Rensselaer

CNGV

Wavelet and Set Partition Coding

©William A. Pearlman

Professor and Director,

Center for Next Generation Video

Rensselaer Polytechnic Institute

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

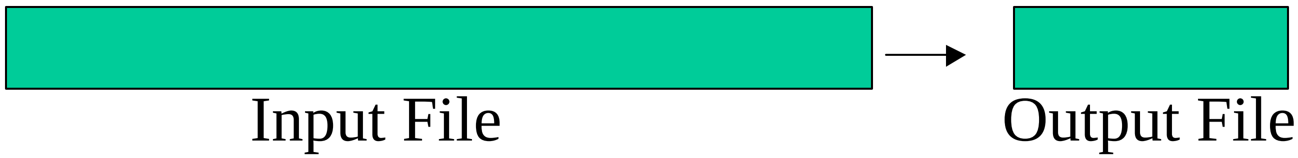
http://www.cipr.rpi.edu/staff/pearlman.html

pearlman@rpi.edu

Outline

- Digital image compression
  - Focus on natural, 8 or more bits per pel sources
- Features expected of modern systems
- Wavelet transform
- Set partition coding methods
- Feature results in video and animation
- Conclusions

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

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

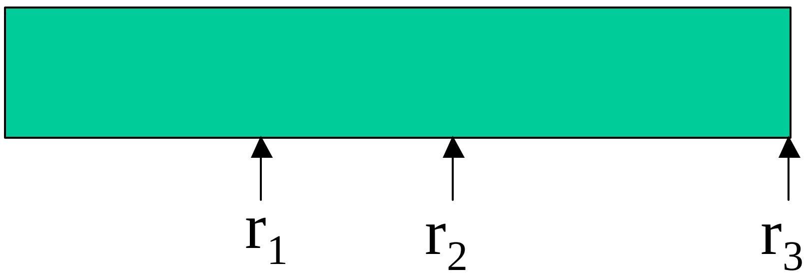
*400 MHz PII, Linux OS

Rensselaer

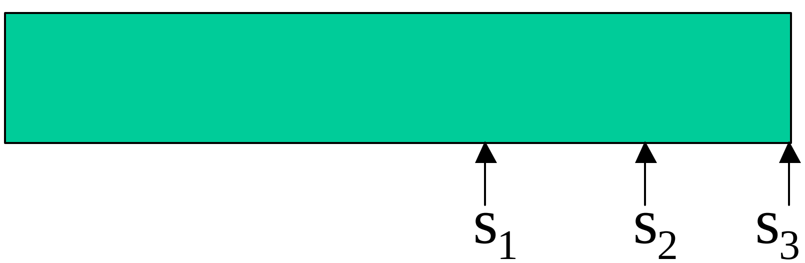
CNGV

Features Requirements: Scalability

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



