Tutorial on
Image Compression

Richard Baraniuk

Rice University
dsp.rice.edu

Agenda

- Image compression problem
- Transform coding (lossy)
- Approximation
  - linear, nonlinear
- DCT-based compression
  - JPEG
- Wavelet-based compression
  - EZW, SFQ, EQ, JPEG2000
- Open issues

Image Compression Problem

Images

- 2-D function f
- Idealized view

$f \in$ some function space defined over $[0,1] \times [0,1]$

- In practice

$$f \in \mathbb{R}^{N \times N}$$

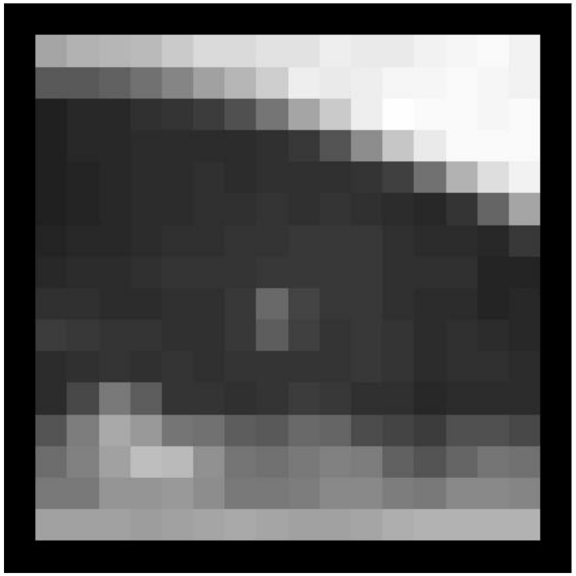
ie: an $N \times N$ matrix

Images

- 2-D function f

- Idealized view

$f \in$ some function space defined over $[0,1] \times [0,1]$



- In practice

$$f \in \mathbb{R}^{N \times N}$$

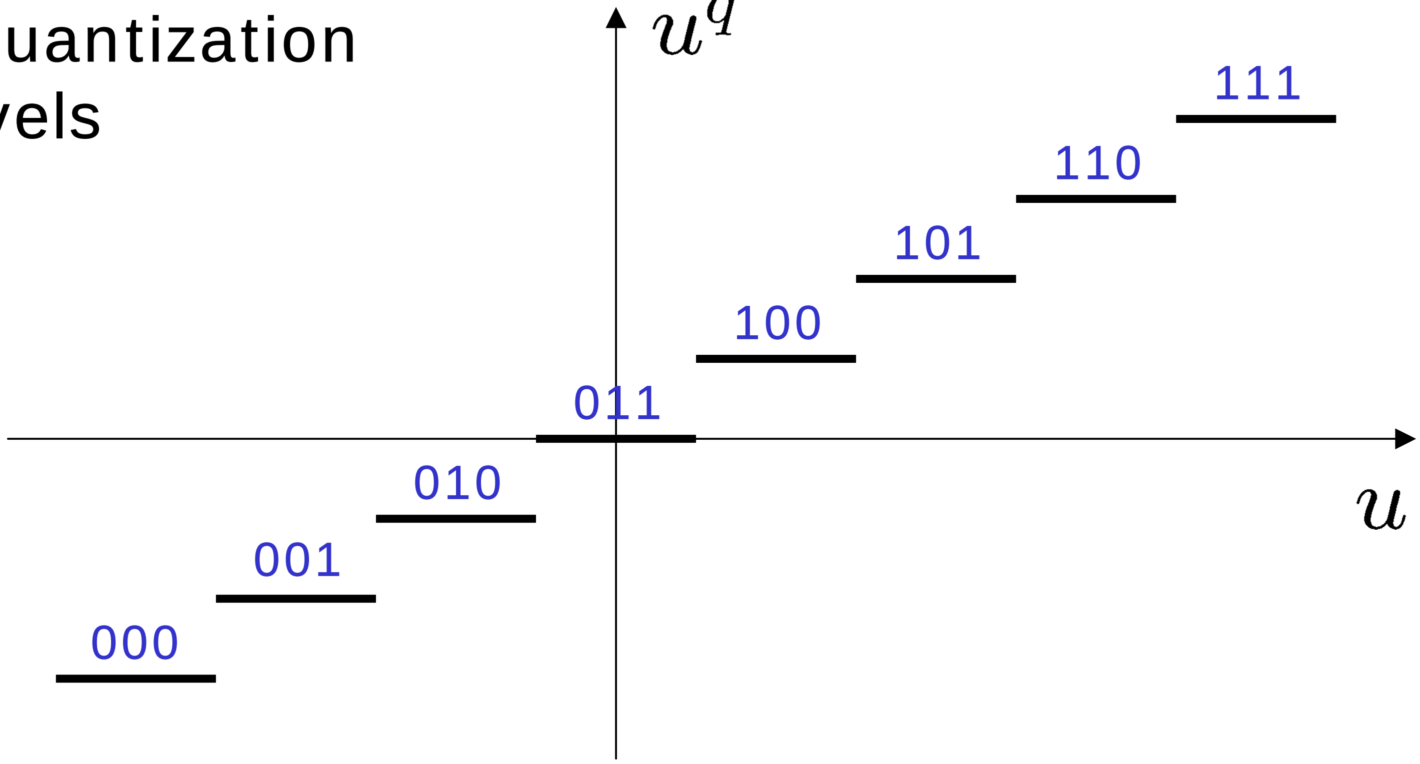
ie: an $N \times N$ matrix (pixel average)

Quantization

- Approximate each continuous-valued *pixel value* u with a discrete-valued variable u^q

- Ex: 3-bit quantization = 8-level approximation

- Human eye "blind" to 8-bit quantization = 256 levels



From Images to Bits

$\approx 0101110001010101\ldots$

The Need for Compression

- Modern digital camera

$$f \in \mathbb{R}^{N \times N}$$

$$N \times N \sim 5 \times 10^6$$

megapixels

$$(N \times N) \times 3 \text{ colors} \times 8 \text{ bits/color}$$

$$= 8 \times 10^8 \text{ bits}$$

$$= 10^8 \text{ bytes}$$

How Much Can We Compress?

[M. Vetterli +]

- $2^{(256x256x8)}$ possible images ~500,000 bits [David Field]
- Dennis Gabor, September 1959 (Editorial IRE)
  "... the 20 bits per second which, the psychologists assure us, the human eye is capable of taking in ..."
- Index all pictures ever taken in the history of mankind
  100 years x 10^{10} ~44 bits
- Search the Web
  google.com: 5-50 billion images online ~33-36 bits
- JPEG on Mona Lisa ~200,000 bits
- JPEG2000 takes a few less, thanks to wavelets ...

Lossy Image Compression

- Given image f
  approximate using R bits

$$\widehat{f}_R$$

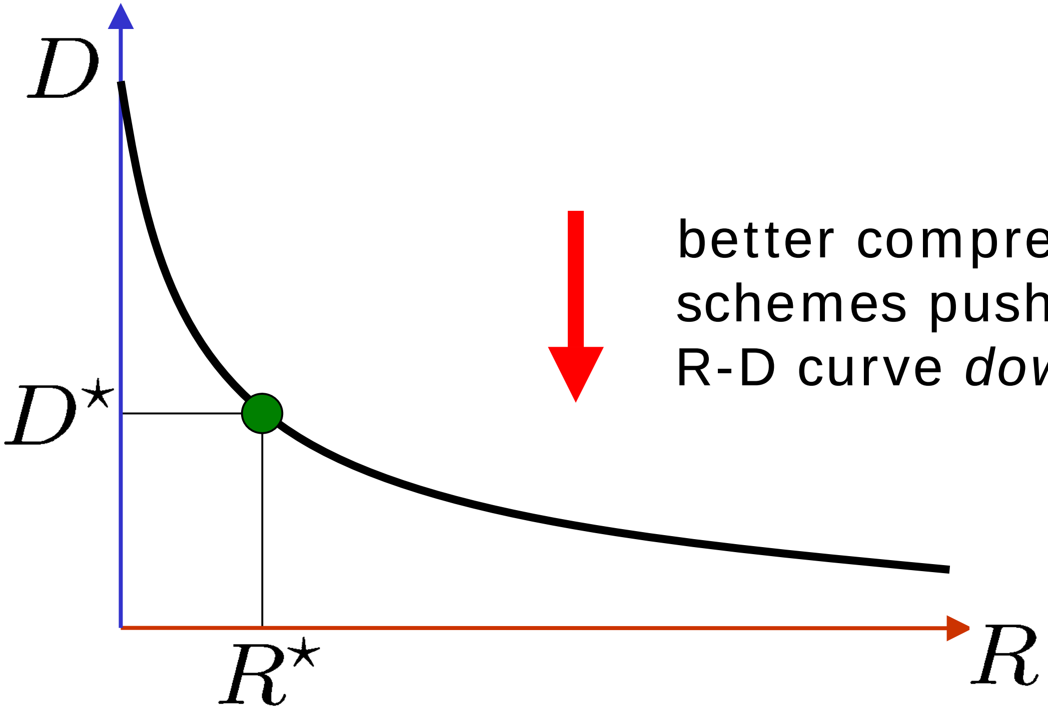
- Error incurred = distortion

Example: squared error

$$D = \|f - \widehat{f}_R\|_2$$

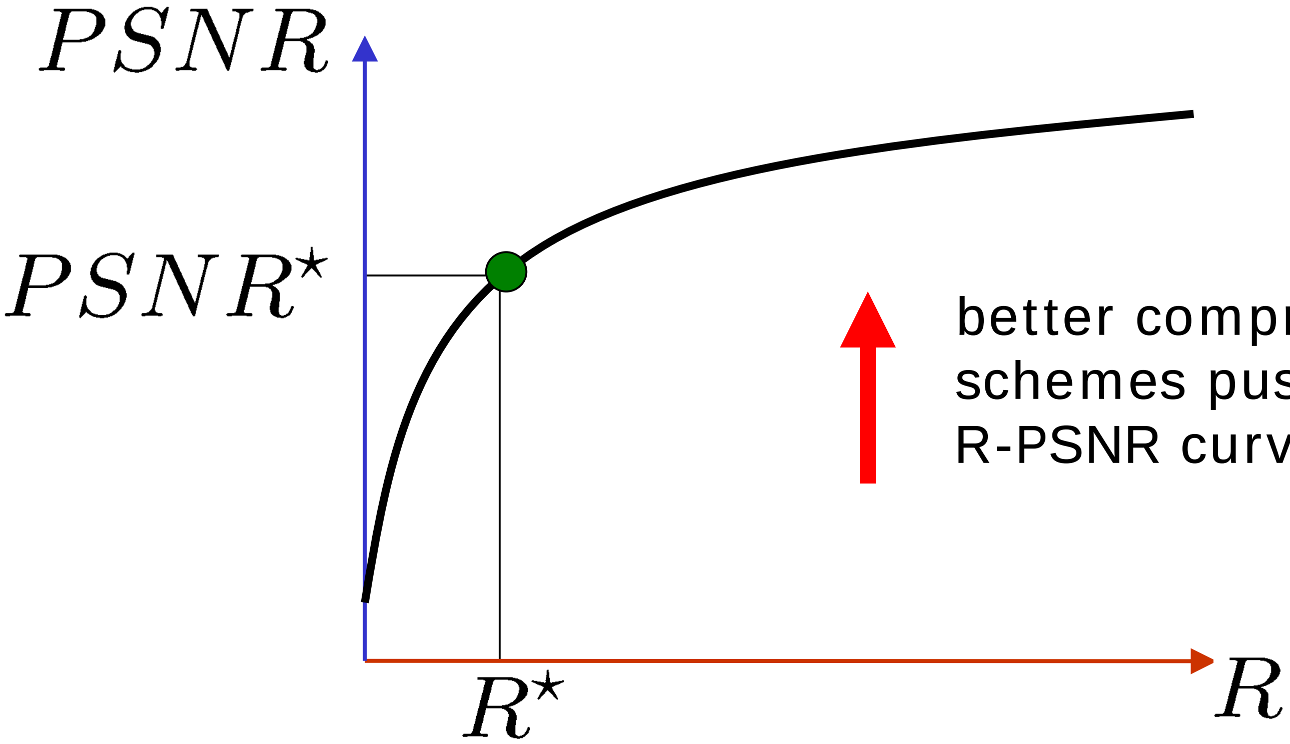
$$PSNR = 20\log_{10}\left(\frac{\max f}{\|(f - \widehat{f}_R)/N\|_2}\right)$$

Rate-Distortion Analysis



better compression schemes push the R-D curve *down*

Rate-Distortion Analysis



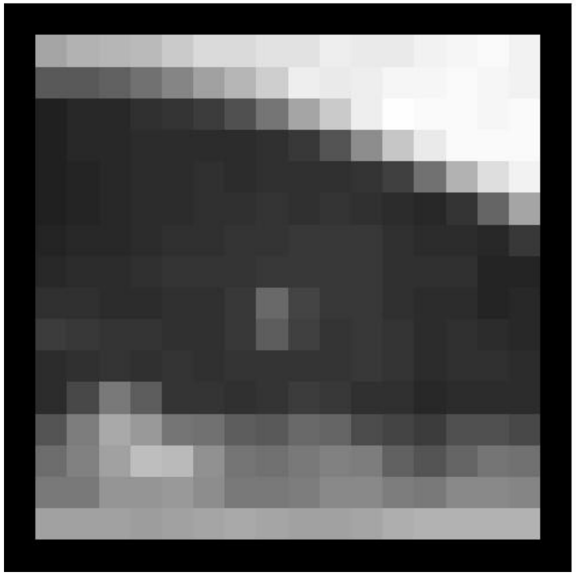
better compression schemes push the R-PSNR curve *up*

Lossy
Transform Coding

Image Compression

- Space-domain coding techniques perform poorly

- Why? smoothness
  $\Rightarrow$ strong correlations
  $\Rightarrow$ redundancies
  $\Rightarrow$ too many bits



$\rightarrow$ 0101110001 . . .

