The Art and Practice of Modern Image Compression

©William A. Pearlman

Professor, RPI and Director, CNGV

pearlman@rpi.edu

http://www.rpi.edu/web/NGV,

http://www.cipr.rpi.edu/research

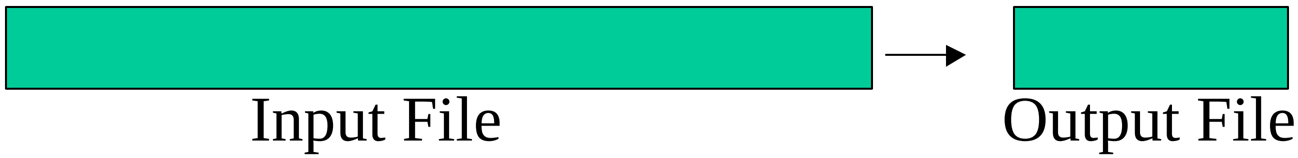
Rensselaer

CNGV

Outline

- Digital image compression
  - Focus on natural, 8 or more bits per pel sources
- Features expected of modern systems
- Compression systems
  - Focus on low complexity
- Extensions to other modes and sources
  - 3D images, 2D and 3D data, etc.
- Future challenges

Image/Video Compression (Coding)



- Transmission over channel of given bandwidth saves time
- Transmission in given time saves bandwidth
- Saves storage space

- Almost every digital image or video viewed has been compressed: GIF, JPEG, MPEG-2 (DVD, HDTV)
- Data collection will continue to outstrip advances in storage technology
  - ❖Estimate 80 billion new images produced last year!

Rensselaer

CNGV

Status of Image Coding

- Current JPEG standard DCT-based
  - blocking artifacts at low to medium bit rates
  - needed features not natural
    - built in with limited capability and flexibility
- Migration to wavelet transform coding
  - eliminates blocking artifacts
  - superior low-medium rate performance
  - natural features and flexibility

Features Requirements

- High performance
  - Low MSE (high PSNR) for given rate
- Fast, simple encoding and decoding
- Idempotency- compressed bitstream to reconstruction perfectly reversible
- Low memory usage
  - Achieved in current JPEG by independent coding of 8x8 image blocks

Rensselaer

CNGV

Features (Cont.)

Random access decoding

Access piece or pieces of file to decompress given image region

Decoding and access time considerably reduced

Rensselaer

Random Access Decoding*

CNGV

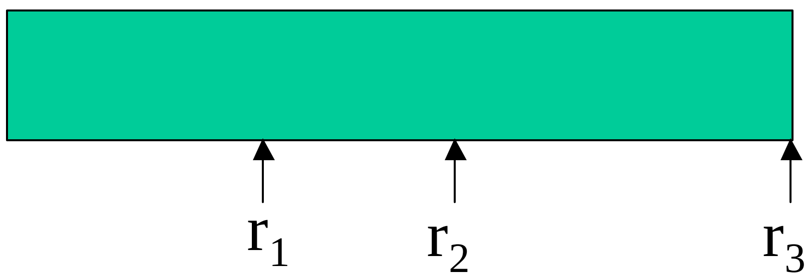
720x576 Goldhill
0.92 bpp
trans 0.66s
comp 0.43s
total 1.57s
----
decomp 0.06s
recover 0.68s
total 1.19s

92x123
decode 0.01s
recover 0.05s
total 0.08s

*400 MHz PII, Linux OS

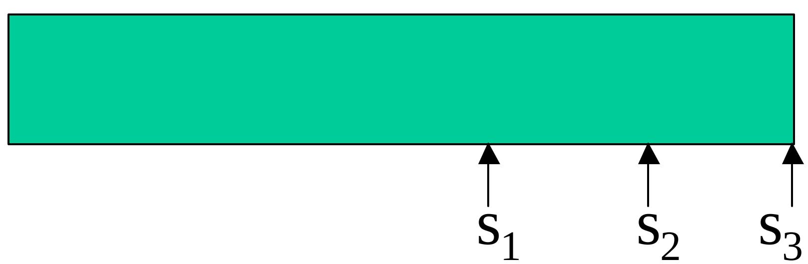
Features Requirements: Scalability

Rate or PSNR scalability: lower rate files embedded in full compressed file



s_2

Spatial (resolution) scalability: lower resolution files embedded in full compressed file



s_1

Rensselaer

ROI Encoding

CNGV

60x103 ROI
0.299 bpp

compress 0.41 s
decompress 0.04 s

ROI bits first,
decompressed first

Rensselaer

CNGV

Transmission Features

- Robust to channel errors/packet loss
  - unprotected bit stream degrades gracefully
  - no overhead required for robustness
- Resilience to channel errors/packet loss
  - redundant bits added for error-correcting channel code
  - reception of acceptable quality in severe environments

Rensselaer

CNGV

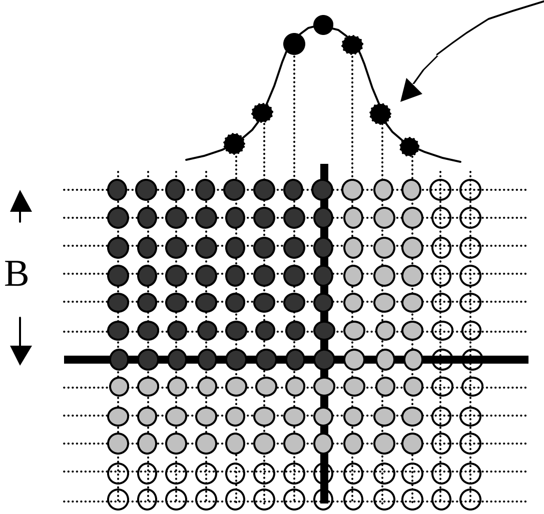
Large Image Issues

- Small dynamic (on-chip or cache) memory
  - line-based or block-based transform
    - same coefficients as full transform
    - processing more complex
  - compromise embedded coding (SNR scalability)
  - tile or stripe the image

Block-Based Transform (SSWT)

1 level of decomposition

Filter Coefficients
Filter Length = 7,
Overlap O_1=3



L - filter length
D - decomposition levels

$$O_D = O_1 \bullet (2^D - 1)$$

$$O_1 = \begin{cases} (L-1)/2, & L \text{ Odd} \\ L/2, & L \text{ Even} \end{cases}$$

● Block Pixels

● Overlap Pixels

○ Pixels not used in transform

Overlap Lengths O_D adjacent to block,
Line-based analogous, uses entire line, overlap vertical only

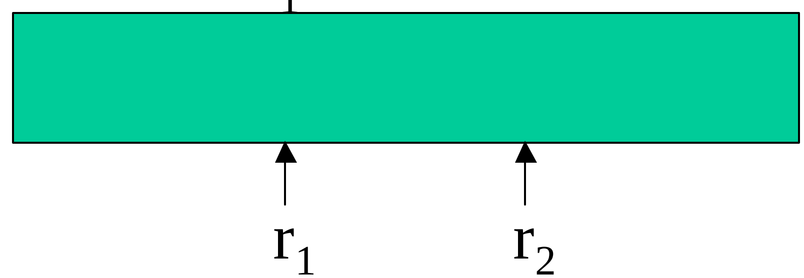
Rensselaer

CNGV

Embedded Coding

Progression from lossy to perfectly lossless enabled with integer-mapping filters

