# Wavelet-based Image Compression

**Amir Said (said@hpl.hp.com)**

**Imaging Technology Department**
**Hewlett Packard Laboratories**
**Palo Alto, CA**

# Overview

- Image compression features
- Principles of image compression
- Transform coding
- Wavelet image transforms
- Properties of image wavelet coefficients
- Efficient coding wavelet coefficients
- Zerotrees and set partitioning
- Application issues
- Conclusions

# Image Compression Standards

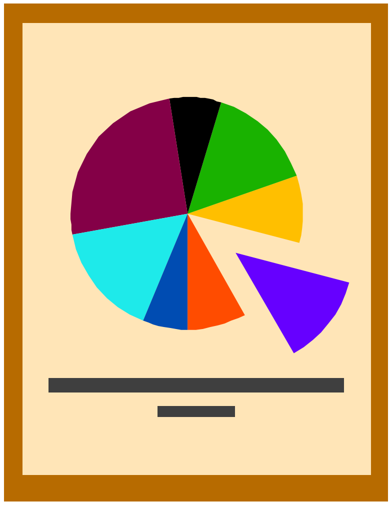
- Good old times: JPEG
  - Choose quality number
  - Compress image
  - Other features proposed, but most not widely supported
- New standard: JPEG 2000
  - Based on HP Labs proposal
  - Wavelet-based compression
  - MANY new features
  - Complex file structure, coding, decoding, etc.

