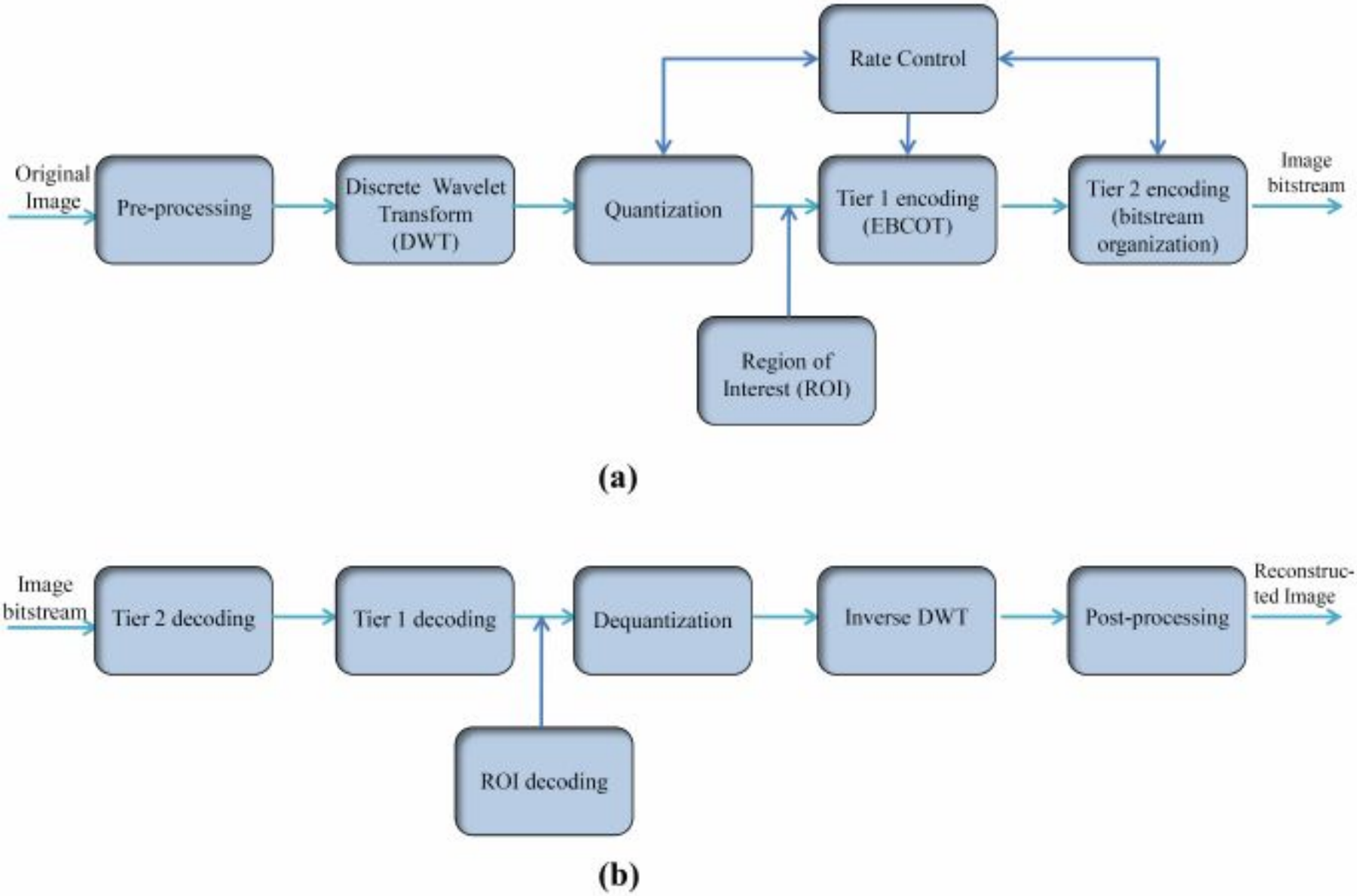
JPEG2000 Overview

- International image compression standard
- Uses wavelet transform instead of DCT
- Supports lossy and lossless compression
- Better quality at high compression ratios