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



s_2

s_1

Rensselaer

ROI Encoding

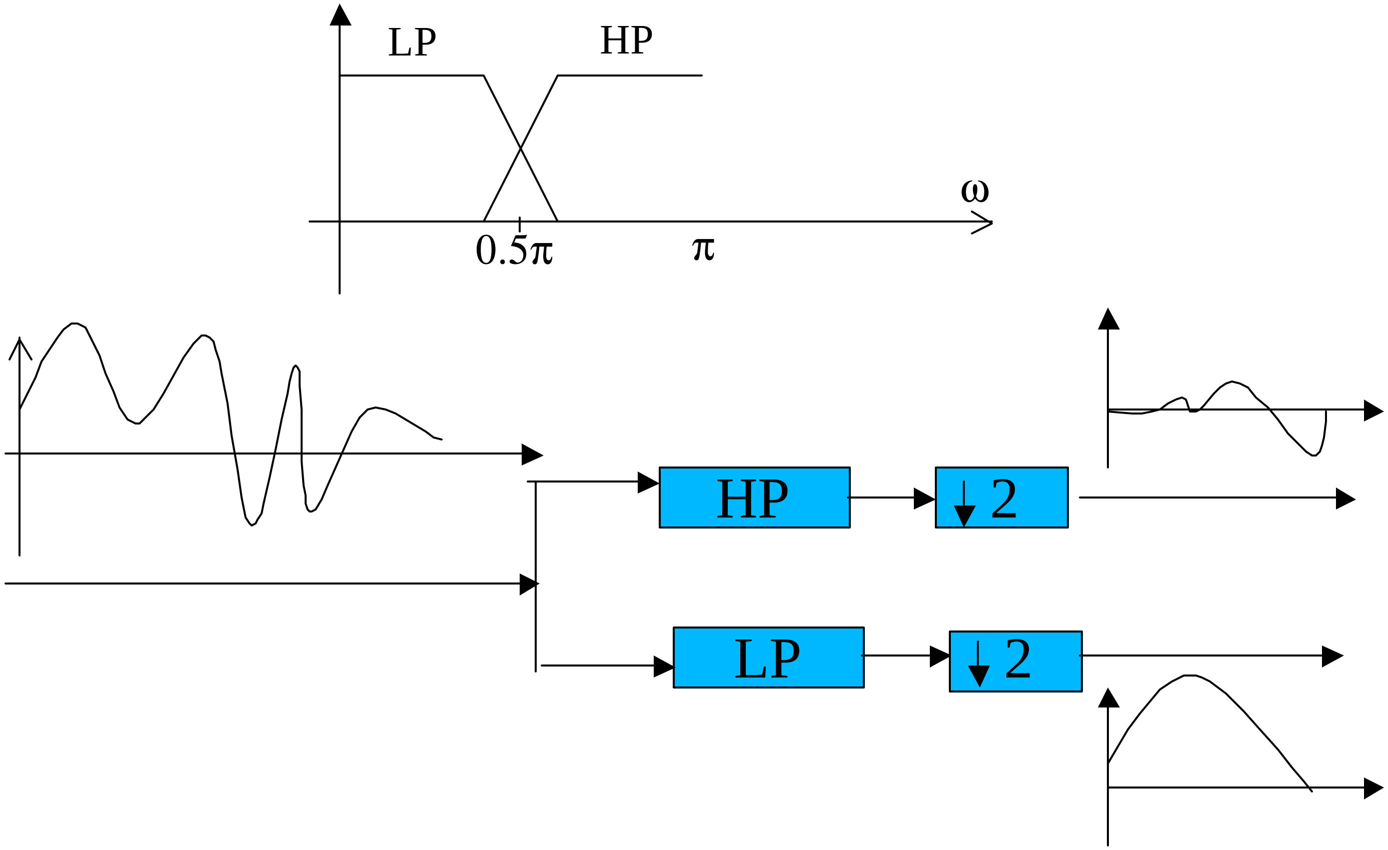
CNGV

60x103 ROI
0.299 bpp

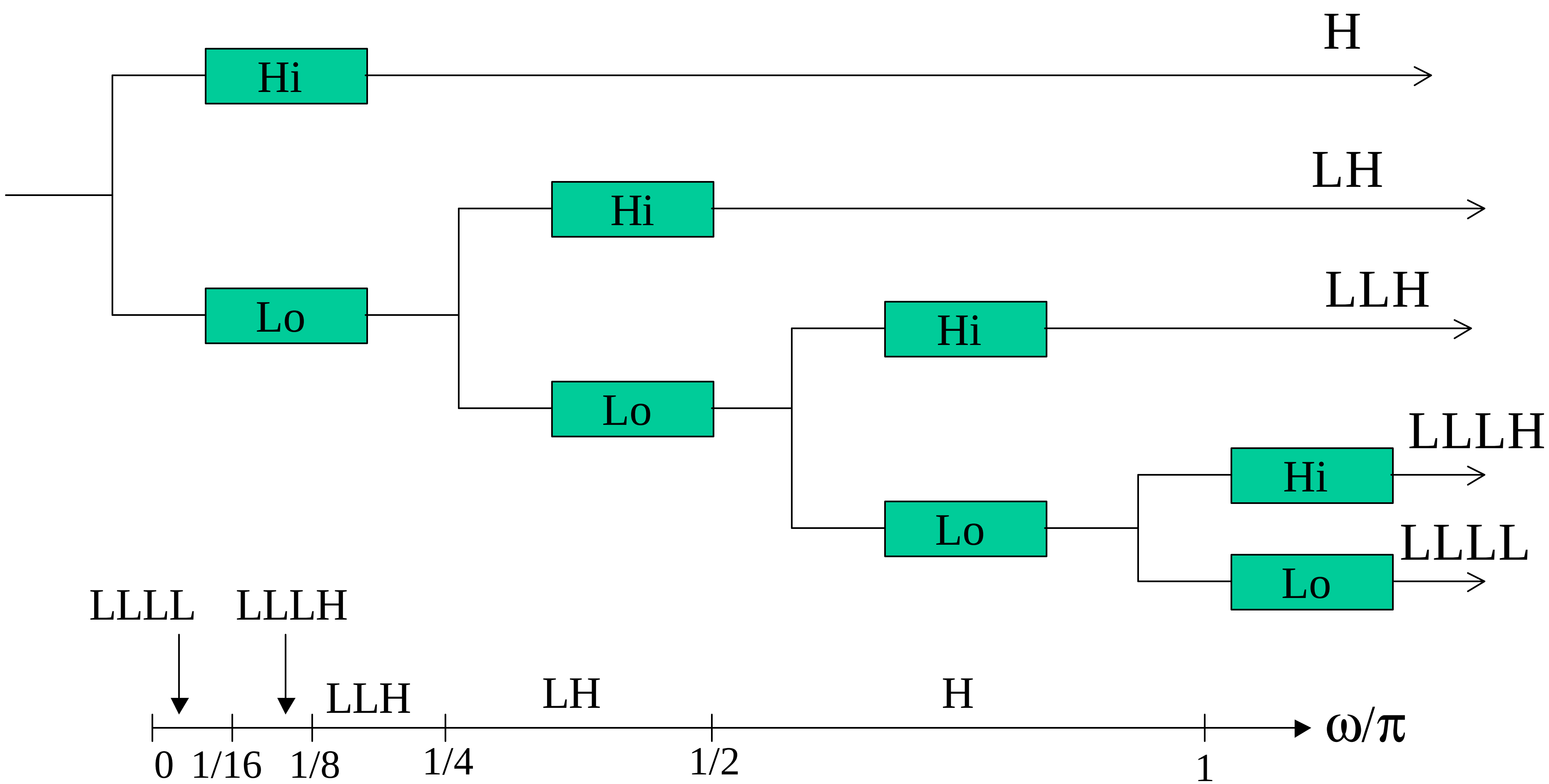
compress 0.41 s
decompress 0.04 s

ROI bits first,
decompressed first

Wavelet Filtering



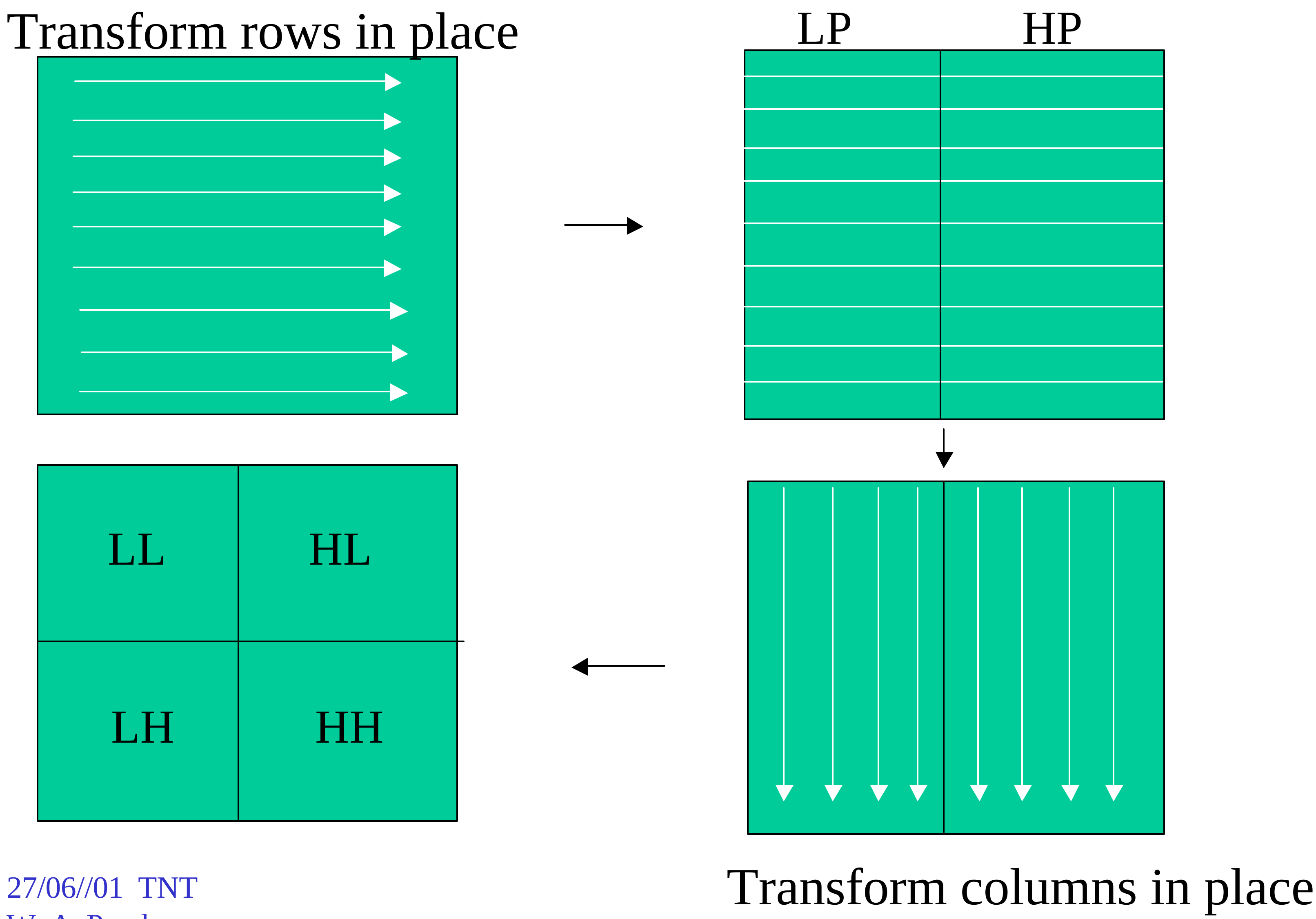
Recursive Low Pass Filtering



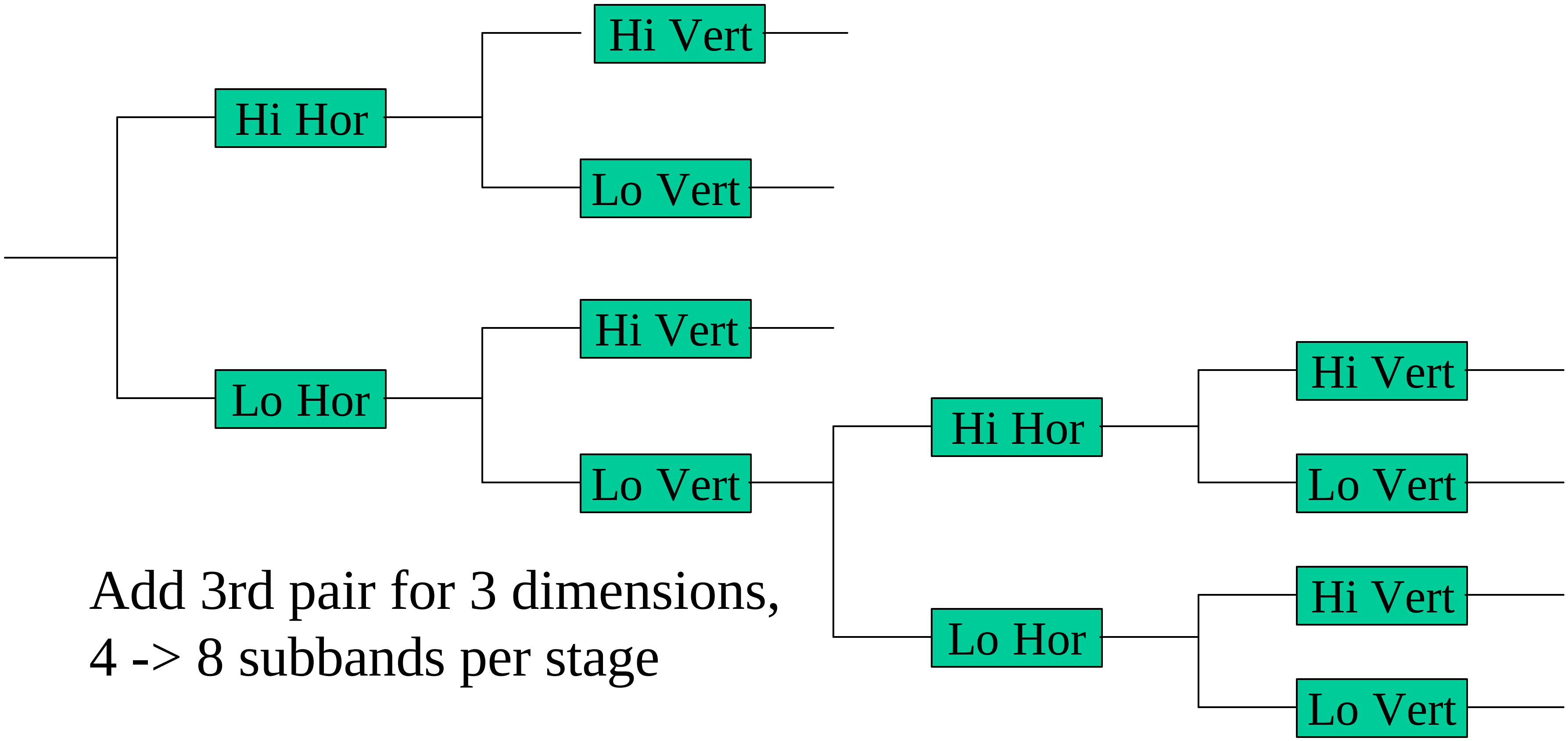
Rensselaer

CNGV

2D Transform



