Low Bit-rate Video Coding with 3D Lower Trees (3D-LTW)

Otoniel López[1]
Miguel Martinez-Rach[1]
Pablo Piñol[1]
José Oliver[2]
Manuel Perez Malumbres[1]

[1]Universidad Miguel Hernández
[2]Universidad Politécnica de Valencia

Universitas Miguel Hernández

INDEX

- 3D-DWT Frame-by-Frame
- LTW
  - Extension to 3D Video
- Results
- Conclusions

UNIVERSITAS Miguel Hernández

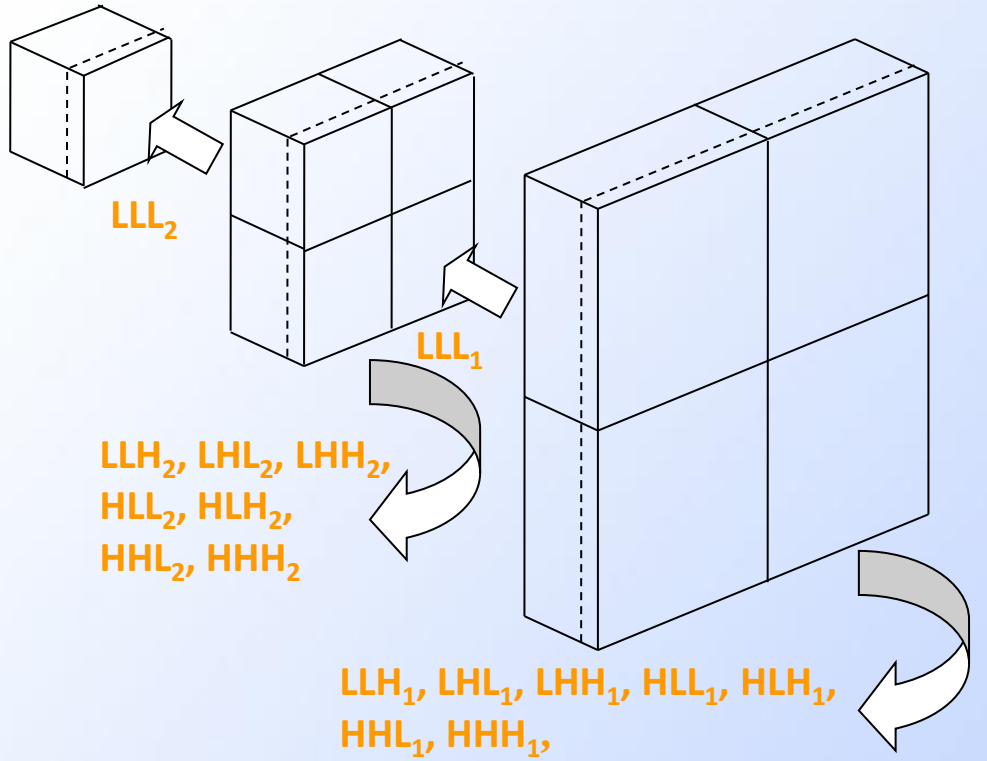
3D-DWT Frame-by-Frame

3D-DWT

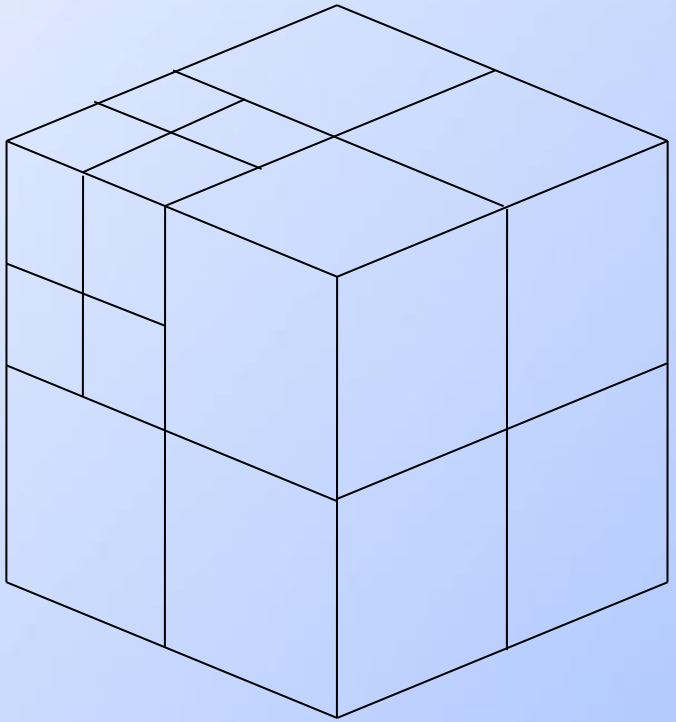
- 3D-DWT coding drawbacks:
  - Extremely high memory consumption with the regular algorithm. (GOPs)
  - Boundary effects between GOPs