All optimal (for that technique) lower rate r_1 files embedded in higher rate file r_2 for all $r_1 < r_2$



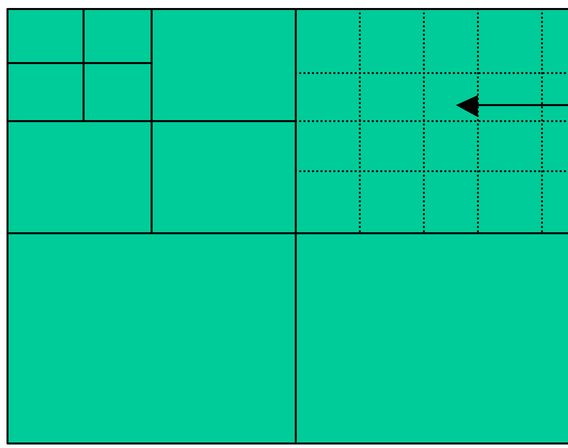
- Requires full transform or full compressed file in memory
  - unfeasible for large images
- For small memory, compromise to some finite set of rates r_1 and r_2

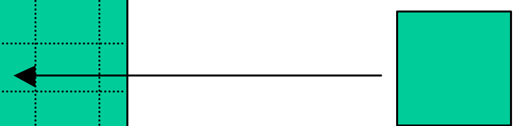
Rensselaer

JPEG2000 Embedded Coding

CNGV

Subbands of wavelet transform tiled into 64x64 (or 32x32) blocks (other sizes for odd-sized or smaller subbands).





Coded independently via context-based bit plane arithmetic coding

1) Each block coded to high rate and then number of optimal smaller size points for different rates determined by rate allocation procedure
2) Bit stream reorganized by decreasing bit planes
3) For given rate, bitstreams truncated at corresponding rate point for each block.

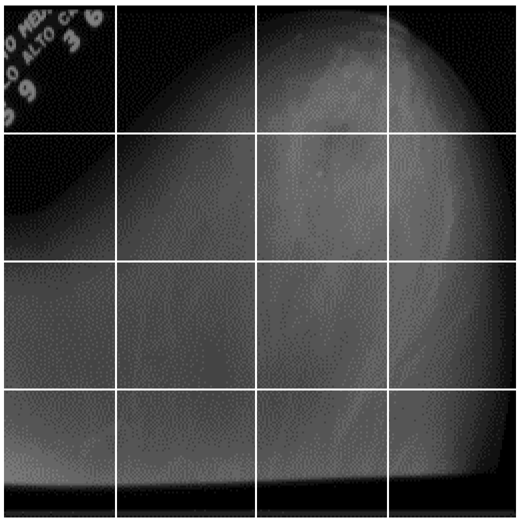
Tiles and Stripes

Original 1024 x 1024 mammogram

(1/4 resolution inverse WT for display)

256

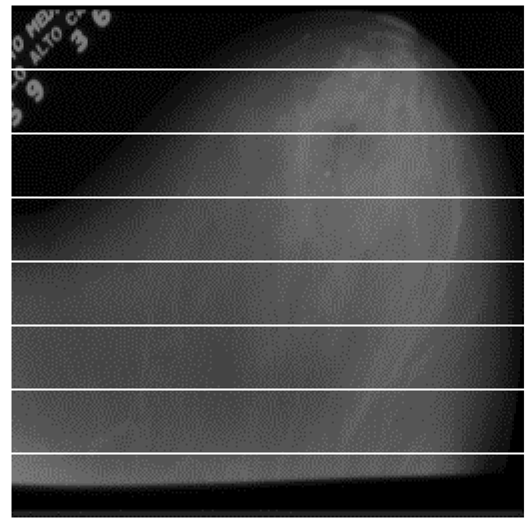
256



Tiles

1024

128



Stripes

Tiles or stripes coded independently w/wo 1 or 3 row and/or column overlap - embedding usually forsaken

Boundary artifacts at lower rates alleviated by overlap

Aim to achieve same quality in each tile or stripe

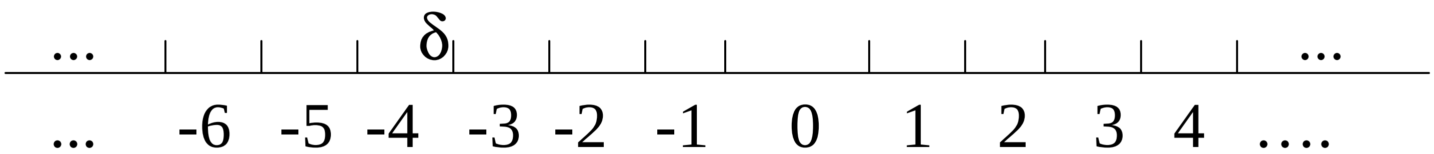
Coding Algorithms

- Progressive lossy-to-lossless
  - High complexity
    - JPEG2000 EBCOT-based coder
  - Low complexity
    - SPECK/SBHP (Set Partitioning Embedded bloCK/Subband Block Hierarchical Partitioning)
    - SPIHT (Set Partitioning In Hierarchical Trees)
- Lossless/Near-lossless
  - High complexity : CALIC by Wu and Memon
  - Low-medium complexity: JPEG-LS (new standard)

Rensselaer JPEG 2000 (EBCOT) Coding CNGV

Quantization and Coding of a Block

-Same uniform, dead zone quantizer for all blocks of all subbands



-Sign-magnitude representation of bin numbers in above

-Do bit plane coding of magnitudes from highest to lowest bit planes, skipping highest all-zero bit planes

-If bit stream truncated, so that last available bit plane is *p*-th

-equivalent to coarser quantizer of step size $\delta 2^p$.

JPEG 2000 (EBCOT) Coding (cont.)

Arithmetic Coding of Bit Planes

Bit planes encoded with context-based, adaptive arithmetic codes.

- uses 3 passes through each bit plane to identify bits in different significance classes, different context for each
- zero run-length coding mode
- Sign bits also encoded by context-based, adaptive arithmetic codes.

Low Complexity Set Partitioning Coders: SPECK and SPIHT

Operate through significance decisions for partitioning sets of wavelet transform coefficients.
Sets and partitioning rules differ between SPECK and SPIHT.

