No Lectures next week

ECE160

Multimedia

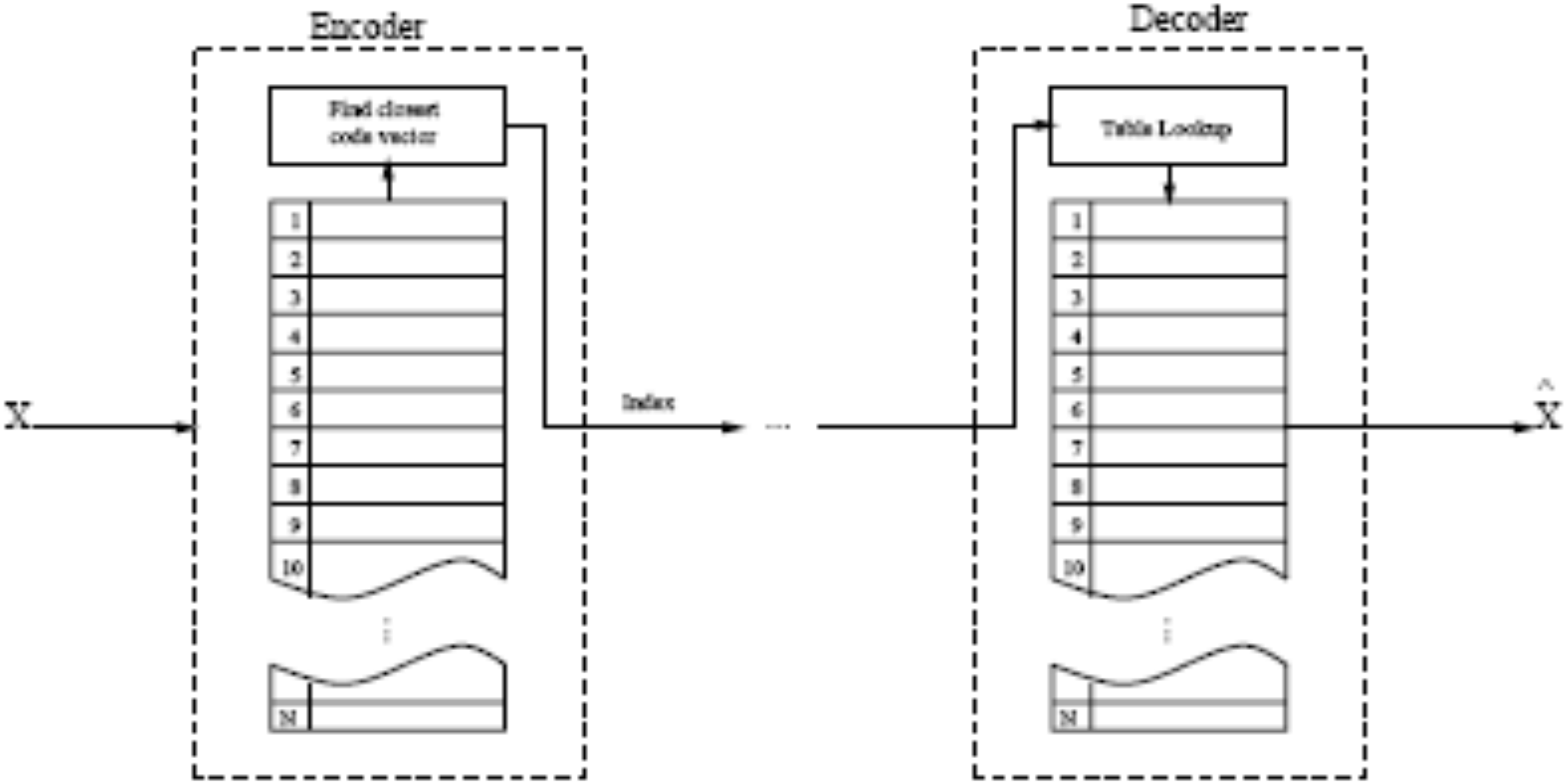
Lecture 8: Spring 2011

Lossy Compression Algorithms

Vector Quantization (VQ)

- According to Shannon's original work on information theory, any compression system performs better if it operates on vectors or groups of samples rather than individual symbols or samples.
- Form vectors of input samples by simply concatenating a number of consecutive samples into a single vector.
- Instead of single reconstruction values as in scalar quantization, in VQ *code vectors* with *n* components are used. A collection of these code vectors form the *codebook*.

Vector Quantization



Transform Coding

The rationale behind transform coding:

- If **Y** is the result of a linear transform **T** of the input vector **X** in such a way that the components of **Y** are much less correlated, then **Y** can be coded more efficiently than **X**.
- If most information is accurately described by the first few components of a transformed vector, then the remaining components can be coarsely quantized, or even set to zero, with little signal distortion.
  - Discrete Cosine Transform (DCT) will be studied first.
  - In addition, we will examine the Karhunen-Loeve Transform (KLT) which *optimally* decorrelates the components of the input **X**.

Spatial Frequency and DCT

- *Spatial frequency* indicates how many times pixel values change across an image block.
- The DCT formalizes this notion with a measure of how much the image contents change in correspondence to the number of cycles of a cosine wave per block.
- The role of the DCT is to *decompose* the original signal into its DC and AC components; the role of the IDCT is to *reconstruct* (re-compose) the signal.
- JPEG and MPEG use image blocks of 8x8 pixels

DCT

- Given an input function *f*(*i,j*) over two integer variables *i* and *j* (a piece of an image), the 2D DCT transforms it into a new function *F*(*u,v*), with integer *u* and *v* running over the same range as *i* and *j*. The general definition of the transform is:

$$F(u,v) = \frac{2\,C(u)\,C(v)}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos\frac{(2i+1)\cdot u\pi}{2M} \cdot \cos\frac{(2j+1)\cdot v\pi}{2N} \cdot f(i,j)$$

- where *i,u* = 0,1,…,*M* − 1; *j,v* = 0,1,…, *N* − 1; and the constants *C*(*u*) and *C*(*v*) are determined by

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & if \quad \xi = 0, \\ 1 & otherwise. \end{cases}$$

