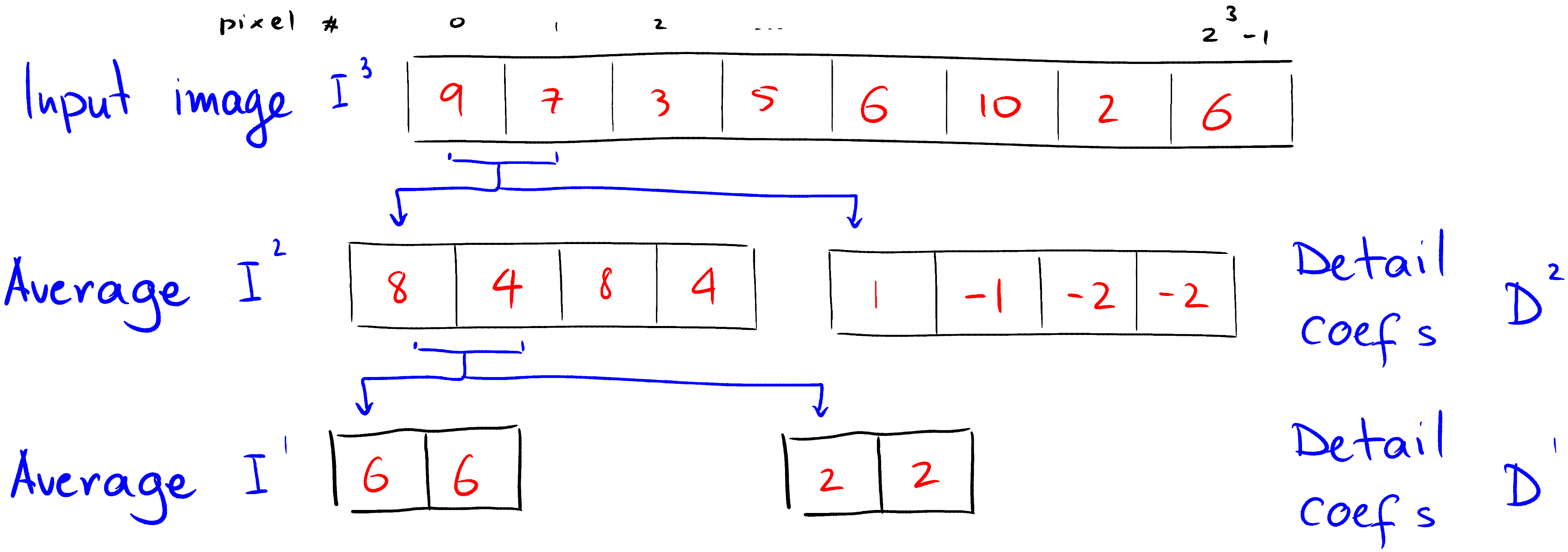
1D Haar Wavelet Transform: Recursive Definition



No need to store the difference of the 2nd pixel from the average!

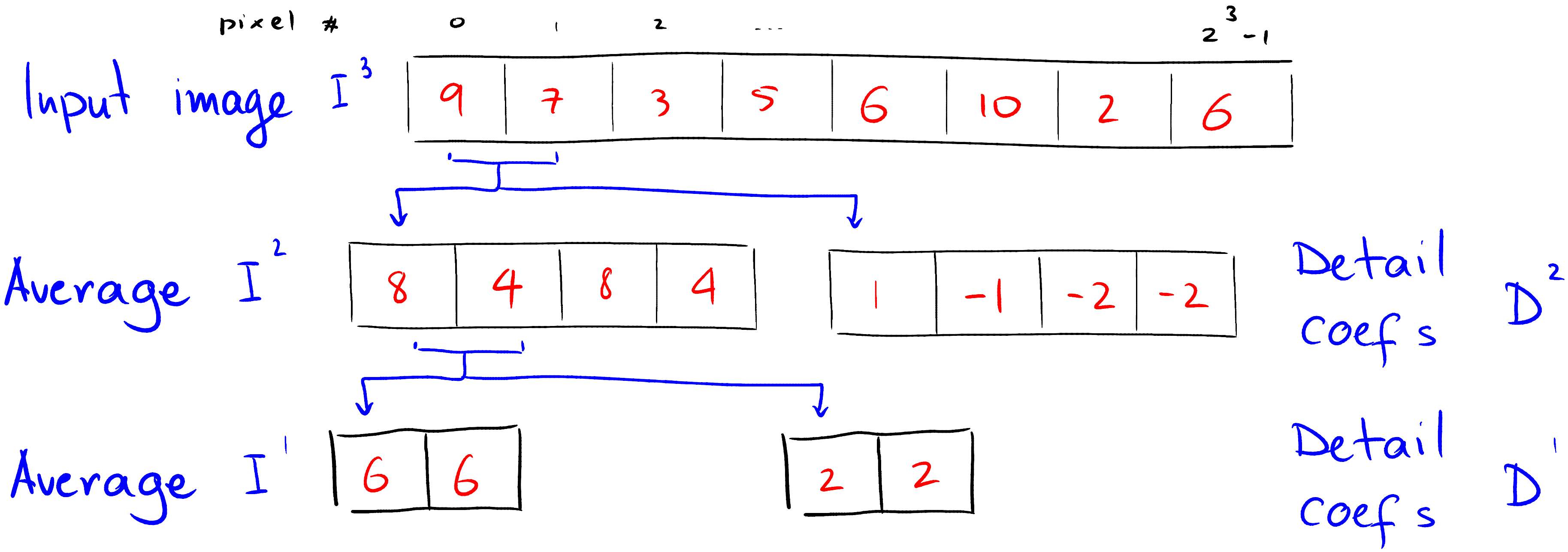
D^0 has ½ the size of the corresponding Laplacian L_0

1D Haar Wavelet Transform: Recursive Definition



Repeat recursively.

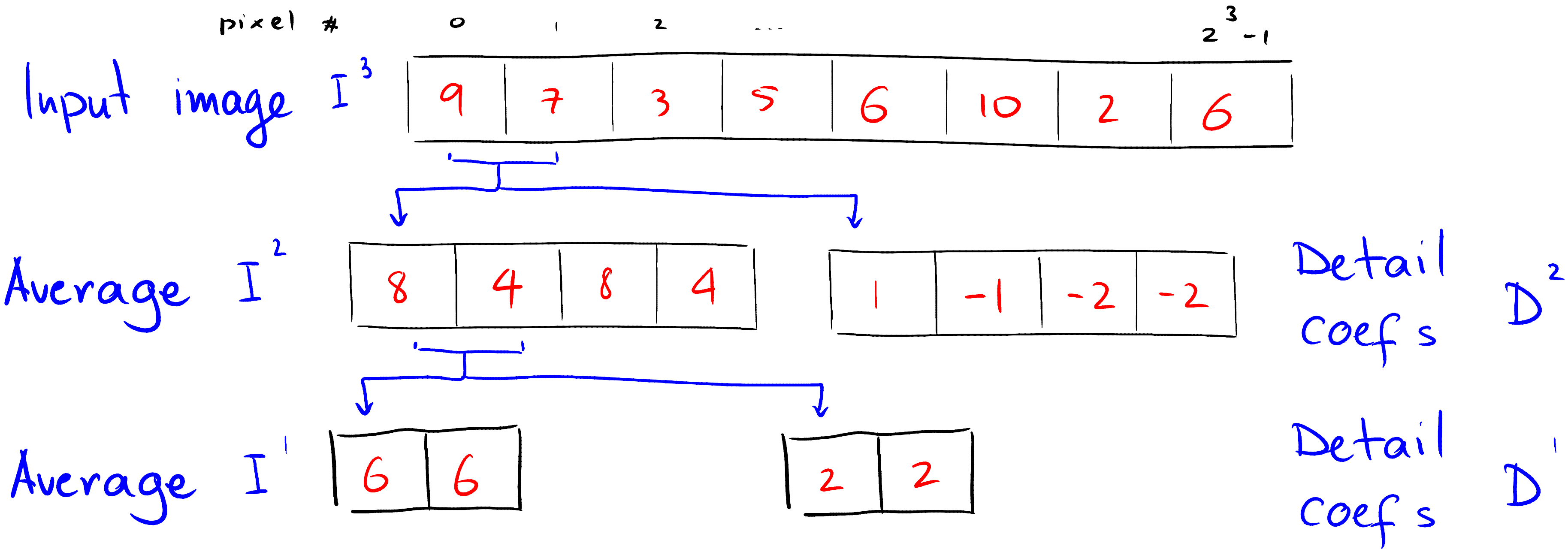
1D Haar Wavelet Transform: Recursive Definition



In general:

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

1D Haar Wavelet Transform: Recursive Definition

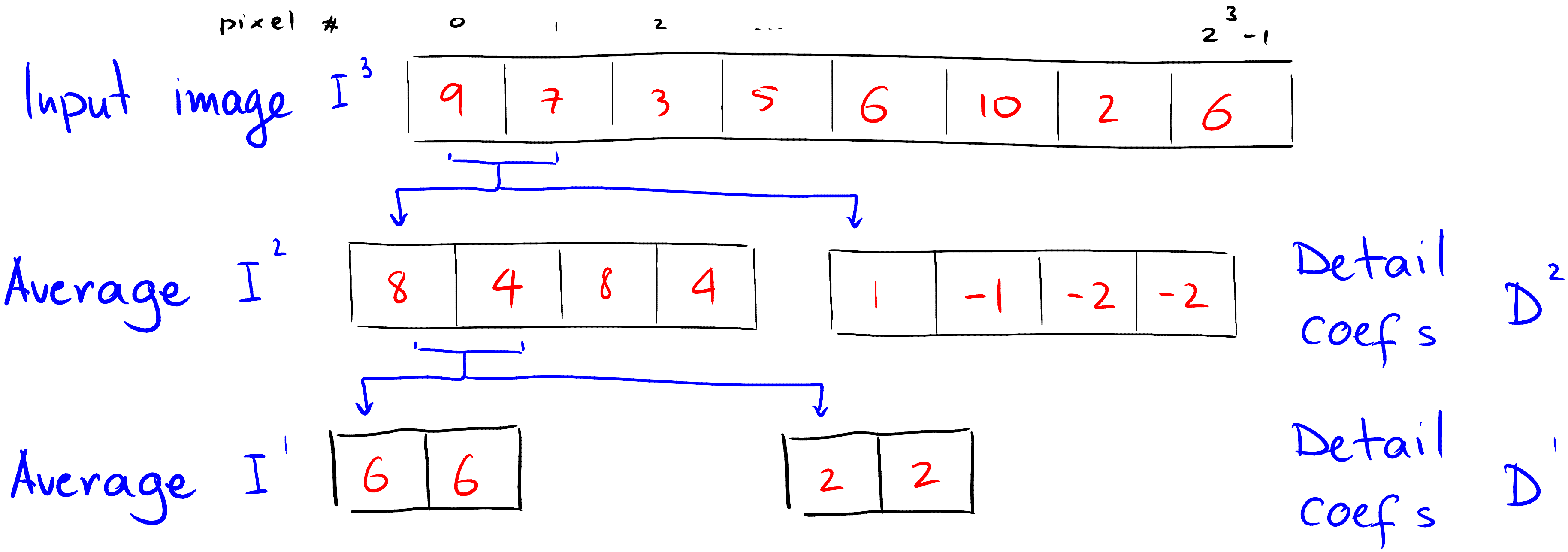


In general:

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

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

1D Haar Wavelet Transform: Recursive Definition



In general:

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

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

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

1D Haar Wavelet Transform: Recursive Definition

Can these two operations be written as convolutions?

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

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

1D Haar Wavelet Transform: Recursive Definition

Can these two operations be written as convolutions?

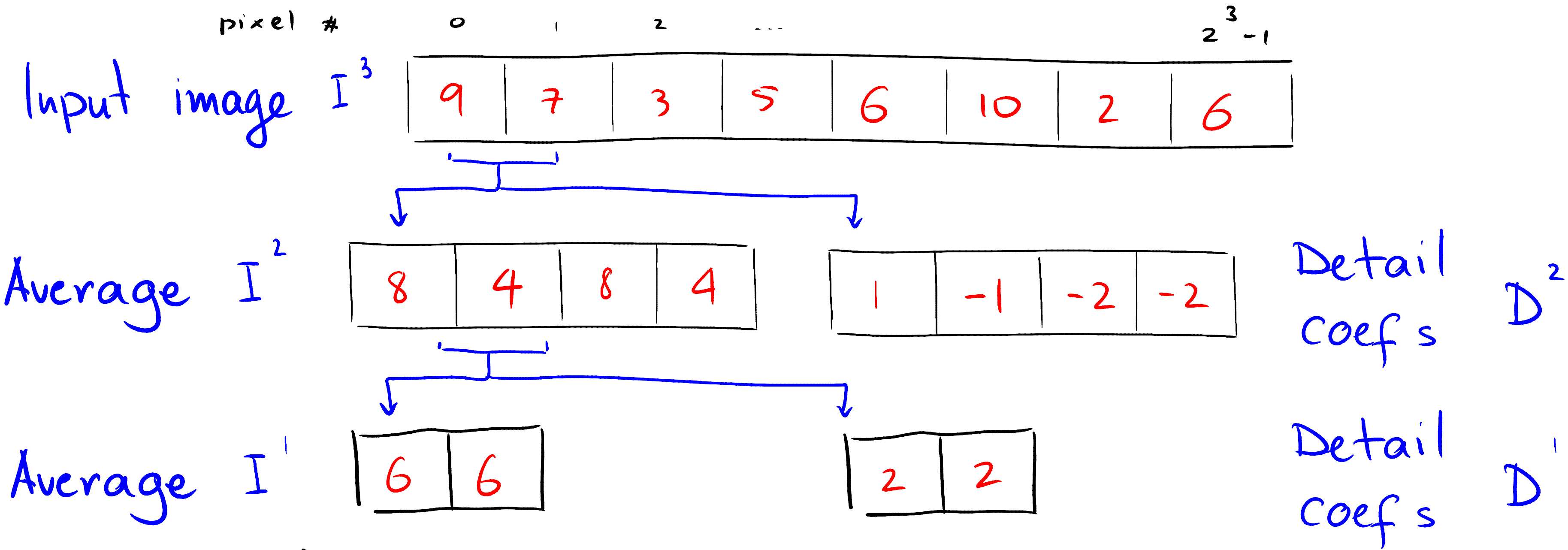
$$I_i^j = \frac{1}{2}\left(I_{2i}^{j+1} + I_{2i+1}^{j+1} \right)$$

and what are the masks?

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

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

1D Haar Wavelet Transform: Recursive Definition



equivalent convolution mask ϕ: | 1/2 | 1/2 |

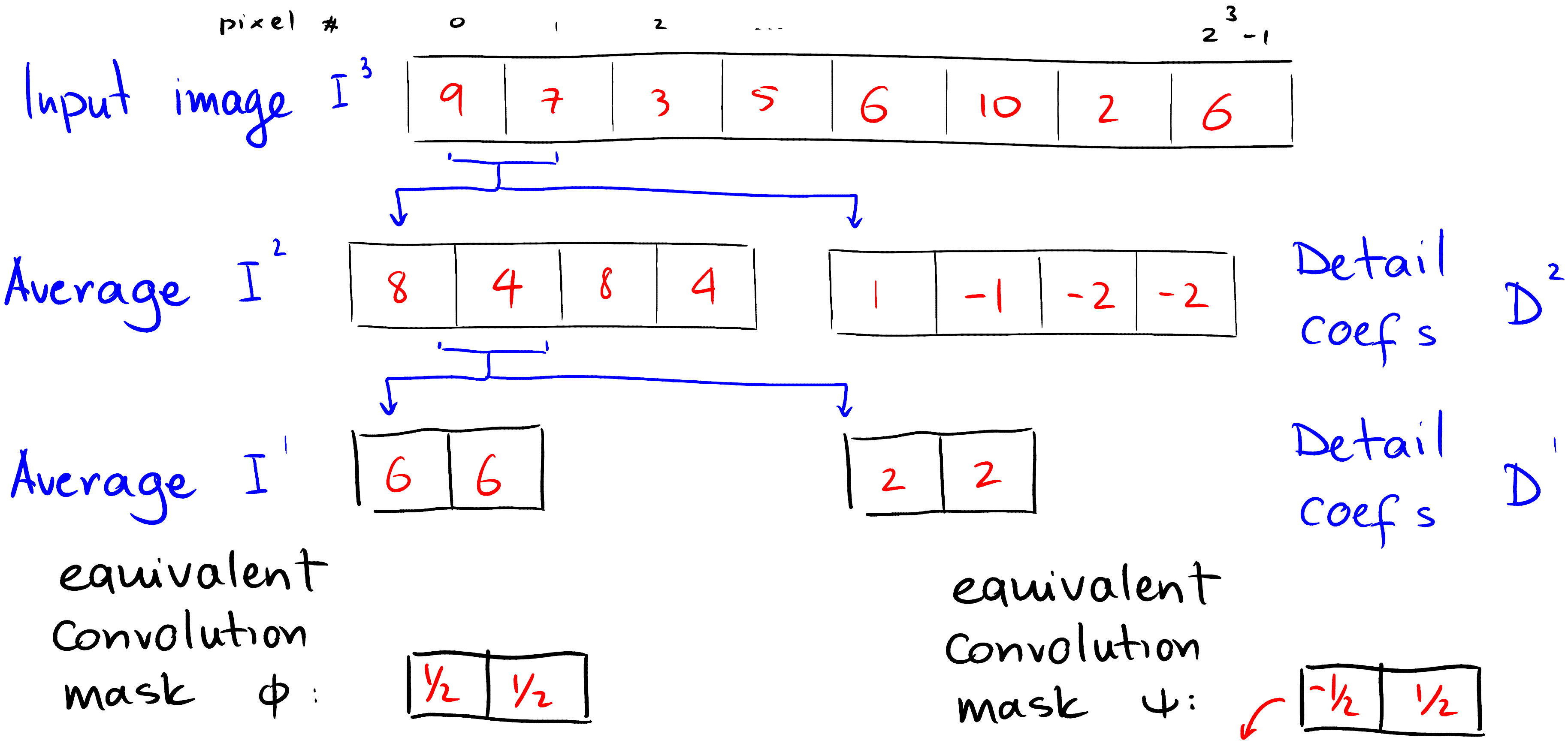
equivalent convolution mask ψ: | -1/2 | 1/2 |