Significance test for pixels $c_{i,j}$ in set B

$$S_n(B) = \begin{cases} 1, & \text{if } \left(\max_{(i,j)\in B} |c_{ij}|\right) \geq 2^n, \\ 0, & \text{otherwise.} \end{cases}$$

SPECK

- SPECK discovered by Islam and Pearlman (VCIP'99) and possibly others
- Partitions and codes sets of coefficients grouped within subbands
- Supports all features of JPEG2000 and more

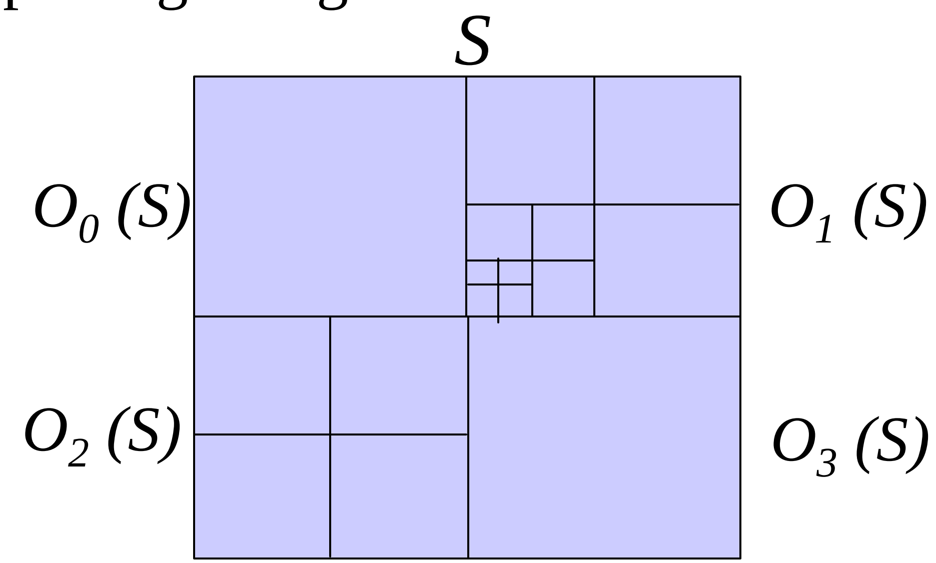
Set Partitioning Rules of SPECK

S Partitioned into Offspring Sets

Recursive quadrature splitting of significant sets

LIS - list of insignificant sets in increasing order of size starting with single element sets

LSP - list of significant single elements.

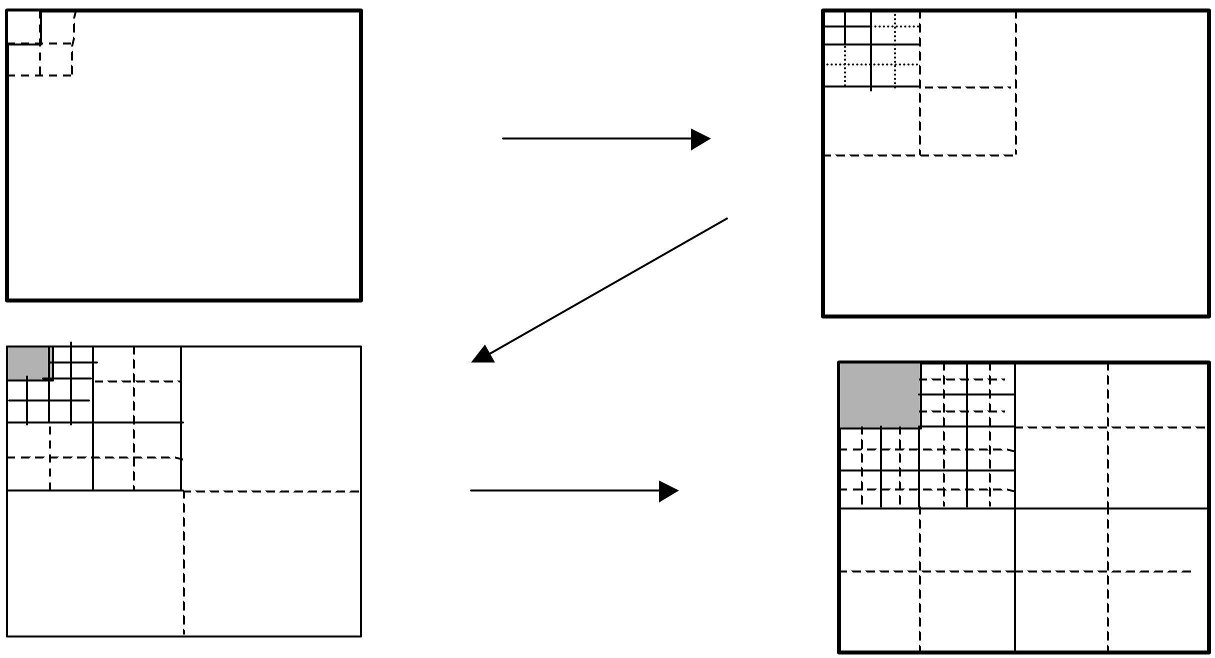


For given threshold or bit plane *n*, insignificant sets and coefficients go to LIS, significant coefficients to LSP. Decrease n by 1, repeat.

Subband Partitioning Order

Full Transform

- Partitioning proceeds from coarse to fine scale



Rensselaer

CNGV

LSP Sorting by Magnitude and Progressive Bit-Plane Transmission

Transmission of magnitude-sorted coefficients

	sign	s	s	s	s	s	s	s	s	s	s	s	s	s
msb	5	1	1	0	0	0	0	0	0	0	0	0	0	0
	4	®	®	1	1	0	0	0	0	0	0	0	0	0
	3	®	®	®	®	1	1	1	1	0	0	0	0	0
	2	®	®	®	®	®	®	®	®	1	1	1	1	1
	1	®	®	®	®	®	®	®	®	®	®	®	®	®
lsb	0	®	®	®	®	®	®	®	®	®	®	®	®	®

Send *n*-th bit of all coefficients found to be significant in previous passes.

Transmission of Bits

- LSP refinement bits sent raw
- Significance decision and sign bits sent raw
  - arithmetic coding in 2x2 sets possible, but not necessary
- Note difference from JPEG 2000 coding, where all bit planes from 1st nonzero one of all coefficients must be arithmetic coded with 3 passes per bit plane

Rensselaer

Embedding Issues

CNGV

- Full transform SPECK fully bit-embedded
  - next bit conveys less value information than previous one
- Full bit-embedding costs in speed and complexity
  - many applications do not require full bit-embedding and/or can not afford it (viz., geographic images, real-time animation)
- SPECK (and SPIHT) allow relaxation to *value embedding* by skipping refinement pass and sending all lower bit planes immediately for each new LSP coefficient.
  - *can not be done in JPEG2000*

SPIHT Algorithm

- Discovered by Said and Pearlman (IEEE CSVT 1996)
- Partitions wavelet transforms into spatial orientation (SO) tree sets
  - SO trees correspond directly to spatial regions and are computed in natural order in wavelet transform from bottom (fine scale) to top (coarse scale).
- Possesses all features of SPECK
  - more convenient and flexible for transform manipulations