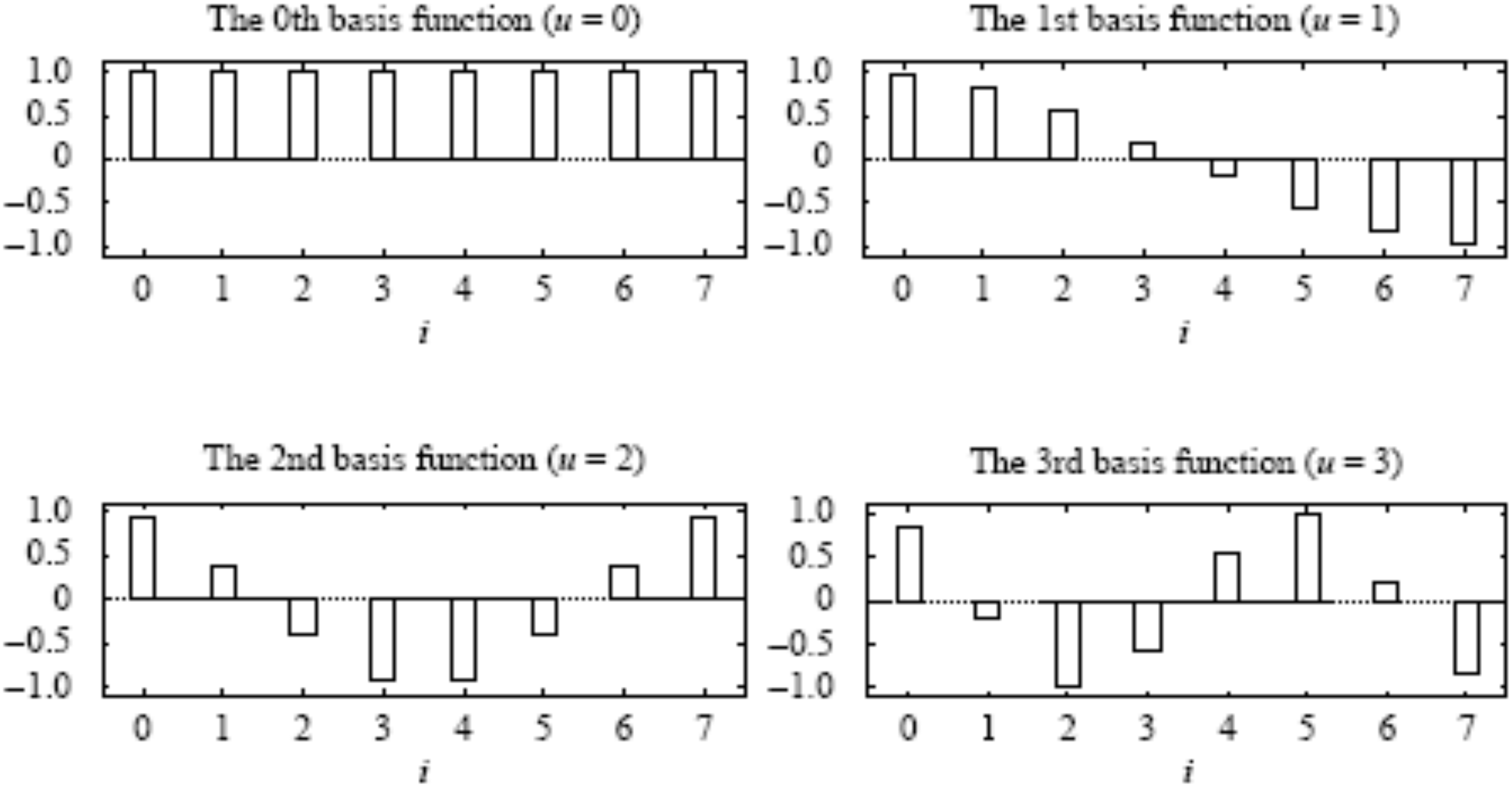
1D Discrete Cosine Transform (1D DCT)

$$F(u) = \frac{C(u)}{2} \sum_{i=0}^{7} \cos \frac{(2i+1)u\pi}{16} f(i)$$

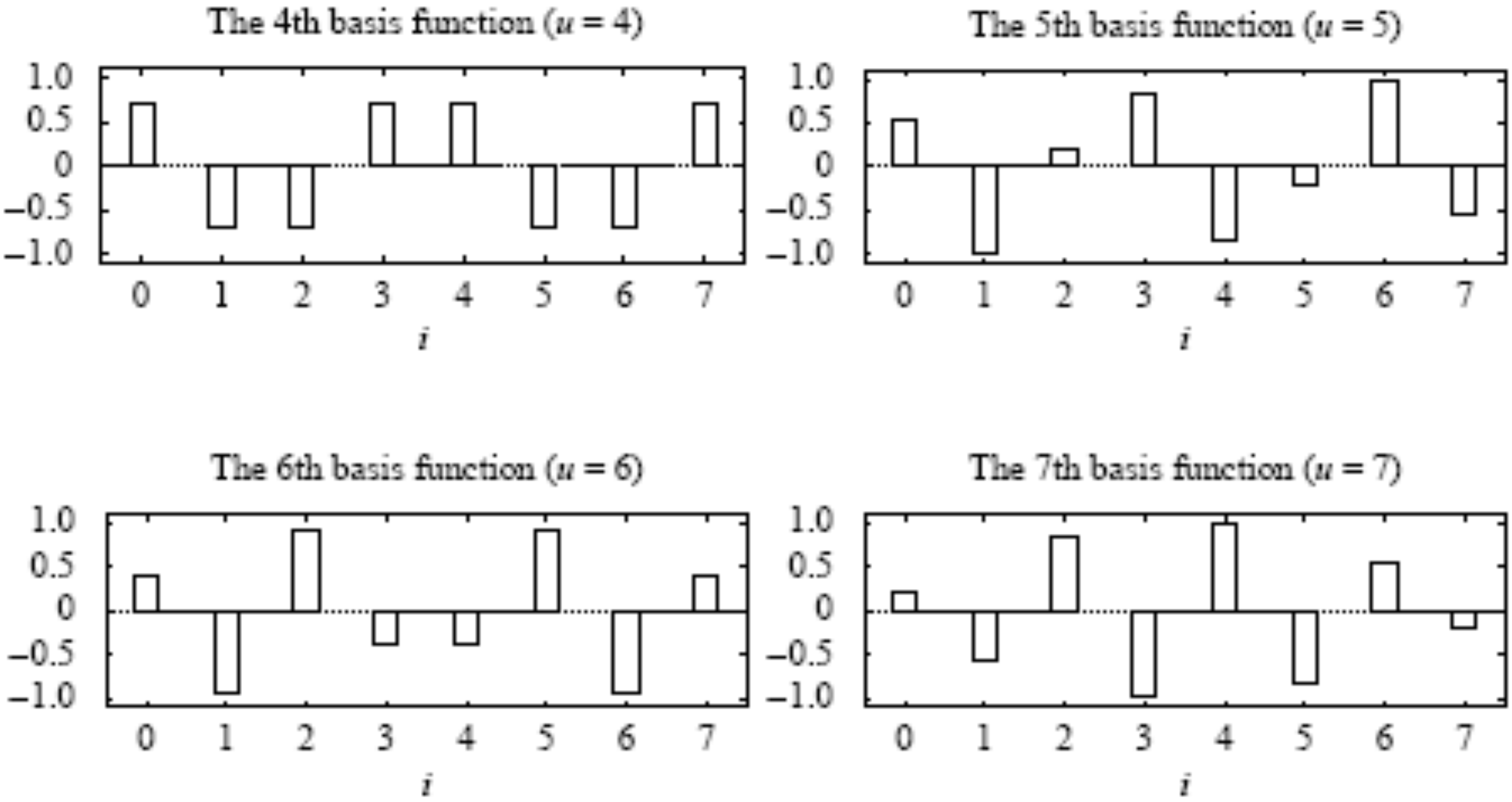
with inverse

$$f(i) = \sum_{u=0}^{7} \frac{C(u)}{2} \cos \frac{(2i+1)u\pi}{16} F(u)$$

1D Discrete Cosine Transform (1D DCT)