Recursive Hi/Lo 2-Stage Filtering

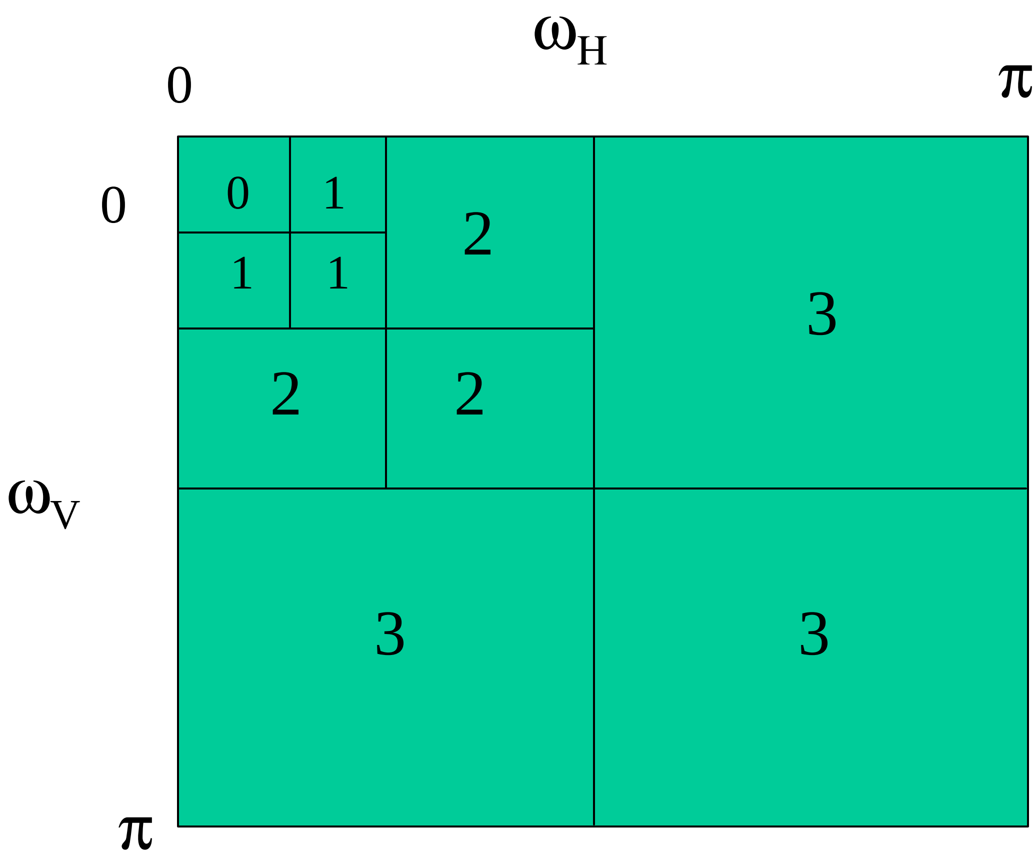


Add 3rd pair for 3 dimensions,
4 -> 8 subbands per stage

Rensselaer

CNGV

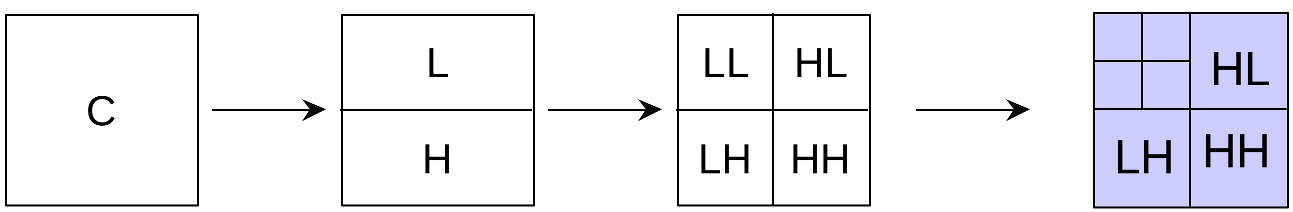
Multi-resolution (Wavelet) Transform



0 -> 1/8 res.
+
1,1,1 -> 1/4 res.
+
2,2,2 -> 1/2 res.
+
3,3,3 -> full res.

Rensselaer

Image Wavelet/Subband Transform

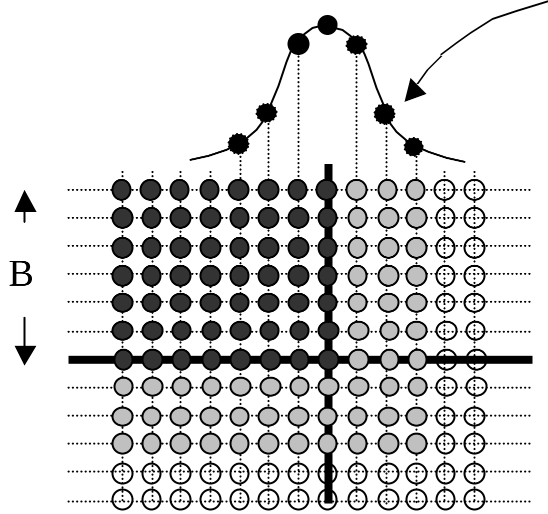


Rensselaer

Block-Based Transform (SSWT)