Frame-by-Frame scheme, based on the 2D-DWT Line-based approach

3D-DWT



frame-by-frame 3D-DWT



Regular 3D-DWT

Frames are continuously input with no need to divide the video in GOPs

3D-DWT

function GetLLLFrame (*level*)

1) First Base Case :

If *No more frames to read at this level* **return EOF**

2) Second Base Case:

IF level = 0 return InputFrame( )

3.1) Recursively fill or update the buffer for this level

IF buffer **EMPTY Fill up** calling**:**

2DFWT(GetLLLFrame (*level−1)*)

ELSE Update buffer calling:

Shift(buffer)

2DFWT(GetLLLFrame (*level−1)*)

3.2) Calculate the WT time direction

{LLL, LLH, LHL, LHH} = Z-axis_FWT_LowPass($buffer_{level}$)

{HLL, HLH, HHL, HHH} = Z-axis_FWT_HighPass($buffer_{level}$)

3.3) Process SubFrames

ProcessSubFrames (LLH, LHL, LHH, HLL, HLH, HHL, HHH)

return LLL

function LowMemUsage3D_FWT(*nlevel)*

REPEAT

LLL = GetLLLFrame(*nlevel*)

if (*LLL!=EOF*)

ProcessSubFrame(LLL)

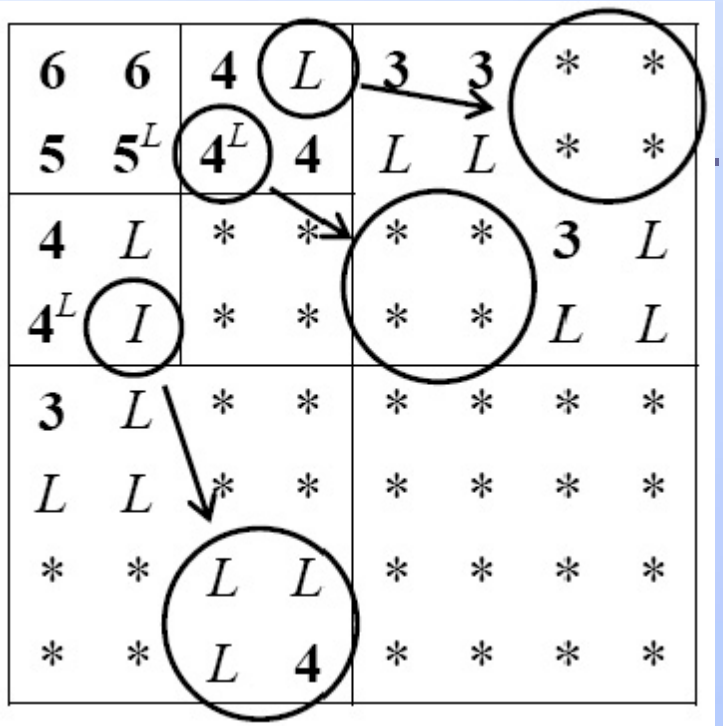
UNTIL *LLL=EOF*

LTW encoder

LTW encoder

- Two strategies for Quantization:
  - Finer: Scalar uniform Quantization (***Q***)
  - Coarser: Removing least significant bit planes (***rplanes***).
- Tree structure called "Lower tree"
  - Reduces redundancy among sub-bands
  - Fast way of grouping coefficients.

51	42	-9	2	4	4	0	-1
25	17	10	11	3	1	0	2
12	3	3	-2	2	-2	-5	3
-9	-3	3	-3	0	3	-1	2
-4	1	1	-2	0	2	1	3
2	-3	0	2	1	-1	-1	-2
1	3	2	1	1	2	-3	1
-2	-3	3	-12	2	0	2	1