1D Discrete Cosine Transform (1D DCT)



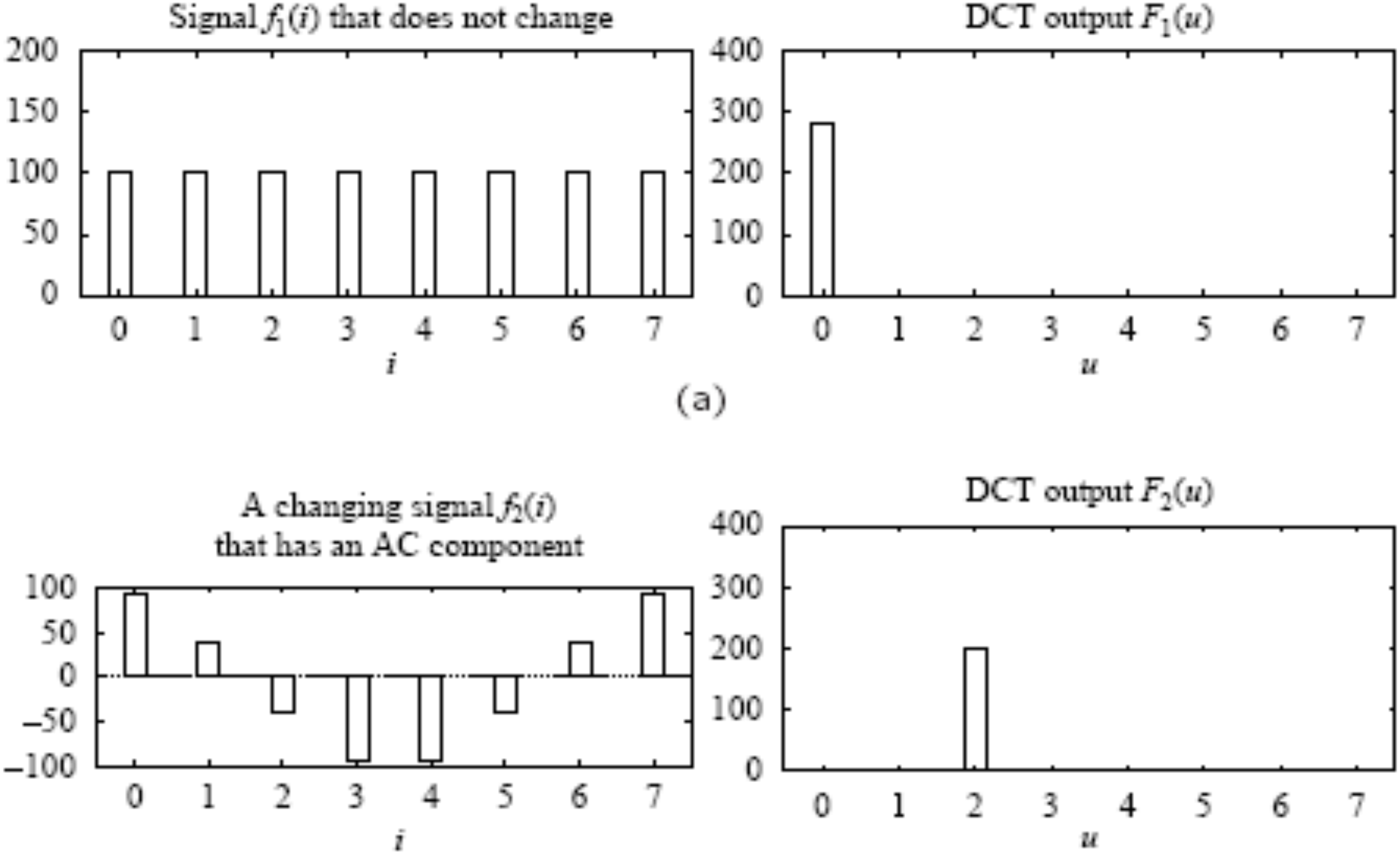
The 4th basis function ($u = 4$)

The 5th basis function ($u = 5$)

The 6th basis function ($u = 6$)

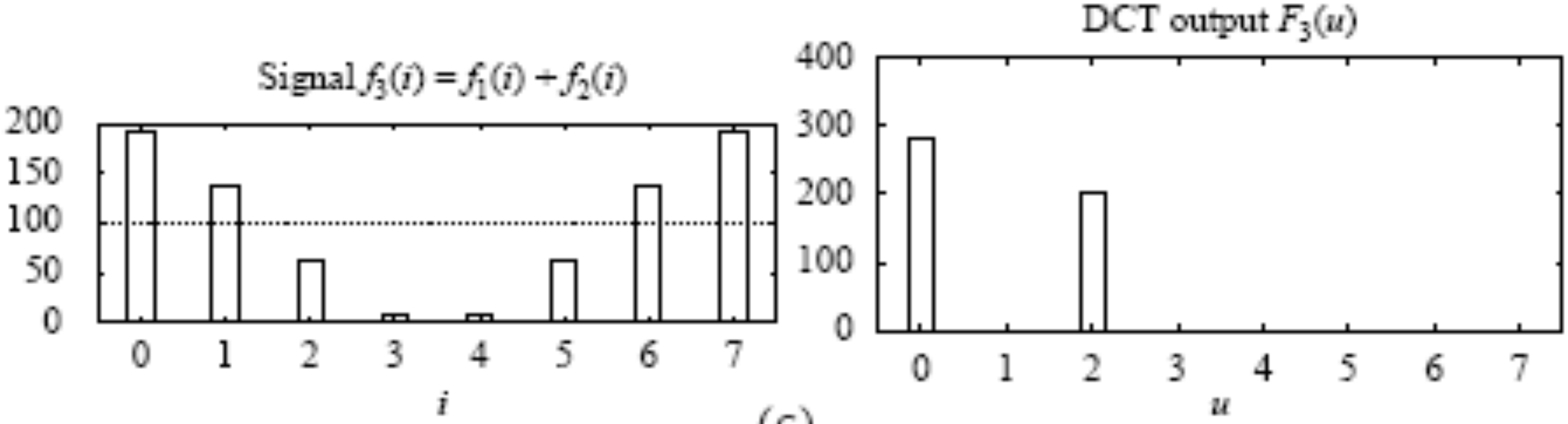
The 7th basis function ($u = 7$)