1 level of decomposition

Filter Coefficients
Filter Length = 7,
Overlap O_1=3



● Block Pixels ● Overlap Pixels ○ Pixels not used in transform

L - filter length
D - decomposition levels

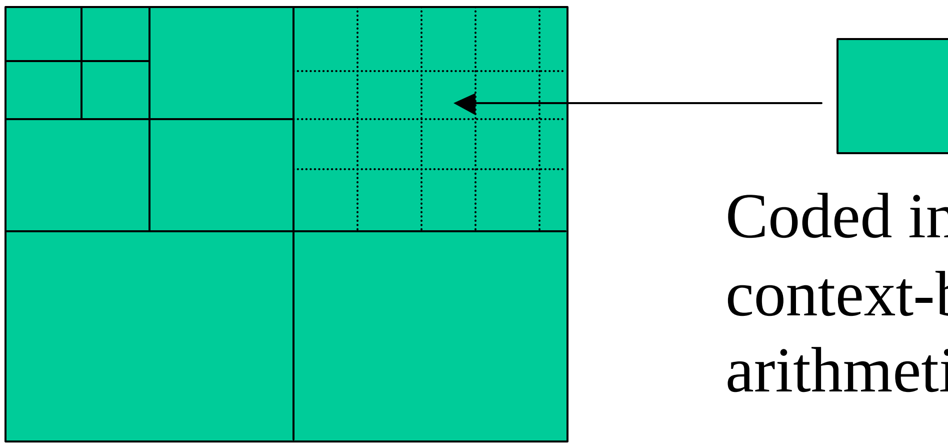
$$O_D = O_1 \bullet (2^D - 1)$$

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

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

JPEG2000 Embedded Coding

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.

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

Rensselaer

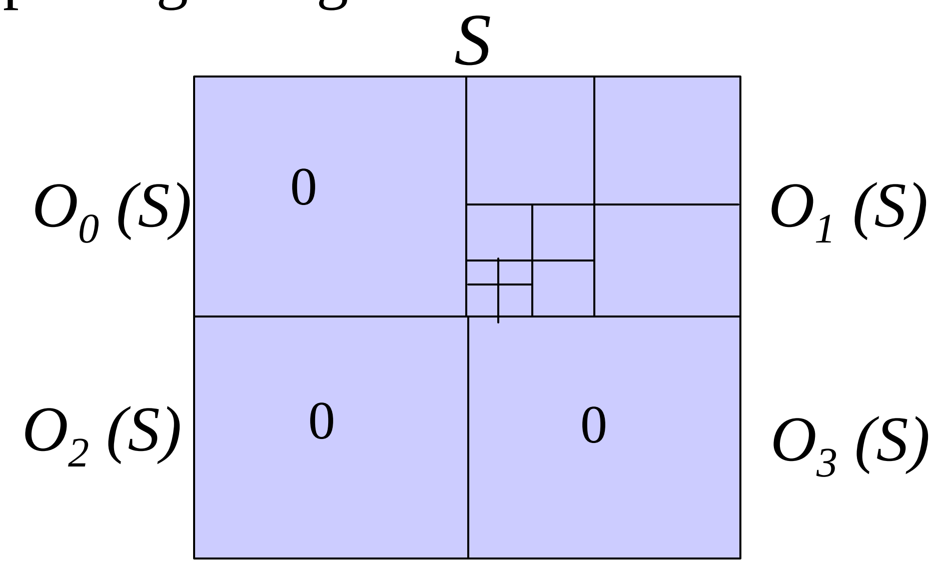
Set Partitioning Rules of SPECK

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.

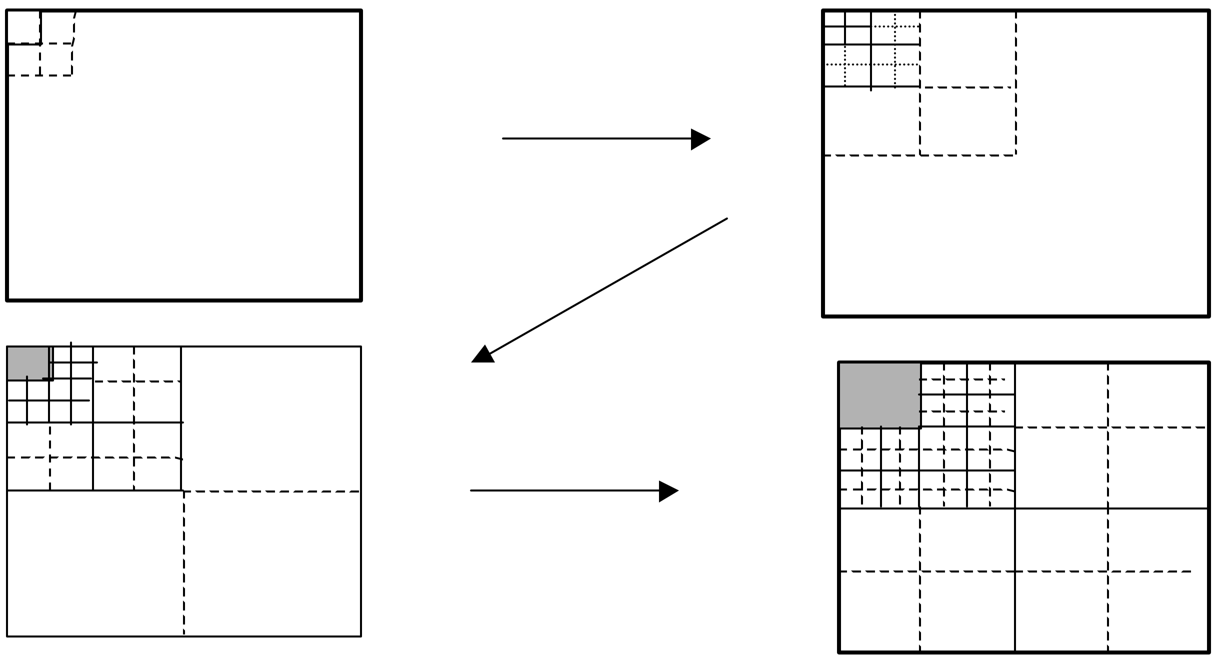
Rensselaer

CNGV

Subband Partitioning Order

Full Transform

- Partitioning proceeds from coarse to fine scale



27/06//01 TNT
W. A. Pearlman

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

Embedding Issues

- Full transform SPECK (and SPIHT) 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*