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

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

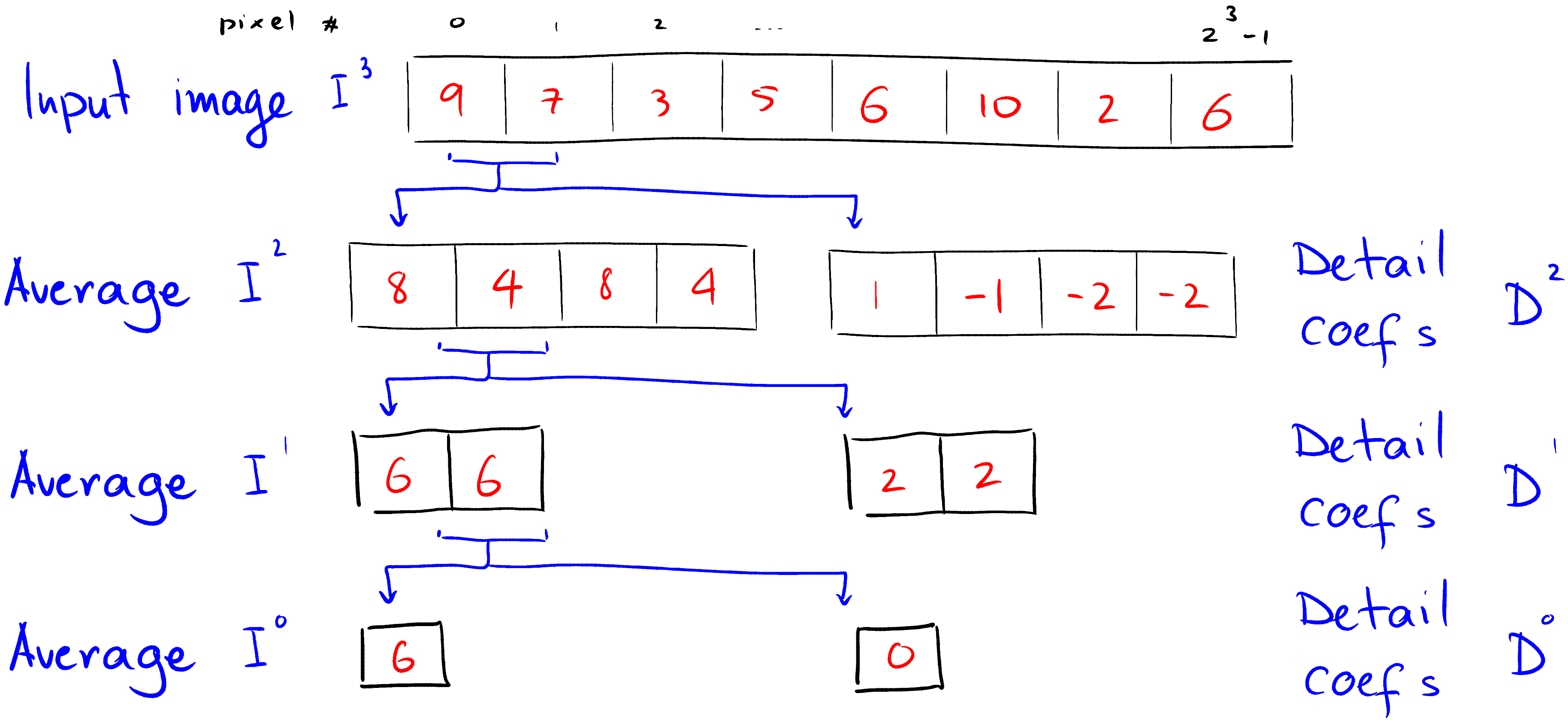
j-th level of "pyramid" contains 2^j pixels

1D Haar Wavelet Transform: Recursive Definition

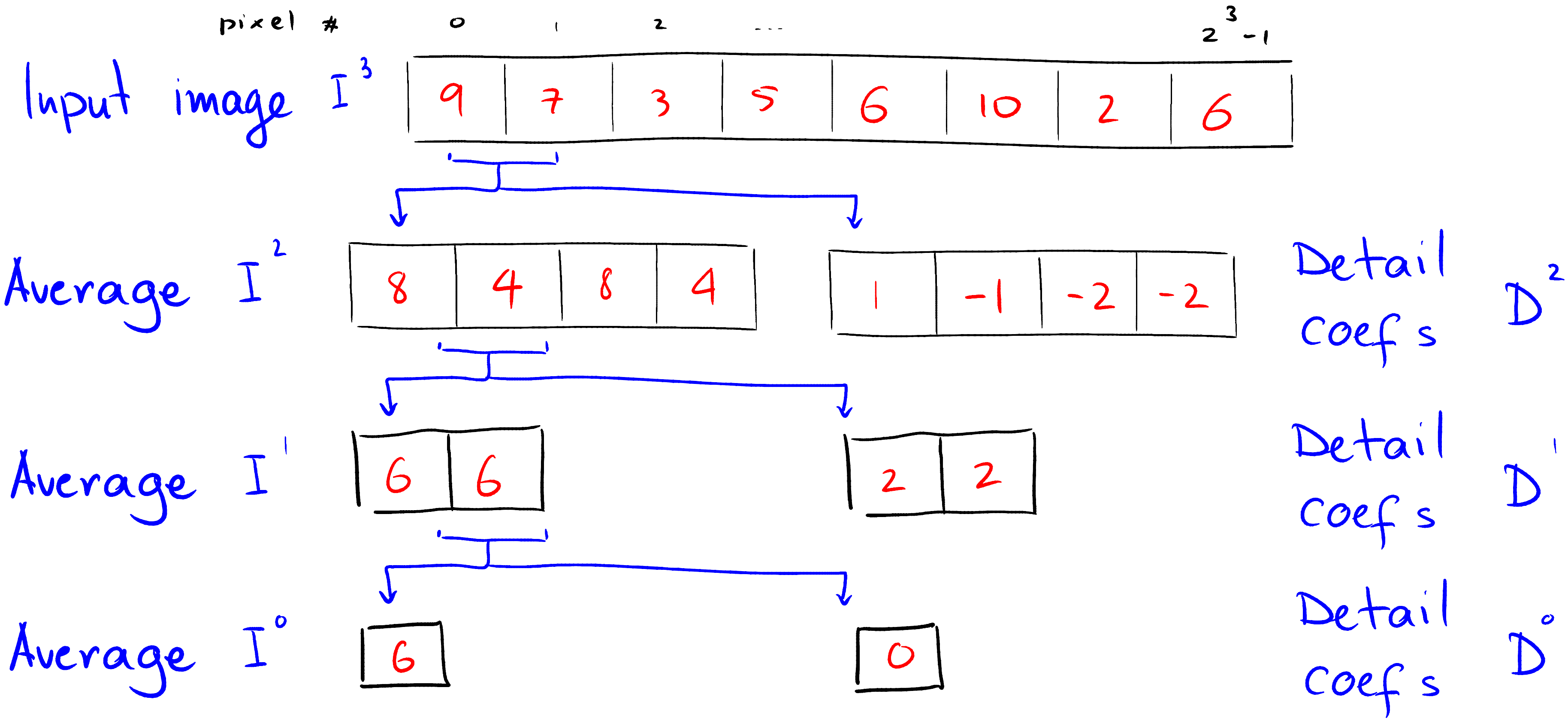


Let's use these masks to estimate the first level

1D Haar Wavelet Transform: Recursive Definition

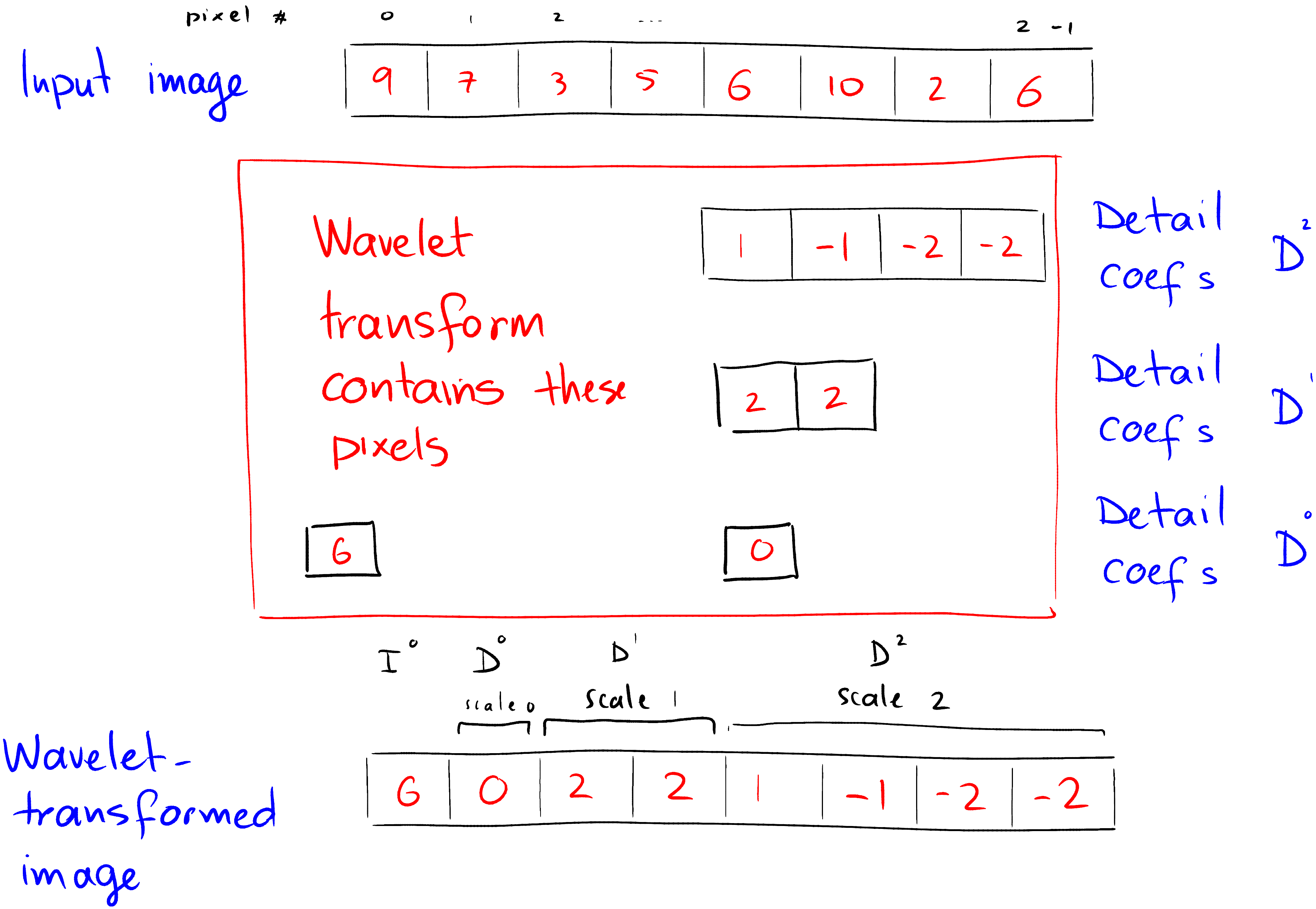


1D Haar Wavelet Transform: Recursive Definition



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

1D Haar Wavelet Transform: Recursive Definition



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 coefs
- 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