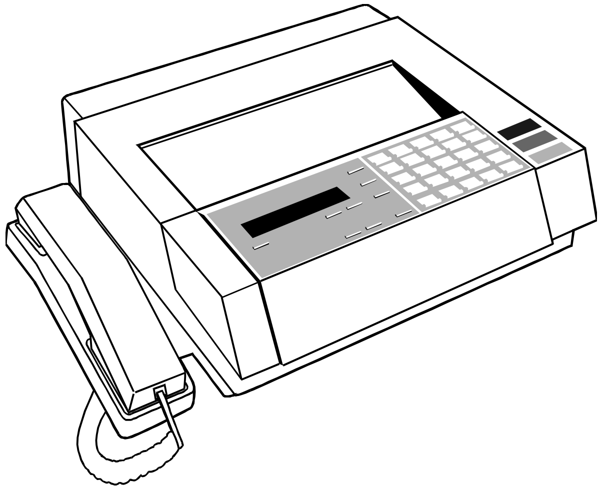
# Image Types

**“Natural”**
**wavelets, JPEG**

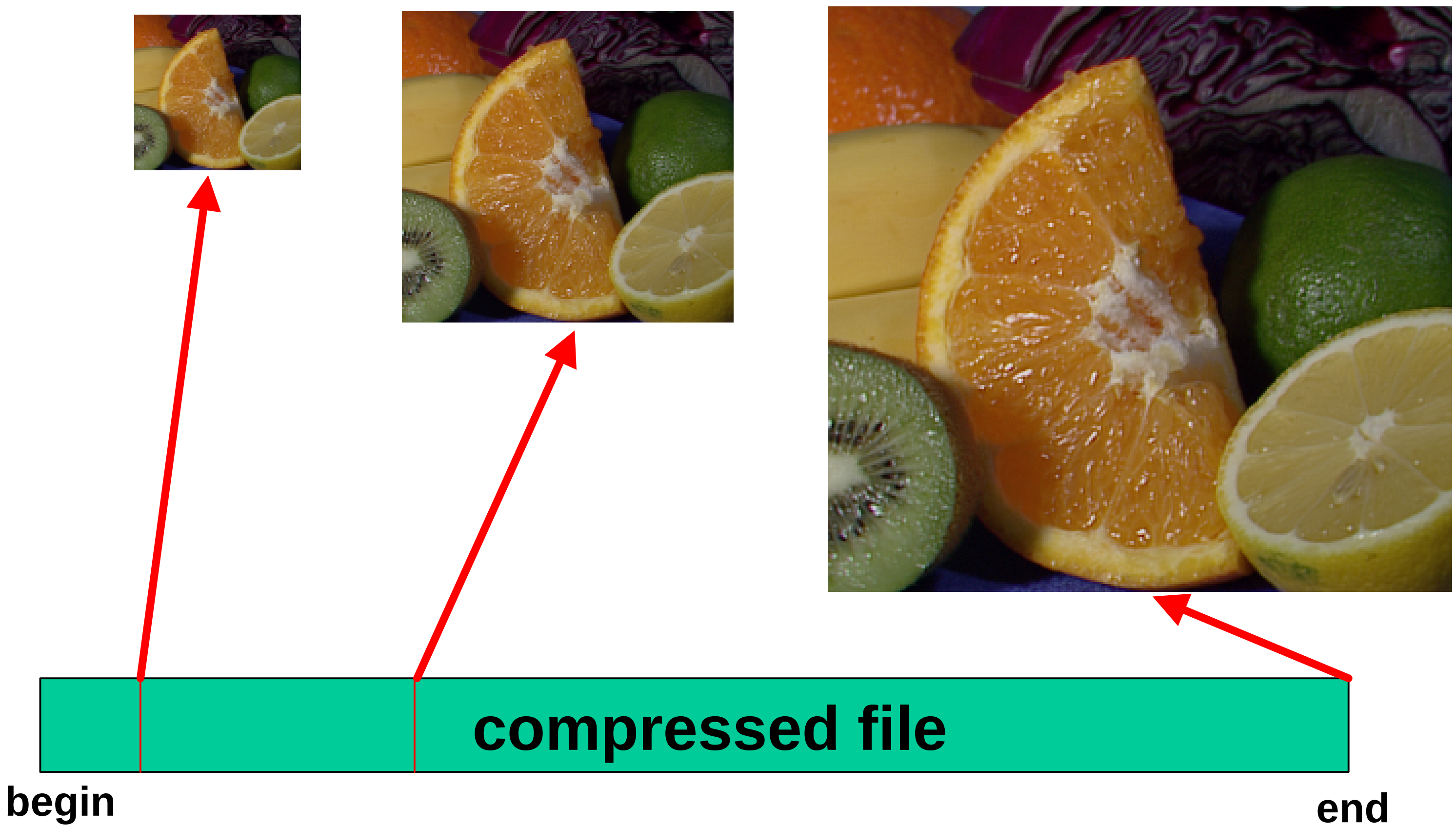
**Graphics**
**GIF**

**Black & White**
**fax**

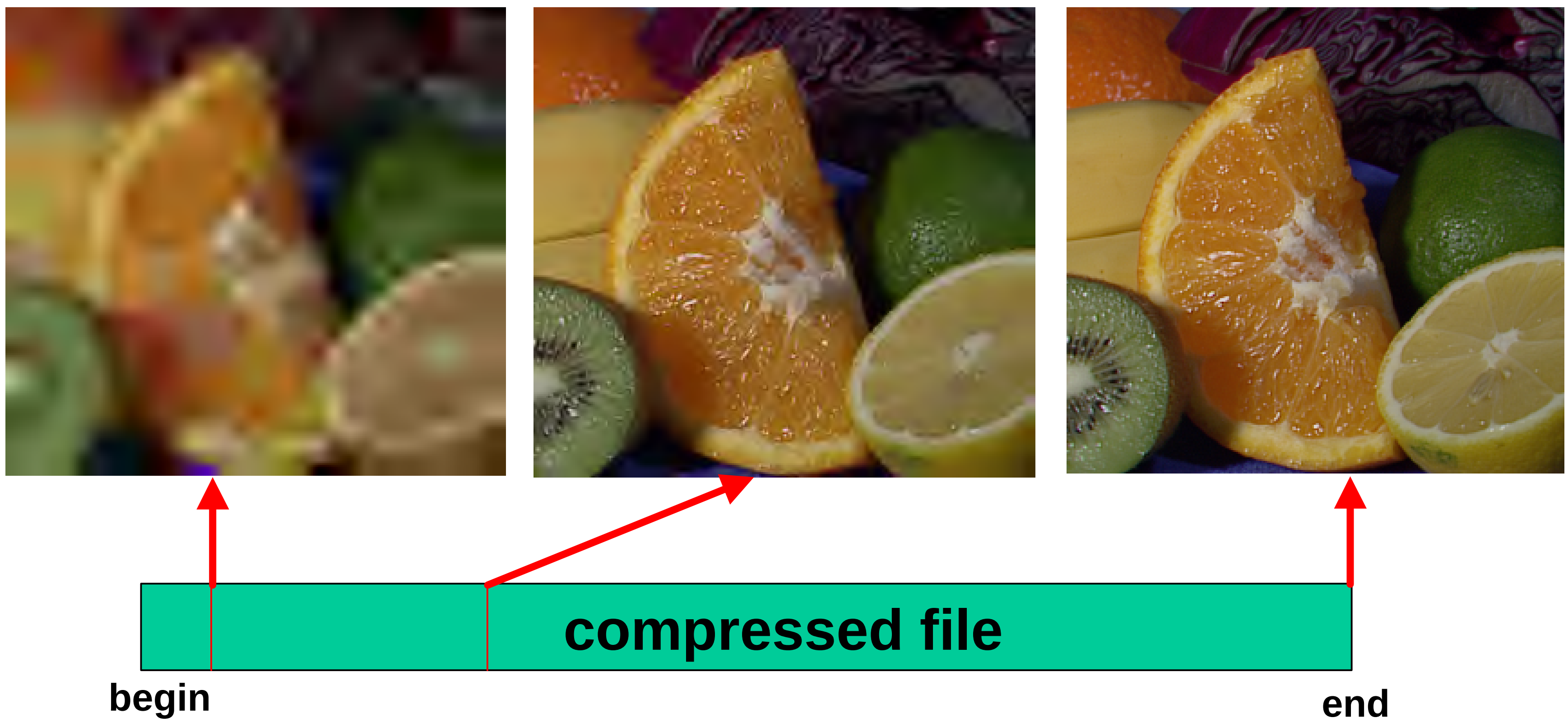




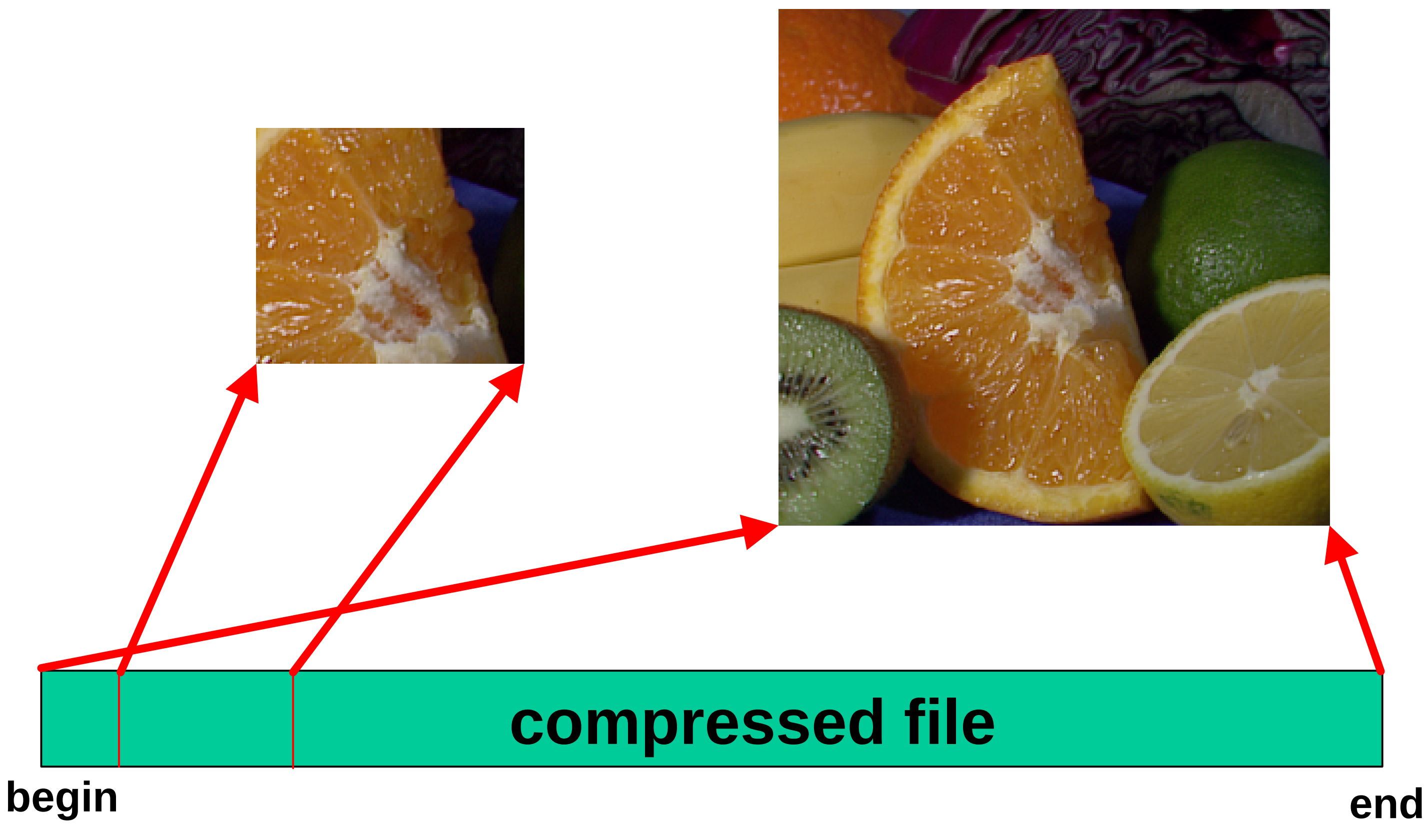
# Progressive Resolution



# Progressive Quality



# Random Access



# Desirable Compression Features

- Scalability
  - same algorithm for wide range of quality, compression ratios
- Flexibility/adaptability
  - efficient compression on wide range of image characteristics
- Automatic rate and quality control
  - one-pass creation of compressed file with desired size or quality
- Same algorithm for lossy and lossless compression
- Support for Region-of-Interest (ROI) decoding
- Low complexity (speed, memory)
- Efficient compression
- Error resilience
- Good visual quality

# Image Compression

- Based on the elimination of data that is
  1. Redundant
  2. Irrelevant
- Redundancy is reduced by using more efficient representation
  - lossless process
  - *entropy-coding*