Rensselaer

CNGV

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

SPIHT Tree Structure

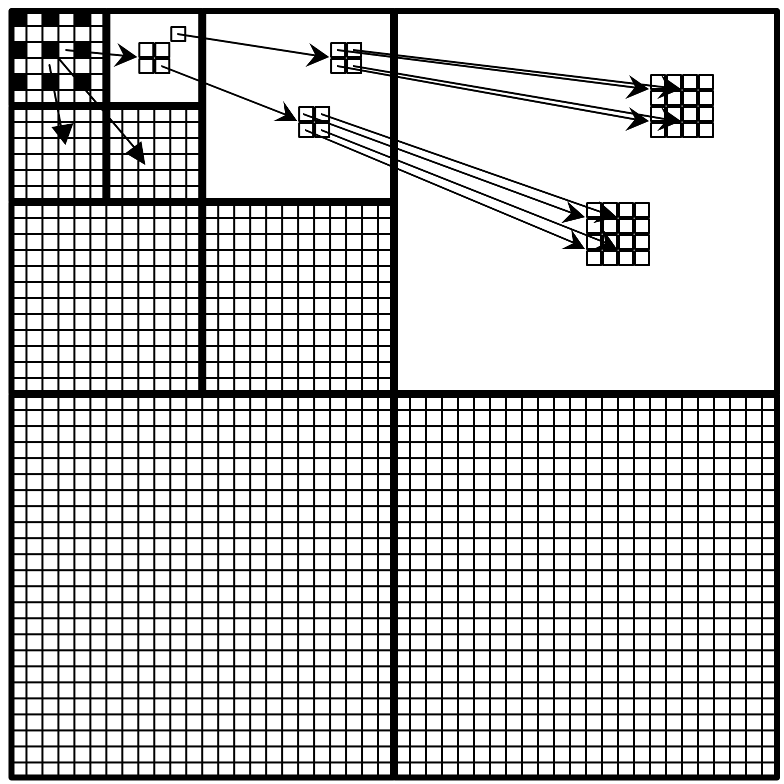
3 level dyadic subband transform shown
Arrows depict parent-child relationships in the SPIHT tree structure
Coefficients are grouped to exploit magnitude dependence
Each subband coefficient has four children
Some coefficients in the DC subband have no children

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)



Coding Sequence

- 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.

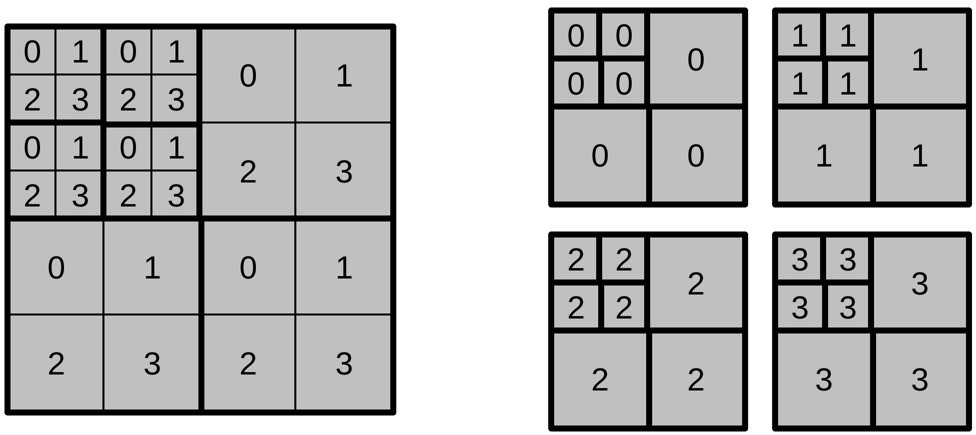
Small Memory Implementations

- 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

Rensselaer

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

Progressive Bit Stream

- 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

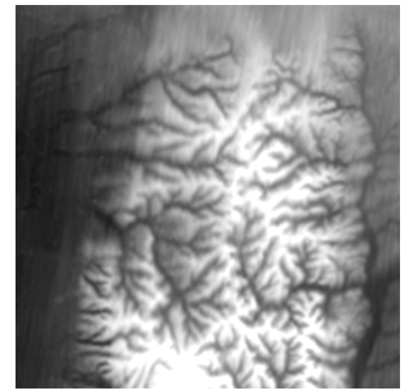
2	data for 2	3	data for 3	3	data for 3	1	data for 1	2	data for 2

Rensselaer

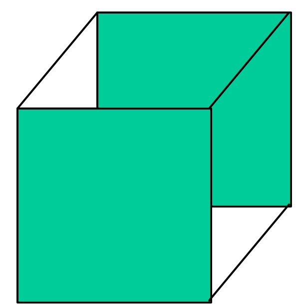
Trends for Image Compression

CNGV

- 2D data: values associated with 2D grid
  - example: terrain elevation data →



- 2D electron microscope images
- Video, Digital Cinema
- 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



Trends for Image Compression (cont.)