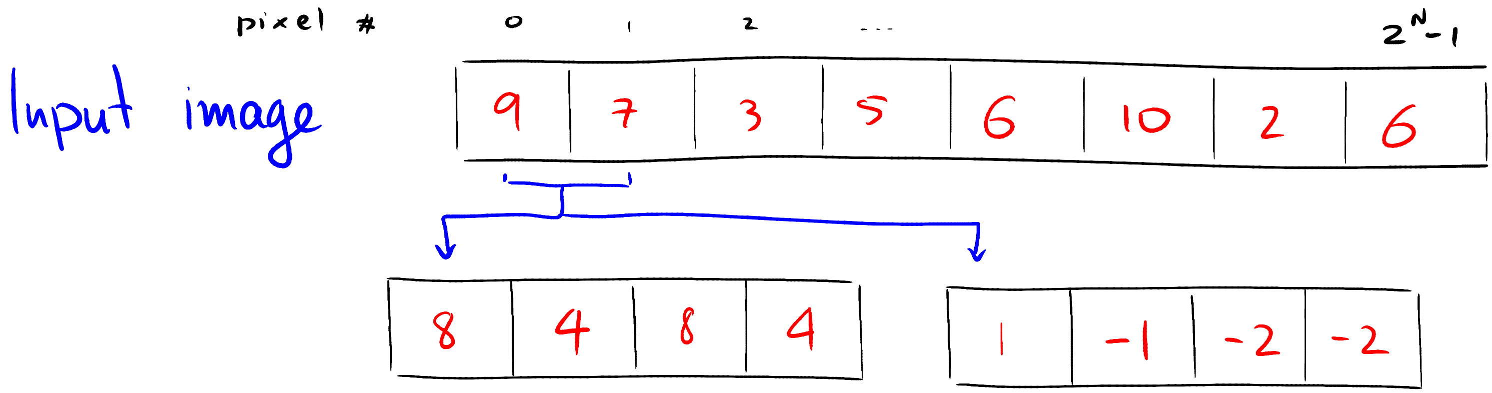
We know the result (because we just computed it), so let's start from the finest detail coefficients D^2. *(Remember that the convolution mask was:* $\boxed{-\frac{1}{2}}\boxed{\frac{1}{2}}$ *)*