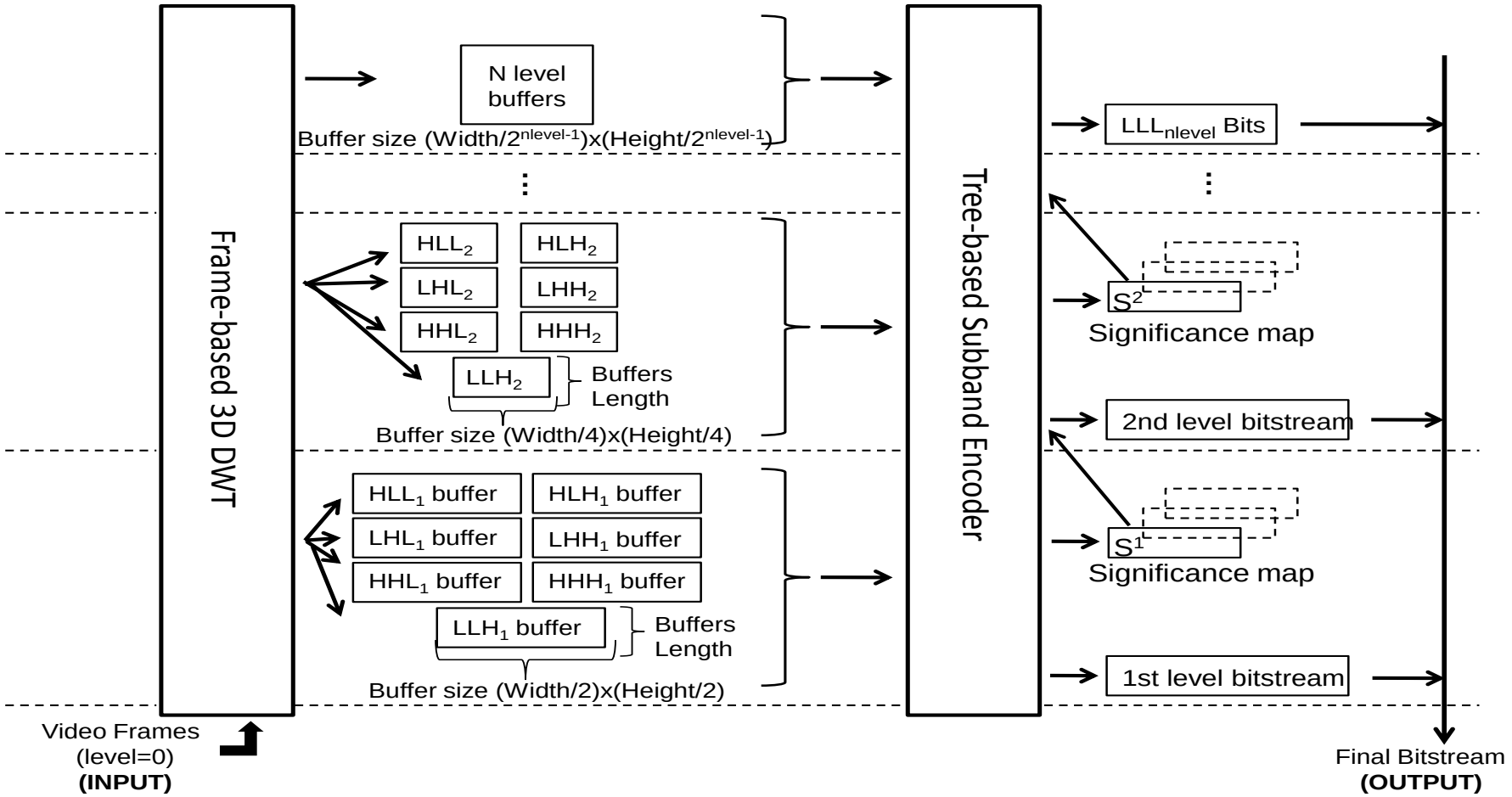


- Two stage Algorithm:
  - First: Create symbol Map from leaves. (classic quad-trees)
  - Second: Subbands encoded from LL_n to first-level wavelet sub-bands.
    - Resolution scalability
- In each sub-band, 2x2 blocks are entropy encoded → arithmetic encoder
- Significant coefficients
  - Significant bits and sign are raw encoded.