Transform Coding

- Quantize coefficients $\{a_k\}$ of an image expansion

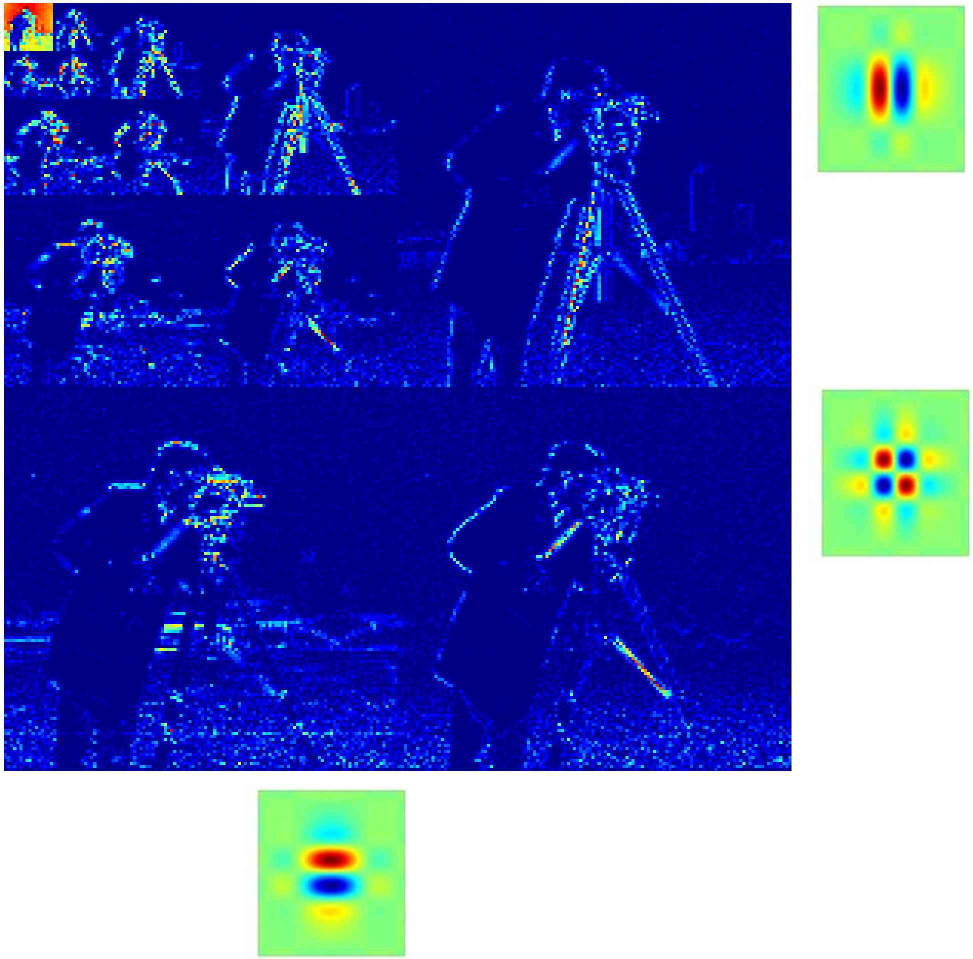
$$f = \sum_k a_k \, \mathbf{b}_k$$

coefficients basis, frame

quantize *to* R *total bits*

$$\widehat{f}_R = \sum_k a_k^q \, \mathbf{b}_k$$

Wavelet Transform

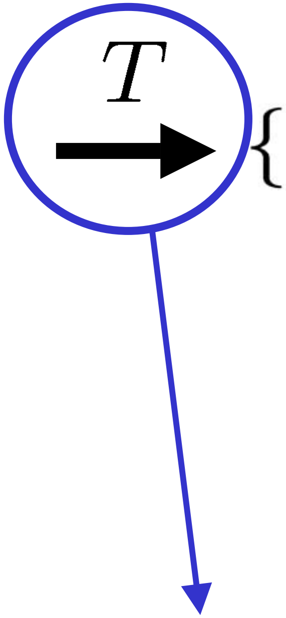


- Standard 2-D tensor product *wavelet transform*

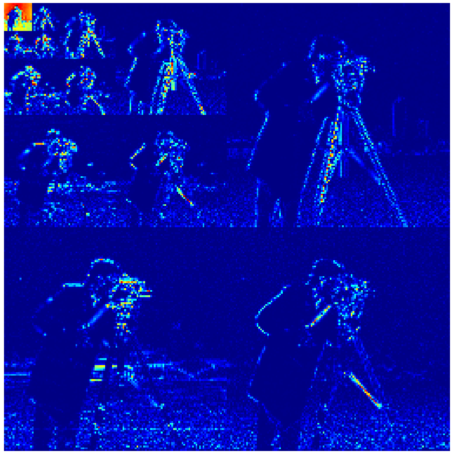
$$f = \sum_k a_k \mathbf{b}_k$$

Transform Coding

f



$\xrightarrow{T} \{a_k\}$



$= \{12.35, -9.11, 0.1, -0.03, \ldots\}$

Transform

– *sparse* set of coefficients
(many ≈ 0)

Transform Coding

f

$$\xrightarrow{T} \{a_k\} \xrightarrow{Q} \{a_k^q\}$$

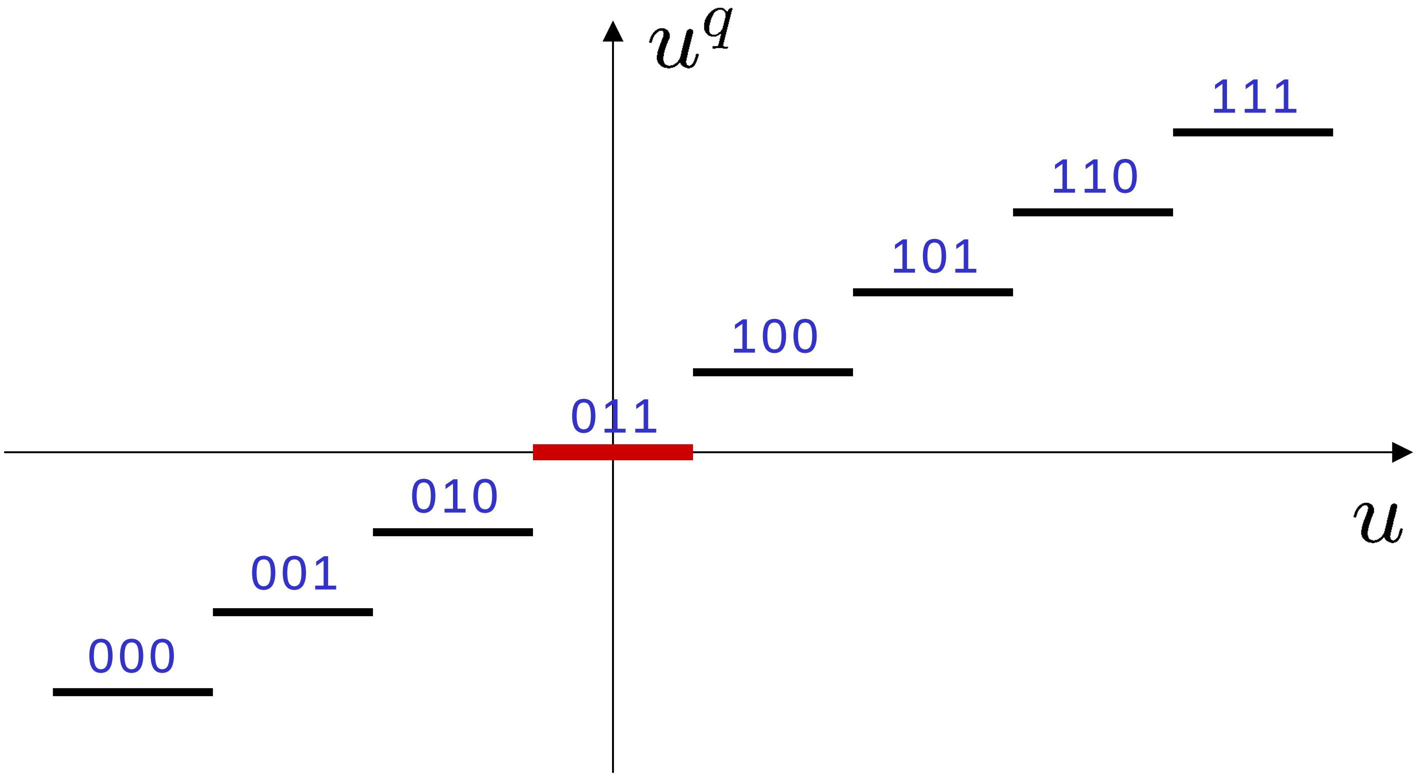
$$= \{011, 11, 00, 00, \ldots\}$$

Quantize

- approximate real-valued coefficients using bits
- sets small coefficients = 0

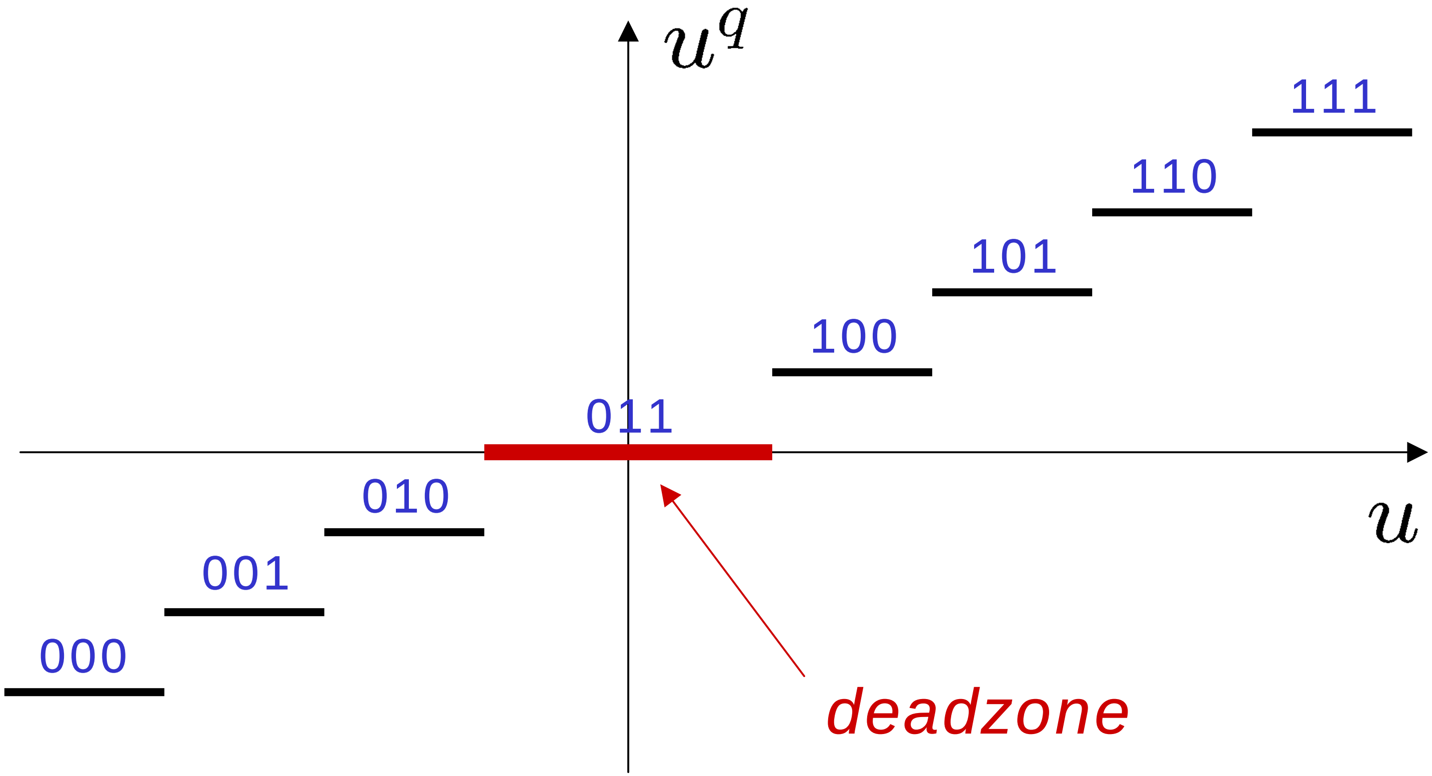
Quantization and Thresholding

- Quantization thresholds *small* coefficients to *zero*



Quantization and Thresholding

- Quantization thresholds *small* coefficients to *zero*



Transform Coding

f

$$\xrightarrow{T} \{a_k\} \xrightarrow{Q} \{a_k^q\}$$

$$= \{011, 11, 00, 00, \dots\}$$

Quantize

- approximate real-valued coefficients using bits
- sets small coefficients = 0

Transform Coding

f

$$f \xrightarrow{T} \{a_k\} \xrightarrow{Q} \{a_k^q\} \xrightarrow{C} R \text{ bits}$$

$$= \{1, 0, 01, \ldots\}$$

Entropy code

– reduce excess redundancy in the bitstream

Ex: Huffman coding, arithmetic coding, gzip, ...

