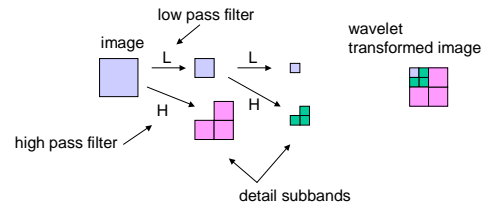


# CSE 490 GZ Introduction to Data Compression Winter 2002

## Wavelet Transform Coding SPIHT

### Wavelet Transform

- Wavelet Transform
  - A family of transformations that filters the data into low resolution data plus detail data.



CSE 490gz - Lecture 15 - Winter 2002

2

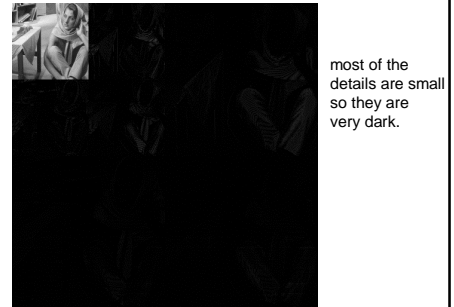
### Wavelet Transformed Barbara (Enhanced)



CSE 490gz - Lecture 15 - Winter 2002

3

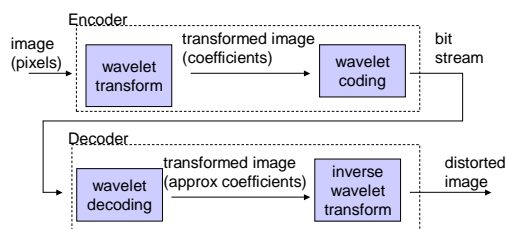
### Wavelet Transformed Barbara (Actual)



CSE 490gz - Lecture 15 - Winter 2002

4

### Wavelet Transform Compression

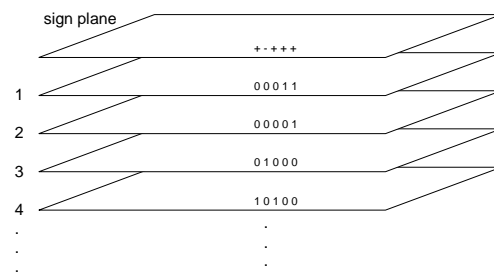


Wavelet coder transmits wavelet transformed image in bit plane order with the most significant bits first.

CSE 490gz - Lecture 15 - Winter 2002

5

### Bit Planes of Coefficients

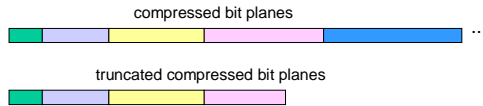


CSE 490gz - Lecture 15 - Winter 2002

6

## Why Wavelet Compression Works

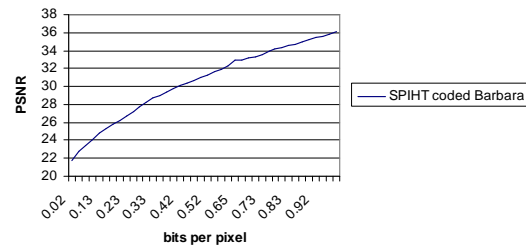
- Wavelet coefficients are transmitted in bit-plane order.
  - In most significant bit planes most coefficients are 0 so they can be coded efficiently.
  - Only some of the bit planes are transmitted. This is where fidelity is lost when compression is gained.
- Natural progressive transmission



CSE 490gz - Lecture 15 - Winter 2002

7

## Rate-Fidelity Curve



More bit planes of the wavelet transformed image that is sent the higher the fidelity.

CSE 490gz - Lecture 15 - Winter 2002

8

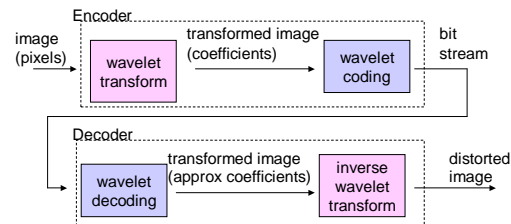
## Wavelet Coding Methods

- EZW** - Shapiro, 1993
  - Embedded Zerotree coding.
- SPIHT** - Said and Pearlman, 1996
  - Set Partitioning in Hierarchical Trees coding. Also uses "zerotrees".
- ECECOW** - Wu, 1997
  - Uses arithmetic coding with context.
- EBCOT** - Taubman, 2000
  - Uses arithmetic coding with different context.
- JPEG 2000** - new standard based largely on EBCOT
- GTW** - Hong, Ladner 2000
  - Uses group testing which is closely related to Golomb codes

CSE 490gz - Lecture 15 - Winter 2002

9

## Wavelet Transform

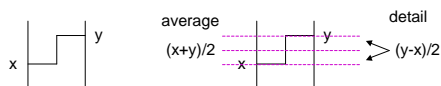


A wavelet transform decomposes the image into a low resolution version and details. The details are typically very small so they can be coded in very few bits.