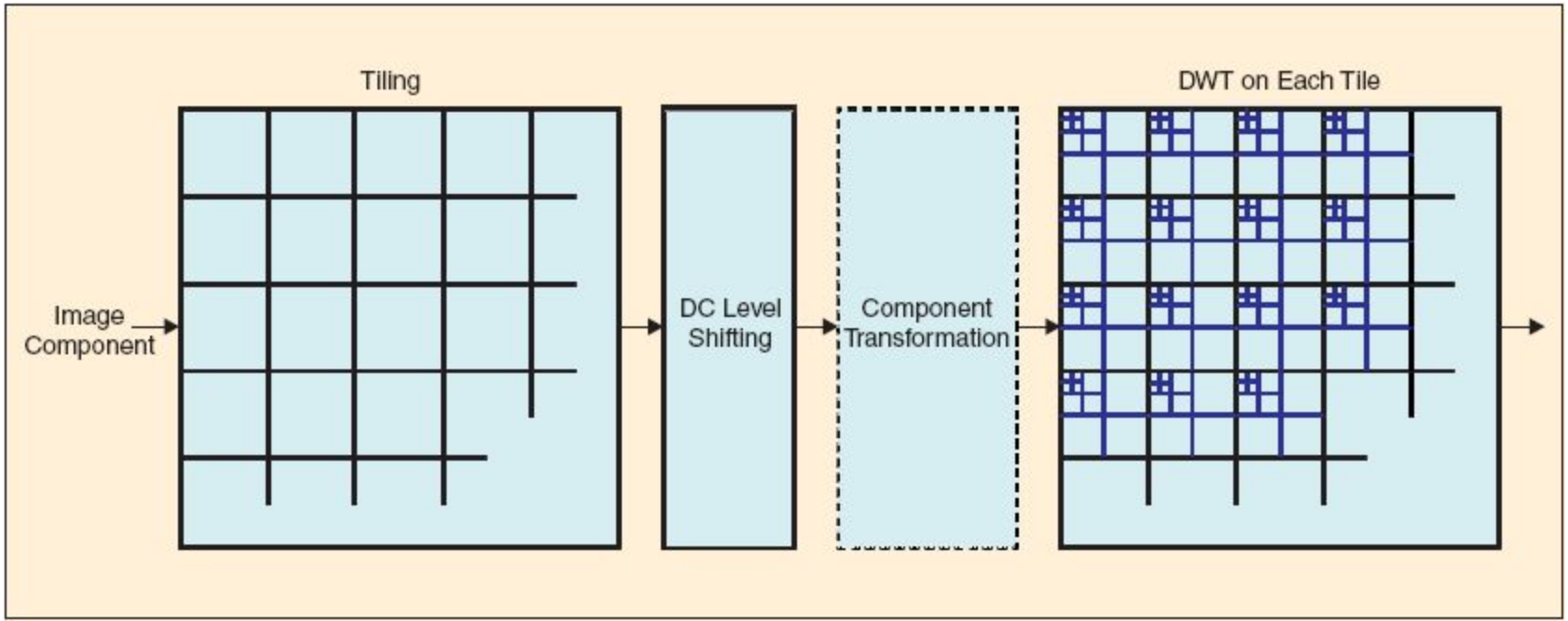
a) Block diagram of the JPEG2000 encoder algorithm. b) Block diagram of the JPEG2000 decoder algorithm.

JPEG2000 Encoding Pipeline

1. Image Tiling
2. Level Offset & DC Shift
3. Color Space Transformation
4. Discrete Wavelet Transform (DWT)
5. Quantization
6. Bit-plane Coding (EBCOT)
7. Rate Control
8. File Formatting

Image Tiling

- Image divided into rectangular tiles
- Each tile processed independently
- Enables memory-efficient processing



Why tiling is REQUIRED in JPEG2000

Memory limitation

- Wavelet transform works on entire data.

  For large images (satellite, medical, maps):

  10000×10000 pixels $\approx$ 100 million pixels

- Cannot load full image into memory.
- Tiling allows processing one tile at a time
- Without tiling → JPEG2000 would be impractical for large images.