Transform Coding/Decoding

f

$$\xrightarrow{T} \{{\color{red}a_k}\} \xrightarrow{Q} \{{\color{red}a_k^q}\} \xrightarrow{C} \quad R$$

bits

$\widehat{f}_R$

$$\xleftarrow{T^{-1}} \{{\color{red}a_k}\} \xleftarrow{Q^{-1}} \{{\color{red}a_k^q}\} \xleftarrow{C^{-1}} \quad R$$

bits

Sparse Approximation

Computational Harmonic Analysis

- Representation

$$f = \sum_k \underset{\substack{\uparrow \\ \text{coefficients}}}{a_k} \; \underset{\substack{\uparrow \\ \text{basis, frame}}}{\mathbf{b}_k}$$

- **Analysis** study f through *structure* of $\{a_k\}$
  $\{\mathbf{b}_k\}$ should *extract features* of interest

- **Approximation** $\widehat{f}_N$ uses just a few terms N
  exploit *sparsity* of $\{a_k\}$

Wavelet Transform Sparsity

$$f = \sum_k a_k \mathbf{b}_k$$

- Many $a_k \approx 0$

(blue)

Nonlinear Approximation

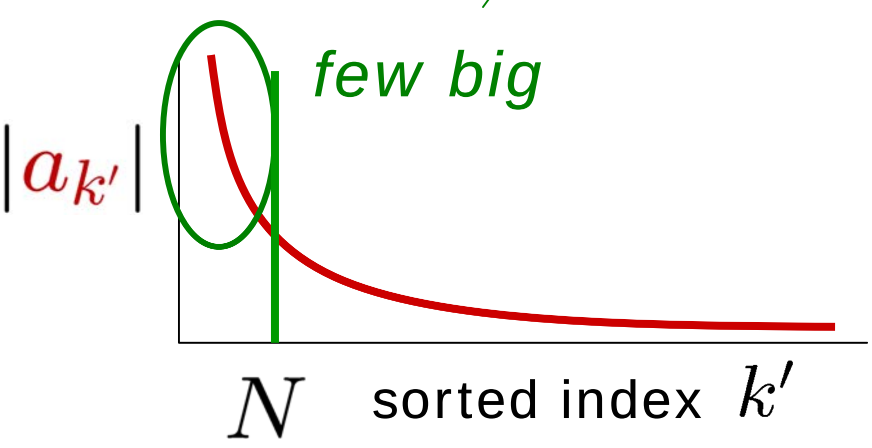
$$f \;=\; \sum_k a_k \, \mathbf{b}_k$$

- N-*term approximation*:

use *largest* a_k *independently*

$$\widehat{f}_N \;:=\; \sum_{k'=1}^{N} a_{k'} \, \mathbf{b}_{k'}$$

- Greedy / *thresholding*

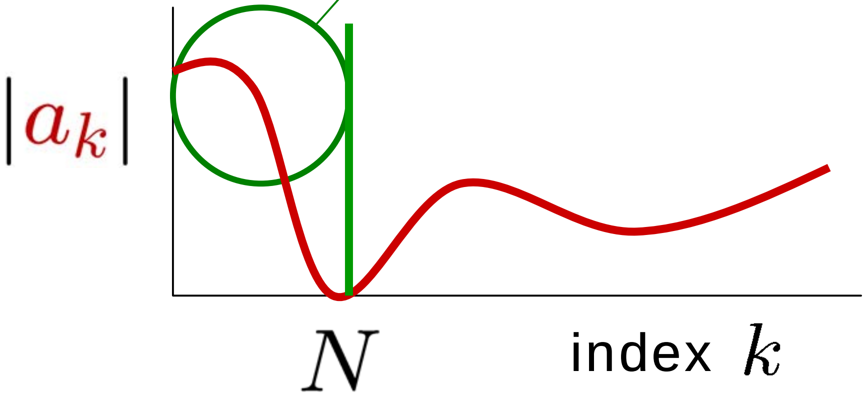


Linear Approximation

$$f \;=\; \sum_k a_k \, \mathbf{b}_k$$

- N-*term approximation*: use "*first*" a_k

$$\widetilde{f}_N \;:=\; \sum_{k=1}^{N} a_k \, \mathbf{b}_k$$



Error Approximation Rates

$$f \;=\; \sum_k a_k \, \mathbf{b}_k$$

$$\widehat{f}_N \;=\; \sum_{k'=1}^{N} a_{k'} \, \mathbf{b}_{k'}$$

$$\|f - \widehat{f}_N\|_2^2 \;<\; C\, N^{-\alpha} \qquad \text{as } N \longrightarrow \infty$$

- Optimize asymptotic *error decay rate* α
- Nonlinear approximation works better than linear

Compression is Approximation

- Lossy compression of an image creates an approximation

$$f = \sum_k a_k \, \mathbf{b}_k$$

coefficients basis, frame

quantize *to R total bits*

$$\widehat{f}_R = \sum_k a_k^q \, \mathbf{b}_k$$

NL Approximation is *not* Compression

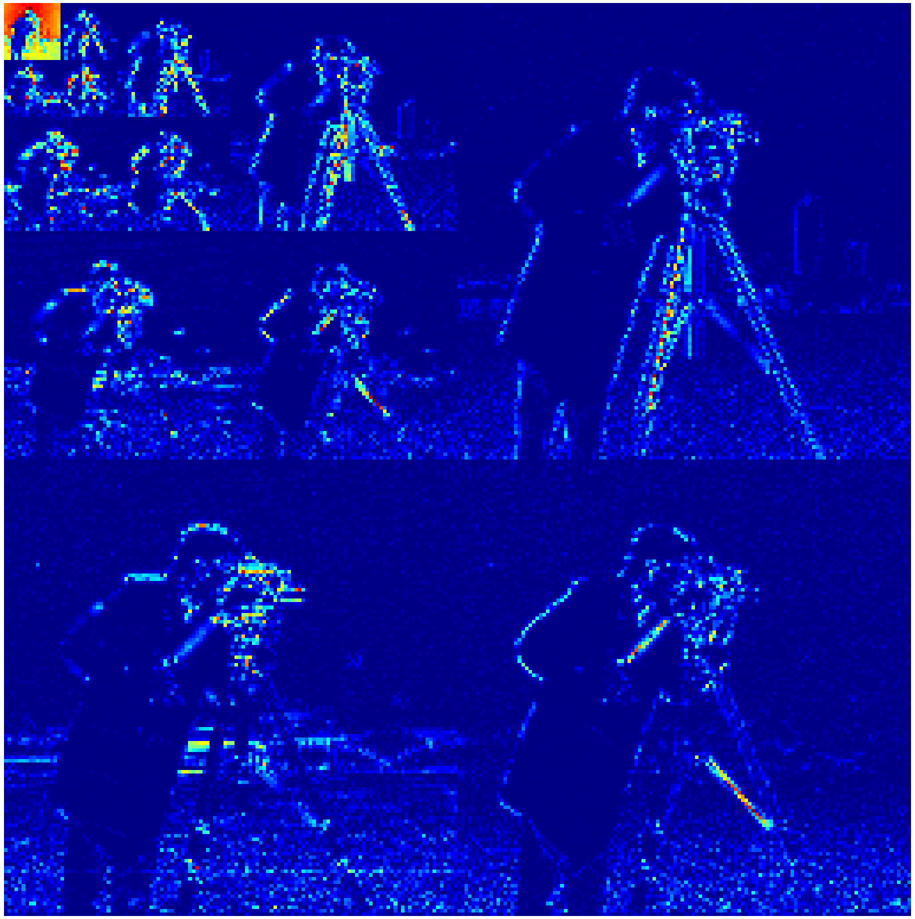
- Nonlinear approximation chooses coefficients but does not worry about their *locations*

$$\begin{tikzcd} f = \sum_k a_k \, \mathbf{b}_k \arrow[d, "\textit{threshold}"] \\ \widehat{f}_N = \sum_{k'=1}^{N} a_{k'} \, \mathbf{b}_{k'} \end{tikzcd}$$

Location, Location, Location

- Nonlinear approximation selects N largest a_k to minimize error (easy – threshold)

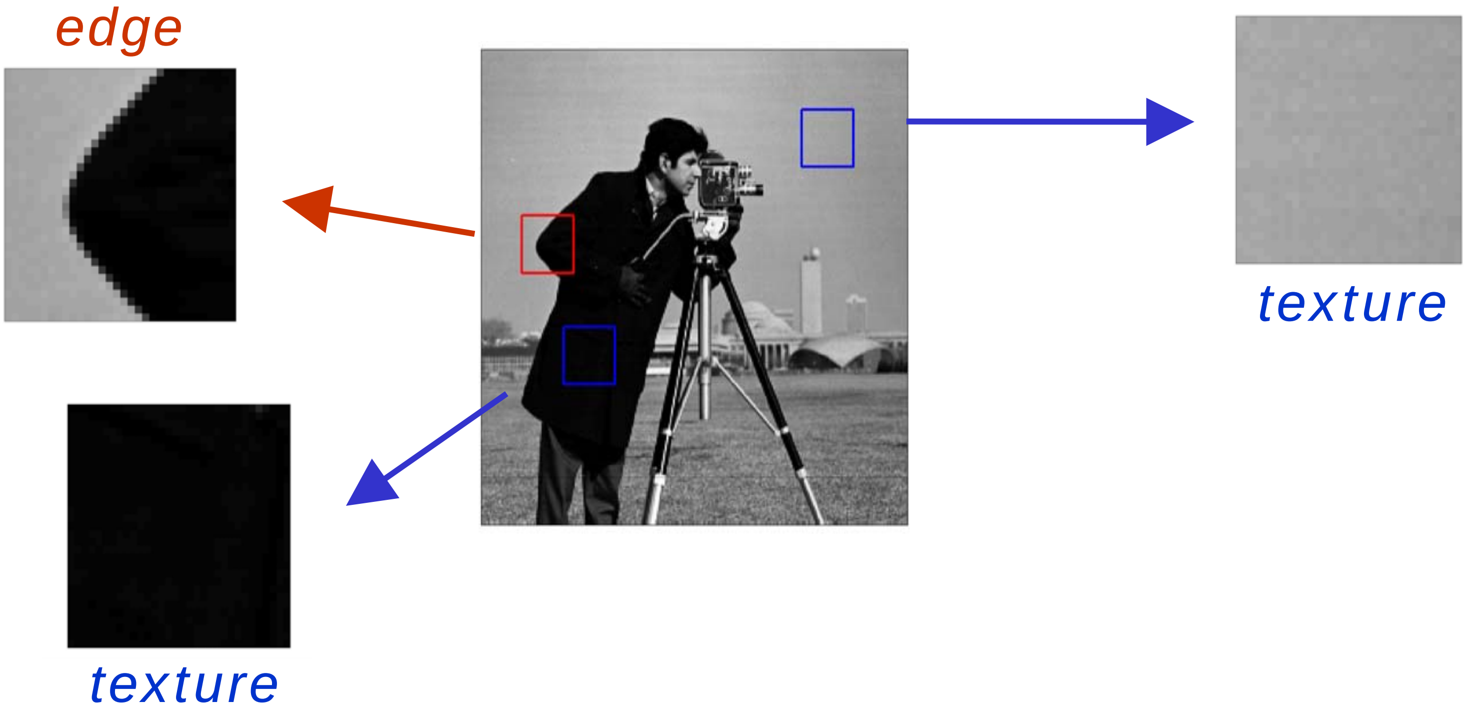
- Compression algorithm must encode *both* a set of a_k *and* their locations (harder)



Local Fourier Compression
JPEG

JPEG Motivation

- Image model: images are *piecewise smooth*
- Transform: *Fourier* representation sparse for smooth signals



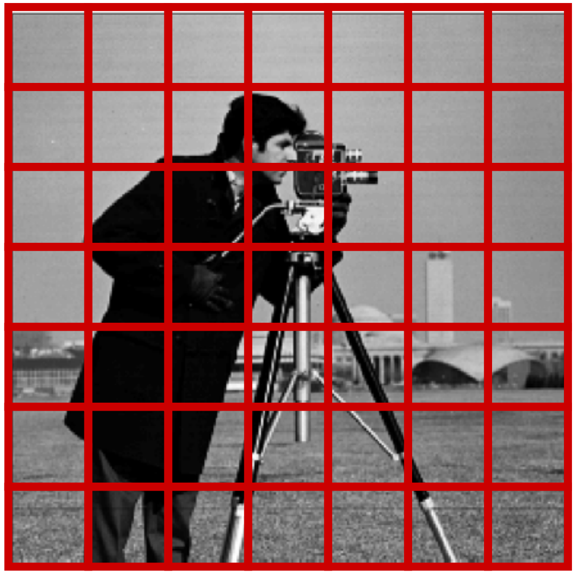
edge

texture

texture

JPEG Motivation

- Image model: images are *piecewise smooth*
- Transform: *Fourier* representation sparse for smooth signals
- Deal with edges: *local Fourier* representation (DCT on 8x8 blocks)