CSE 490gz - Lecture 15 - Winter 2002

10

## One-Dimensional Average Transform (1)

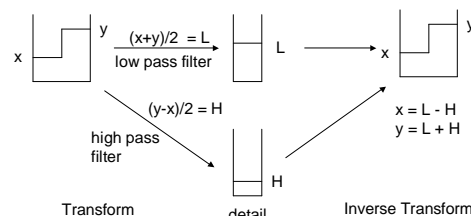


How do we represent two data points at lower resolution?

CSE 490gz - Lecture 15 - Winter 2002

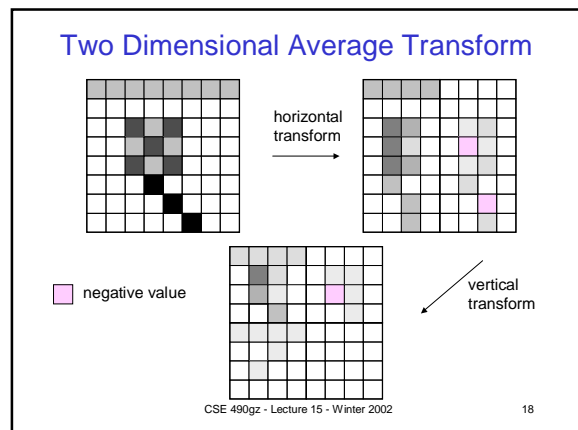
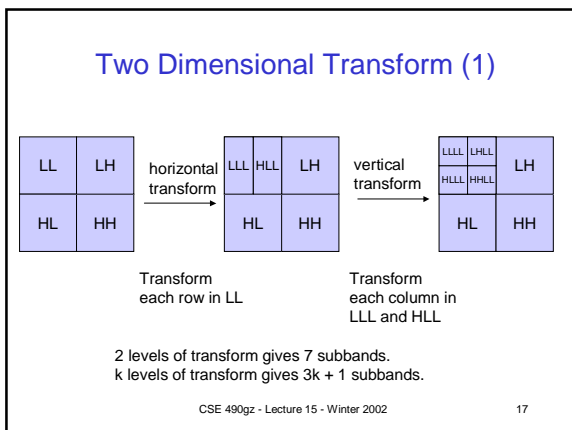
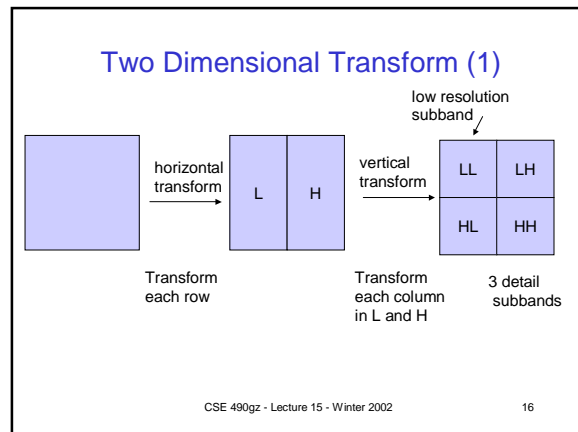
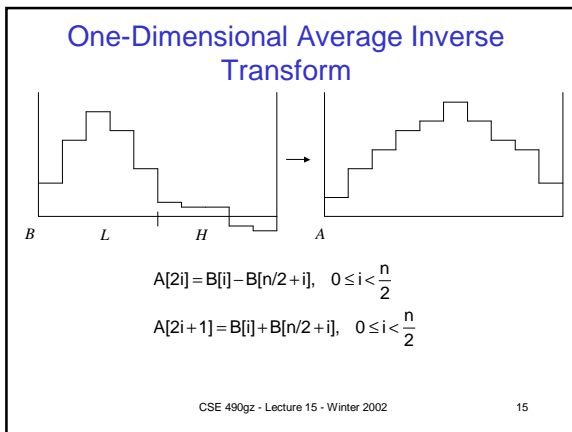
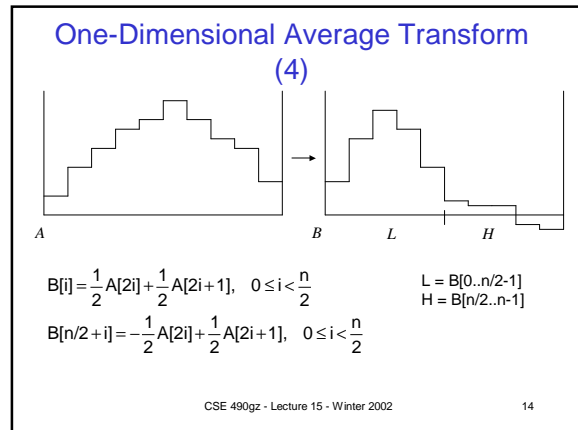
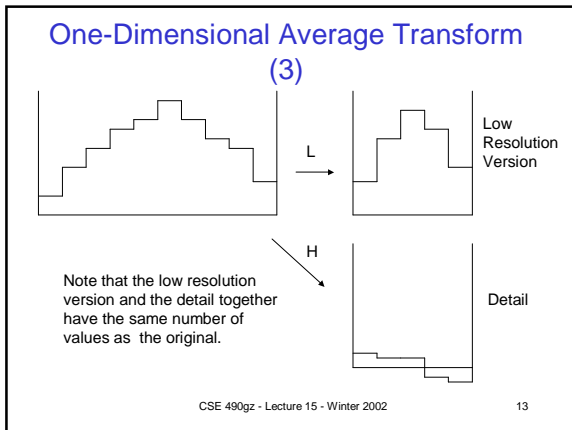
11