- Irrelevant data is discarded
  - lossy process
  - depends on image use
    - subjective visual quality
    - maximum error

# Entropy Coding

- **Standard coding techniques**
  - **Huffman codes (good compression, fast)**
  - **Arithmetic codes (better compression, slower)**
  - **Lempel-Ziv (not so good)**

- **Techniques specialized for images**
  - **Exploit two-dimensional structure**
    - **Example: code large single-color square by identifying (origin, size, color)**
  - **Use properties that are present in normal images**
  - **Exploit structures on different scales**

# Quantization

- Reduction of allowable images to a much smaller set
  - Example: set least significant bit in pixel values to zero
- The new set should contain most important cases
  - Difference should be the irrelevant data
- Quantization is tightly connected to entropy coding
  - Smaller number of possible outcomes means less bits
- Scalar quantization

reconstructed pixel value

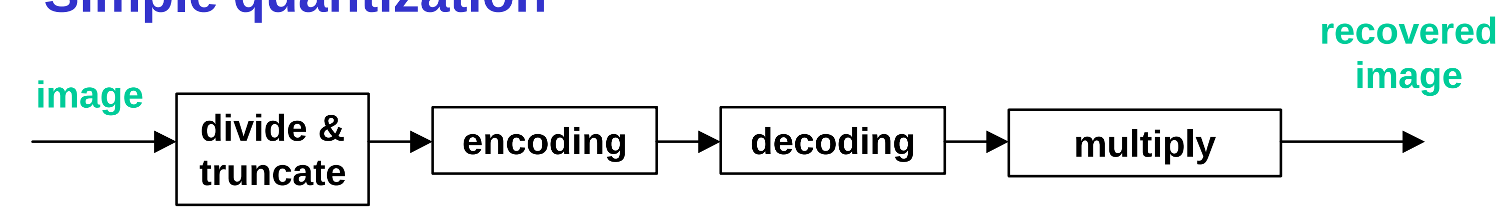
$$\hat{a} = d\left\lfloor \frac{a + d/2}{d} \right\rfloor$$