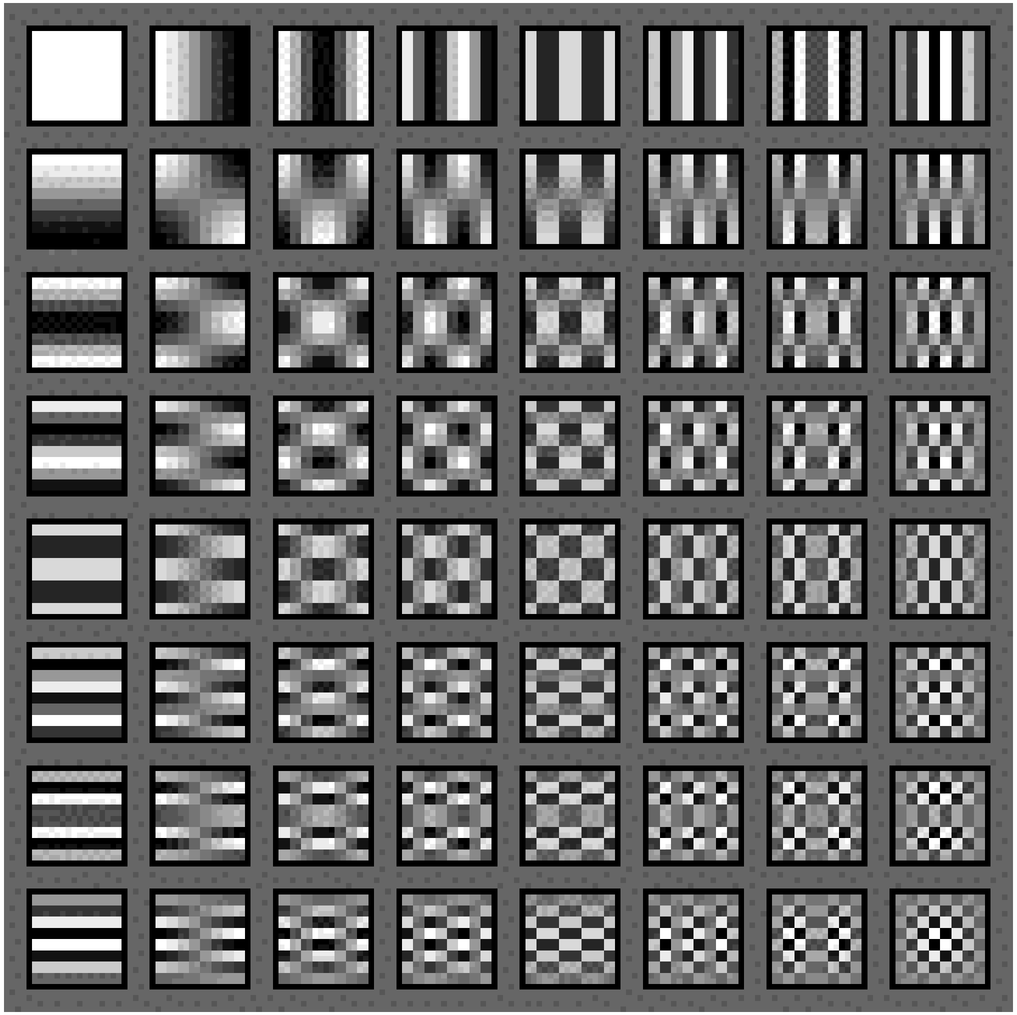


JPEG and DCT

- Local DCT
  (Gabor transform with square window or wavelet packets)
- Divide image into 8x8 blocks
- Take Discrete Cosine Transform (DCT) of each block

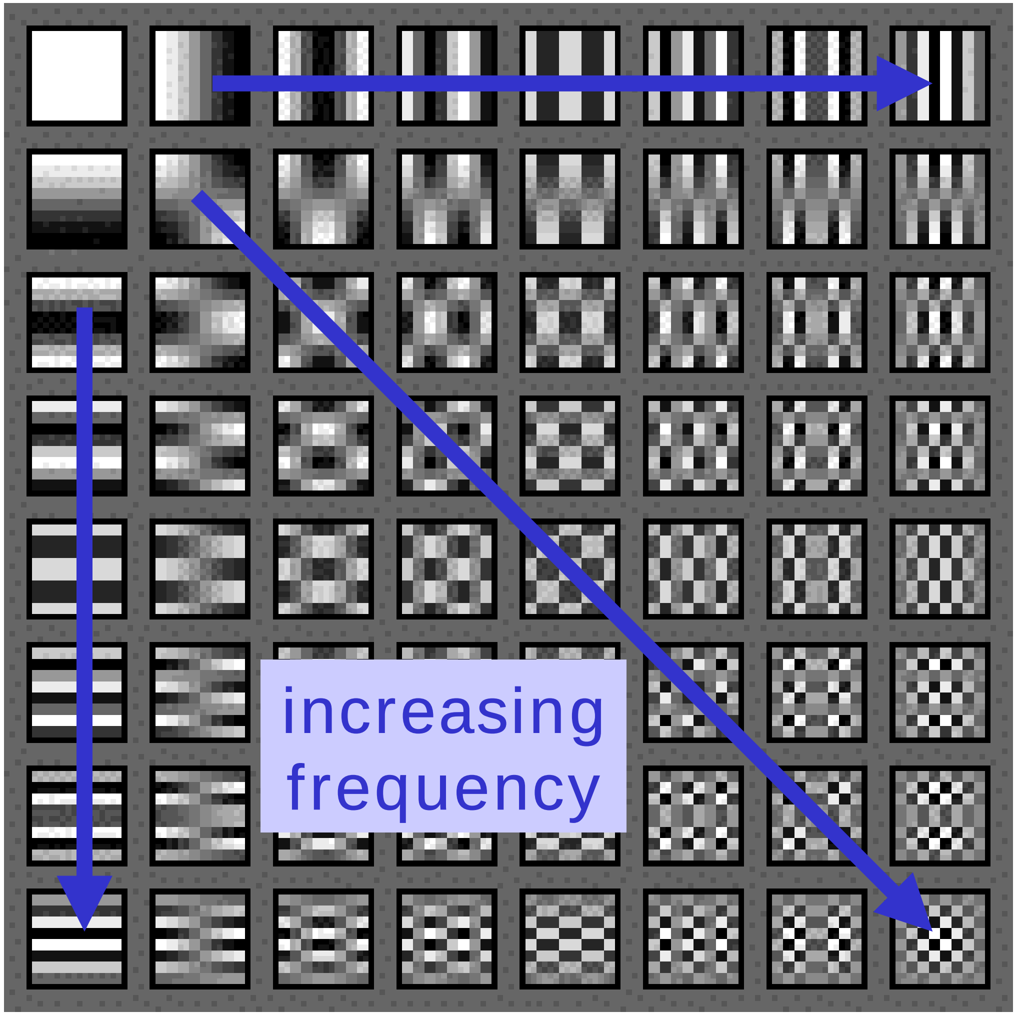
Discrete Cosine Transform (DCT)

- 8x8 block 🟥
- Project onto 64 different basis functions (tensor products of 1-D DCT)
- Real valued
- Orthobasis



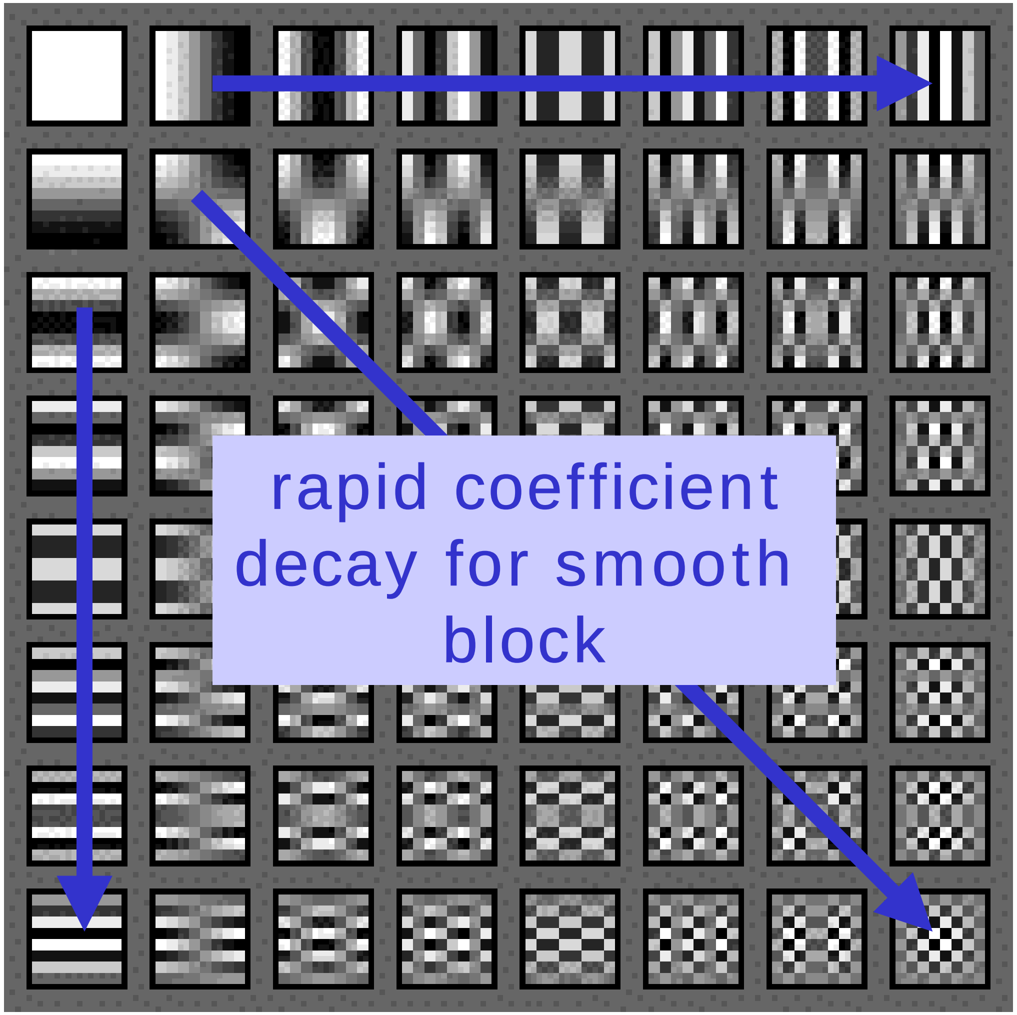
Discrete Cosine Transform (DCT)

- 8x8 block 🟥
- Project onto 64 different basis functions (tensor products of 1-D DCT)
- Real valued
- Orthobasis



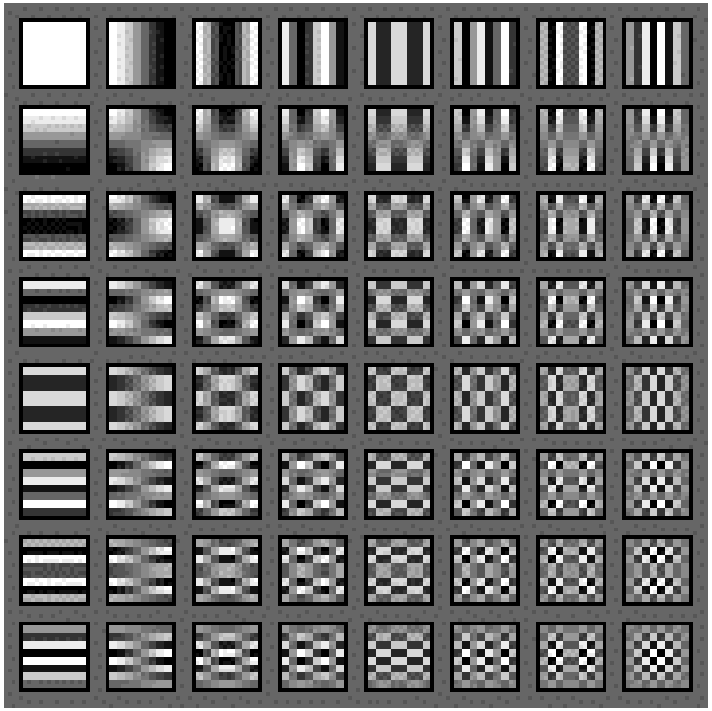
Discrete Cosine Transform (DCT)