## One-Dimensional Average Transform (2)



CSE 490gz - Lecture 15 - Winter 2002

12



## Wavelet Transformed Image



2 levels of wavelet transform

1 low resolution subband

6 detail subbands

CSE 490gz - Lecture 15 - Winter 2002

19

## Wavelet Transform Details

- Conversion to reals.
  - Convert gray scale to floating point.
  - Convert color to Y U V and then convert each to band to floating point. Compress separately.
- After several levels (3-8) of transform we have a matrix of floating point numbers called the **wavelet transformed image (coefficients)**.

CSE 490gz - Lecture 15 - Winter 2002

20

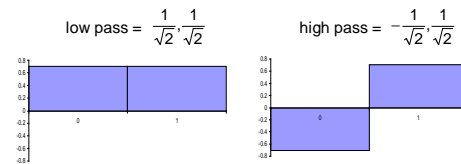
## Wavelet Transforms

- Technically wavelet transforms are special kinds of linear transformations. Easiest to think of them as filters.
  - The filters depend only on a constant number of values. (bounded support)
  - Preserve energy (norm of the pixels = norm of the coefficients)
  - Inverse filters also have bounded support.
- Well-known wavelet transforms
  - Haar – like the average but orthogonal to preserve energy. Not used in practice.
  - Daubechies 9/7 – biorthogonal (inverse is not the transpose). Most commonly used in practice.

CSE 490gz - Lecture 15 - Winter 2002

21

## Haar Filters



$$\text{low pass } B[i] = \frac{1}{\sqrt{2}} A[2i] + \frac{1}{\sqrt{2}} A[2i+1], \quad 0 \leq i < \frac{n}{2}$$

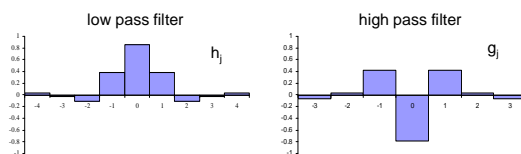
$$\text{high pass } B[n/2+i] = -\frac{1}{\sqrt{2}} A[2i] + \frac{1}{\sqrt{2}} A[2i+1], \quad 0 \leq i < \frac{n}{2}$$