1D Discrete Cosine Transform (1D DCT)

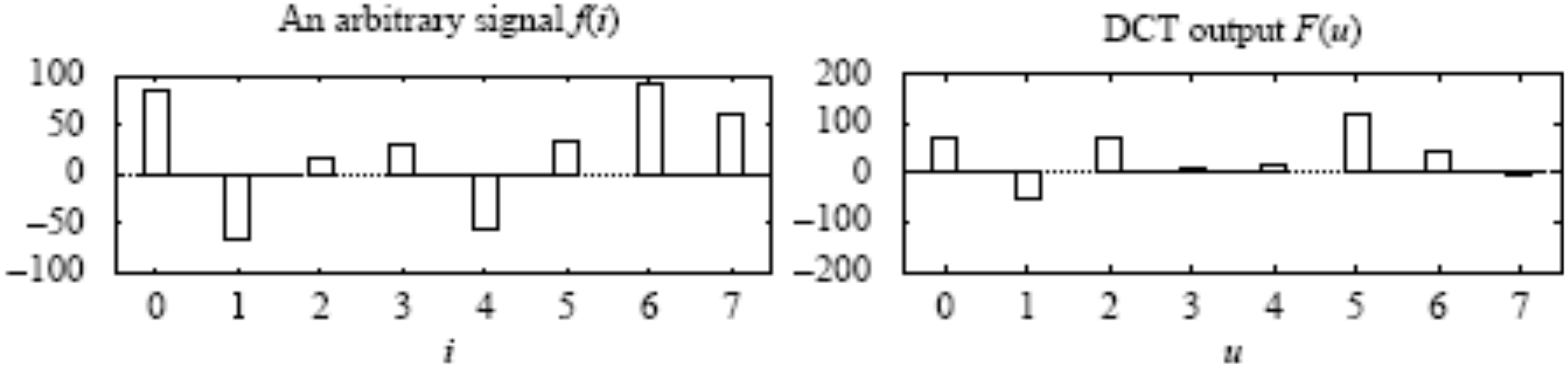


(a)

1D Discrete Cosine Transform (1D DCT)



(c)



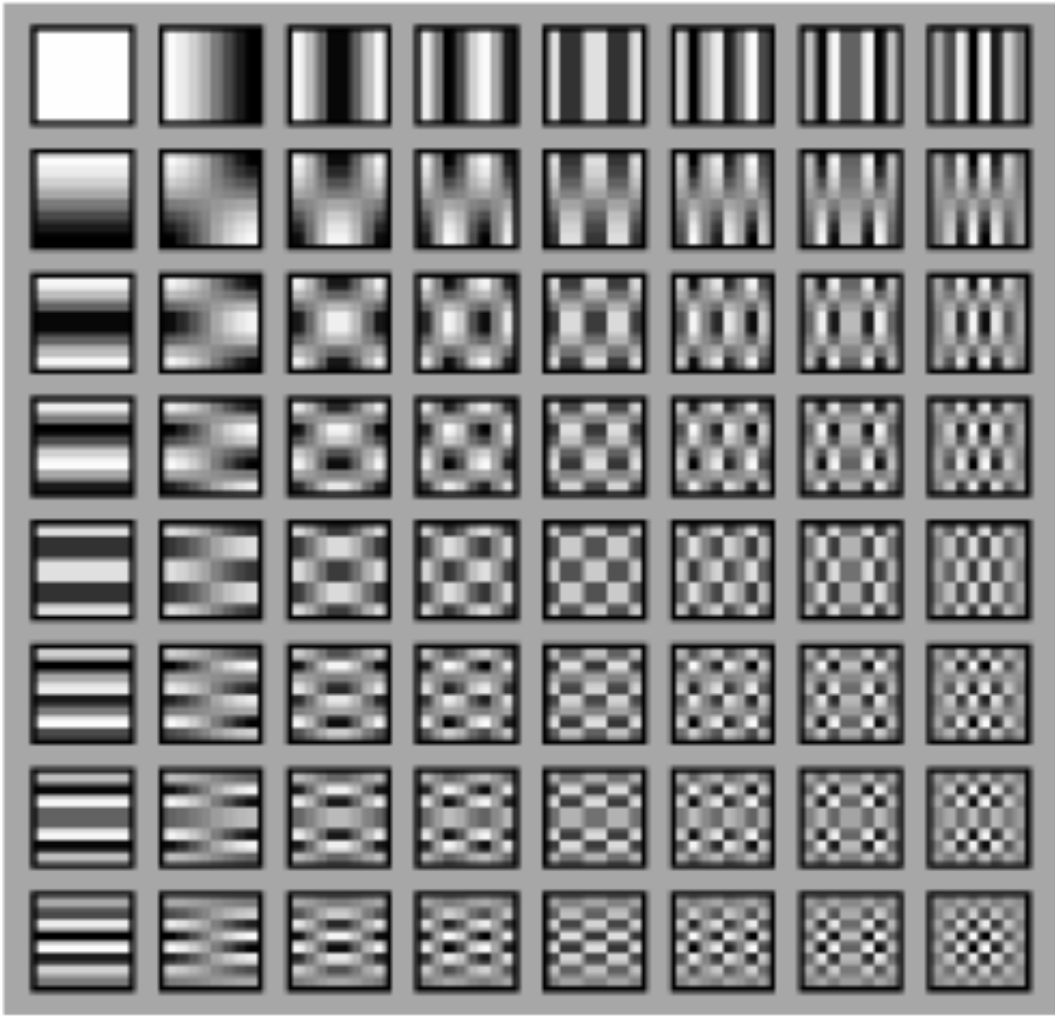
2D Discrete Cosine Transform (2D DCT)

$$F(u,v) = \frac{C(u)\,C(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} f(i,j)$$