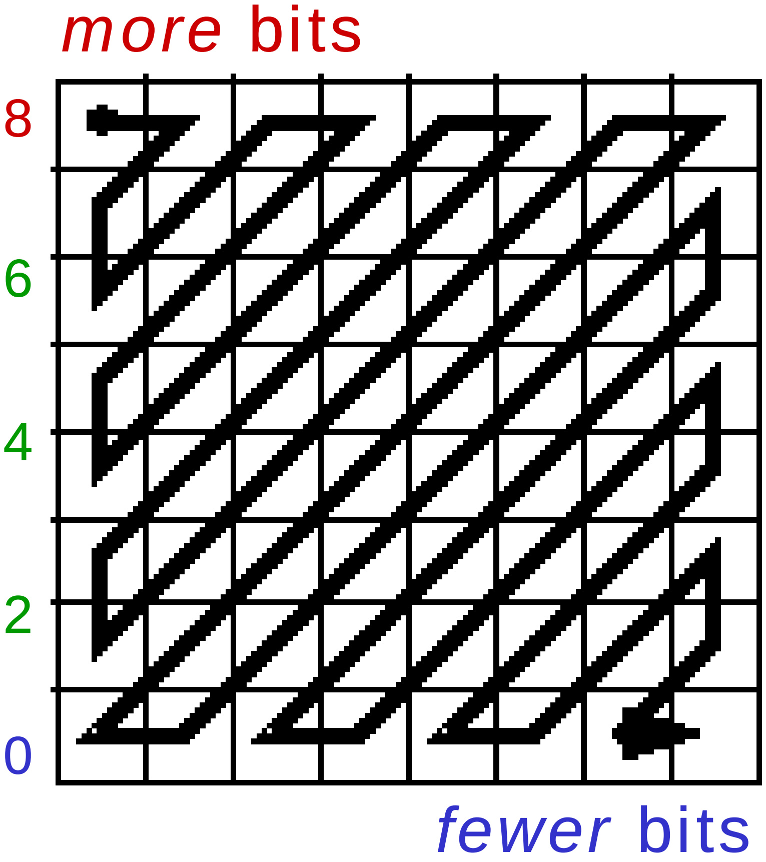
- 8x8 block
- Project onto 64 different basis functions (tensor products of 1-D DCT)
- Real valued
- Orthobasis



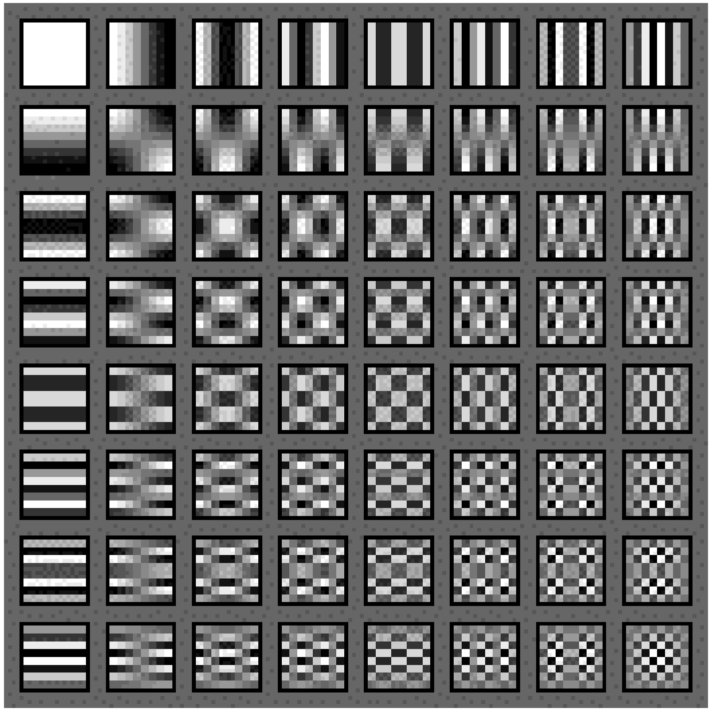
rapid coefficient decay for smooth block

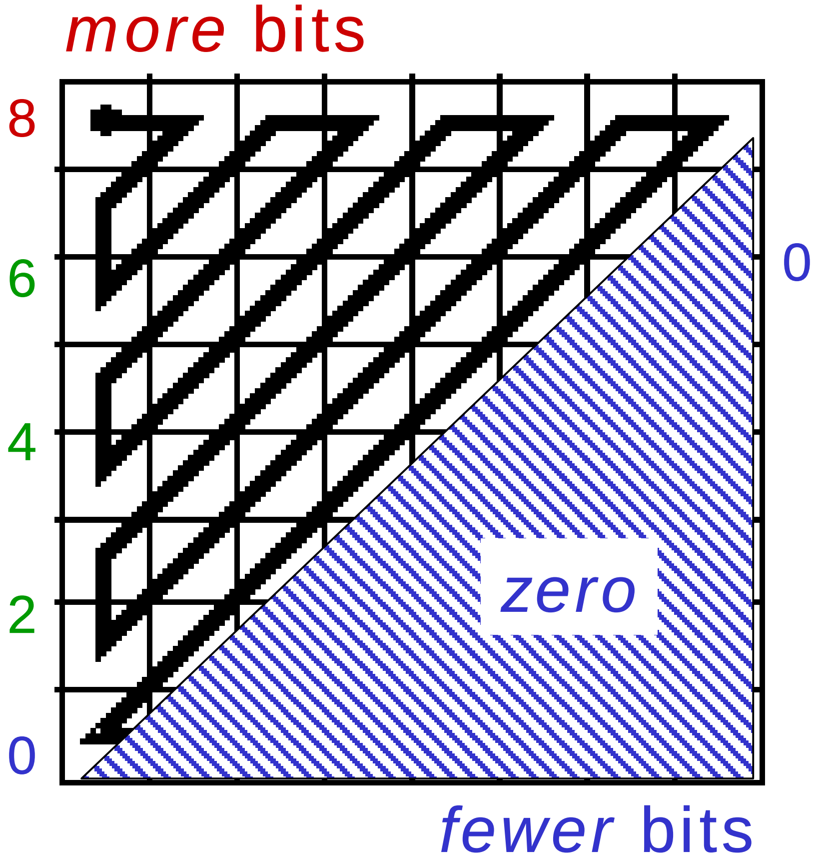
JPEG Quantization





JPEG Quantization





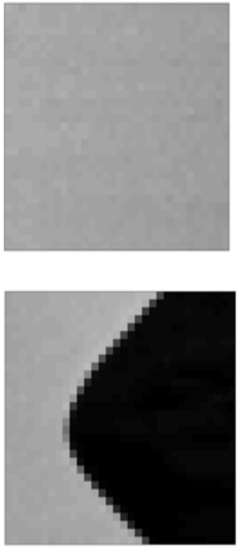
- *Quasi-linear approximation* in each block (fixed scheme)

JPEG Compression

256x256 pixels, 12,500 total bits, 0.19 bits/pixel

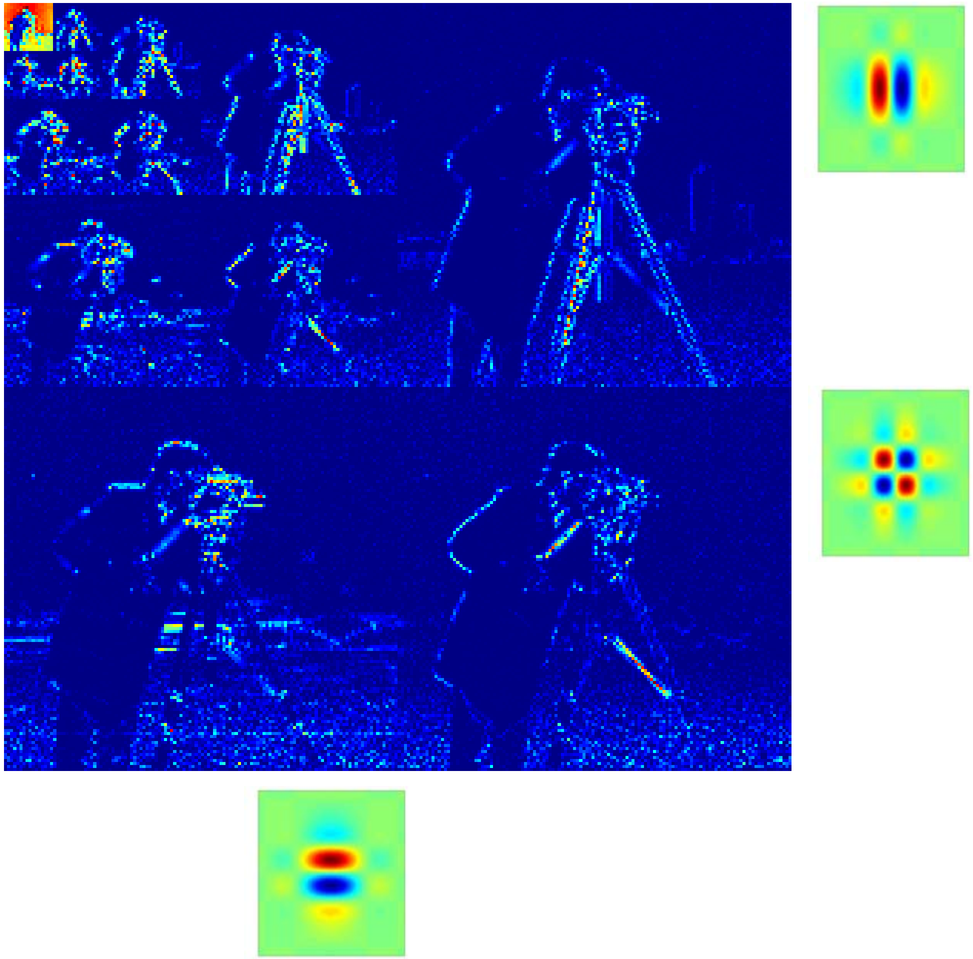
JPEG Compression

- Worldwide coding standard
- Problems
  - – local Fourier representation not sparse for edges so poor approximation at low rates
  - – blocking artifacts (discontinuities between 8x8 blocks)



Wavelet Compression

Enter Wavelets...



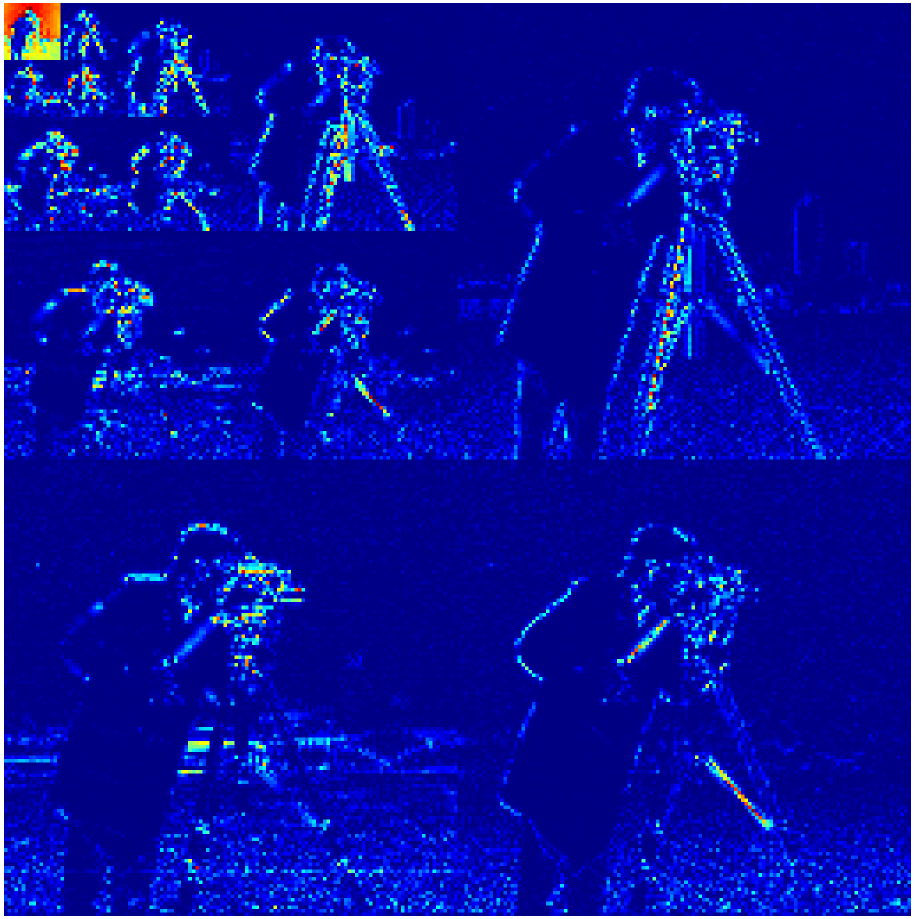
- Standard 2-D tensor product *wavelet transform*

$$f = \sum_k a_k \mathbf{b}_k$$

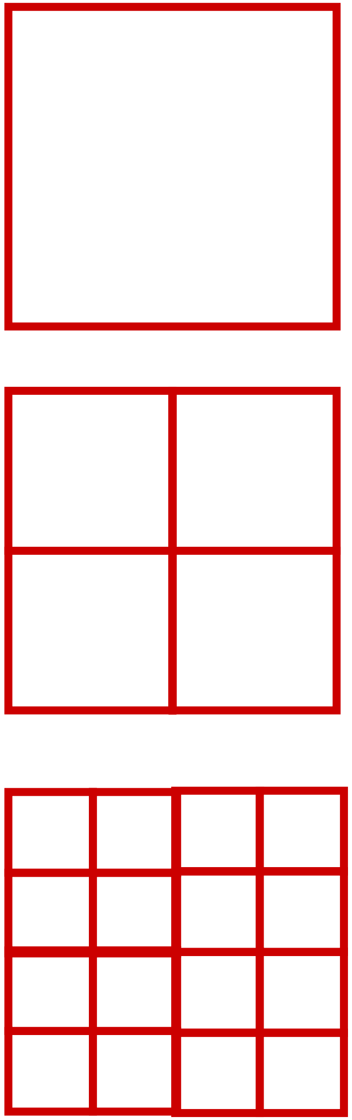
Location, Location, Location

- Nonlinear approximation selects N largest a_k to minimize error (easy – threshold)