Want the sum of squares of the filter coefficients = 1

CSE 490gz - Lecture 15 - Winter 2002

22

## Daubechies 9/7 Filters



$$\text{low pass } B[i] = \sum_{j=-4}^4 h_j A[2i+j], \quad 0 \leq i < \frac{n}{2}$$

$$\text{high pass } B[n/2+i] = \sum_{j=-3}^3 g_j A[2i+j], \quad 0 \leq i < \frac{n}{2}$$

reflection used near boundaries

CSE 490gz - Lecture 15 - Winter 2002

23

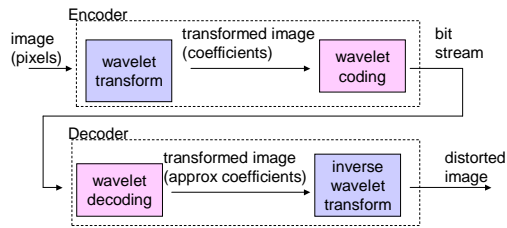
## Linear Time Complexity of 2D Wavelet Transform

- Let  $n$  = number of pixels and let  $b$  be the number of coefficients in the filters.
- One level of transform takes time
  - $O(bn)$
- $k$  levels of transform takes time proportional to
  - $bn + bn/4 + \dots + bn/4^{k-1} < (4/3)bn$ .
- The wavelet transform is linear time when the filters have constant size.
  - The point of wavelets is to use constant size filters unlike many other transforms.

CSE 490gz - Lecture 15 - Winter 2002

24

## Wavelet Transform



Wavelet coder transmits wavelet transformed image in bit plane order with the most significant bits first.

CSE 490gz - Lecture 15 - Winter 2002

25

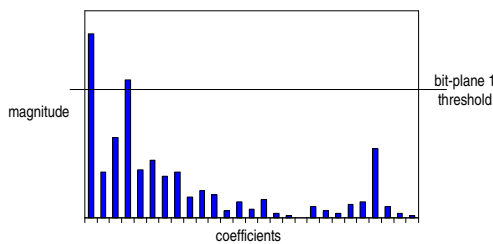
## Wavelet Coding

- Normalize the coefficients to be between  $-1$  and  $1$
- Transmit one bit-plane at a time
- For each bit-plane
  - **Significance pass:** Find the newly significant coefficients, transmit their signs.
  - **Refinement pass:** transmit the bits of the known significant coefficients.

CSE 490gz - Lecture 15 - Winter 2002

26

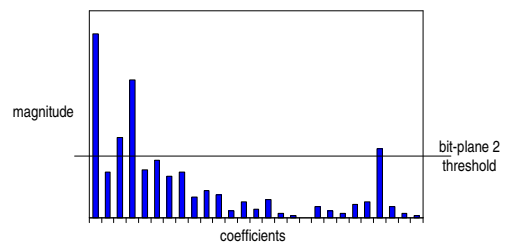
## Significant Coefficients



CSE 490gz - Lecture 15 - Winter 2002

27

## Significant Coefficients



CSE 490gz - Lecture 15 - Winter 2002

28

## Significance & Refinement Passes

- Code a bit-plane in two passes
  - Significance pass
    - codes previously insignificant coefficients
    - also codes sign bit
  - Refinement pass
    - refines values for previously significant coefficients
- Main idea:
  - Significance-pass bits likely to be 0;
  - Refinement-pass bits are not

Coefficient List	
#	value
1	010010010110
2	001011011110
3	000001001001
4	000000010110
5	000100111101
6	000000100101
7	101101110101
8	010010011111
9	001011101101
10	000010100101