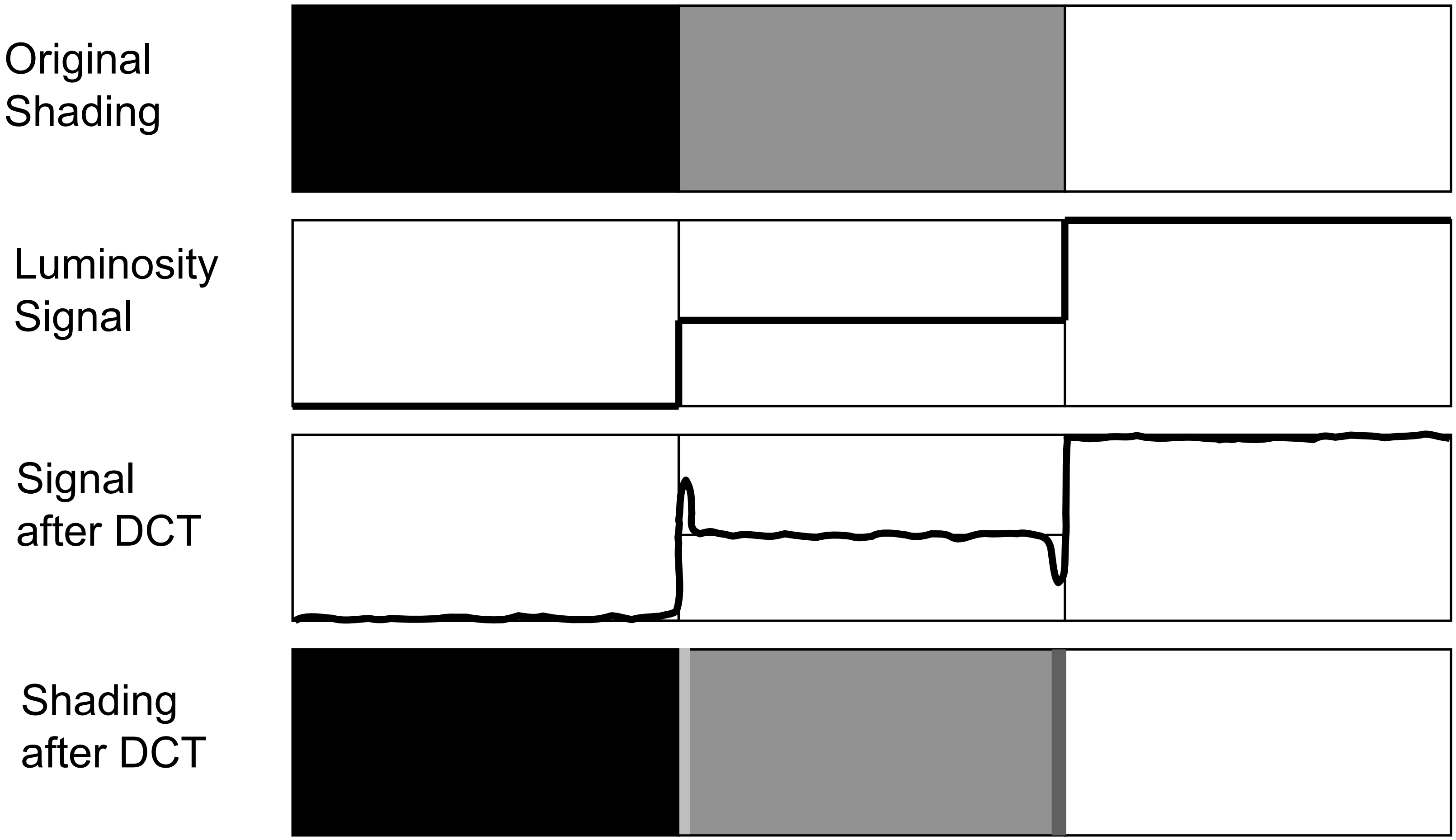
with inverse

$$\tilde{f}(i,j) = \sum_{u=0}^{7} \sum_{v=0}^{7} \frac{C(u)\,C(v)}{4} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} F(u,v)$$

2D Discrete Cosine Transform (2D DCT)



DCT Artifact



Original Shading

Luminosity Signal

Signal after DCT

Shading after DCT

Karhunen-Loeve Transform (KLT)

- The Karhunen-Loeve transform is a reversible linear transform that exploits the statistical properties of the vector representation.
- It optimally decorrelates the input signal.
- To understand the optimality of the KLT, consider the autocorrelation matrix **RX** of the input vector **X** defined as:

$$\mathrm{R}_{\mathrm{X}} = E[\mathrm{X}\mathrm{X}^T]$$

$$= \begin{bmatrix} R_X(1,1) & R_X(1,2) & \cdots & R_X(1,k) \\ R_X(2,1) & R_X(2,2) & \cdots & R_X(2,k) \\ \vdots & \vdots & \ddots & \vdots \\ R_X(k,1) & R_X(k,2) & \cdots & R_X(k,k) \end{bmatrix}$$

KLT Example

- To illustrate the mechanics of the KLT, consider the four 3D input vectors **x**1 = (4,4,5), **x**2 = (3,2,5), **x**3 = (5,7,6), and **x**4 = (6,7,7).
- Estimate the mean:

$$\mathbf{m}_x = \frac{1}{4}\begin{bmatrix} 18 \\ 20 \\ 23 \end{bmatrix}$$

- Estimate the autocorrelation matrix of the input:

$$\mathrm{R_X} = \frac{1}{M}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T - \mathbf{m}_x \mathbf{m}_x^T$$

$$= \begin{bmatrix} 1.25 & 2.25 & 0.88 \\ 2.25 & 4.50 & 1.50 \\ 0.88 & 1.50 & 0.69 \end{bmatrix}$$

KLT Example

- The eigenvalues of $\mathbf{R_X}$ are $\lambda_1 = 6.1963$, $\lambda_2 = 0.2147$, and $\lambda_3 = 0.0264$. The corresponding eigenvectors are:

$$\mathbf{u}_1 = \begin{bmatrix} 0.4385 \\ 0.8471 \\ 0.3003 \end{bmatrix}, \quad \mathbf{u}_2 = \begin{bmatrix} 0.4460 \\ -0.4952 \\ 0.7456 \end{bmatrix}, \quad \mathbf{u}_3 = \begin{bmatrix} -0.7803 \\ 0.1929 \\ 0.5949 \end{bmatrix}$$

- The KLT is given by the matrix

$$\mathbf{T} = \begin{bmatrix} 0.4385 & 0.8471 & 0.3003 \\ 0.4460 & -0.4952 & 0.7456 \\ -0.7803 & 0.1929 & 0.5949 \end{bmatrix}$$

KLT Example

- Subtracting the mean vector from each input vector and apply the KLT

$$y_1 = \begin{bmatrix} -1.2916 \\ -0.2870 \\ -0.2490 \end{bmatrix}, \quad y_2 = \begin{bmatrix} -3.4242 \\ 0.2573 \\ 0.1453 \end{bmatrix}$$

$$y_3 = \begin{bmatrix} 1.9885 \\ -0.5809 \\ 0.1445 \end{bmatrix}, \quad y_4 = \begin{bmatrix} 2.7273 \\ 0.6107 \\ -0.0408 \end{bmatrix}$$