- Compression algorithm must encode *both* a set of a_k *and* their ***locations*** (harder)

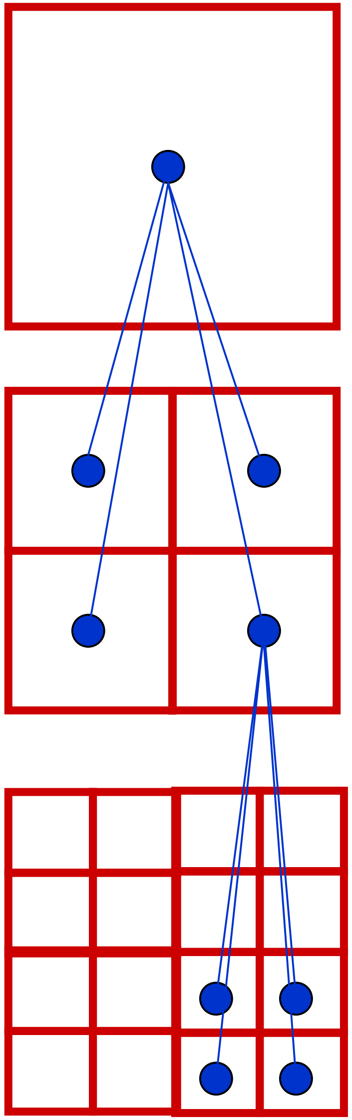


2-D Dyadic Partition



- *Multiscale* analysis
- Zoom in by factor of 2 each scale

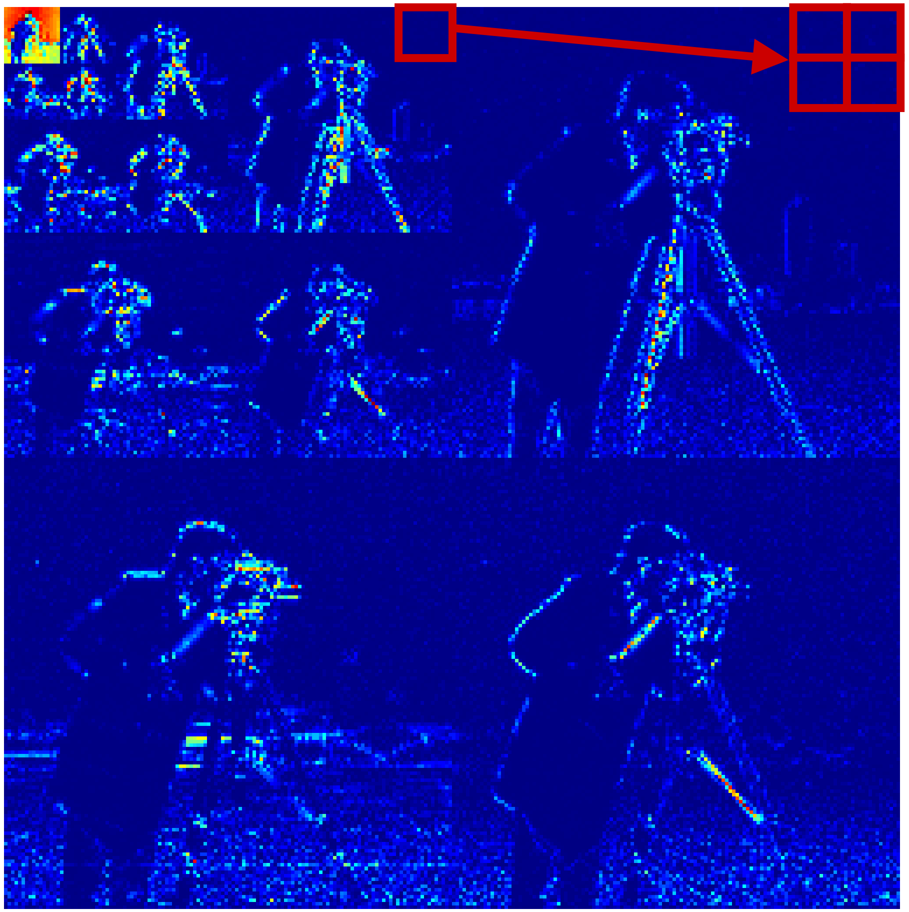
2-D Dyadic Partition = *Quadtree*



- *Multiscale* analysis
- Zoom in by factor of 2 each scale
- Each *parent* node has *4 children* at next finer scale

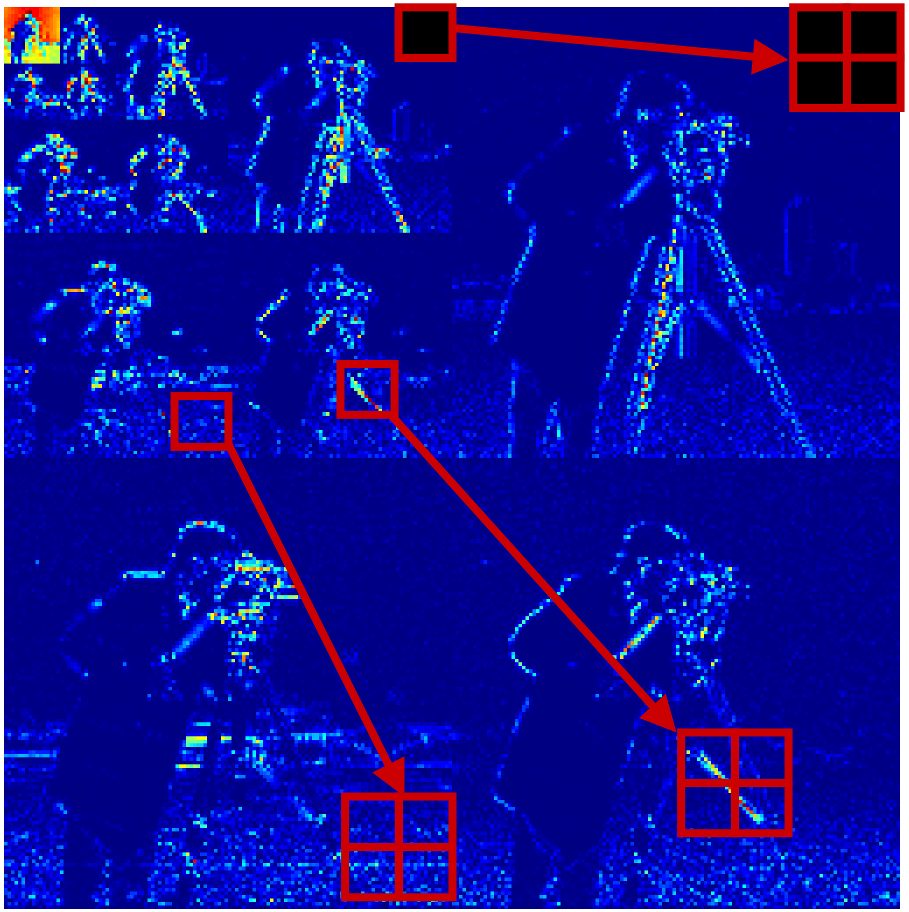
Wavelet Quadtrees

- Wavelet coefficients structured on *quadtree*
  - each *parent* has *4 children* at next finer scale



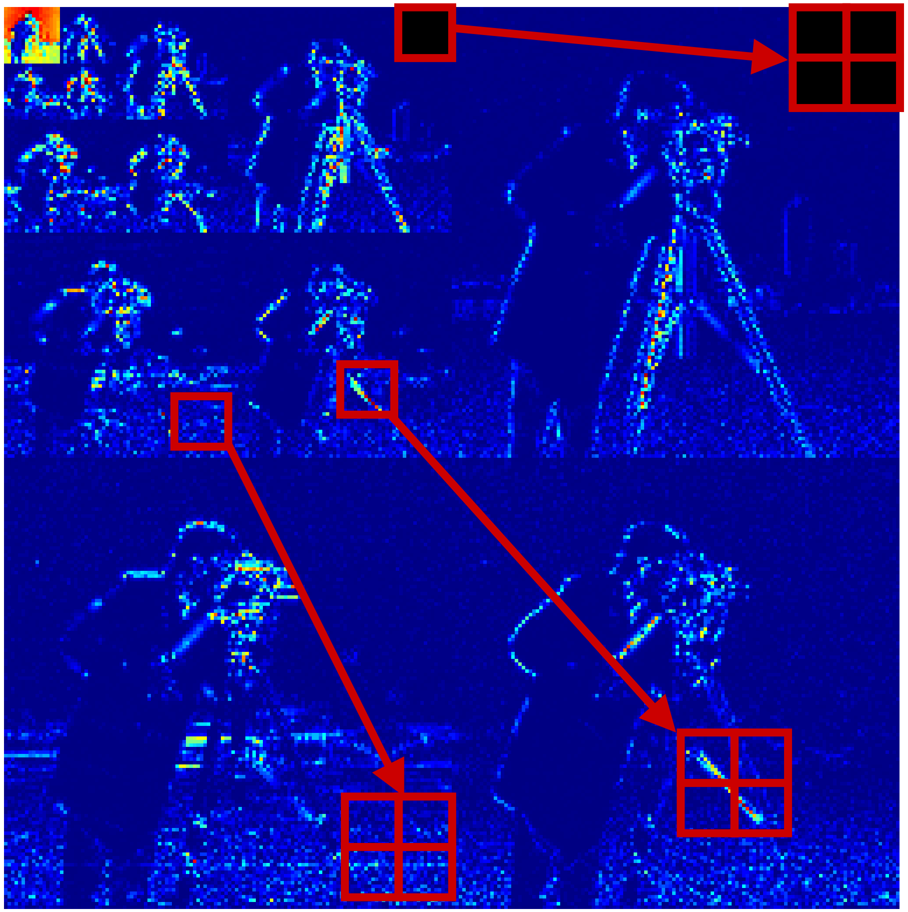
Wavelet Persistence

- *Smooth* region - *small* values down tree
- *Singularity*/*texture* - *large* values down tree



Zero Tree Approximation

- Idea: *Prune* wavelet subtrees in smooth regions
  - *tree-structured thresholding*



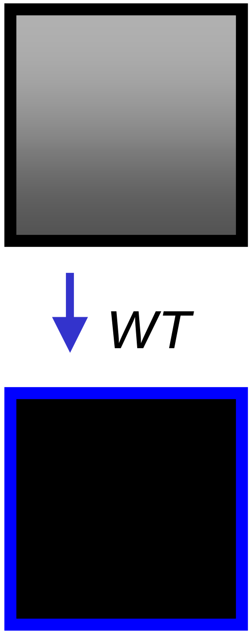
Zero Tree Approximation

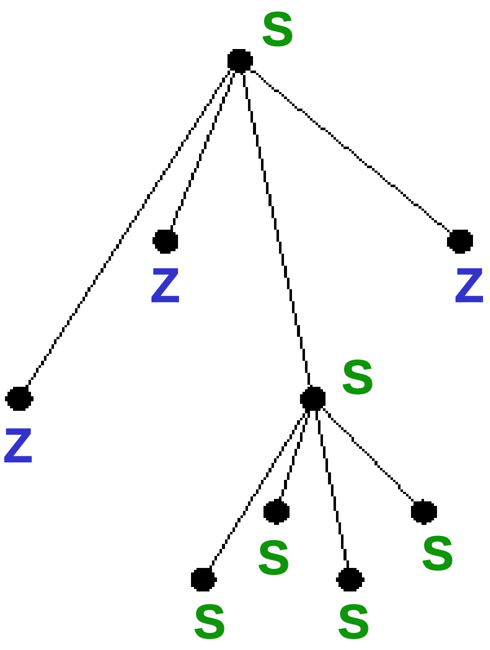
- Prune wavelet quadtree in smooth regions

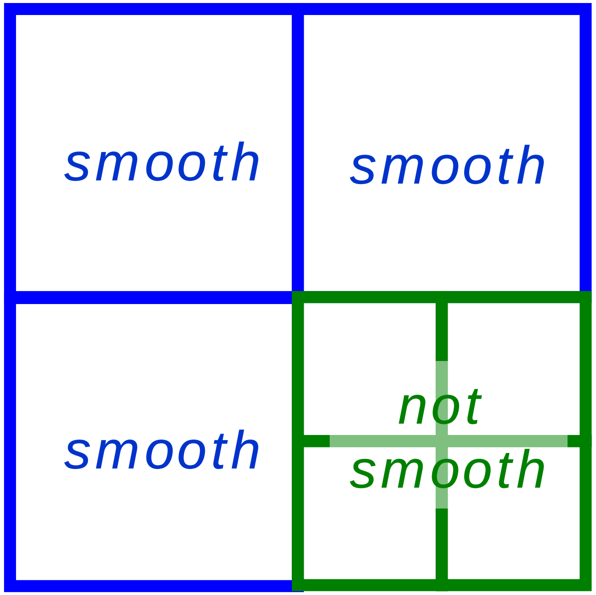
zero-tree	- *smooth region*	(prune)
significant	- *edge/texture region*	(keep)