refinement bits

Bit-plane 3

CSE 490gz - Lecture 15 - Winter 2002

29

bit plane 1  
bpp  
.0014  
PSNR  
15.3



CSE 490gz - Lecture 15 - Winter 2002

30

bit planes  
1 - 2

bpp  
.0033

PSNR  
16.8



CSE 490gz - Lecture 15 - Winter 2002

31

bit planes  
1 - 3

bpp  
.0072

PSNR  
18.8



CSE 490gz - Lecture 15 - Winter 2002

32

bit planes  
1 - 4

bpp  
.015

ratio  
533 : 1

PSNR  
20.5



CSE 490gz - Lecture 15 - Winter 2002

33

bit planes  
1 - 5

bpp  
.035

ratio  
229 : 1

PSNR  
22.2



CSE 490gz - Lecture 15 - Winter 2002

34

bit planes  
1 - 6

bpp  
.118

ratio  
68 : 1

PSNR  
24.8



CSE 490gz - Lecture 15 - Winter 2002

35

bit planes  
1 - 7

bpp  
.303

ratio  
26 : 1

PSNR  
28.7



CSE 490gz - Lecture 15 - Winter 2002


36

bit planes  
1 – 8

bpp  
.619

ratio  
13 : 1

PSNR  
32.9




CSE 490gz - Lecture 15 - Winter 2002 37

bit planes  
1 – 9

bpp  
1.116

ratio  
7 : 1

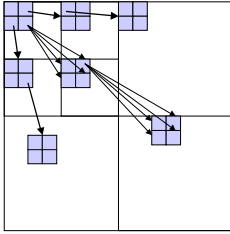
PSNR  
37.5



CSE 490gz - Lecture 15 - Winter 2002 38

### The Zero-Tree Method

- Invented by Shapiro, 1993, and refined by Said and Pearlman, 1996.




If a bit plane value in a low resolution subband is insignificant then it is likely that the corresponding values in higher subbands are also insignificant in the same bit plane.

Such groups of insignificant values are called **zero-trees**.

CSE 490gz - Lecture 15 - Winter 2002 39

### Zero-Tree Example



Values in a zero-tree are correlated.

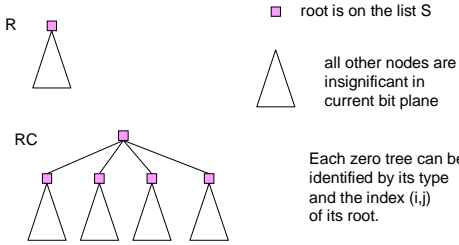
CSE 490gz - Lecture 15 - Winter 2002 40

### Simplified SPIHT Coding

- Runs in passes - one for each bit plane.
- $C[i,j]$  is the coefficient at index  $(i,j)$  and  $C[i,j,k]$  is the  $k$ -th bit of  $C[i,j]$ .
- Encoder maintains two data structures.
  - $S$ , a list of indices  $(i,j)$  such that  $C[i,j]$  is declared significant in the current bit plane.
  - $Z$ , a stack of zero trees of two types.
    - rootless ( $R$ )
    - root-and-childless ( $RC$ )
- The nodes in a zero tree are insignificant in the current bit plane. (ignore root in  $R$  and root and children in  $RC$ )

CSE 490gz - Lecture 15 - Winter 2002 41

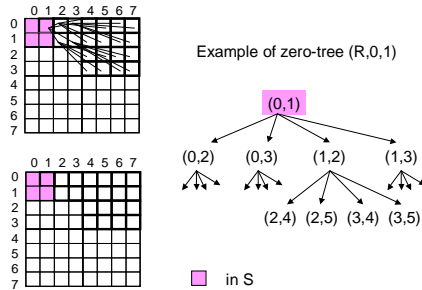
### SPIHT Zero-Trees



Each zero tree can be identified by its type and the index  $(i,j)$  of its root.

CSE 490gz - Lecture 15 - Winter 2002 42

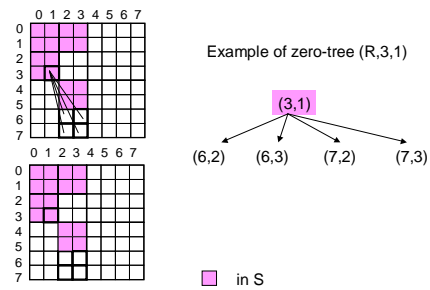
### R-Tree Example (1)



CSE 490gz - Lecture 15 - Winter 2002

43

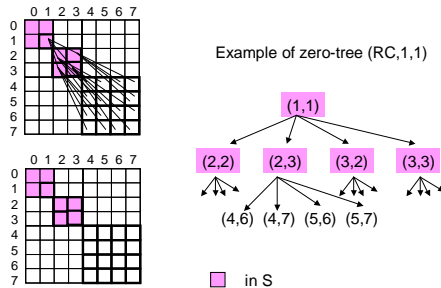
### R-Tree Example (2)



CSE 490gz - Lecture 15 - Winter 2002

44

### RC-Tree Example



CSE 490gz - Lecture 15 - Winter 2002

45

### Initialization of SPIHT

- The lowest subband indices are put into S.
  - If  $(i,j)$  in lowest subband then output sign (0 for - and 1 for +) of  $C[i,j]$  and put  $(i,j)$  into S.
- A stack Z of zero trees is formed using the lowest resolution subband indices as roots.
  - If  $(i,j)$  in the lowest subband is a root of a zero tree of type R if i is odd or (i is even and j is odd).