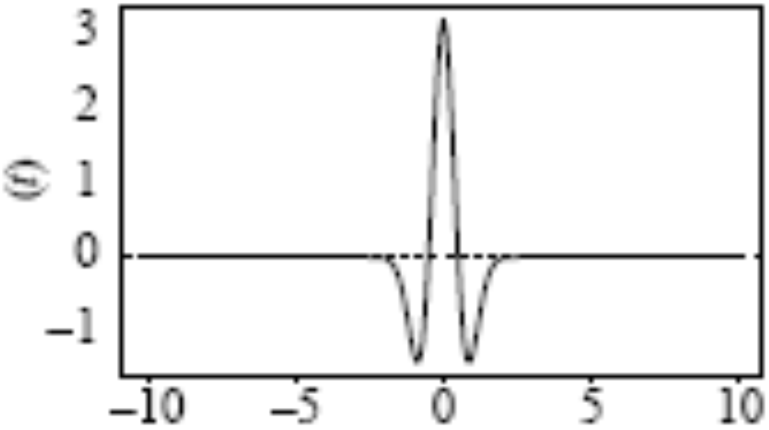
- Since the rows of **T** are orthonormal vectors, the inverse transform is just the transpose: $\mathbf{T}-1 = \mathbf{T}\mathit{T}$, and

$$\mathbf{x} = \mathbf{T}\mathit{T}\mathbf{y}+\mathbf{m}x$$

- In general, after the KLT most of the “energy" of the transform coefficients are concentrated within the first few components. This is the “energy compaction" property of the KLT.

Wavelet-Based Coding

- The objective of the wavelet transform is to decompose the input signal into components that are easier to deal with, have special interpretations, or have some components that can be thresholded away, for compression purposes.
- We want to be able to at least approximately reconstruct the original signal given these components.
- The basis functions of the wavelet transform are localized in space, time and frequency.
- There are two types of wavelet transforms: the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT).



The Discrete Wavelet Transform

- Discrete wavelets are formed from a mother wavelet, with scale and shift in discrete steps.
- The DWT makes the connection between wavelets in the continuous time domain and “filter banks" in the discrete time domain in a multiresolution analysis framework.
- It is possible to show that the dilated and translated family of wavelets

$$\left\{\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\,\psi\left(\frac{t-2^j n}{2^j}\right)\right\}_{(j,n)\in\mathbf{Z}^2}$$

form an *orthonormal basis* of **L**2(*R*).

Multiresolution Analysis in the Wavelet Domain

- Multiresolution analysis provides the tool to *adapt signal resolution to only relevant details* for a particular task.
- The approximation component is then recursively decomposed into approximation and detail at successively coarser scales.
- Wavelet functions $\psi(t)$ are used to characterize detail information. The averaging (approximation) information is formally determined by a kind of dual to the mother wavelet, called the “scaling function" $\varphi(t)$.
- Wavelets are set up such that the approximation at resolution 2^{-j} contains all the necessary information to compute an approximation at coarser resolution $2^{-(j+1)}$.

Multiresolution Analysis in the Wavelet Domain

- The scaling function must satisfy the so-called **dilation equation**:

$$\phi(t) \;=\; \sum_{n\in\mathbf{Z}} \sqrt{2}h_0[n]\phi(2t-n)$$

- The *wavelet* at the coarser level is also expressible as a sum of translated scaling functions:

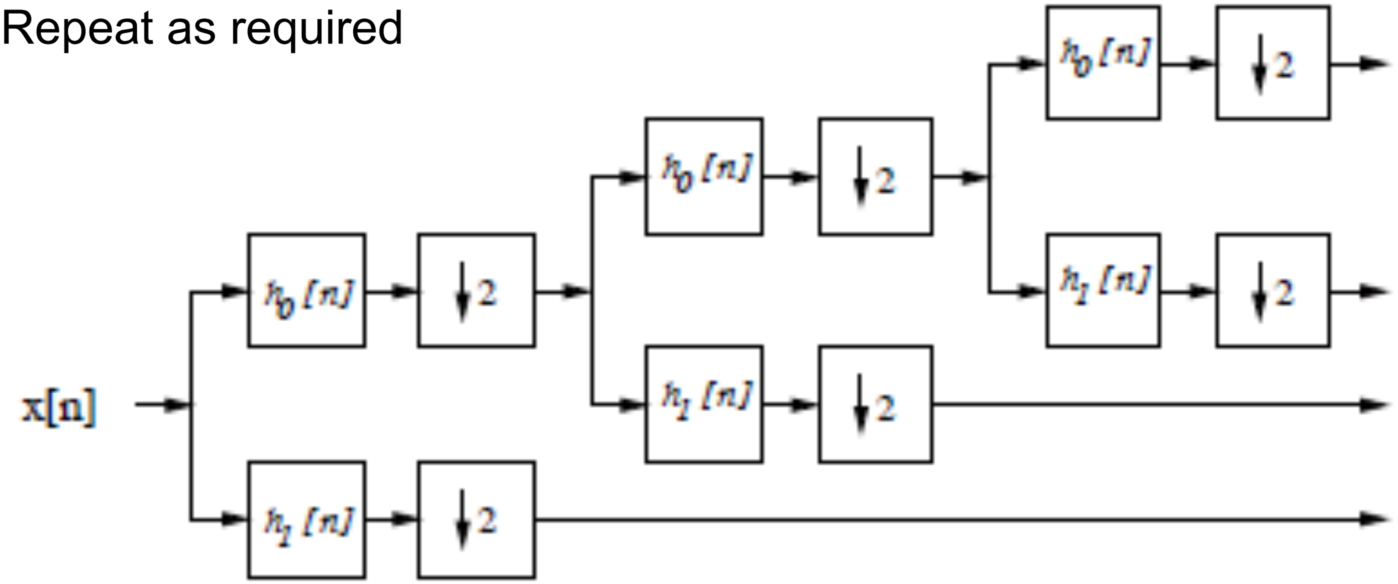
$$\psi(t) = \sum_{n\in\mathbf{Z}} \sqrt{2}h_1[n]\phi(2t-n)$$