data coded

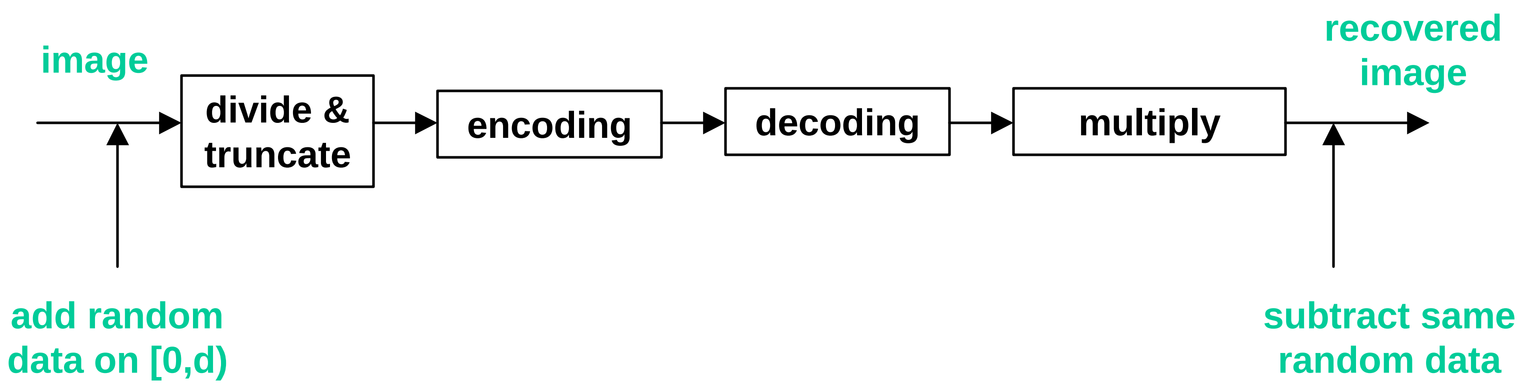
Where $d$ is the quantization factor

# Visual Properties

- Simple quantization




- Same + “error shaping”



# Quantized Image

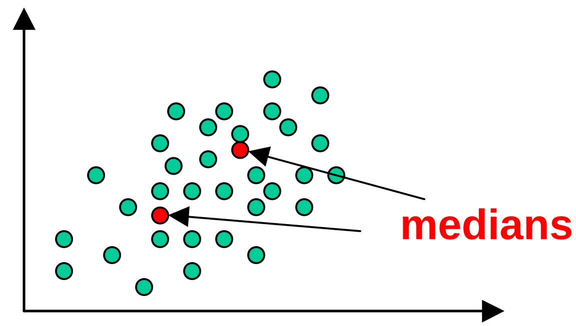
**Lena image after setting 4 least significant bits to zero (direct 2:1 compression)**

# Quantized Image

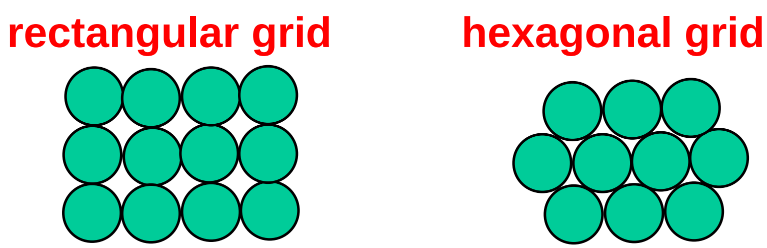
**Lena image after removing 4 least significant bits (2:1) + dithered quantization**

# Vector Quantization

- Exploit clustered data



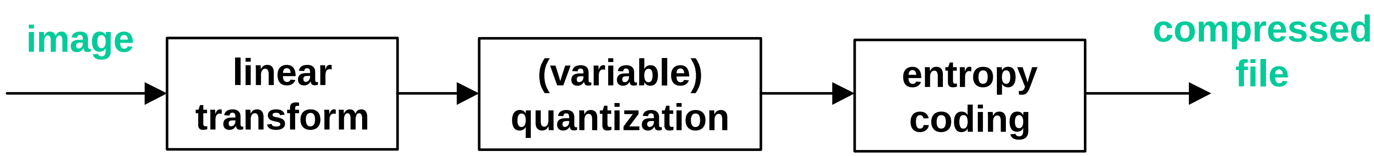

- More efficient “sphere packing”




- *Real* advantage for image compression unproved

# Transform Coding

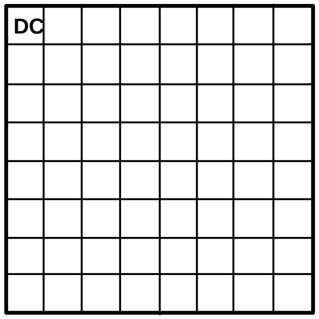
- New block




- Example: 8 x 8 discrete cosine transform (DCT)

$$T_{m,n} = \frac{1}{2} \sum_{i=0}^{7} \sum_{j=0}^{7} p_{i,j} \cos[(2m+1) i p / 16] \cos[(2n+1) j p / 16]$$

# Properties of Transform Coding




- Unitary transform: MSE conservation
- Energy compaction
  - Easier, more efficient entropy coding
- Good error shaping
  - Inverse transform greatly reduces quantization artifacts
- Clustering vector quantization less effective
- Small "sphere-packing" gains
- Block-based transform yields
  - Blocking artifacts
  - Potentially worse compression