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

Wavelet transformed image

Original image

1D Haar Wavelet Transform as a Matrix Product

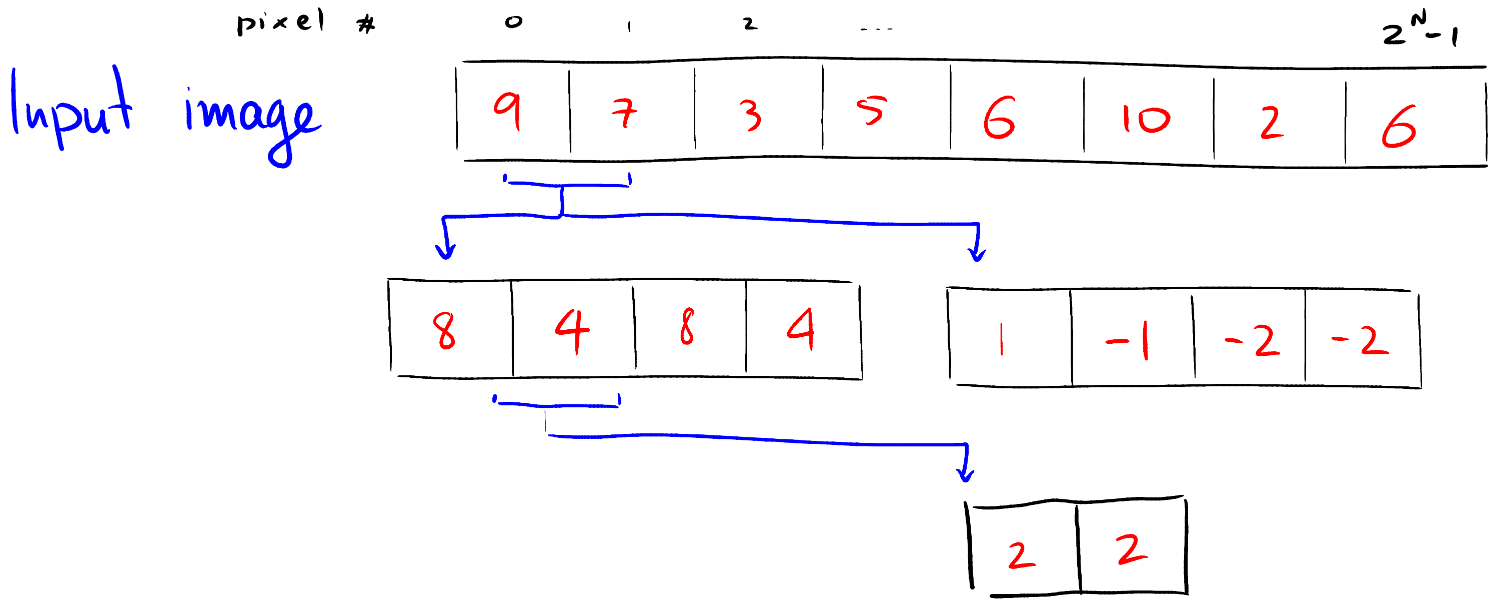


Wavelet transformed image

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

Original image

1D Haar Wavelet Transform as a Matrix Product

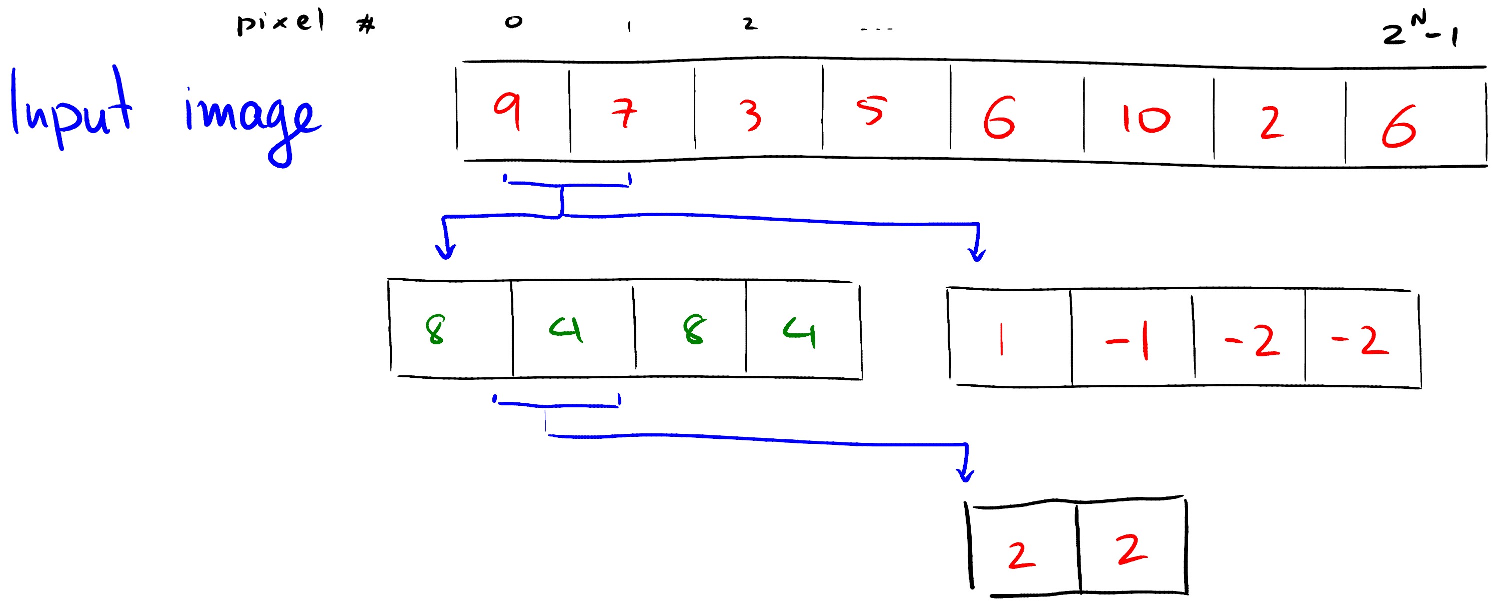


Wavelet transformed image

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

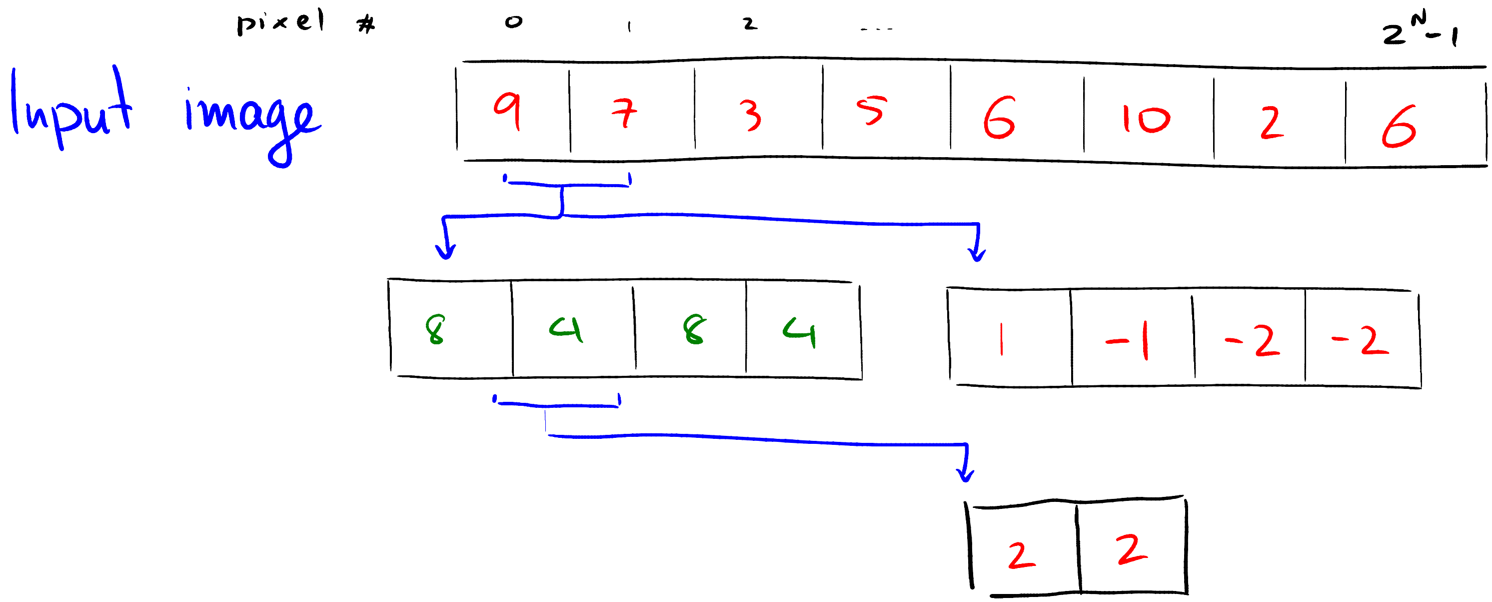
Original image

1D Haar Wavelet Transform as a Matrix Product



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

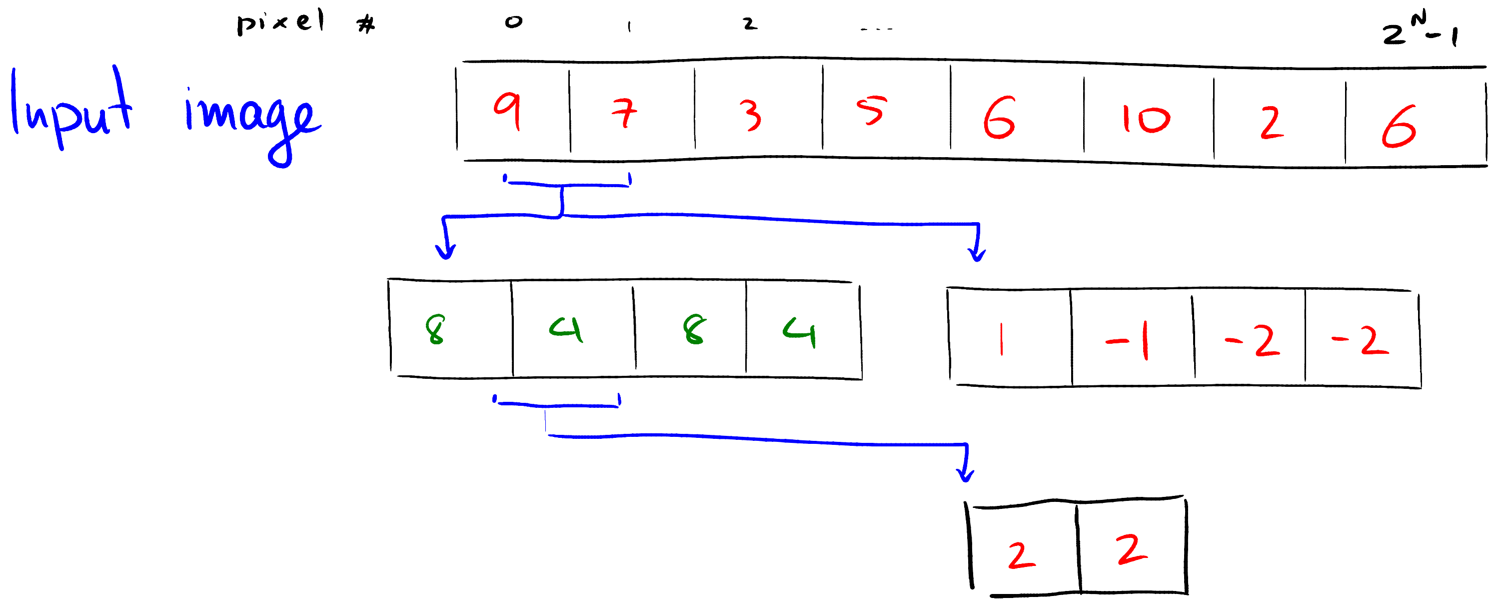
1D Haar Wavelet Transform as a Matrix Product



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

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

1D Haar Wavelet Transform as a Matrix Product



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

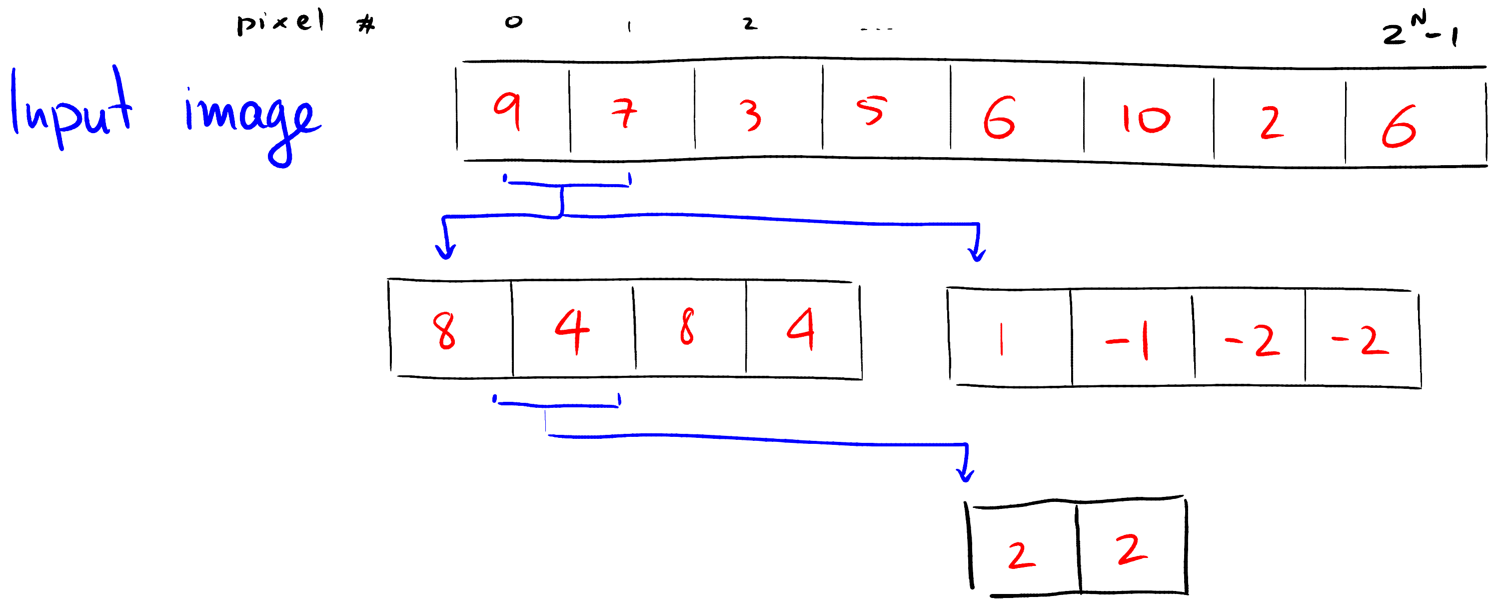
1D Haar Wavelet Transform as a Matrix Product

3rd & 4th rows of product

$$
\begin{bmatrix}
 & & & & & & & \\
 & & & & & & & \\
\tfrac{1}{4} & \tfrac{1}{4} & -\tfrac{1}{4} & -\tfrac{1}{4} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \tfrac{1}{4} & \tfrac{1}{4} & -\tfrac{1}{4} & -\tfrac{1}{4} \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & &
\end{bmatrix}
$$