- Image Sequence Animation
  - Motion JPEG current workhorse
  - Motion JPEG 2000 (MJP2) emerging
    - Complexity and IPR issues unresolved
  - Motion SPECK worthy candidate for future systems
    - Additional feature of embedded color coding
      - Superior bit allocation among color planes and frames

Rensselaer

Comparison of various color codecs

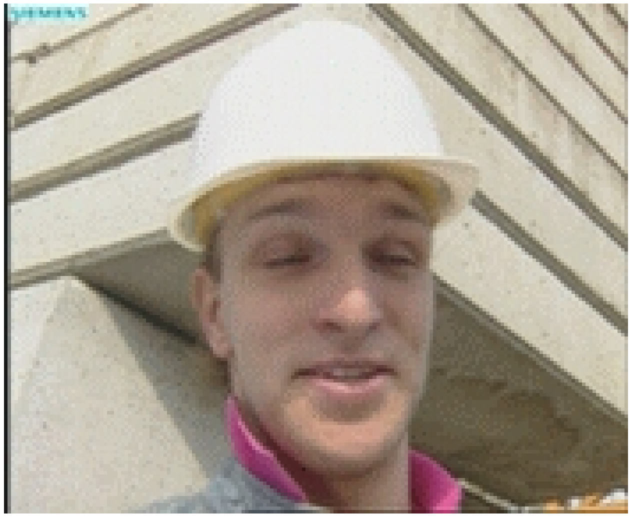
KBits	Y/C	JPEG 2000	SPIHT	CSPECK	MPEG-4
20.9	Y	34.65	35.40	**35.83**	35.09
(Hall)	C	**41.27**	39.73	37.62	39.78
28.5	Y	37.73	38.05	**38.50**	37.82
(Hall)	C	**43.14**	40.71	39.79	41.16
19.3	Y	33.65	34.33	**34.77**	33.95
(foreman)	C	**42.54**	40.32	37.42	40.56
28.7	Y	37.42	37.53	**37.71**	37.11
(foreman)	C	**43.70**	43.04	40.61	42.08

Figures indicate PSNR (in dB)

W. A. Pearlman

Rensselaer

Original Image “Foreman.QCIF”



CSPECK (19.3 Kbits)

SPIHT (19.3 Kbits)

JPEG2000 (19.3 Kbits)

Rensselaer

CNGV

Motion-SPECK vs. Motion-JPEG2000

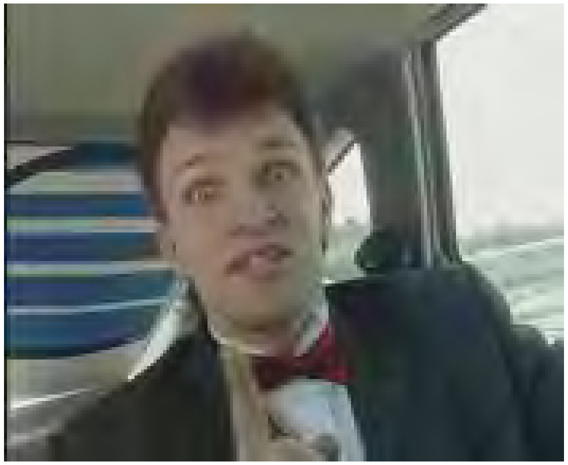
Codec	Mean PSNR in dB			Decoding Times * (sec)
	Y	U	V	
"Carphone"(97 frames) at 0.5 bits/pixel/frame				
Motion-JPEG2000	31.23	38.28	39.17	8.43
Motion-SPECK	32.56	35.97	36.25	8.74
"Trevor"(150 frames) at 0.25 bits/pixel/frame				
Motion-JPEG2000	28.59	36.73	37.28	11.39
Motion-SPECK	30.59	34.26	35.14	12.10

*Decoding time is based on Sun SPARC 10

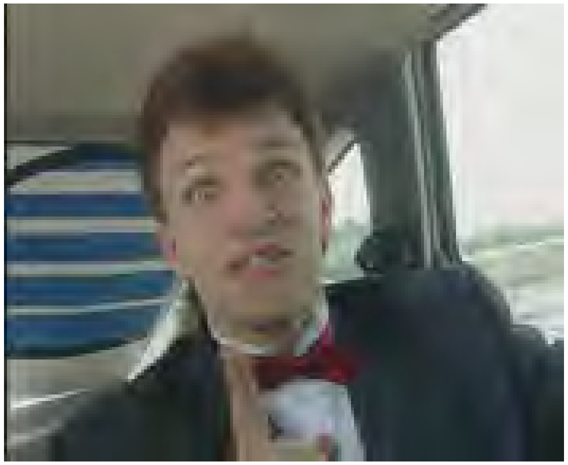
Rensselaer

CNGV

(a)



(b)



(c)

(a) Original image, frame no. 42 of “Carphone” sequence

(b) Motion-SPECK reconstructed (0.5bpp/frame)

(c) Motion-JPEG2000 reconstructed (0.5bpp/frame)