Rensselaer

CNGV

SPIHT Tree Structure

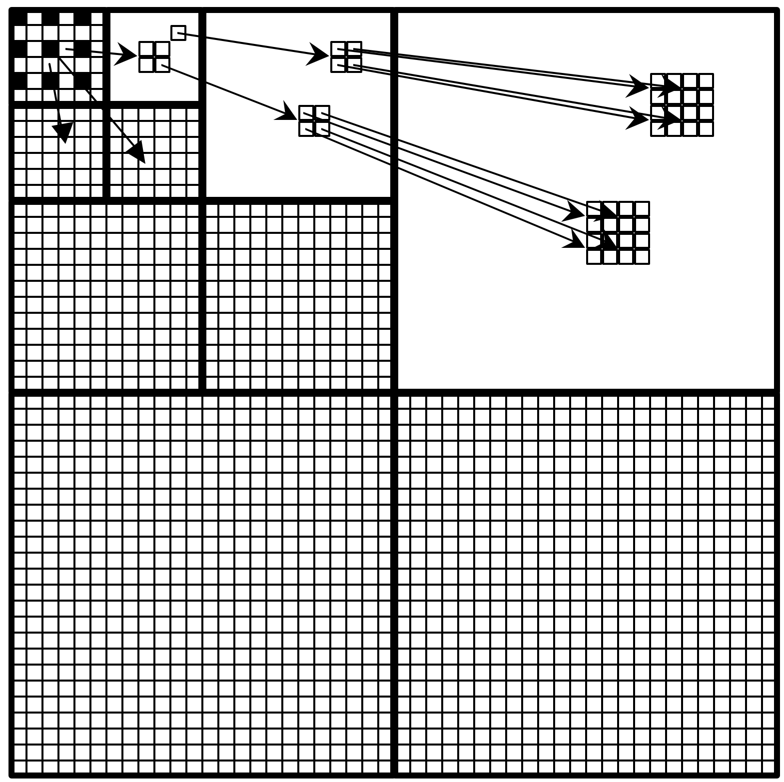
3 level dyadic subband transform shown
Arrows depict parent-child relationships in the SPIHT tree structure
Coefficients are grouped to exploit magnitude dependence
Each subband coefficient has four children
Some coefficients in the DC subband have no children

Each coefficient is denoted by it coordinates (i,j)

Set Types

(i,j): Single coefficient
C(i,j): Children of (i,j)
D(i,j): Descendants of (i,j)
G(i,j): D(i,j) - C(i,j)

Significant sets D(i,j) partitioned C(i,j) and G(i,j)



Rensselaer

Coding Sequence

CNGV

- SPIHT uses three lists LIP, LIS, LSP visited for significance testing in that order, for a given bit-plane (*n*).
  - LIP: list of coordinates of insignificant pixels
    - Initialized by highest level low-pass (DC) subband
  - LIS: list of coordinates of insignificant sets and their type (D or G)
    - Initialized by coordinates in DC subband with descendants as D type
  - LSP: list of coordinates of significant pixels
    - Send n-th bit of all pixels found to be significant in previous passes.

Rensselaer

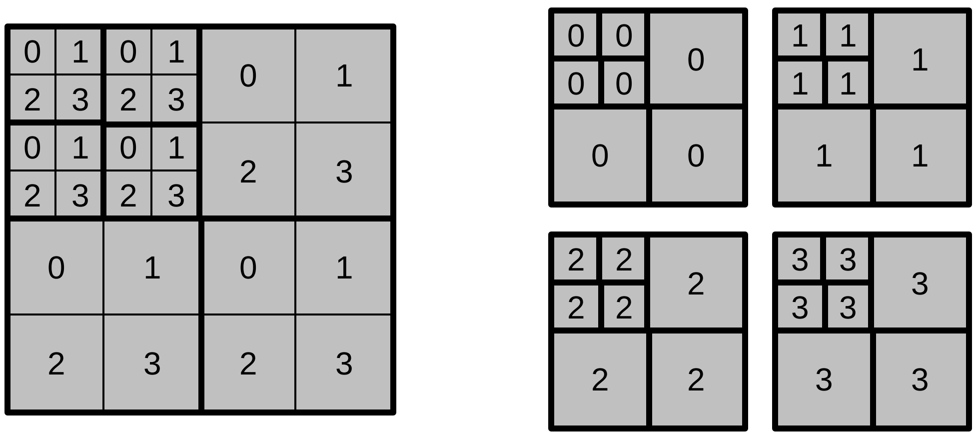
CNGV

Small Memory Implementations

- SPECK used in small subband blocks (SBHP)
  - replaced coding of subband blocks in JPEG2000 framework
- SPIHT implemented in JPEG2000 framework
  - SPIHT coding applied to small transform tree-blocks independently
- Embedded bit streams from independent tree-block or subband block encoders are packetized and multiplexed to generate a progressive main bit stream

Tree-block Partitioning for Small Memory

- Works with line- or block-based wavelet transform
- Transform will deliver subband partition corresponding to image block
- Subband partition is organized into tree-blocks, if necessary
- SPIHT encoder is applied to trees
- Example shows a 2 level subband partition organized into 4 trees forming a tree-block



In actual experiments a 5 level transform is used and partitioning generates 128 by 128 or 64 by 64 tree-blocks

Rensselaer

Progressive Bit Stream

CNGV

- The SPIHT/SPECK algorithm generates a progressive bit stream for each tree/subband block
- As the SPIHT/SPECK algorithm progresses it periodically records the current bit plane and list (LIP, LIS or LSP) being processed
- Once a sub-bit stream has been generated for each sub-image they are packetized and multiplexed to generate the primary bit stream
- Priority is given to packets with data for more significant bit planes
- Within the same bit plane, priority is given to packets with data for the LIP, then LIS then LSP
- The packetized bit stream is progressive
- There is no Huffman or arithmetic coding for SPIHT. SPECK/SBHP uses 2 fixed Huffman codes

Example main bit stream structure:

hdr	0	data for 0	1	data for 1	2	data for 2	3	data for 3

2	data for 2	3	data for 3	3	data for 3	1	data for 1	2	data for 2

Rensselaer

CNGV

PSNR Results

- 9/7 tap filters, dyadic decomposition, 64x64 tree-blocks

Rate (bpp)	2	1	0.5	0.25	0.125	0.0625
Aerial2						
VM3.2A	38.401	33.440	30.713	28.662	26.587	24.717
IPSPIHT	37.274	32.664	30.038	27.837	25.680	23.374
PRSPIHT	37.376	32.532	29.889	27.670		
Bike						
VM3.2A	44.302	38.295	33.659	29.730	26.456	23.852
IPSPIHT	41.545	36.452	31.776	27.857	24.381	21.567
PRSPIHT	42.810	36.713	31.915	27.829		
Café						
VM3.2A	39.443	32.268	26.990	23.273	20.874	19.099
IPSPIHT	37.698	30.630	25.612	22.028	19.709	17.969
PRSPIHT	37.936	30.756	25.588			
Woman						
VM3.2A	44.329	38.648	33.807	30.014	27.453	25.672
IPSPIHT	42.643	37.075	32.296	28.664	26.113	24.161
PRSPIHT	43.013	37.389	32.433	28.552		

Bike, Café, Woman:2048x2520   Aerial2: 2048x2048

Rensselaer

Time Measurements

CNGV

Bike 2.0 b/p Sun Ultra 20 300 MHz CPU		
	Encode (sec)	Decode (sec)
VM3.2A	12.20	7.43
IPSPIHT	4.73	3.37
PRSPIHT	1.50	0.70