CSE 490gz - Lecture 15 - Winter 2002

46

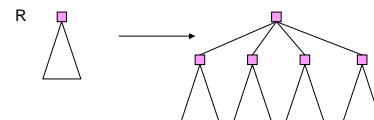
### Iteration of SPIHT Encoder

**k-th iteration**  
 We have list S of significant values and a stack Z of zero trees from the previous pass or the initialization.  
**Significance Pass.**  
 while Z is not empty do  
    $T := \text{pop}(Z)$ ;  
   if T has an index that becomes significant in bit plane k then  
     output 1;  
     decompose(T);  
   else  
     output 0;  
     push T on Z'  
 Z := Z'; (At this point all indices in zero trees in Z are insignificant)  
**Refinement Pass.**  
 for each  $(i,j)$  in S output the k-th significant bit,  $C[i,j,k]$ .

CSE 490gz - Lecture 15 - Winter 2002

47

### Decomposition of R



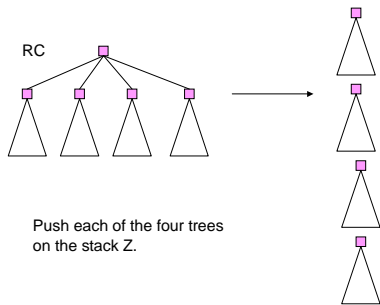
Output the sign (0 for - and 1 for +) of each of the children of the root and put them in S. Push the RC tree on the stack Z. Exception is when tree has no grandchildren. In this case, the tree dies.

CSE 490gz - Lecture 15 - Winter 2002

48



## Decomposition of RC



CSE 490gz - Lecture 15 - Winter 2002

49

## SPIHT Coding Example: Initialization

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

■ in S

Initial data structure:

$S = (0,0), (0,1), (1,0), (1,1)$

$Z = (R,0,1), (R,1,0), (R,1,1)$

Initial output:

0 1 1 1

$\text{sign}(0,0) = -$

$\text{sign}(0,1) = +$

$\text{sign}(1,0) = +$

$\text{sign}(1,1) = +$

CSE 490gz - Lecture 15 - Winter 2002

50

## SPIHT Coding Example: Pass 1, Significance Pass (1)

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1)$

$Z = (R,0,1), (R,1,0), (R,1,1)$

(R,0,1) is significant  
output 1

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3)$   
output 1101 for signs of these

$Z = (RC,0,1), (R,1,0), (R,1,1)$

CSE 490gz - Lecture 15 - Winter 2002

51

## SPIHT Coding Example: Pass 1, Significance Pass (2)

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3)$

$Z = (RC,0,1), (R,1,0), (R,1,1)$

(RC,0,1) is not significant  
output 0

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3)$

$Z = (R,1,0), (R,1,1)$   
 $Z' = (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

52

## SPIHT Coding Example: Pass 1, Significance Pass (3)

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3)$

$Z = (R,1,0), (R,1,1)$   
 $Z' = (RC,0,1)$

(R,1,0) is significant  
output 1

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3),$   
 $(2,0), (2,1), (3,0), (3,1)$   
output 1100 for signs of these  
 $Z = (RC,1,0), (R,1,1)$   
 $Z' = (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

53

## SPIHT Coding Example: Pass 1, Significance Pass (4)

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3),$   
 $(2,0), (2,1), (3,0), (3,1)$

$Z = (RC,1,0), (R,1,1)$   
 $Z' = (RC,0,1)$

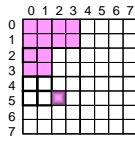
(RC,1,0) is significant  
output 1

$S = (0,0), (0,1), (1,0), (1,1),$   
 $(0,2), (0,3), (1,2), (1,3),$   
 $(2,0), (2,1), (3,0), (3,1),$   
 $(2,2), (2,3), (3,2), (3,3),$   
 $(R,3,1), (R,1,1)$   
 $Z = (R,2,0), (R,2,1), (R,3,0),$   
 $(R,3,1), (R,1,1)$   
 $Z' = (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

54

### SPIHT Coding Example: Pass 1, Significance Pass (5)



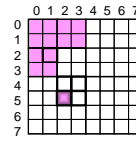
■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1)$   
 $Z = (R,2,0), (R,2,1), (R,3,0), (R,3,1), (R,1,1)$   
 $Z' = (RC,0,1)$   
 $(R,2,0)$  is not significant  
 output 0  
 $S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1)$   
 $Z = (R,2,1), (R,3,0), (R,3,1), (R,1,1)$   
 $Z' = (R,2,0), (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