UNIVERSITAS Miguel Hernández

LTW encoder

3D Video extension

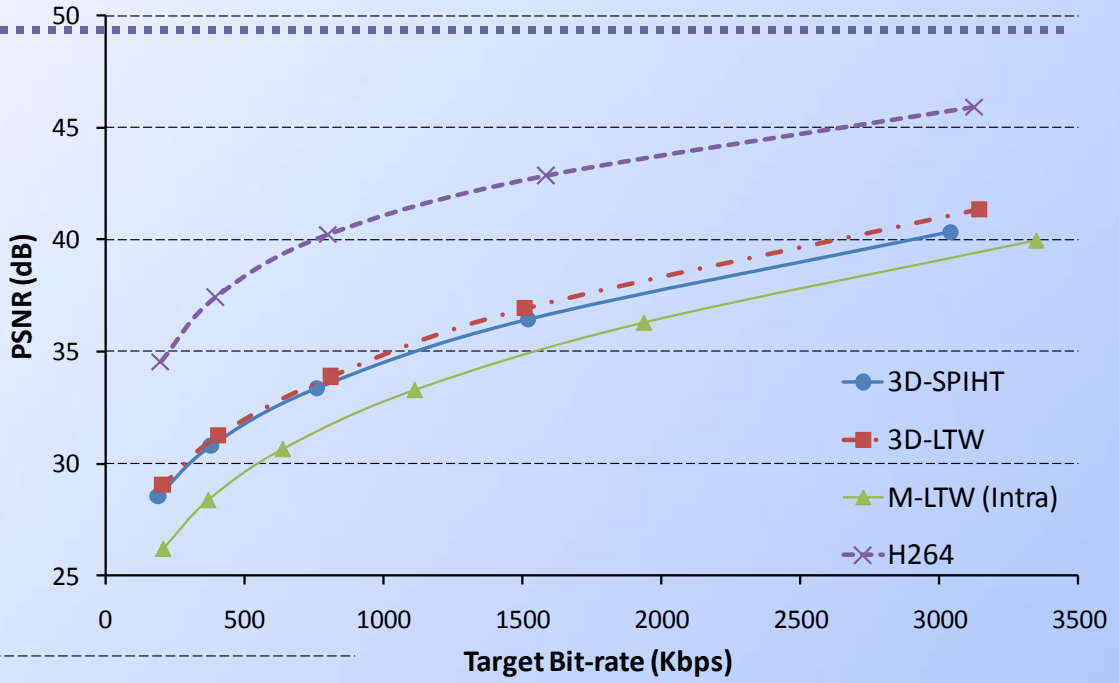


- 2 frames per subband type at each level.
- We need binary significance map for both 2x2 blocks of each subband type.

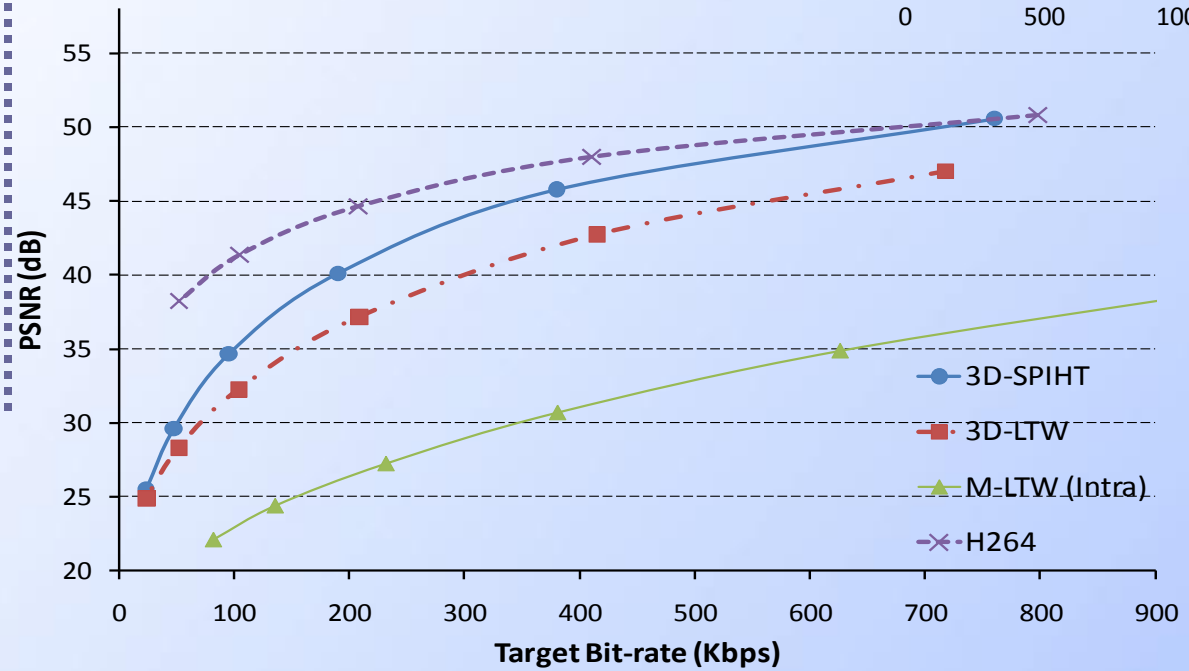
Results

Results: R/D

Up to 11 dB compared to M-LTW (Intra)