Bike 0.5 b/p Sun Ultra 20 300 MHz CPU		
	Encode (sec)	Decode (sec)
VM3.2A	12.11	2.37
IPSPIHT	2.41	1.39
PRSPIHT	1.00	0.20

Bike 1.0 b/p Sun Ultra 20 300 MHz CPU		
	Encode (sec)	Decode (sec)
VM3.2A	12.21	4.31
IPSPIHT	3.95	2.00
PRSPIHT	1.18	0.37

Bike 0.25 b/p Sun Ultra 20 300 MHz CPU		
	Encode (sec)	Decode (sec)
VM3.2A	12.08	1.26
IPSPIHT	1.55	0.95
PRSPIHT	0.90	0.12

Rensselaer

Summary Results with SPIHT

- The SPIHT algorithm can be operated in small tree blocks within JPEG 2000 framework
  - complexity reduction gives
    - average encoder speedup of 4.6 for IP (embedded) and 11 for PR (non-embedded)
    - average decoder speedup of 1.9 for IP (embedded) and 9.9 for PR (non-embedded)
    - average PSNR reduction of 1.4 dB
- Makes full-featured compression feasible for today's larger images in embedded applications (printers, scanners, digital cameras)

* Special thanks to Amir Said, Fred Wheeler

SPECK/SBHP in JPEG2000 VM 4.2*

- Bit rates : 1/16, 1/8, 1/4, 1/2, 1, 2, lossless
- Modes:
  - single layer: one pass rate control
  - seven layer scalable: bit stream optimized for 7 rates, same bit stream decoded for each rate
  - generic scalable:bit stream optimized for 60 rates, same bit stream decoded for each rate
- 16 images x 7 rates x 3 modes x 3 block sizes x 2 filters x 2 VM-coders = 4032 points

*ICASSP 2000, Chrysafis et al.

Rensselaer

CNGV

Encoding Times

Bike (2048x2560) 64x64 blocks, single layer mode, 300 MHz Pentium		
	JPEG2K VM4.2	**SBHP**
Rate (bpp)	Entropy Encode. (sec.)	Entropy Encode. (sec.)
0.061	9.88	2.33
0.123	9.83	2.29
0.248	9.88	2.55
0.498	9.83	2.39
0.997	9.82	2.51
1.998	9.84	2.43

Rensselaer

Entropy Decoding Times

Entropy Decoding Times, Single Layer Mode Bike (2048x2560), 64 x 64 blocks					
	300 MHZ Pentium II, Win NT 4.0			300 MHz, Sun Ultra 10, Solaris 2.6	
Rate (bpp)	VM4.2 times (sec.)	SBHP times (sec.)		VM4.2 times (sec.)	SBHP times(sec)
0.061	0.2510	0.0233		0.25	0.02
0.123	0.5790	0.0548		0.48	0.07
0.248	0.9860	0.1203		0.98	0.19
0.498	2.0610	0.2385		1.88	0.27
0.997	3.8460	0.4295		3.37	0.54
1.998	6.9500	0.8197		6.00	1.02
Lossless 5x3					
4.5238	14.0630	1.8090			2.58

SBHP Performance Summary

- PSNR loss to VM4.2
  - Photographic and Medical/Satellite: 0.4 - 0.5 dB
- Bit Rate loss to VM4.2
  - Photographic and Medical/Satellite: 5-10 %
    - Lossless compression : 1-2%
- Speed Gain

Encoder **400%**

Decoder:

* embedded **800 %**

***** non-embedded **1100 %**

Rensselaer CNGV

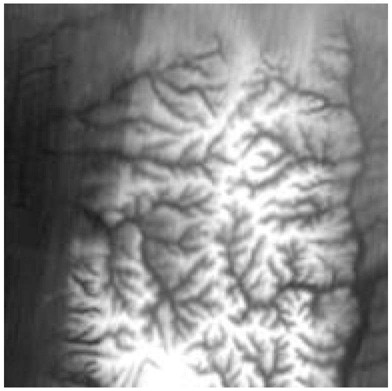
Summary of SBHP/SPECK

- SBHP/SPECK has small loss, but great speed over VM4.2
  - supports all features (and more), operates in EBCOT framework
  - broadens market and application base of JPEG2000
  - low cost hardware implementation (cameras, mobile devices, real-time animation, etc.)

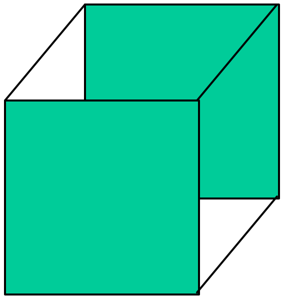
* Special thanks to Asad Islam, Amir Said, Christos Chrysafis, Alex Drukarev

Rensselaer

Trends for Image Compression



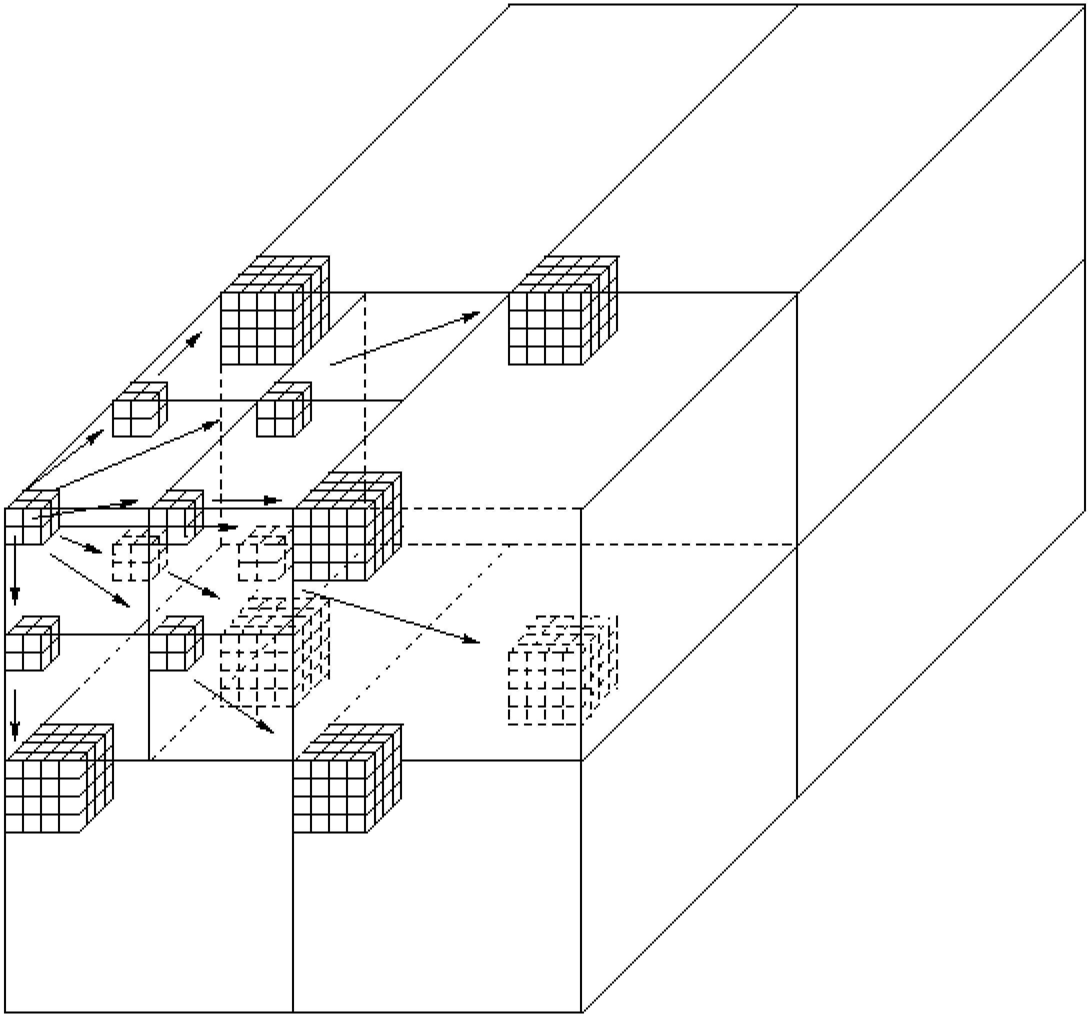
- 2D data: values associated with 2D grid
  - example: terrain elevation data
- 2D electron microscope images
- 3D images:
  - multi-component images
  - tomographic images - medical, materials, geological
- 3D data: values associated with 3D grid
  - temperature, pressure, moisture, etc. at points in atmospheric volume
  - multi-spectral data
- MSE criterion may be inappropriate for compression of data