# Blocking Artifacts

# Haar Transform

- **Definition for one-dimensional array**

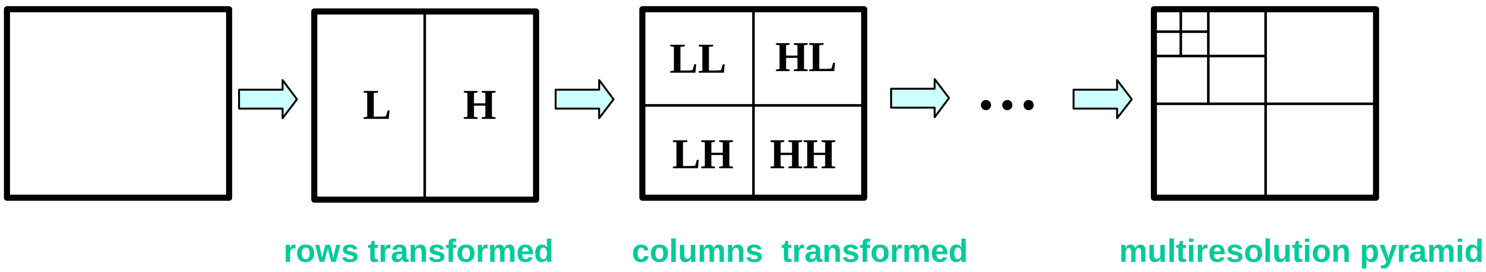
$$p_i^{k+1} = \frac{\sqrt{2}}{2}\left(p_{2i}^{k} + p_{2i+1}^{k}\right), \quad h_i^{k+1} = \frac{\sqrt{2}}{2}\left(p_{2i}^{k} - p_{2i+1}^{k}\right)$$

- **Recursive computation**

$$\begin{matrix} p_0^0 & p_1^0 & p_2^0 & p_3^0 & p_4^0 & p_5^0 & p_6^0 & p_7^0 \\ p_0^1 & p_1^1 & p_2^1 & p_3^1 & h_0^1 & h_1^1 & h_2^1 & h_3^1 \\ p_0^2 & p_1^2 & h_0^2 & h_1^2 & h_0^1 & h_1^1 & h_2^1 & h_3^1 \\ p_0^3 & h_0^3 & h_0^2 & h_1^2 & h_0^1 & h_1^1 & h_2^1 & h_3^1 \end{matrix}$$