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

1D Haar Wavelet Transform as a Matrix Product

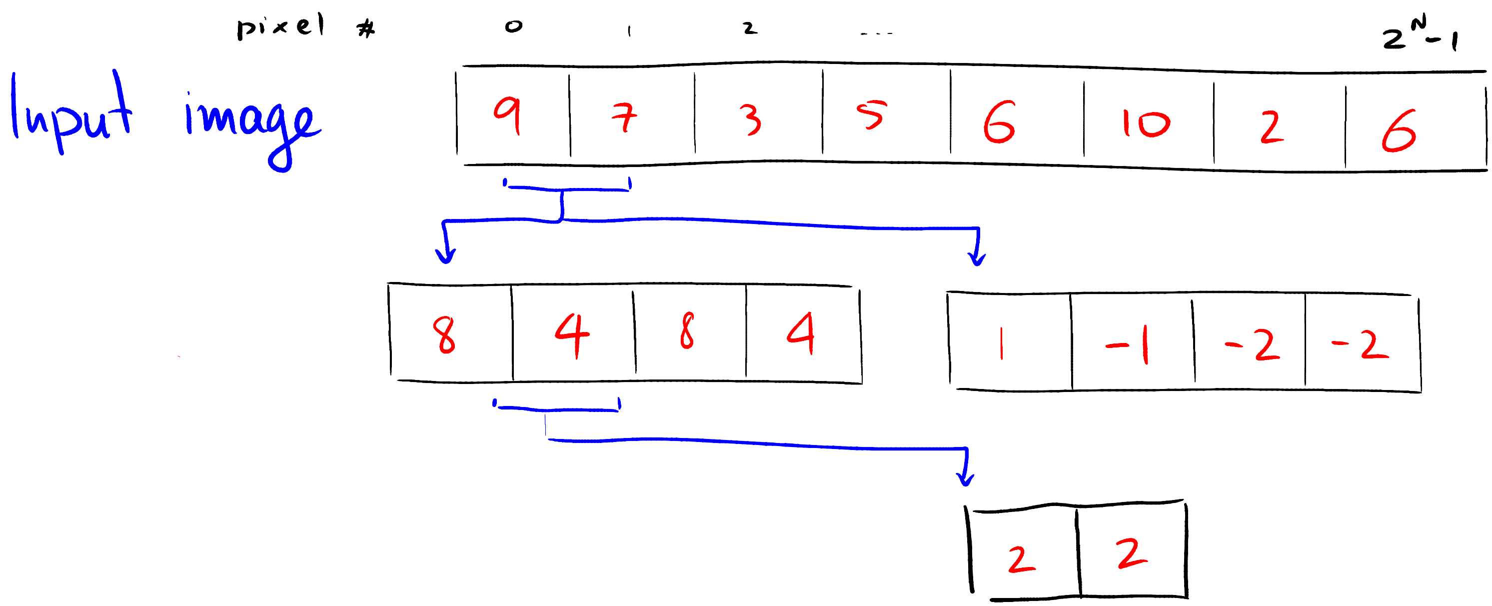


Wavelet transformed image

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

Original image

1D Haar Wavelet Transform as a Matrix Product



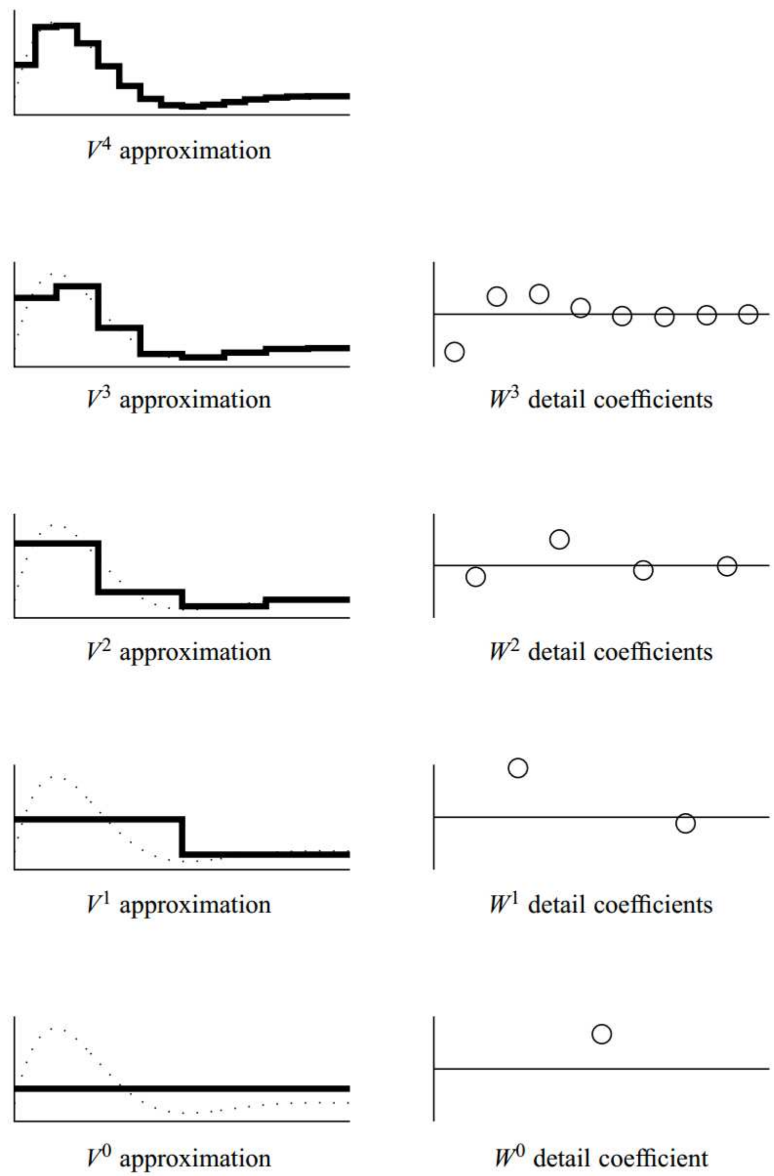
Wavelet transformed image

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

Original image

1D Haar Wavelet Transform as a Matrix Product

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

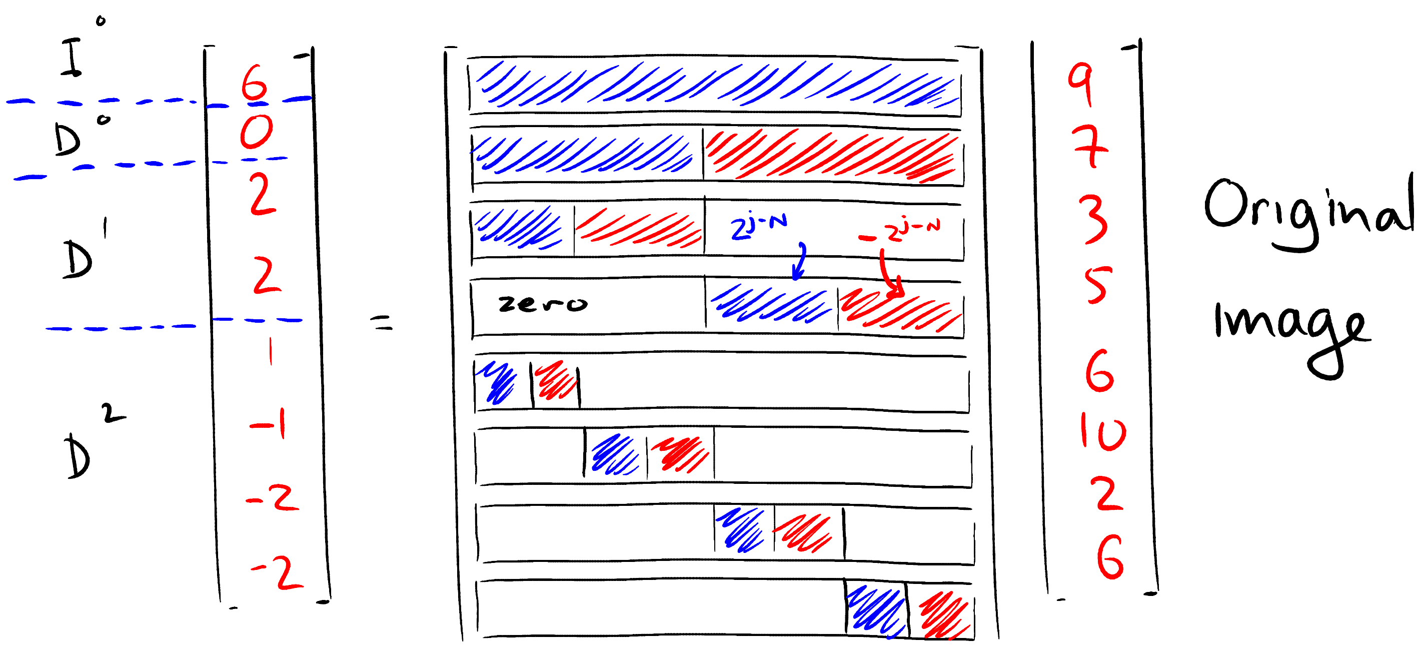


The 1D Haar Wavelet Transform Matrix W

And the matrix W has interesting properties.

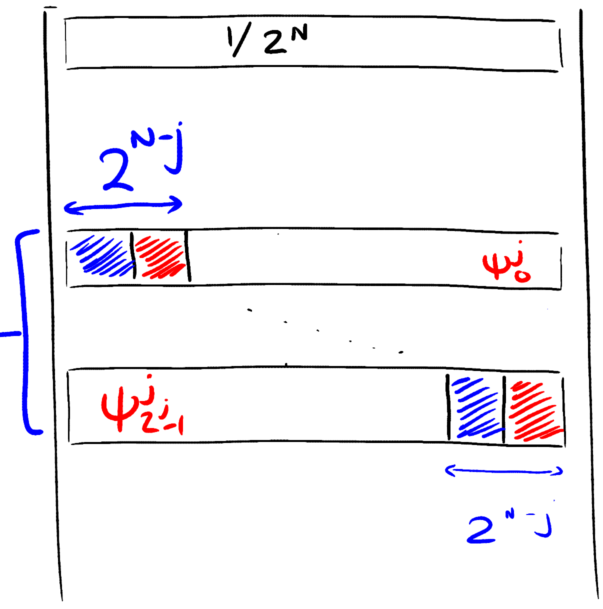
$$
\text{Wavelet transformed image}\quad
\begin{array}{c} I^0 \\ D^0 \\ D^1 \\ \\ D^2 \\ \\ \\ \\ \end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{array}{c} 1/8 \\ 1/8 \\ 1/4 \\ \\ 1/2 \\ \\ \\ \\ \end{array}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1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