3D Image/Data Coding

- 3D Wavelet Subbands
  - groups of 16 frames (slices) or rolling wavelet
  - axial subbanding decorrelates and compacts energy into low frequency bands
- SPIHT coding on 3D s-t orientation trees
- SPECK 2D coding of frames of 16 slice 3D transform

3D SPIHT Tree Branching



3D Wavelet Transform/2D Coding

- Use 3D wavelet transform as before
- Encode axial (intra-slice) subbands with 2D SPECK compression algorithm
- Method emulates JPEG2000 with different, lower complexity entropy coding engines
  - transaxial line transform with 4 levels of decomposition
  - axial transform with log2 (GOS) -1 levels

Lossless Volume Image Compression

MR Chest 256x256x64, CT skull 256x256x128, 8 b/p

Method	GOS	Filters	Bit Rate (bits per pel)	
			MR Chest	CT Skull
3D SPIHT	16	I(4,2)	1.78	2.04
3D WT/2D SPECK	16	I(5,3)	2.176	2.47
2D SPIHT	1	I(4,2)	2.85	2.69
JPEG-LS	1	none	2.93	2.85

2D-3D WT/Coding Improvement

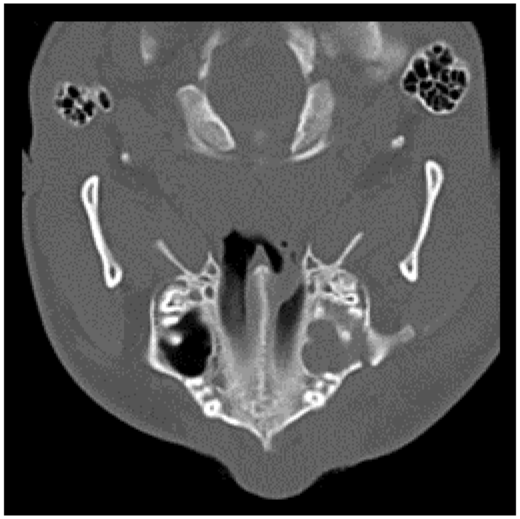
	SPIHT at 0.1 bpp			SPECK at 0.25 bpp		
	GOS 1	GOS 16	D	GOS 1	GOS 16	D
MR Chest*	36.38	42.98	**6.60**	39.46	45.60	**6.14**
CT Skull*	26.38	33.98	**7.60**	29.66	36.51	**6.85**