$$\psi(t) = \sum_{n\in\mathbf{Z}} (-1)^n h_0[1-n]\phi(2t-n)$$

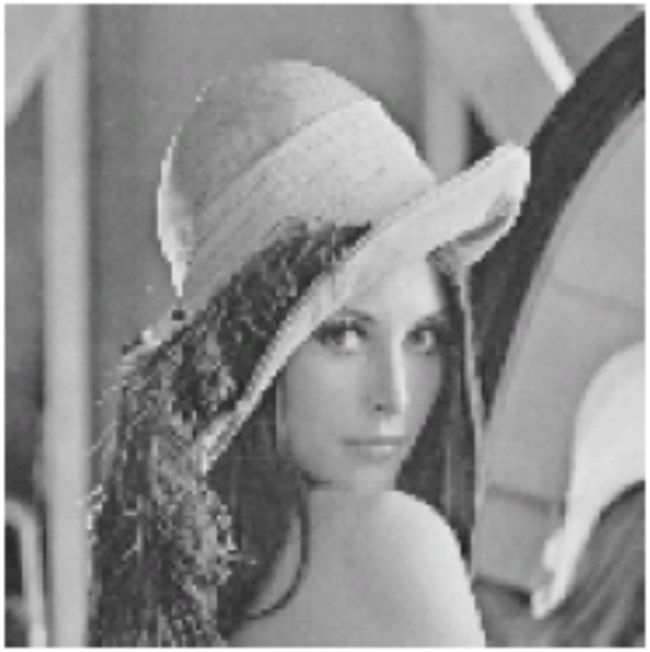
- The vectors *h*0[*n*] and *h*1[*n*] are called the low-pass and highpass *analysis* filters. To *reconstruct* the original input, an inverse operation is needed. The inverse filters are called *synthesis* filters.

Wavelet Transform

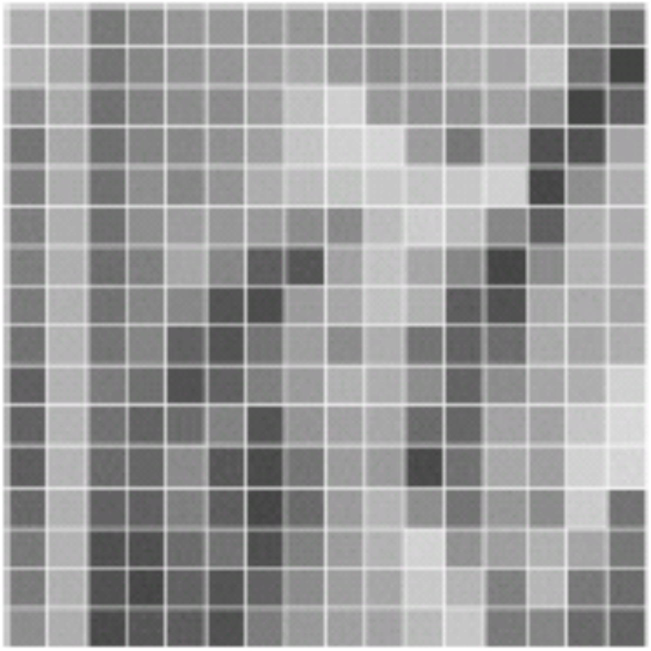
High pass and low pass filters
Down sample
Repeat as required



Down Sampling



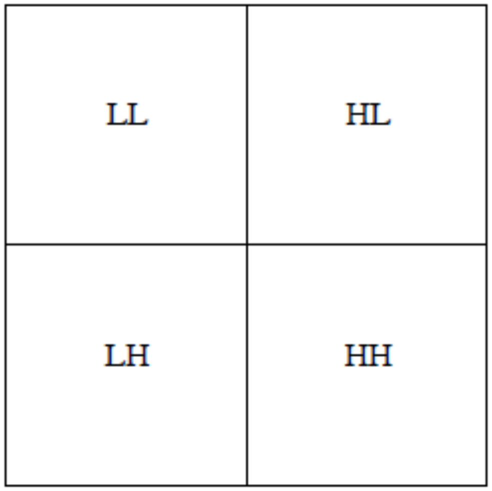
128x128 image



Downsampled to 16x16

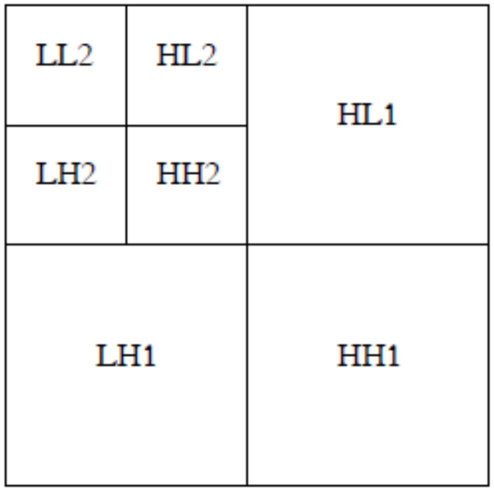
Wavelet Transform

- Low pass and high pass in both x and y followed by down sampling by 2



LL HL

LH HH



LL2 HL2

LH2 HH2

HL1

LH1 HH1

Embedded Zerotree of Wavelet Coefficients

- Effective and computationally efficient for image coding.
- The EZW algorithm addresses two problems:
  1. Obtain the best image quality for a given bit-rate, and
  2. Accomplish this task in an embedded fashion.
- Using an embedded code allows the encoder to terminate the encoding at any point. Hence, the encoder is able to meet any target bit-rate exactly.
- Similarly, a decoder can cease to decode at any point and can produce reconstructions corresponding to all lower-rate encodings.