The 1D Haar Wavelet Transform Matrix W

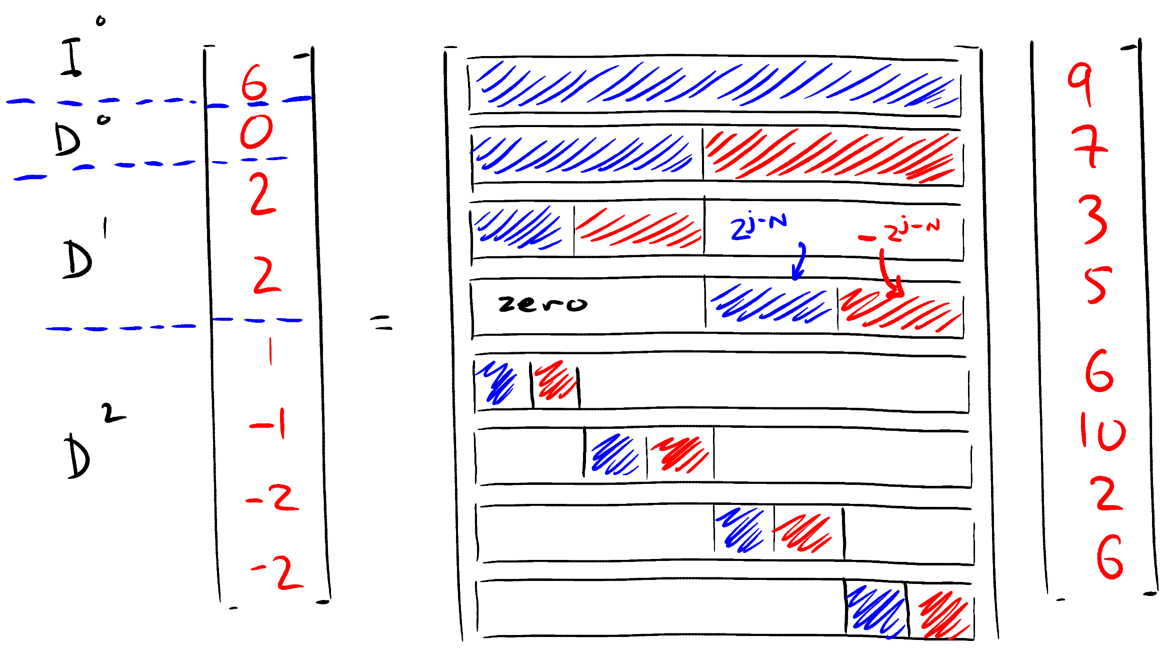


The 1D Haar Wavelet Transform Matrix W

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



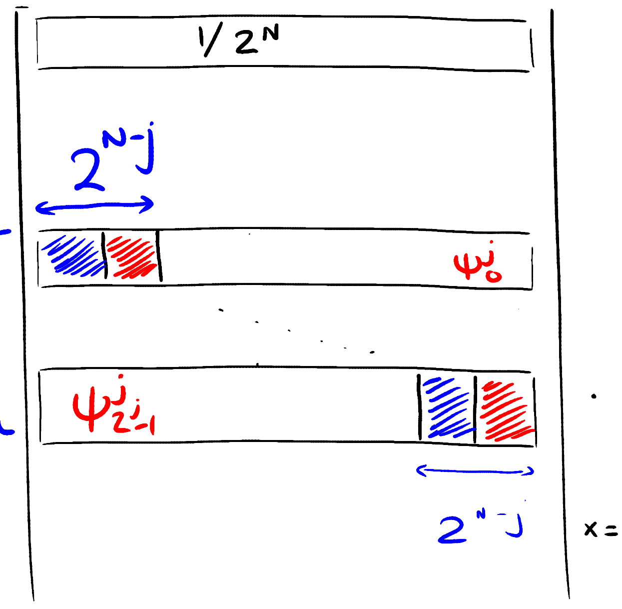
Wavelet transformed image



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 ψ_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 coefs
- Wavelet-based image compression
- The 2D Haar wavelet transform

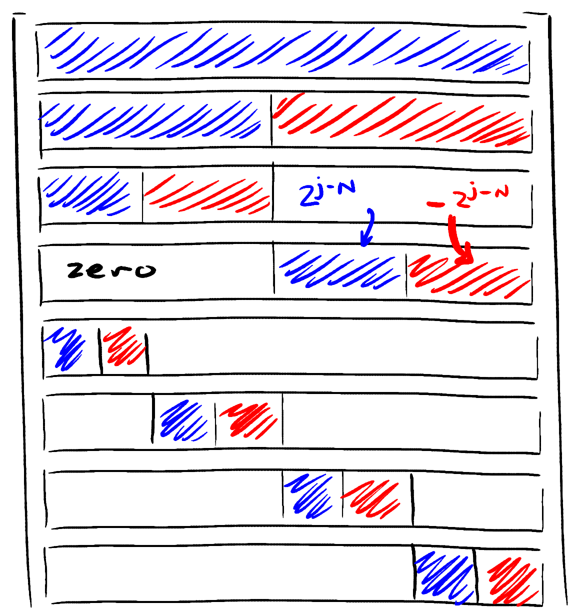
Reconstructing an Image from its Wavelet Coefs

What is the dot product

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

of two distinct rows of W?

$W =$



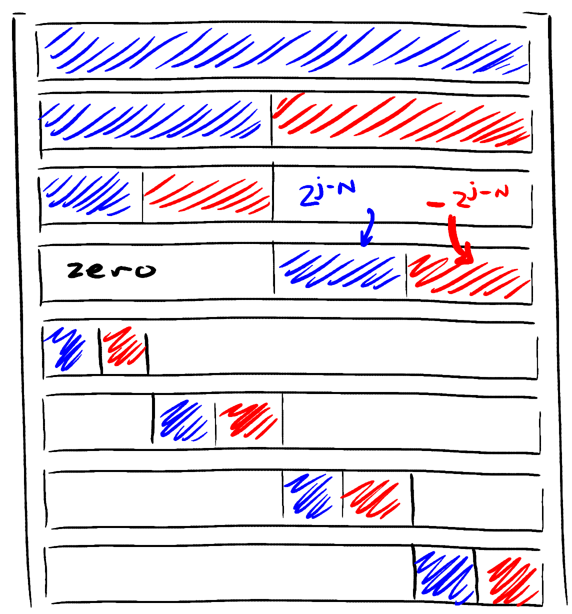
Reconstructing an Image from its Wavelet Coefs

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.

$$WW^T = \text{diagonal}$$

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

Reconstructing an Image from its Wavelet Coefs

Define $\Lambda = WW^T$ with

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_{2^N-1} \end{bmatrix}$$

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

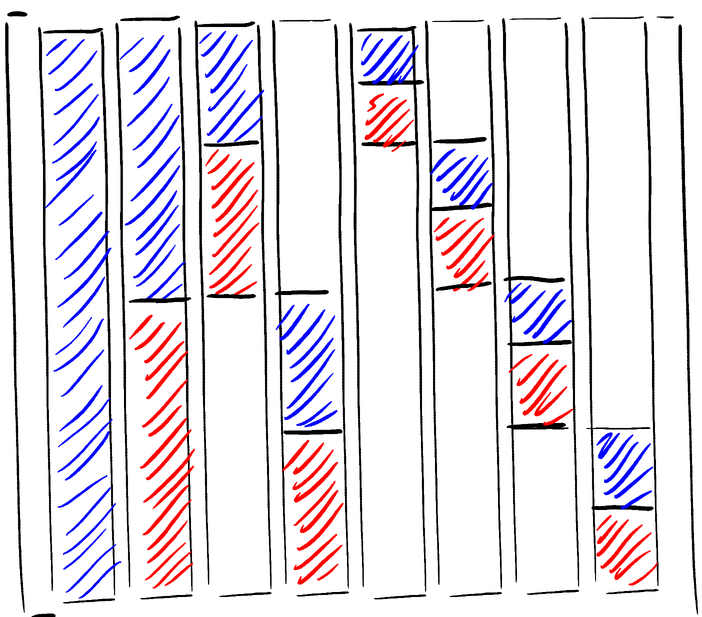
$$I = \Lambda^{-1} W^T \Lambda^{-1} C,$$

where C are the Haar wavelet coefficients.

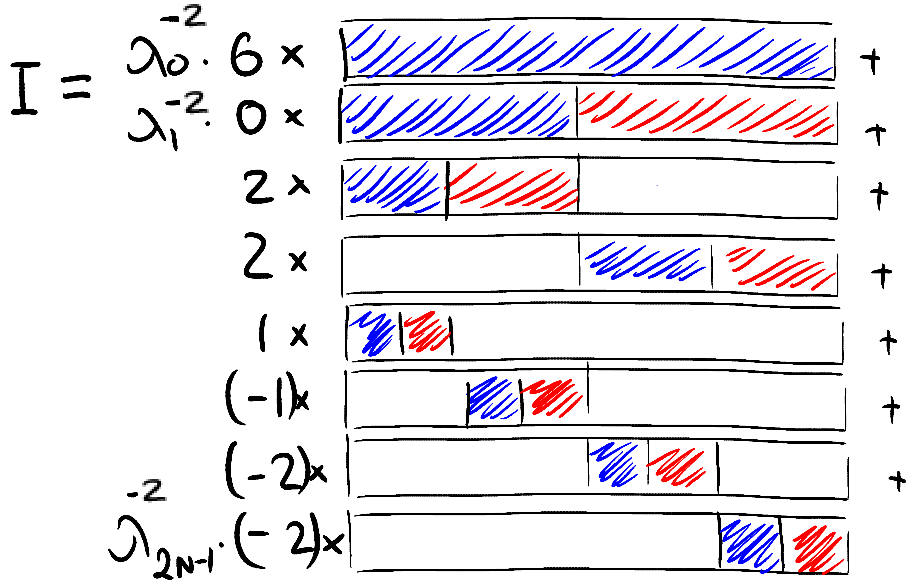
Reconstructing an Image from its Wavelet Coefs

$$\Lambda^{-1} = \begin{bmatrix} \lambda_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \lambda_{2^N}^{-1} \end{bmatrix}$$