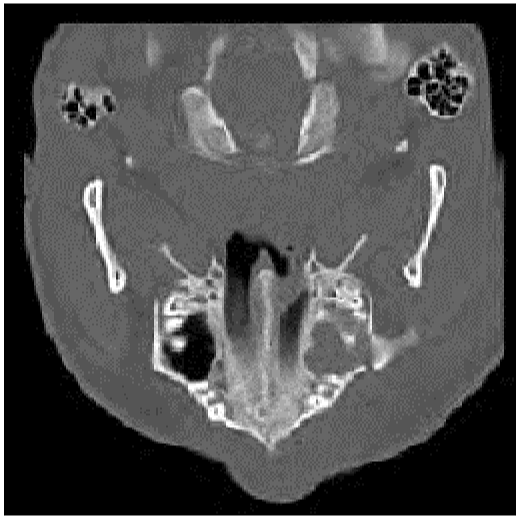
*Numbers in rows are PSNR in dB

Rensselaer

CT Skull Coding Comparisons

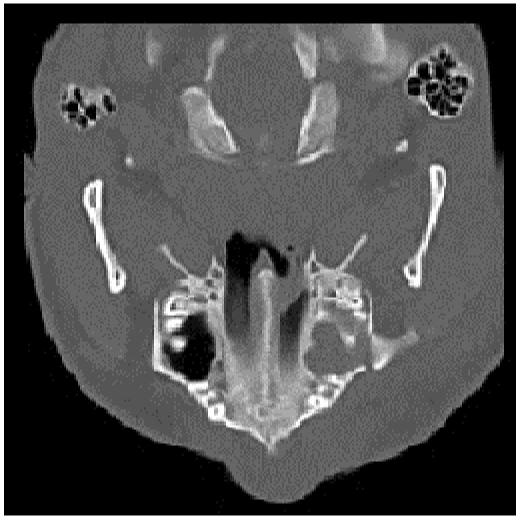
Original
slice 32





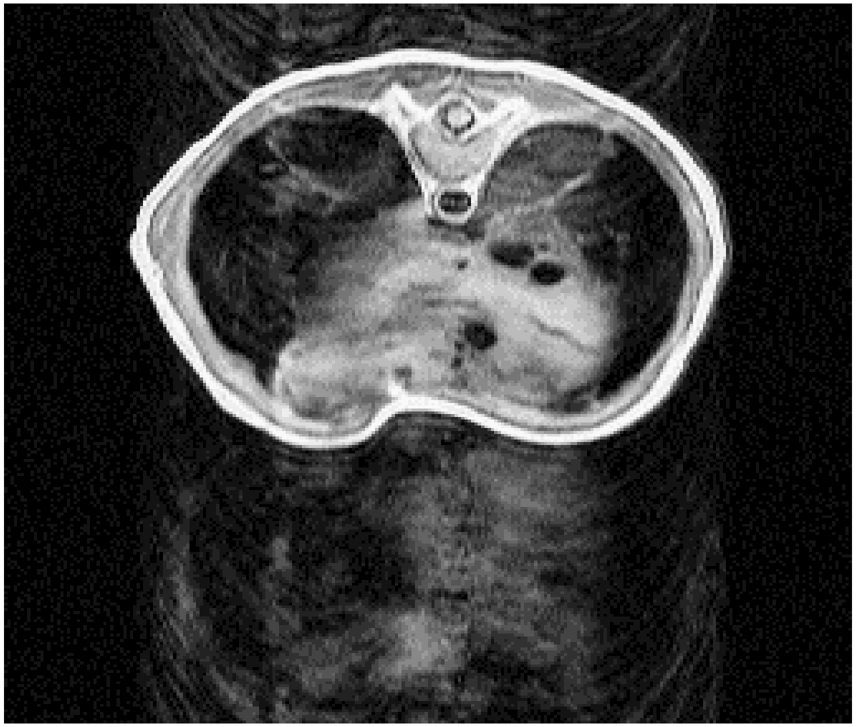
SPIHT
0.1 bpp
5 levels

OL SPIHT
0.25 bpp
5 levels

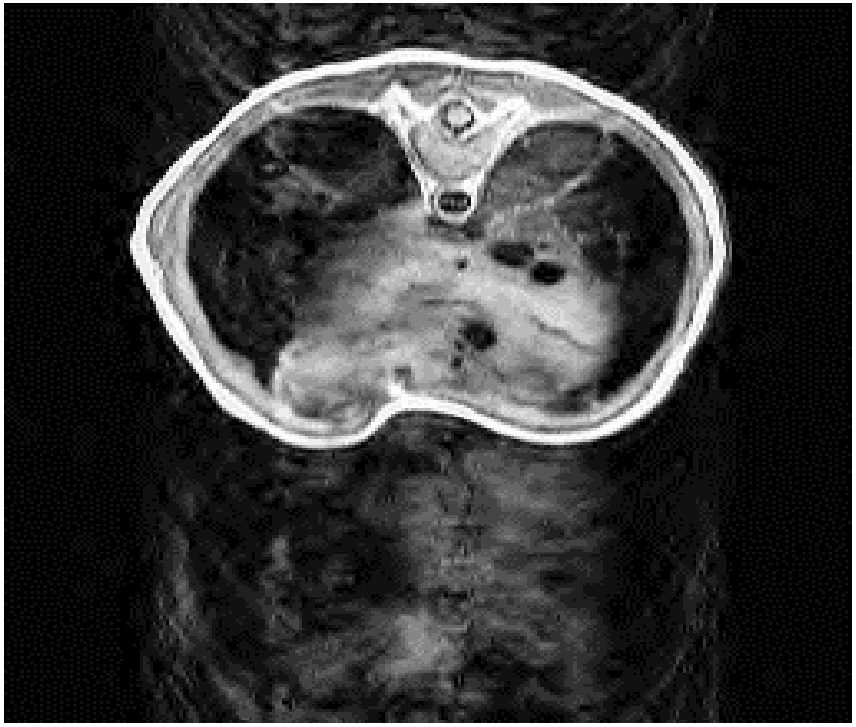


Progression to lossless
coding attainable

MR Chest Sequence



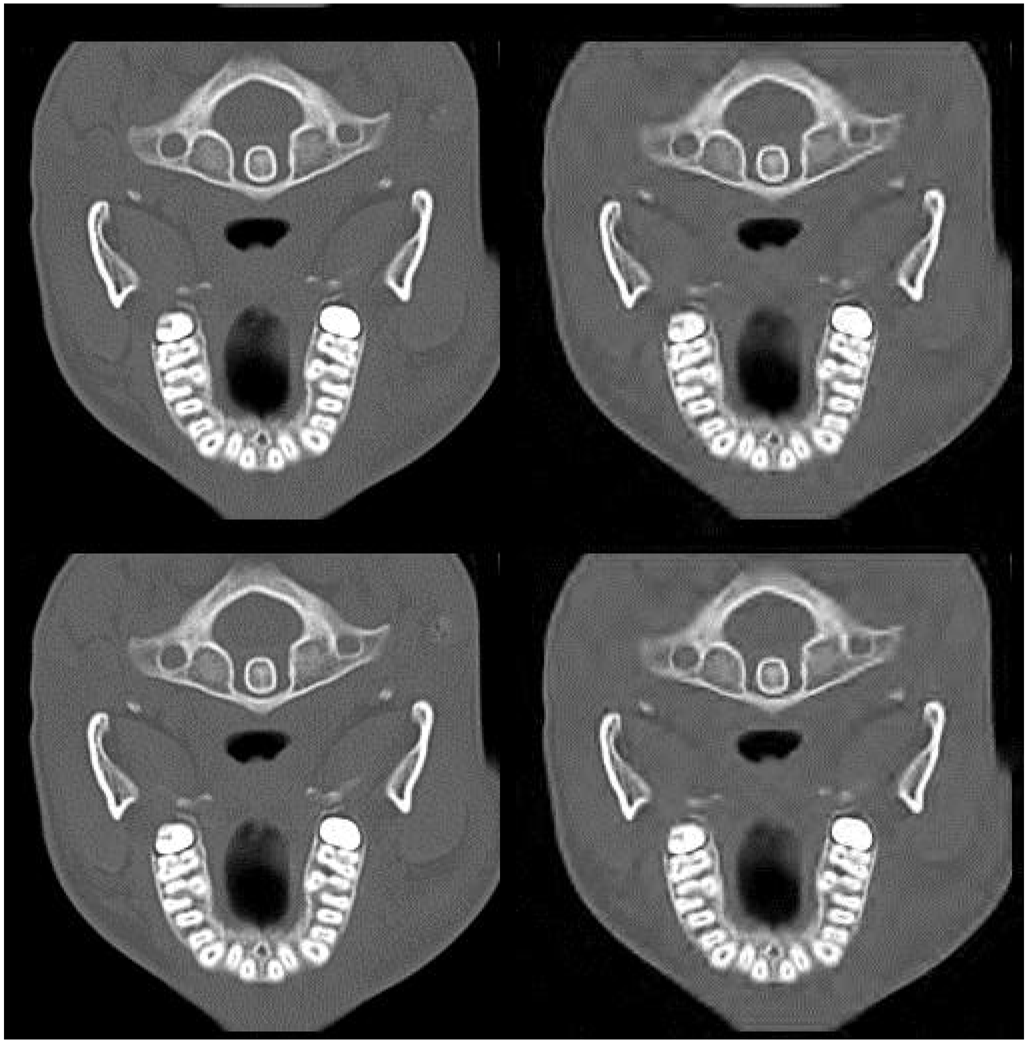
Original



0.1 bpp, 16- slice segments,
no stripes, no overlap

Lossy 3D Transform/2D Speck Coding Results

Original Slice 13



Slice 13,
0.15 bpp

Original Slice 14

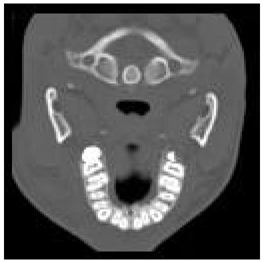
Slice 14,
0.15 bpp

Lossy Decompression at Reduced Resolution

Slice 1 of CT Skull Volume

Original

0.25 bpp

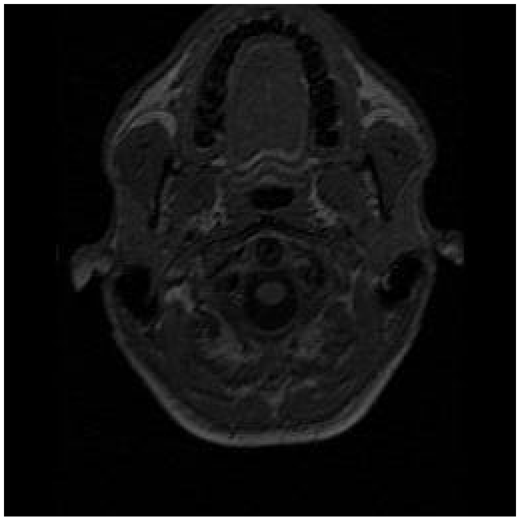


1/2 Resolution from 0.25 bpp file

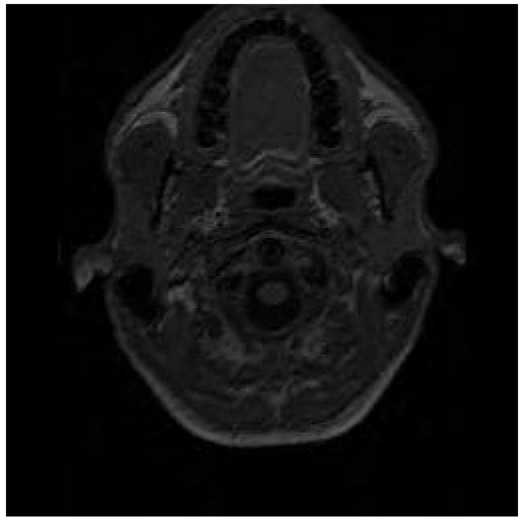
Rensselaer

CNGV

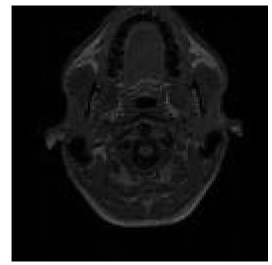
Reduced Resolution Decompression



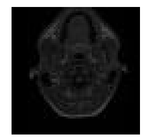
Original 8 bpp,
256x256 MR



0.25 bpp



128x128



64x64

1/2 and 1/4 resolutions
from 0.25 bpp
full resolution file

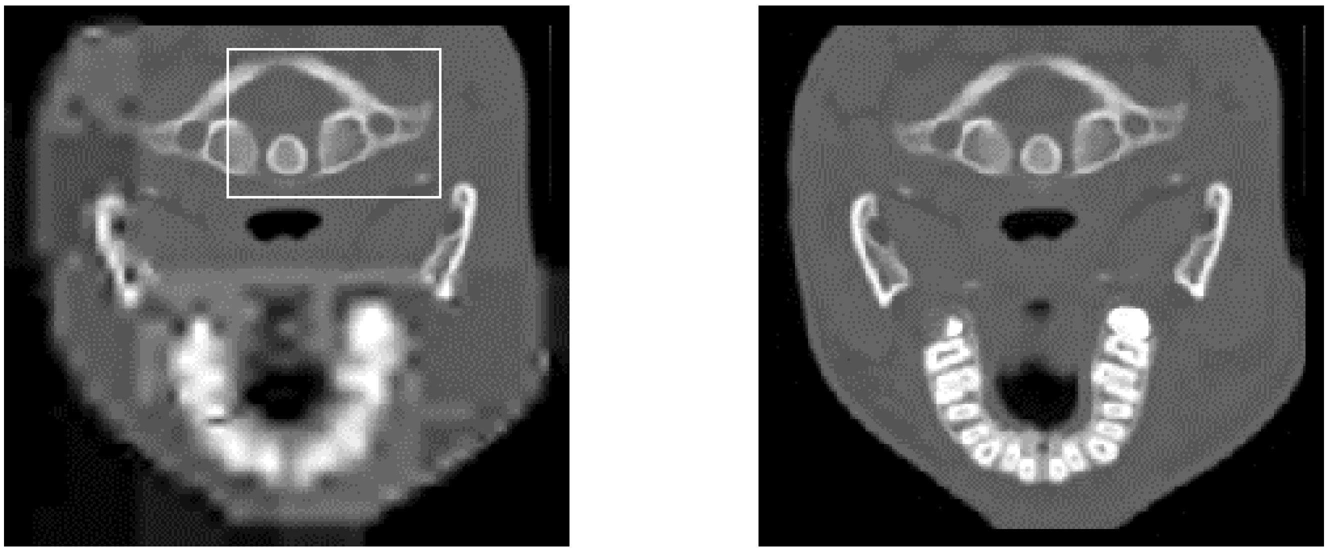