55

### SPIHT Coding Example: Pass 1, Significance Pass (6)



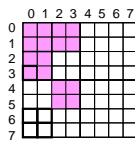
■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1)$   
 $Z = (R,2,1), (R,3,0), (R,3,1), (R,1,1)$   
 $Z' = (R,2,0), (RC,0,1)$   
 $(R,2,1)$  is significant  
 output 1  
 $S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 output 1010 for signs of these  
 $Z = (R,3,0), (R,3,1), (R,1,1)$   
 $Z' = (R,2,0), (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

56

### SPIHT Coding Example: Pass 1, Significance Pass (7)



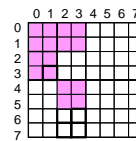
■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,3,0), (R,3,1), (R,1,1)$   
 $Z' = (R,2,0), (RC,0,1)$   
 $(R,3,0)$  is insignificant  
 output 0  
 $S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,3,1), (R,1,1)$   
 $Z' = (R,3,0), (R,2,0), (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

57

### SPIHT Coding Example: Pass 1, Significance Pass (8)



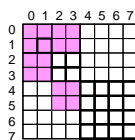
■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,3,1), (R,1,1)$   
 $Z' = (R,3,0), (R,2,0), (RC,0,1)$   
 $(R,3,1)$  is insignificant  
 output 0  
 $S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,1,1)$   
 $Z' = (R,3,1), (R,3,0), (R,2,0), (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

58

### SPIHT Coding Example: Pass 1, Significance Pass (9)



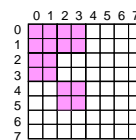
■ became significant  
■ in S

$S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,1,1)$   
 $Z' = (R,3,1), (R,3,0), (R,2,0), (RC,0,1)$   
 $(R,1,1)$  is insignificant  
 output 0  
 $S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,1,1), (R,3,1), (R,3,0), (R,2,0), (RC,0,1)$

CSE 490gz - Lecture 15 - Winter 2002

59

### SPIHT Coding Example: Pass 1, Refinement Step



■ in S

$S = (0,0), (0,1), (1,0), (1,1), (0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1), (4,2), (4,3), (5,2), (5,3)$   
 $Z = (R,1,1), (R,3,1), (R,3,0), (R,2,0), (RC,0,1)$   
 output 1011000100000010  
 one bit for each member of S.  
 37 total bits in pass 1 were output.  
 Initialization was 4 bits.  
 Total of 41 bits to send 64 bits plus 16 sign bits.

CSE 490gz - Lecture 15 - Winter 2002

60

## SPIHT Decoding

- The decoder emulates the encoder.
  - The decoder maintains exactly the same data structures as the encoder.
  - When the decoder has popped the Z stack to examine a zero tree it receives a bit telling it whether the tree is significant. The decoder can then do the right thing.
    - If it is significant then it does the decomposition.
    - If it is not significant then it deduces a number of zeros in the current bit plane.

CSE 490gz - Lecture 15 - Winter 2002

61

## SPIHT Decoder

### k-th iteration

We have list S of significant values and a stack Z of zero trees from the previous pass or the initialization.

### Significance Pass.

while Z is not empty do

  T := pop(Z);

  input := read;

  if input = 1 then decompose(T);

  else push T on Z'

  Z := Z'; {At this point all indices in zero trees in Z are insignificant}

### Refinement Pass.

for each (i,j) in S do C[i,j,k] := read.

In decompose the signs of coefficients are input

CSE 490gz - Lecture 15 - Winter 2002

62

## Notes on SPIHT

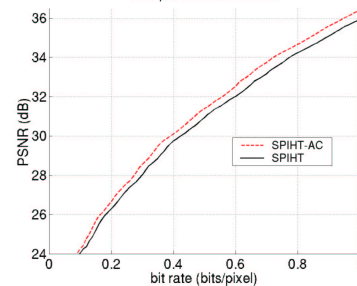
- SPIHT was very influential
  - People really came to believe that wavelet compression can really be practical (fast and effective).
- To yield the best compression an arithmetic coding step is added to SPIHT
  - The improvement is about .5 DB

CSE 490gz - Lecture 15 - Winter 2002

63

## SPIHT-AC

Compression of Barbara



CSE 490gz - Lecture 15 - Winter 2002

64