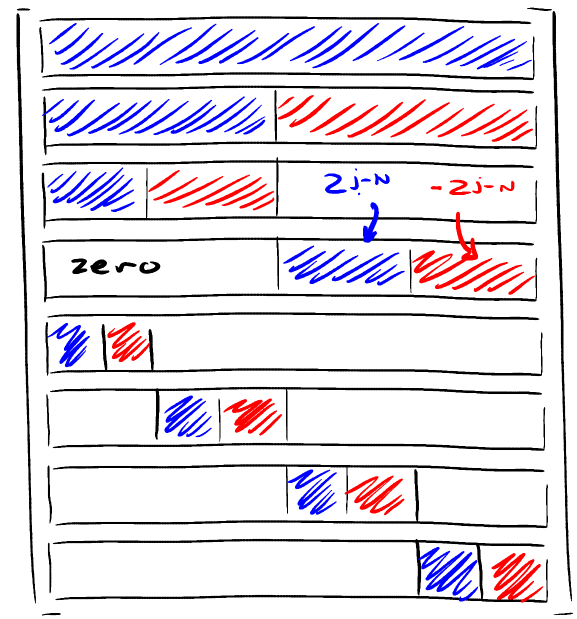
So we have

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

Interpreting the Wavelet Coefficients

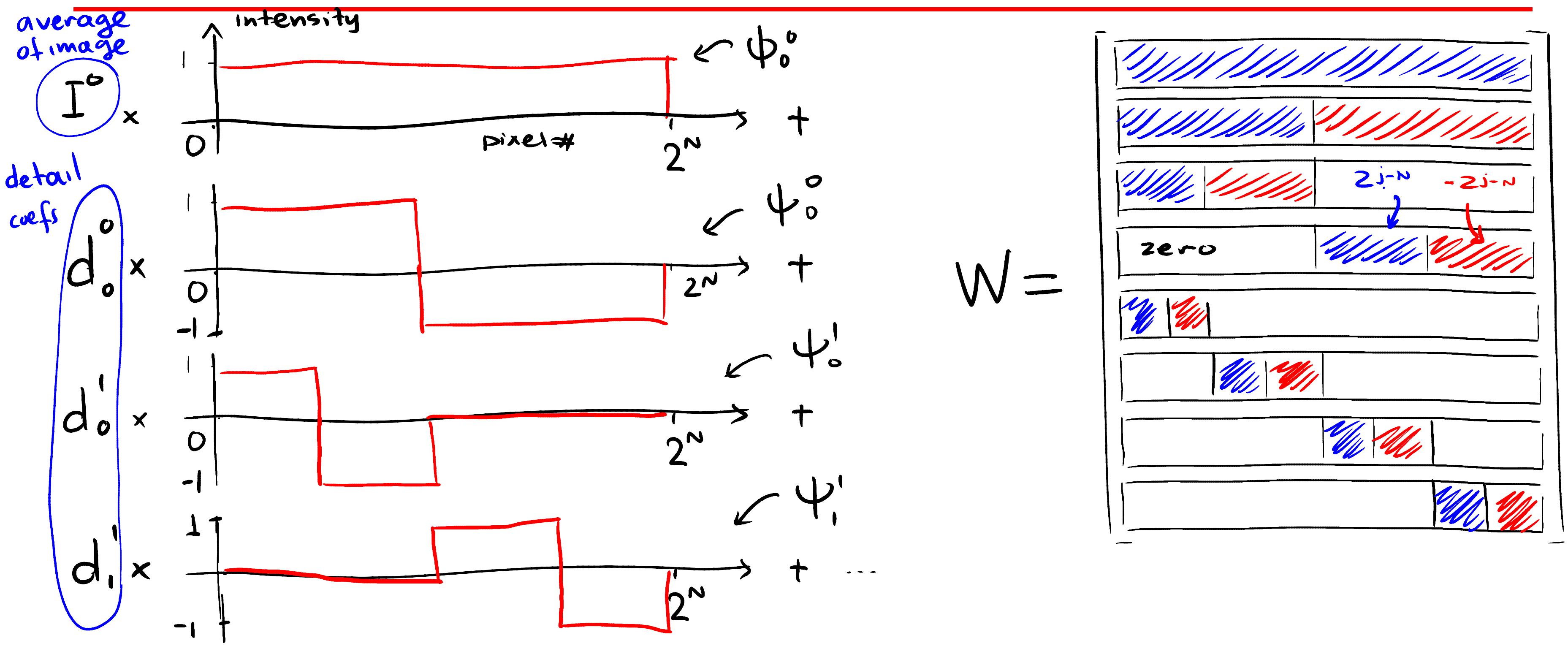


W =



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

Interpreting the Wavelet Coefficients

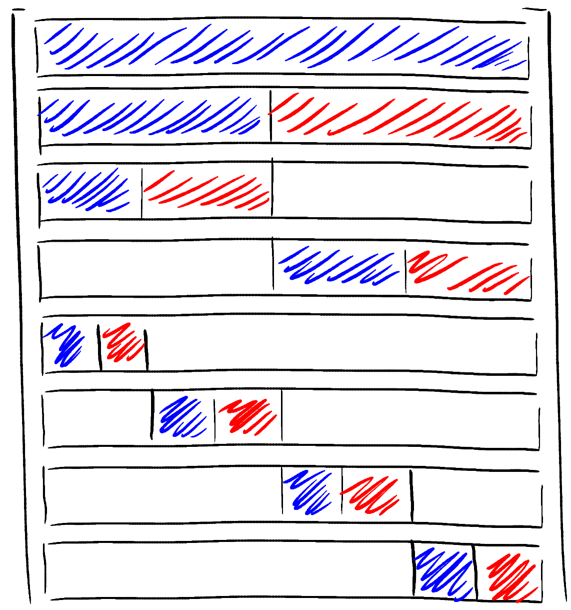


⇒ The wavelet coefficients are the coordinates of the image, considered as a vector in $\mathbb{R}^{2^N}$, in the basis defined by images $\phi_0^0, \psi_0^0, \psi_0^1, \ldots$

The Normalized Haar Wavelet Matrix

We can normalize the wavelet transform matrix by multiplying

$$\tilde{W} = \begin{bmatrix} \sqrt{a_1^{-1}} & & \\ & \ddots & \\ & & \sqrt{\lambda_{2^N-1}^{-1}} \end{bmatrix} \cdot W$$

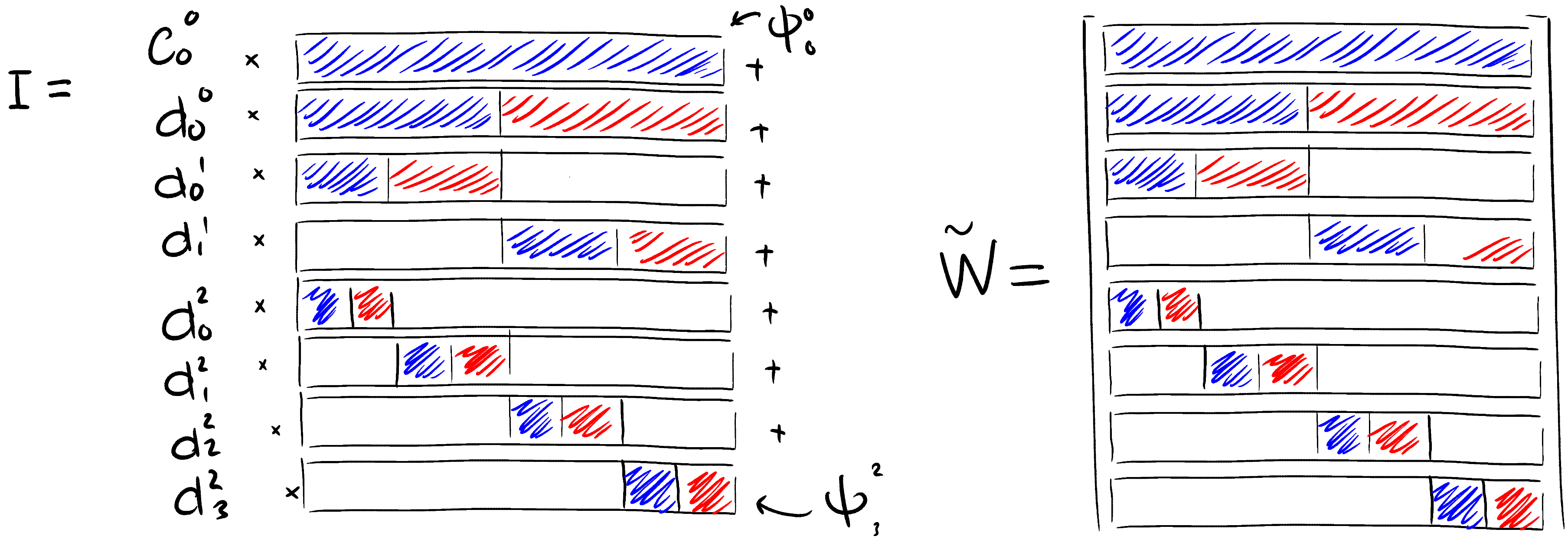
$\tilde{W} =$ 

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

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 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 coefs
- 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 $\tilde{W} I$

2. Sort the coefficients $c_0^0, d_0^1, d_1^1, \ldots$ in order of decreasing absolute value

3. Keep the top $k2^N$ coefficients (we know the basis)

Wavelet Compression Algorithm #1

Input: 1D Image I, desired error ε
Output: $k2^N$ 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 enough coefficients to satisfy $|\tilde{I} - I| < \varepsilon$

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 coefs
- 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 $\tilde{W} I_i$ 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

$$\left(\psi_i^j\right)^T \cdot \left(\psi_{i'}^{j'}\right)$$

where ψ_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 $2^N \times 2^N$