The Zerotree Data Structure

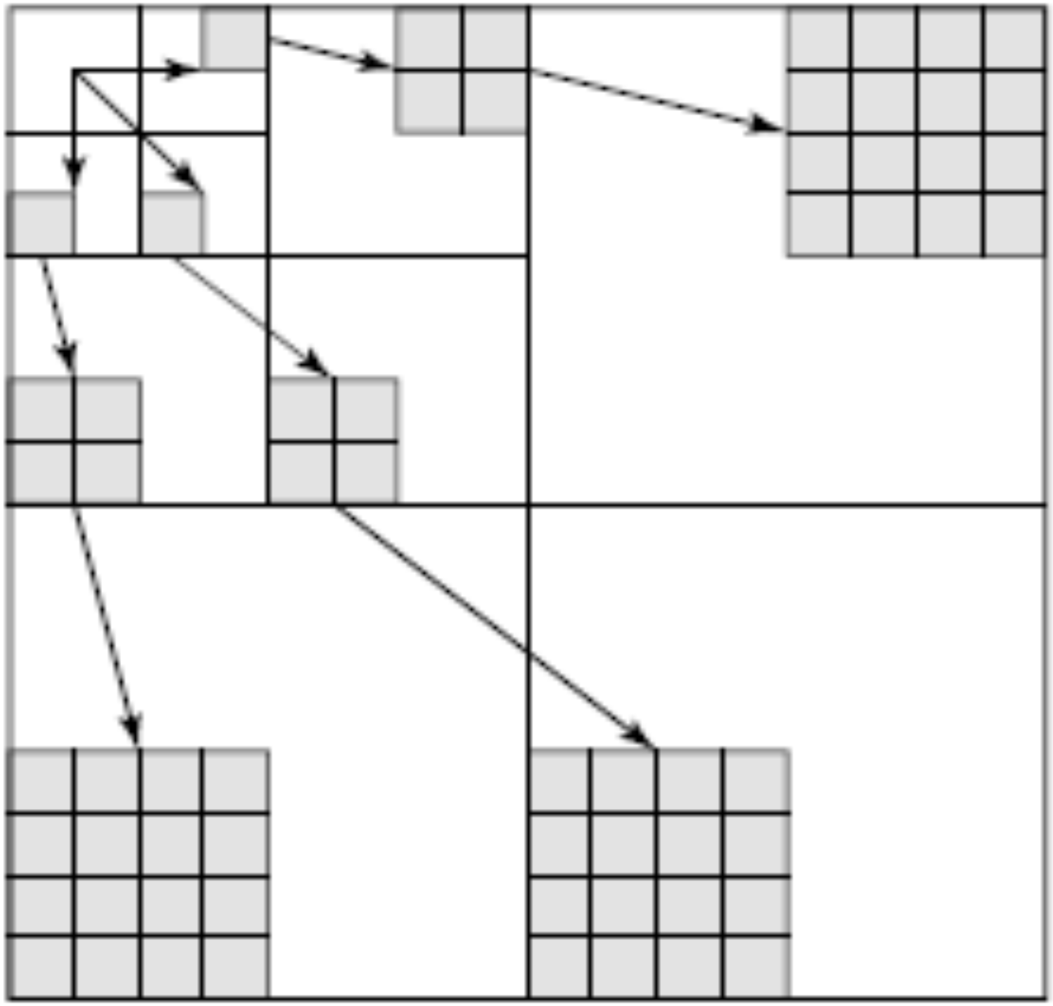
- The EZW algorithm efficiently codes the “significance map“ which indicates the locations of nonzero quantized wavelet coefficients.
- This is is achieved using a new data structure called the *zerotree*.
- Using the hierarchical wavelet decomposition, we can relate every coefficient at a given scale to a set of coefficients at the next finer scale of similar orientation.
- The coefficient at the coarse scale is called the “parent“ while all corresponding coefficients are the next finer scale of the same spatial location and similar orientation are called “children".

The Zerotree Data Structure

- Given a threshold *T*, a coefficient *x* is an element of the zerotree if it is insignificant and all of its descendants are insignificant as well.
- The significance map is coded using the zerotree with a four symbol alphabet:
  - **The zerotree root**: The root of the zerotree is encoded with a special symbol indicating that the insignificance of the coefficients at finer scales is completely predictable.
  - **Isolated zero**: The coefficient is insignificant but has some significant descendants.
  - **Positive significance**: The coefficient is significant with a positive value.
  - **Negative significance**: The coefficient is significant with a negative value.

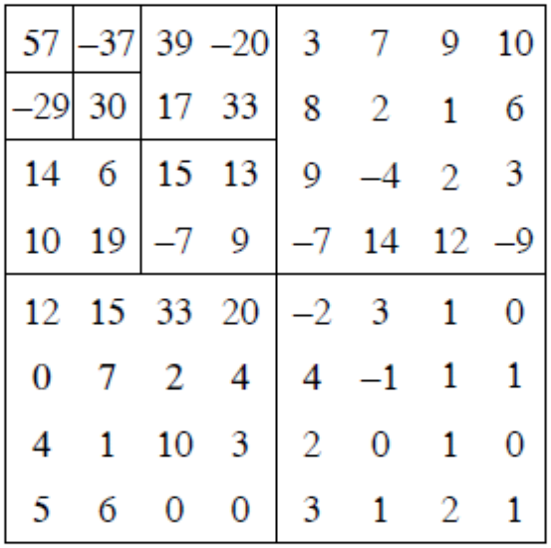
The Zerotree Data Structure

Nodes are:
Root node
Positive P
Negative N
Zero but has significant descendants Z
Zero and all descendants are zero T



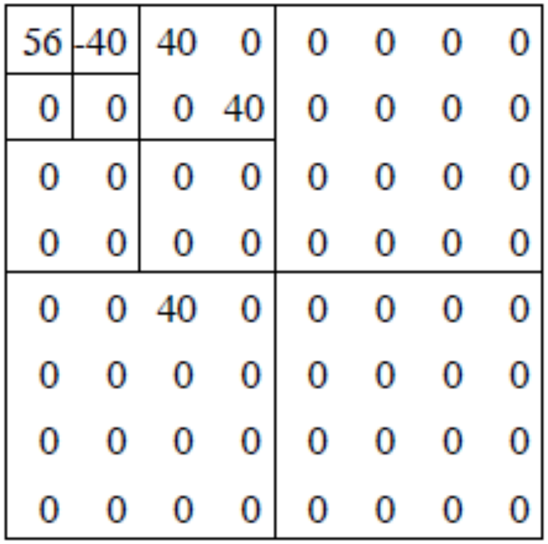
First stage of Zero Tree

Three stage wavelet transform



57	–37	39	–20	3	7	9	10
–29	30	17	33	8	2	1	6
14	6	15	13	9	–4	2	3
10	19	–7	9	–7	14	12	–9
12	15	33	20	–2	3	1	0
0	7	2	4	4	–1	1	1
4	1	10	3	2	0	1	0
5	6	0	0	3	1	2	1

Threshold is 32



56	-40	40	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	40	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

PNZT PTTP TZTT PTTT

Next threshold is 48

Successive Approximation Quantization

Motivation:

- Takes advantage of the efficient encoding of the significance map using the zerotree data structure by allowing it to encode more significance maps.
- Produces an embedded code that provides a coarse-to-fine, multiprecision logarithmic representation of the scale space corresponding to the wavelet-transformed image.
- The SAQ method sequentially applies a sequence of thresholds $T_0,\ldots,T_{N-1}$ to determine the significance of each coefficient.
- A *dominant list* and a *subordinate list* are maintained during the encoding and decoding process.

Set Partitioning in Hierarchical Trees (SPIHT)

- The SPIHT algorithm is an extension of the EZW algorithm.
- The SPIHT algorithm significantly improved the performance of its predecessor by changing the way subsets of coefficients are partitioned and how refinement information is conveyed.
- A unique property of the SPIHT bitstream is its compactness. The resulting bitstream from the SPIHT algorithm is so compact that passing it through an entropy coder would only produce very marginal gain in compression.
- No ordering information is explicitly transmitted to the decoder. Instead, the decoder reproduces the execution path of the encoder and recovers the ordering information.