Z: all wc's below=0

ie: wc's of







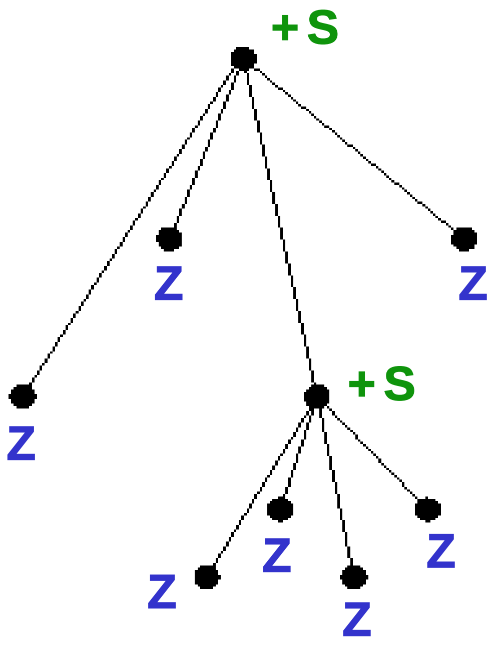
EZW Compression

[Shapiro '92]

- Set threshold $\tau = \max_k \{a_k\}$

- Iterate:

1. Reduce $\tau \leftarrow \tau/2$
2. Threshold $\{a_k\}$
3. Assign labels
   +S, **–S**, **Z**, **I**



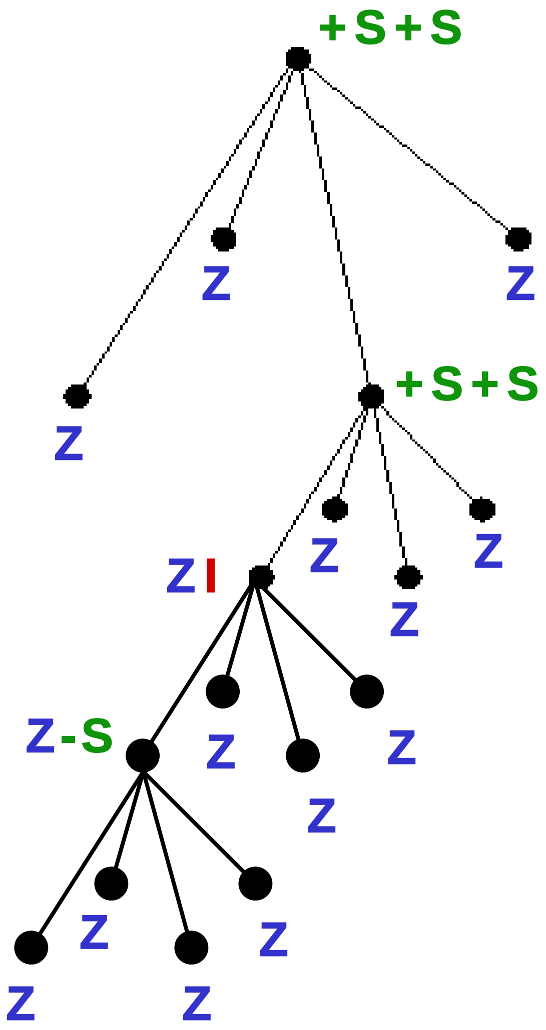
- Encode symbols with arithmetic coder

EZW Compression

[Shapiro '92]

- Set threshold $\tau = \max_k\{a_k\}$
- Iterate:

1. Reduce $\tau \leftarrow \tau/2$
2. Threshold $\{a_k\}$
3. Assign labels **+S**, **–S**, **Z**, **I**



- Encode symbols with arithmetic coder

EZW Compression

[Shapiro ‘92]

- Greedy algorithm based on “persistence” heuristic
- Encodes larger coefficients with more bits
- Progressive encoding (embedded)
    - adds one bit of information to each significant coefficient per iteration
- SPIHT similar

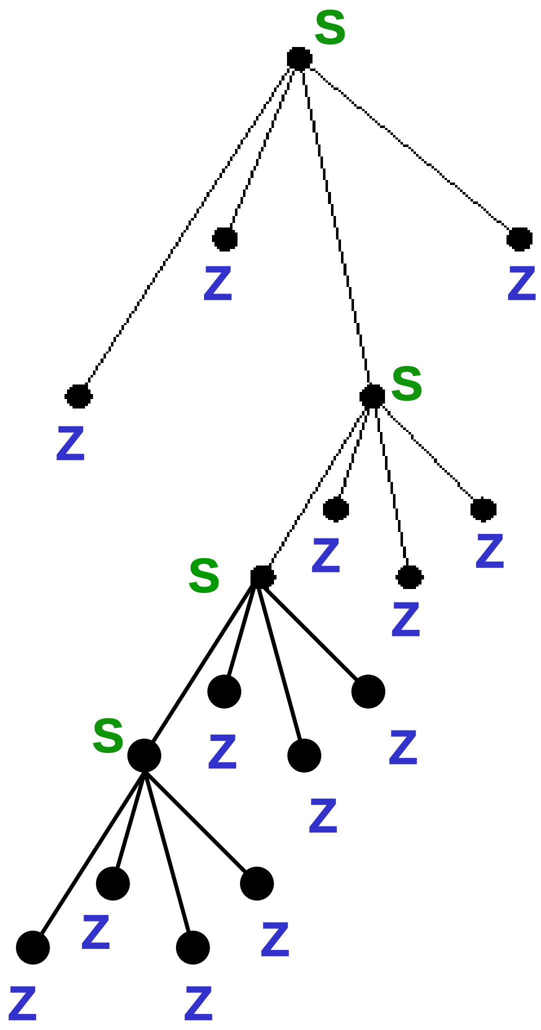
EZW Compression

256x256 pixels, 9,800 total bits, 0.15 bits/pixel

SFQ Compression

[Orchard, Ramchandran, Xiong]

- "Space Frequency Quantization"
- EZW is a greedy algorithm
- SFQ – optimize placement of **S** and **Z** symbols by *dynamic programming*
- Rate-distortion "optimal"
- Not progressive



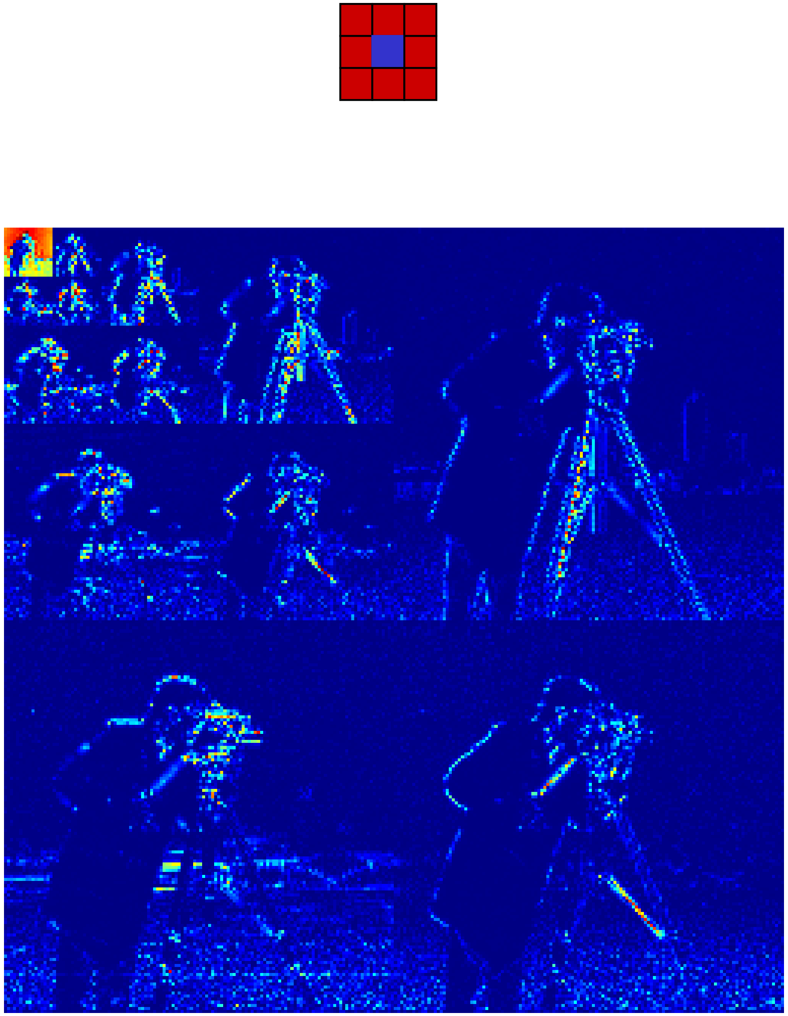
SFQ Compression

256x256 pixels, 9,500 total bits, 0.145 bits/pixel

EQ Compression

[Orchard, Ramchandran, LoPresto]

- “Estimation Quantization”
- *Not* tree-based
- Scans thru each wavelet subband and estimates variance of each wc from its neighbors
- Quantize wc as a Gaussian rv with this variance
- Not progressive



EQ Compression

256x256 pixels, 10,100 total bits, 0.169 bits/pixel

JPEG2000 Compression

- *Not* tree-based
- Similar to JPEG applied to wavelet transform
- Can be progressive

JPEG2000 Compression

256x256 pixels, 9,400 total bits, 0.144 bits/pixel

Discussion and Conclusions

Summary

- Compression is approximation, but approximation is *not* (quite) compression
- Modern image compression techniques exploit piecewise *smooth* image model
  - smooth regions yield small transform coefficients and sparse representation

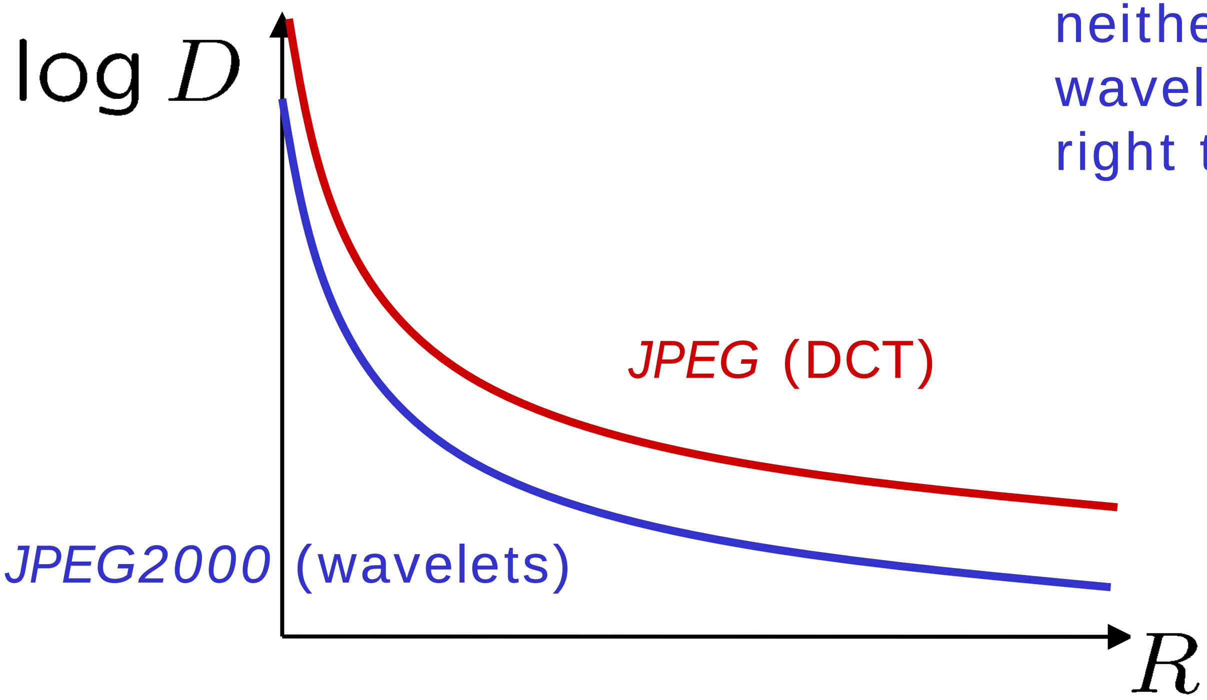
Issues

- Why L_2 distortion metric?
- Pixelization at fine scales

Issues

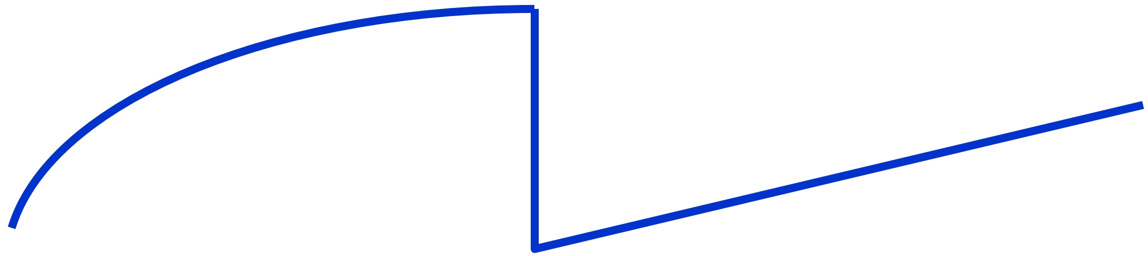
- Current wavelet methods do *not* improve on decay rate of JPEG!