# Multiresolution Transforms

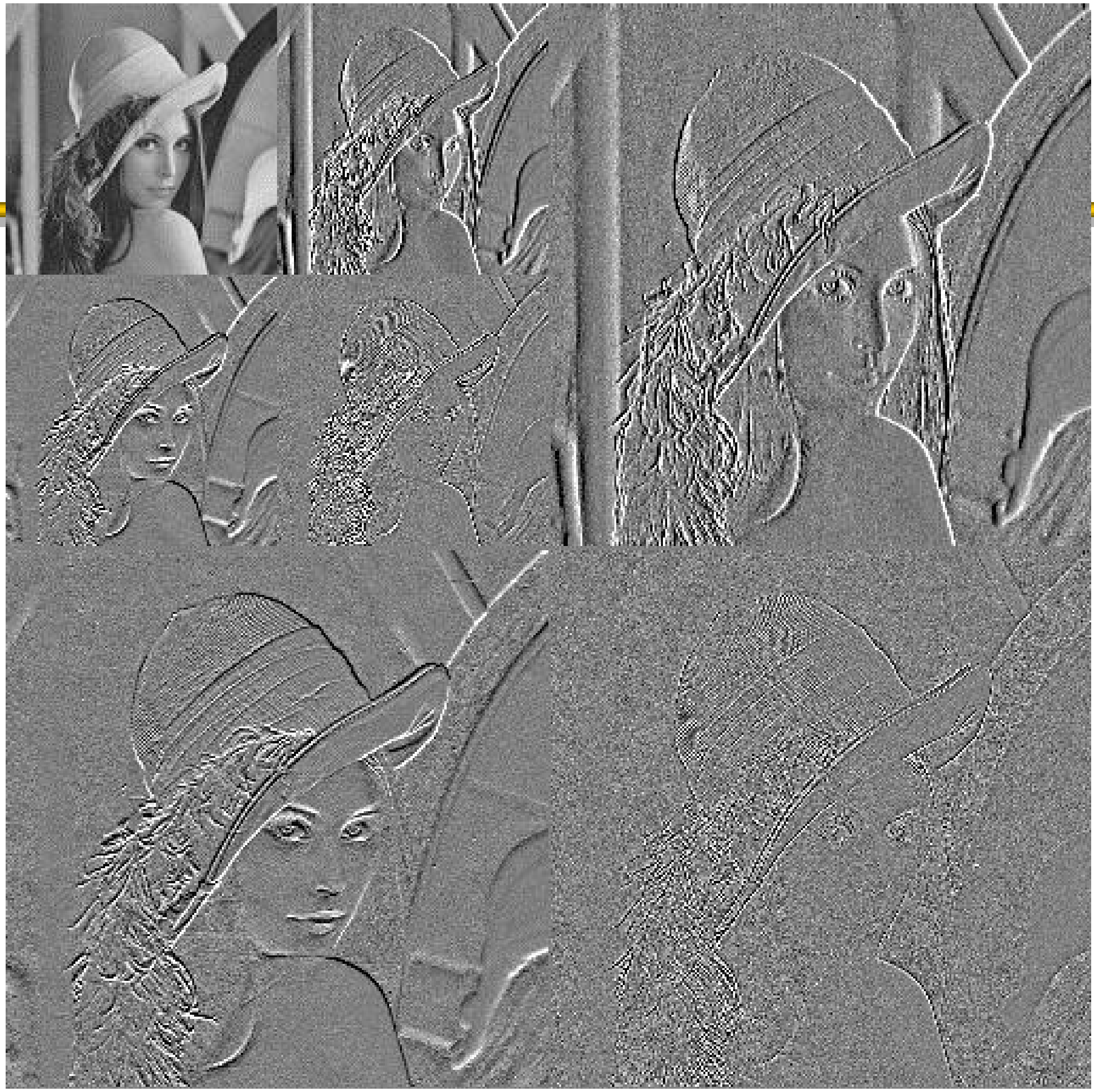
- Two-dimensional computation



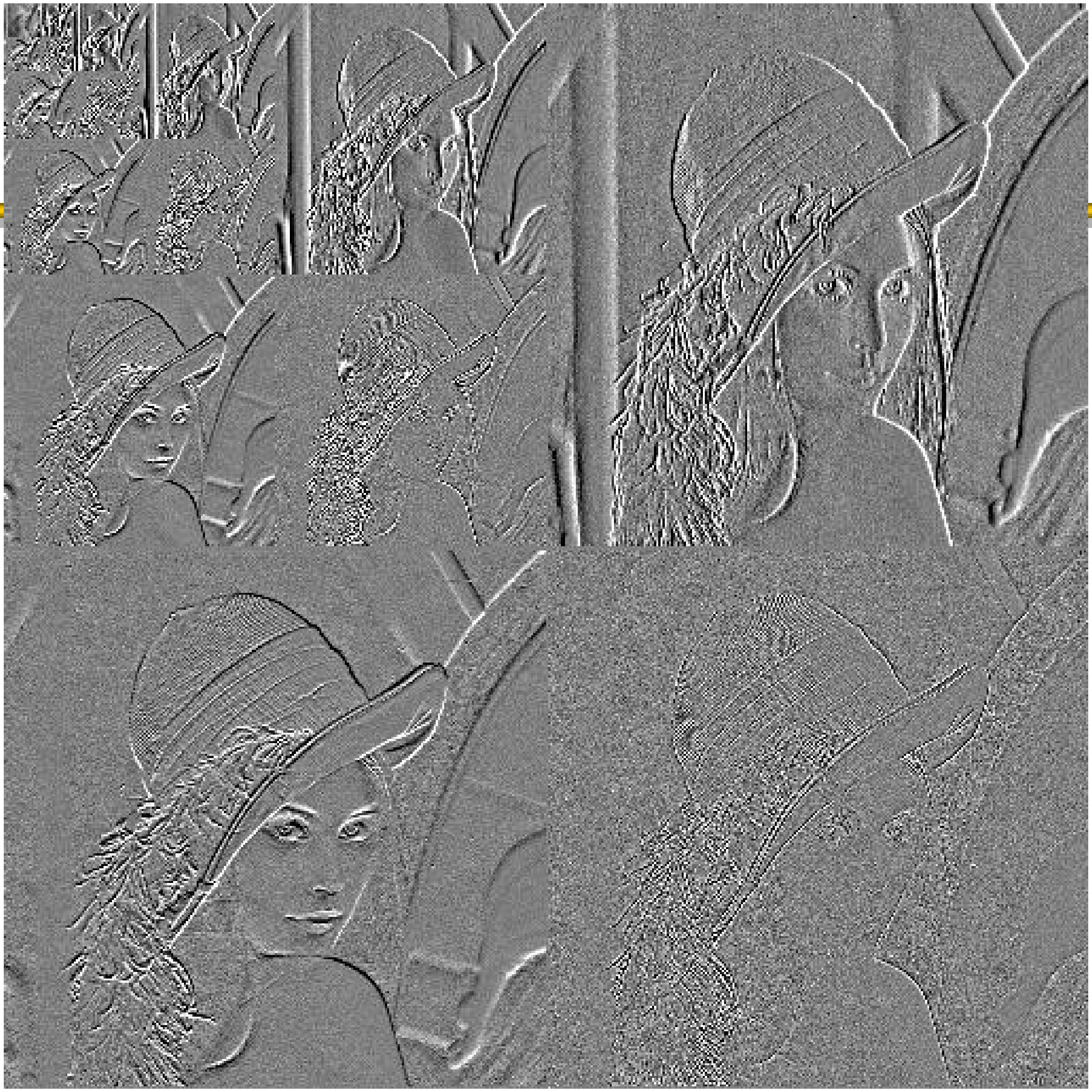

- Properties
  - Can exploit structures on several scales (large, small)
  - Hierarchical decomposition - progressive transmission
  - Good energy compaction
  - Preliminary classification for entropy coding (subbands)
  - Haar transform produces blocking artifacts

# 1st stage

## 2nd stage



## 5th stage



# Overlapping Kernels

- One formula for overlapping multiresolution transform

$$p_i^{k+1} = \frac{\sqrt{2}}{8}\left(-p_{2i-2}^{k} + 2\,p_{2i-1}^{k} + 6\,p_{2i}^{k} + 2\,p_{2i+1}^{k} - p_{2i+2}^{k}\right)$$

$$h_i^{k+1} = \frac{\sqrt{2}}{8}\left(-2\,p_{2i}^{k} + 4\,p_{2i+1}^{k} - 2\,p_{2i+2}^{k}\right)$$

- Inverse transform

$$p_{2i}^{k} = \frac{\sqrt{2}}{8}\left(-2\,h_{i-1}^{k+1} + 4\,p_i^{k+1} - 2\,h_i^{k+1}\right)$$