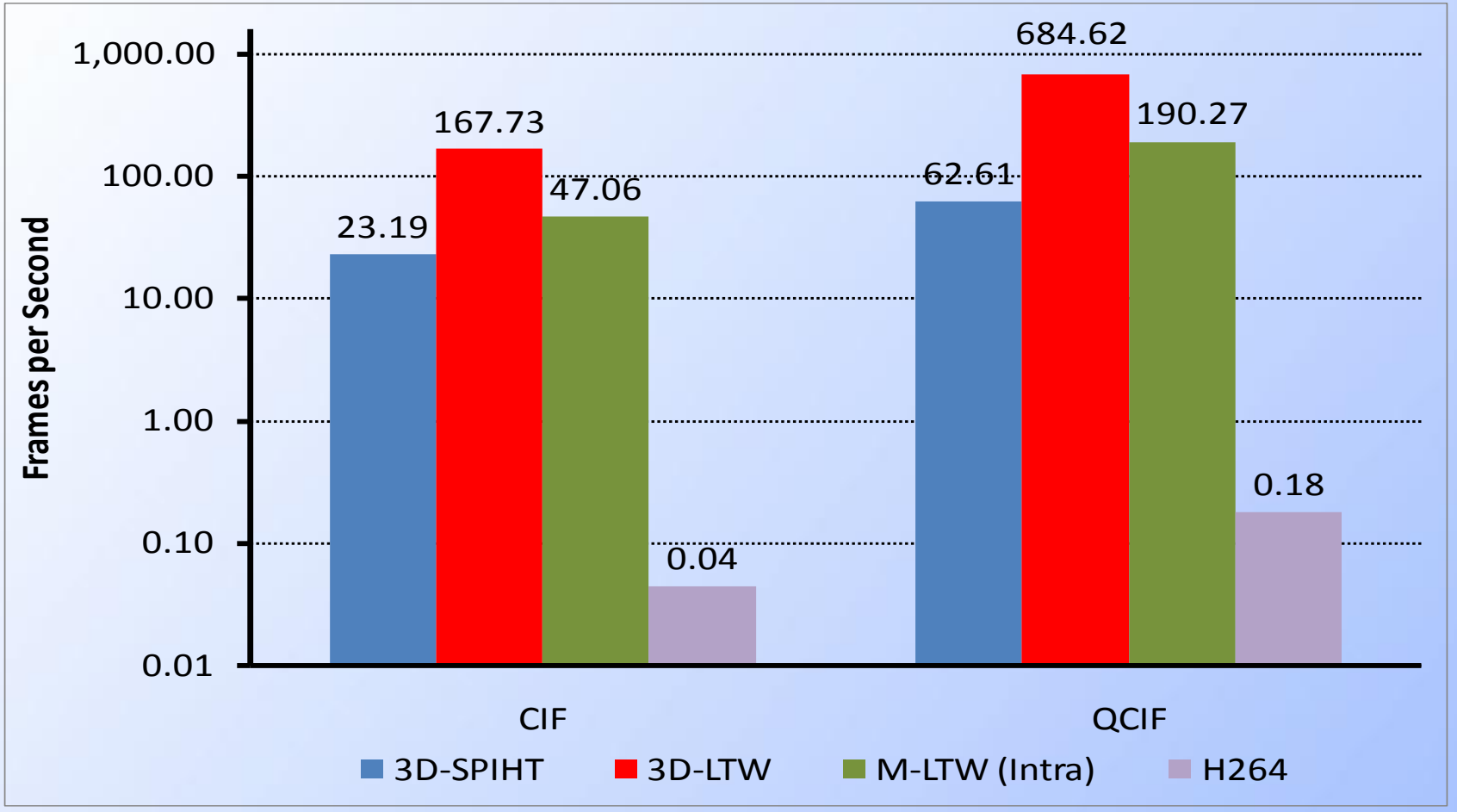


Foreman (CIF)



Container (QCIF)

Results: Coding Delay



Up to 10 times as fast as 3D-SPIHT

Results: Memory(KB)

Codec/ Format	H.264	3D-SPIHT	3D-LTW	M-LTW
QCIF	35824	10152	4008	1104
CIF	86272	34504	10644	1540

- Up to 3.5 times less memory than 3D-SPIHT
- Up to 10 times less memory than H.264

Conclusions

Conclusions

- Low memory demanding 3D-DWT based encoder
  - Reduces memory requirements
- 3D-LTW :
  - 10 times as fast as 3D-SPIHT
  - R/D improvement up to 11 dB compared to M-LTW (intra)
  - Similar R/D behavior than 3D-SPIHT

Thank you

otoniel@umh.es

Thanks to Spanish Ministry of Science and Innovation under grant TIN2009-05737-E for funding

Computer Architecture Group

http://atc.umh.es