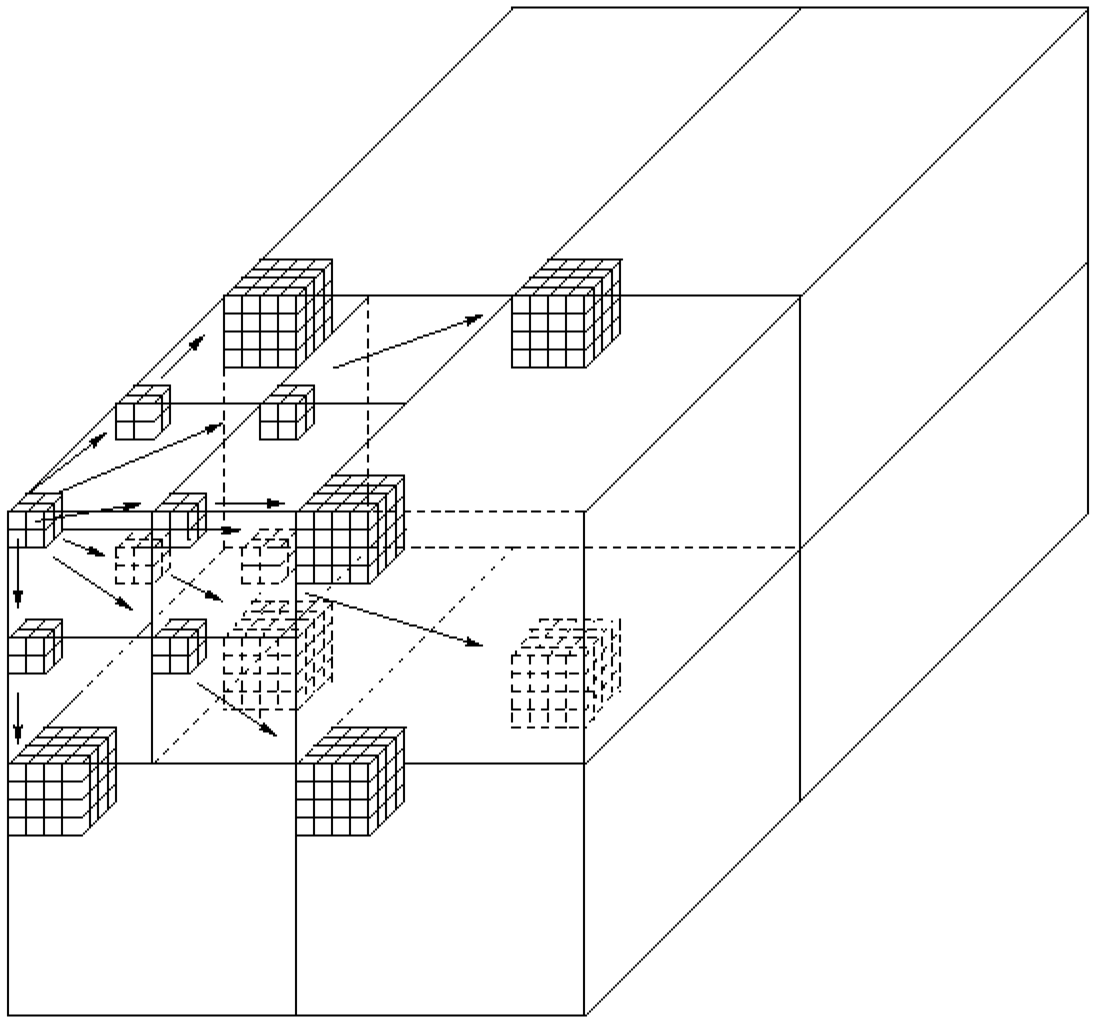
3D Image or Video Coding

- 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

Rensselaer

CNGV

3D SPIHT Tree Branching



Rensselaer Normal, Multiscale, and Error Resilient Coding

Original Frame 15

Normal 3D SPIHT, 2.53 Mbps

ERC SPIHT,
BER = 10^{-2},
2.53 Mbps

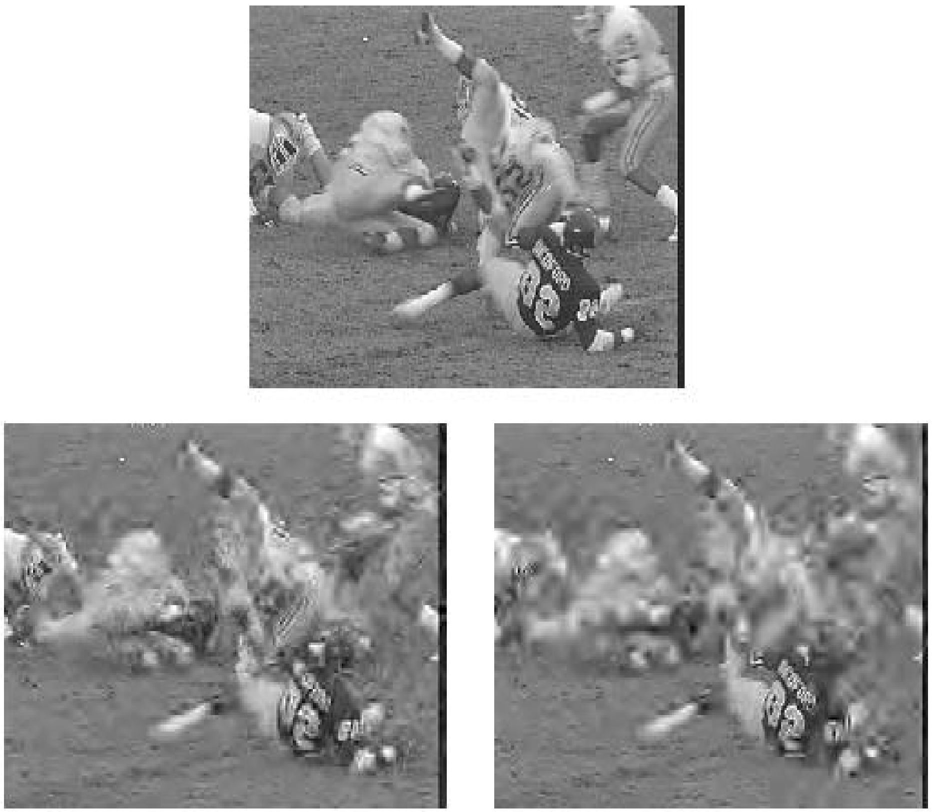
½ scale

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

Rensselaer

ROI Video Coding



55 sub-bitsreams

- 352x240 Football sequence (frame 15)
- (a)Top : Original sequence
- (b) Bottom-left : 3-D SPIHT compressed at 0.0845 bpp
- (c)Bottom-right : region-based STTP-SPIHT result, requiring an overall rate of 0.0845 bpp

Conclusions

- Wavelet transform and set partition coding bring many desirable features:
  - Scalability in resolution and fidelity
  - Region of interest (ROI) fidelity enhancement
  - Low complexity in computation and memory
    - Fast encoding and decoding
- Key to low complexity
  - set partitioning and testing for set significance
    - Adaptive techniques with higher complexity
      - SLCC (Chai et al), PACC (Marpe & Cycon), GTW (Hong & Ladner)
- Other features
  - Error resilient bitstream
  - Parallelizable for embedded hardware applications