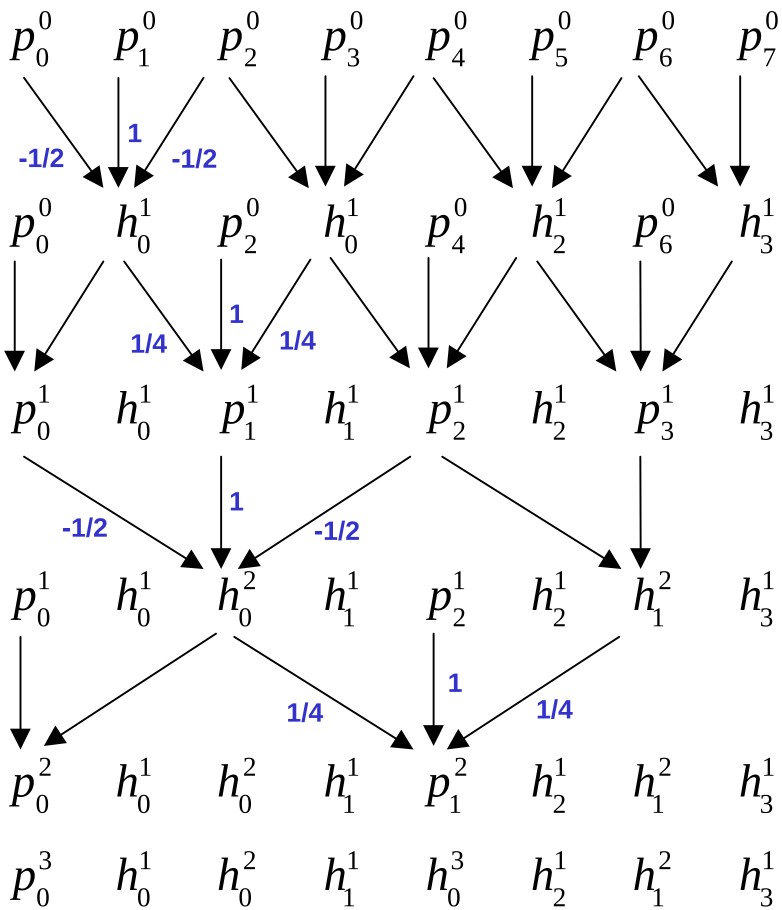
$$p_{2i+1}^{k} = \frac{\sqrt{2}}{8}\left(-h_{i-1}^{k+1} + 2\,p_i^{k+1} + 6\,h_i^{k+1} + 2\,p_{i+1}^{k+1} - h_{i+1}^{k+1}\right)$$

# Matrix Formulation

- **Example: 10 samples, cyclic convolution, 5/3 filters**

$$
\begin{bmatrix}
\mathbf{4} & -\mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{2} \\
\mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 & 0 & 0 & 0 & 0 & -\mathbf{1} \\
0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} \\
\mathbf{2} & -\mathbf{1} & 0 & 0 & 0 & 0 & 0 & -\mathbf{1} & \mathbf{2} & \mathbf{6}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 & 0 & 0 & 0 & 0 & -\mathbf{1} & \mathbf{2} \\
-\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} & -\mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{2} & \mathbf{4} & -\mathbf{2} & 0 \\
-\mathbf{1} & 0 & 0 & 0 & 0 & 0 & -\mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{2} \\
-\mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{2} & \mathbf{4}
\end{bmatrix}
= 32\mathbf{I}
$$

# The “Lifting” Technique




add neighbors, divide by 2, subtract

add neighbors, divide by 4, add

add neighbors, divide by 2, subtract

add neighbors, divide by 4, add

use bit reversal to reorder

# Properties of Wavelet Coefficients

- Residual correlation too small for any practical use
- Different distribution on different parts (subbands)
- Stationary assumption quite unrealistic
- Most coefficient are zero after quantization
- Distribution somehow replicated on resolution hierarchy
- Variable quantization needs to be addressed (bit allocation)

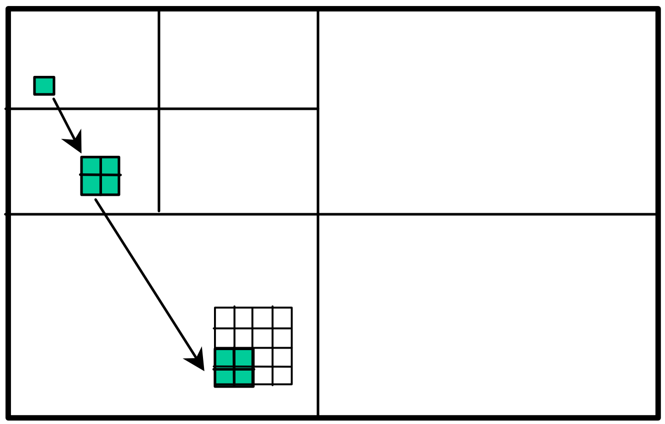
- What is the best coding method for wavelet coefficients?

# Zerotrees and Set Partitioning

- **Some efficient methods for coding wavelet coefficients**
  - **A.S. Lewis & g. Knowles (1991) - first to use, for efficient coding, trees defined on the multiresolution pyramid**
  - **J.M. Shapiro (1992) - EZW (embedded zerotrees of wavelets) - developed method of progressive refinement for fully embedded coding, used efficient entropy-coding**
  - **A. Said & W.A. Pearlman (1993) - SPIHT (set partitioning in hierarchical trees) - more efficient coding, generalization of set-partitioning, equivalence to sorting, lossless compression, *public domain demos***

# Significance Trees

- Sets of insignificant coefficients
  - All magnitudes smaller than a threshold
  - "Zerotrees" when sets are defined as trees
- Spatial orientation trees



# Basic Algorithm Ideas

- Use one bit to indicate if all wavelet coefficients in a tree are zero
  - If all zero, nothing else to do
  - If not
    - Subdivide tree in several (4) parts
    - Apply same test to new parts
- Repeat until all nonzero coefficients found
  - Apply simple entropy-coding to nonzero coefficients

# Set-partitioning Properties

- Very simple entropy coding (mostly partitioning data)
- No explicit bit allocation
- Only simple scalar (uniform) quantization used
- Low encoding complexity
- *Very* efficient decoding

- **Best compression when first published**
- **Efficient compression from high rates up to lossless recovery**