- WHY?
  neither DCT nor wavelets are the right transform



1-D Piecewise Smooth Signals

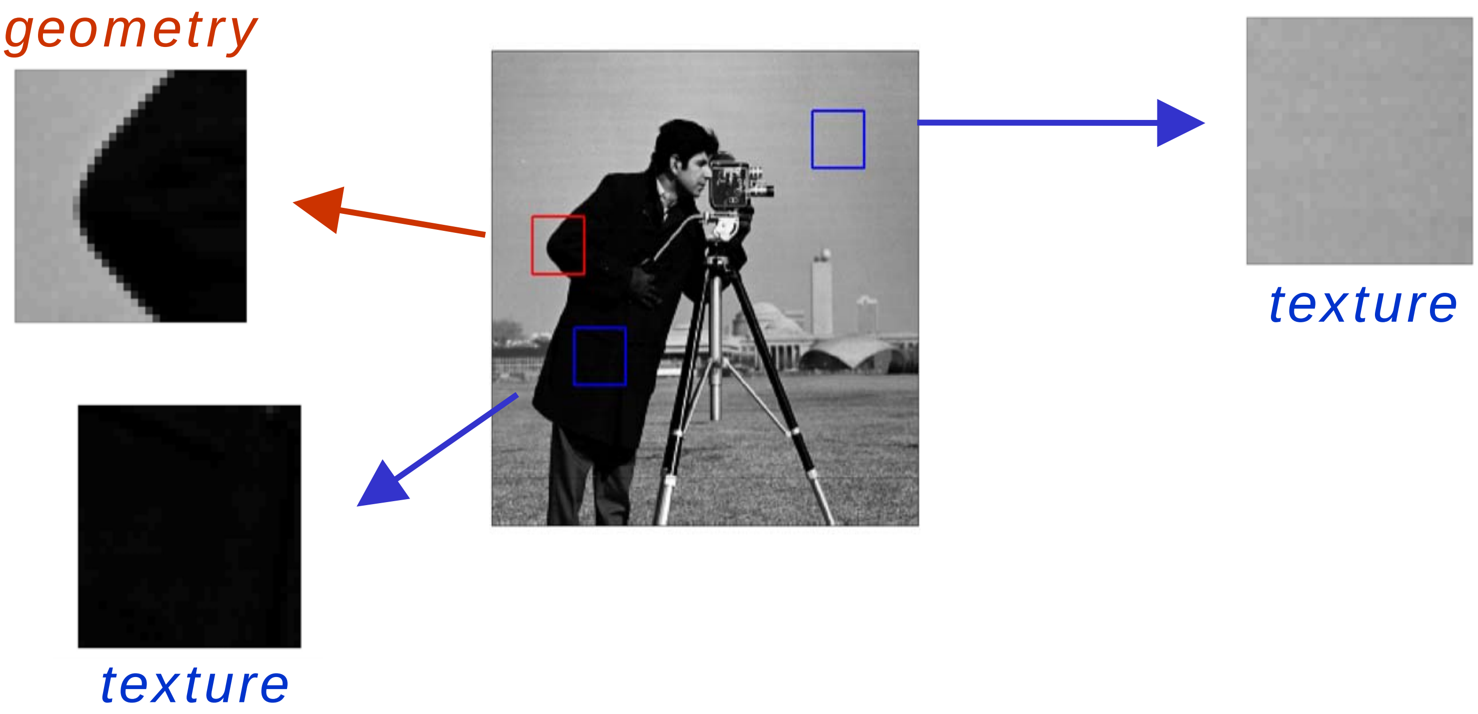
- f *smooth* except for *singularities* at a finite number of 0-D *points*



Fourier sinusoids:	suboptimal greedy approximation and extraction
wavelets:	*optimal* greedy approximation extract singularity structure

2-D Piecewise Smooth Signals

- f *smooth* except for *singularities* along a finite number of smooth 1-D *curves*

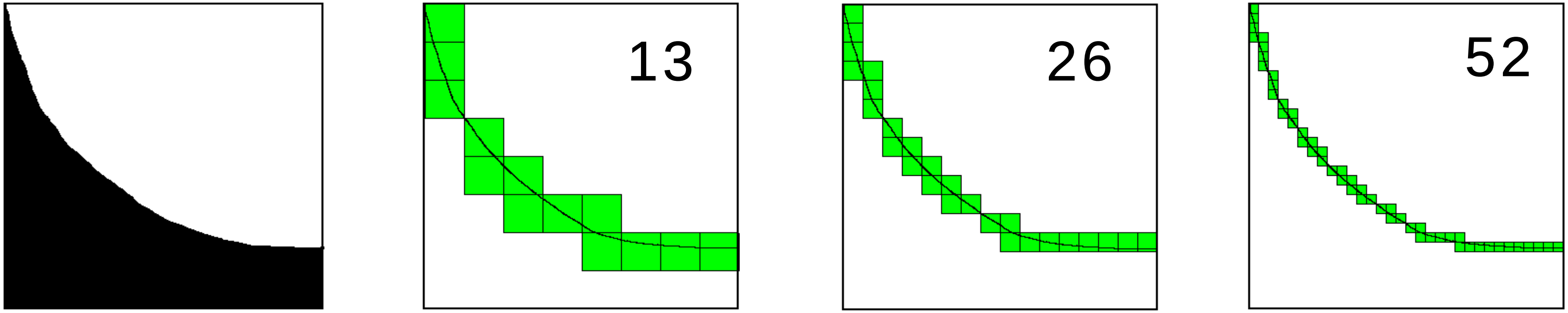


- Challenge: analyze/approximate ***geometric*** *structure*

- *Inefficient* - large number of significant WCs cluster around edge contours, no matter how smooth

2-D Wavelets: Poor Approximation

- Even for a smooth C^2 contour, which straightens at fine scales...



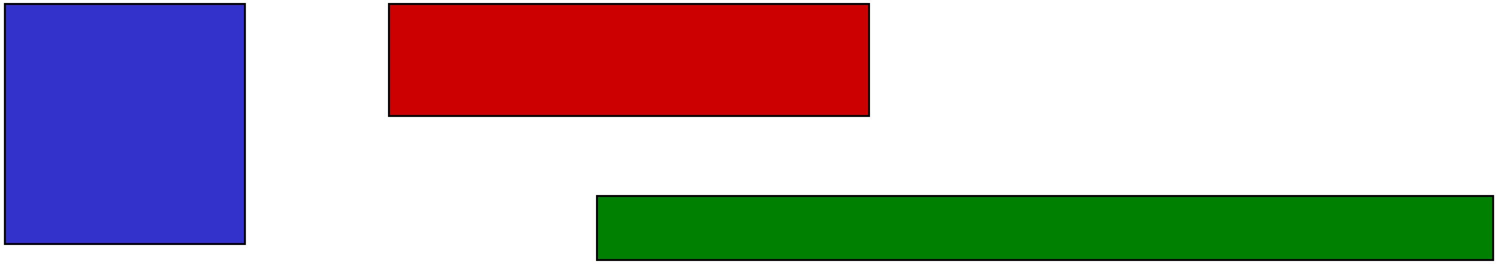
- *Too many wavelets required!*

$$\widehat{f}_N \;:=\; N\text{-term wavelet approximation}$$

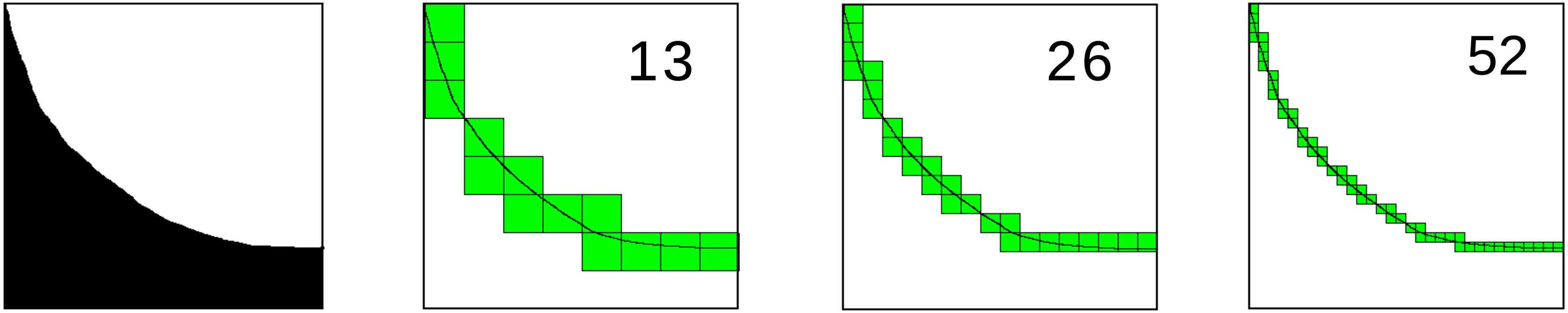
$$\|f - \widehat{f}_N\|_2^2 < C\,N^{-1} \quad \textit{not} \quad N^{-2}$$

Solution 1: Upgrade the *Transform*

- Introduce *anisotropic transform*
  - – curvelets, ridgelets, contourlets, ...

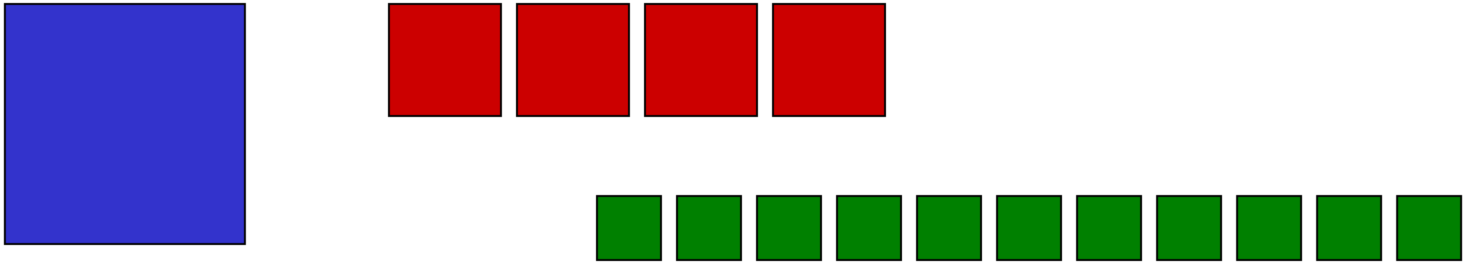


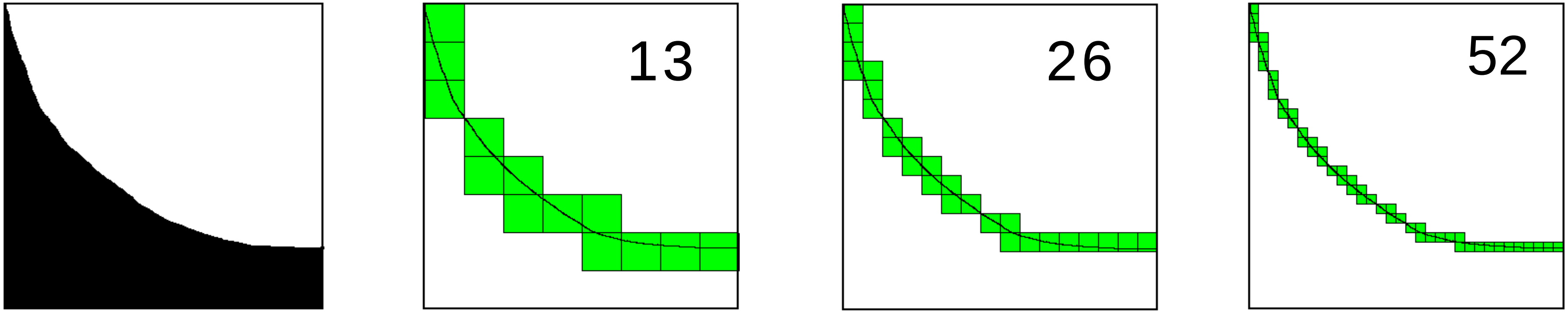
- Optimal error decay rates for cartoons +



Solution 2: Upgrade the *Processing*

- Replace coefficient thresholding by a new wavelet coefficient *model* that captures *anisotropic spatial correlations* of wavelet coefficients





Richard Baraniuk
Mike Wakin
Hyeokho Choi
Justin Romberg

richb@rice.edu

Web: **dsp.rice.edu**