Rensselaer

CNGV

Region of Interest (ROI) Coding

- 3D wavelet transform reorganized into blocks of trees
  - each block corresponds to a cubic region of the volume
  - Select ROI and determine set of contiguous blocks comprising it
- Encode ROI blocks with higher rate than rest
  - bit stream packetized and reorganized to achieve fidelity and/or resolution scalability
  - higher bit planes of ROI sent first to bit stream

Example of ROI Coding

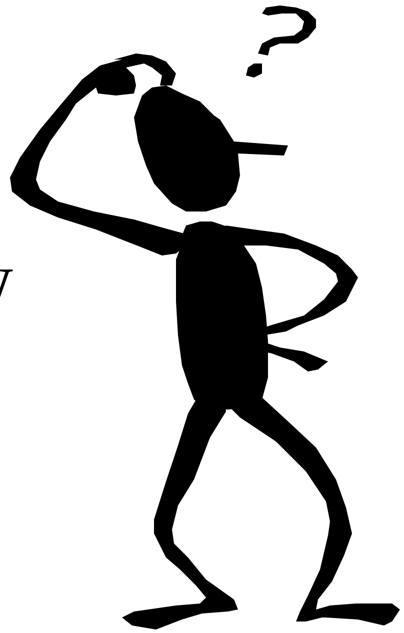


ROI-1 bpp, Background (BD)-0.01 bpp (left) at [110-200, 20-90] for ROI 3-D SPIHT requiring an overall rate of 0.37 bpp and 0.37 bpp (right) of slice 1 of CT skull volume by using traditional 3-D SPIHT

Rensselaer

CNGV

The Challenges



- Retain desirable features, especially low complexity
  - Key is set partitioning
- Produce bit streams resilient to

  channel errors and packet loss
- Discover better low complexity algorithms
  - Need bounded error criterion for data
  - Reconstruction of compressed bit stream for real-time playback of digital cinema
    - D-cinema is focus of a future standard

❖ Homework assignment for next lecture