## Wavelets & SPIHT: compressed 50:1

**Baseline JPEG: compressed 45:1**

original

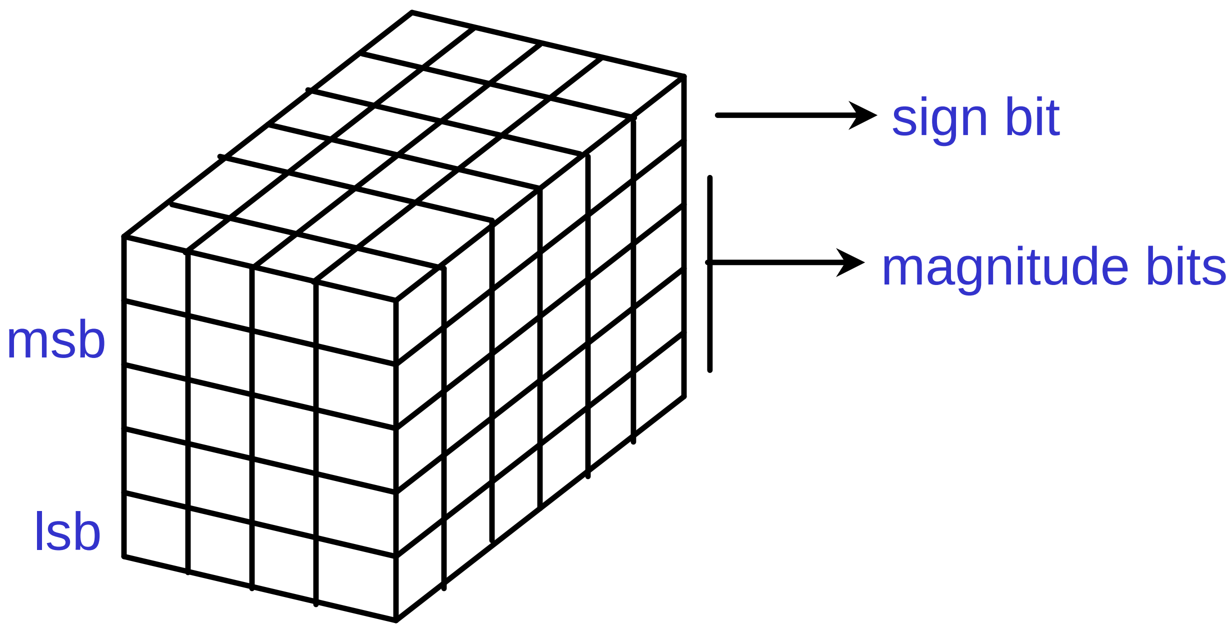
Wavelet & SPIHT: compressed 100:1

original


Wavelet & SPIHT: compressed 50:1

# Bit-plane Coding

- **Progressive coding of wavelet coefficients**



# Lessons Learned

- **Most important problem: efficient coding of the location of zero coefficients**
- **Vector quantization, optimal bit allocation hardly necessary**
- **Exploits spatial clustering of *magnitude* distribution**
- **Efficient entropy coding does not have to be complex**
- **Multiresolution properties can be effectively exploited**

# Application Issues - I

- Complexity of the transform
  - DCT
    - O(log(b)) operations per pixel, b = block size
    - Inverse transform frequently skipped or simplified
    - All block data on L1 cache
    - Efficient hardware implementations
  - Wavelet
    - O(t) operations per pixel, t = average kernel size
    - Inverse transform = smoothing, cannot be skipped
    - Lifting reduces the number of operations
    - Bandwidth may be more important than number of operations

# Application Issues - II

- Memory usage
  - DCT
    - Minimum b·b block
    - Commonly width·b
  - Wavelet
    - Simplest implementation: full image
  - "Rolling wavelet"
    - Minimum k·t (typical k = 8)
    - Commonly width·k·t
  - Fully embedded coding
    - Must keep full image buffered *or* compressed image buffered

# Application Issues - III

- Quantization & entropy-coding complexity
  - Basically independent of the image transforms
- Compression efficiency & visual appearance
  - Wavelets do yield best results
  - Poor visual quality due to obsession with MSE
- Versatility
  - Wavelets naturally support progressive resolution
  - Easy combination of lossy and lossless compression
  - Efficient methods for embedded coding
  - Region-of-interest modes supported

# Application Issues - IV

- Error resiliency
  - Very complicated problem
    - Protocols
    - Error propagation
    - Error detection and error correction
  - Hierarchical structure allows sorting data by importance, for unequal error protection
  - Simple entropy-coding allows identification of bits that do not lead to catastrophic error propagation
  - Overlapping kernels produce smooth error artifacts

# Conclusions

- Wavelet transform has several features required for effective image compression
  - Error shaping yield good visual quality
  - Efficient energy compaction
  - Can exploit image features in several scales
- Its structure allows
  - Easy implementation of progressive transmission and multiresolution
  - More efficient compression
  - Better error resiliency
- Not all application issues solved, but significant progress recently

# Wavelet & Image Compression Links

- http://www.cipr.rpi.edu/research/SPIHT/EE_Forum.pdf
- http://www.cipr.rpi.edu/research/SPIHT/
- http://www.cipr.rpi.edu/research/SPIHT/spiht8.html
- http://www.wavelet.org/wavelet/index.html
- http://www.code.ucsd.edu/~jkrogers/Papers/
- http://cm.bell-labs.com/who/wim/papers/papers.html#iciam95
- http://cm.bell-labs.com/who/wim/papers/papers.html
- http://www.mat.sbg.ac.at/~uhl/wav.html
- http://www.jpeg.org
- http://www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/top.html
- http://www.mathsoft.com/wavelets.html
- http://biron.usc.edu/~chrysafi/Publications.html