Independent processing

- Each tile goes through:

  Tile → DWT → Quantization → EBCOT → Bitstream

- Advantages:
  - Parallel processing
  - Easier hardware implementation
  - Faster encoding/decoding

Region of Interest (ROI) decoding

If user wants only a small region:

- Decode only those tiles, and Ignore rest of image

Used in: Google Maps, Medical imaging, Satellite imagery

Reason 4: Error resilience

If bitstream is corrupted:

- Damage is limited to one tile
- Other tiles remain intact

Aspect	Significance
Memory	Reduces memory requirement
Speed	Enables parallelism
Flexibility	Partial decoding
Robustness	Limits error spread
Scalability	Supports large images

- Tiling is not for any compression gain. It is for **practical feasibility**.
- Tiling has a trade-off: Compression efficiency decreases slightly

- It is because Wavelet transform cannot cross tile boundaries
- Correlation across tiles is lost
  - Smaller tiles → worse compression
  - Larger tiles → better compression

What is tile size? Tile size = dimensions of each rectangular block

There is no fixed value to decide tile size?. It depends on application constraints.

Example: Image size = 1024 × 1024 Tile size = 256 × 256 → 16 tiles

Case 1: Small images (photographs)Similar to “no tiling”

Tile size = full image

Reason: No memory issue, Best compression

Case 2: Medium images

Balanced choice: Moderate memory, Good compression

Typical tile sizes:

128×128

256×256

512×512

Case 3: Very large images (satellite / medical)

Tile size depends on: Available RAM, Cache size, Real-time constraints

Larger tiles → better wavelet efficiency

Smaller tiles → lower memory

Level offset

Subtracting a fixed value from all pixel samples so that the signal is centered around zero.

For n-bit images

Offset value = 2^(n−1),

if n=8, Offset value = 128

New_pixel = Original_pixel − 128

Original pixel values: [0 64 128 192 255]

After level offset: [-128 -64 0 64 127]

Why level offset is REQUIRED

Efficient wavelet transform

- Low-pass filters compute averages
- Zero-centered data → smaller coefficients → fewer bits

Better energy compaction

- Before offset: Mean ≈ 128, After offset: Mean ≈ 0

Improves entropy coding (EBCOT)

- Bit-plane coding is more efficient
- MSB planes contain more zeros
- Arithmetic coding compresses better

DC Shift is the EFFECT of level offset

- DC = **average (zero-frequency) component,** Level offset **shifts the DC component to zero**
- **Level offset is the operation and DC shift is the the result of that operation.**

Without level offset

[120 130
125 135] Average (DC): (120+130+125+135)/4 = 127.5

With level offset After subtracting 128:

[-8 2 (-8+2−3+7)/4 = -0.5 ≈ 0
-3 7]

DC component is now near zero: This is called DC shift.

Color Space Transformation

- RGB $\longrightarrow$ YCbCr (lossy)
- Reversible Color Transform (lossless)
- Separates luminance and chrominance

Color Space Transformation in JPEG2000

Process of converting an image from **RGB** to another color space, it is used to **improve compression efficiency**

Why Color Space Transformation is Needed

- RGB components are **highly correlated**
- All three components carry brightness information
- Human eye is **more sensitive to brightness than color**
- Separating brightness and color improves compression

Y → (brightness) Cb → Blue chrominance Cr → Red chrominance

Luminance preserved with higher quality and Chrominance can be compressed more

JPEG2000 supports two transforms, and Choice depends on lossless or lossy mode

- Reversible Color Transform (RCT)
- Irreversible Color Transform (ICT)

Lossy Vs Loseless

Reversible Color Transform (RCT) Equations

Used in lossless JPEG2000
Integer-only operations
Exact reconstruction possible
No floating-point errors

Y = floor((R + 2G + B) / 4)
Cb = B − G
Cr = R − G

R = 100, G = 150, B = 200
Y = 150, Cb = 50, Cr = −50

Original RGB can be perfectly recovered

Irreversible Color Transform (ICT) Equations

Used in lossy JPEG2000

Floating-point computation

Y = 0.299R + 0.587G + 0.114B
Cb = −0.1687R − 0.3313G + 0.5B
Cr = 0.5R − 0.4187G − 0.0813B

Not exactly reversible

Discrete Wavelet Transform (DWT)

- Core of JPEG2000
- Converts image into frequency subbands
- Uses:
  - 5/3 wavelet (lossless)
  - 9/7 wavelet (lossy)

5/3 and 9/7 indicate the number of filter coefficients used in the wavelet transform:

First number → Low-pass filter length

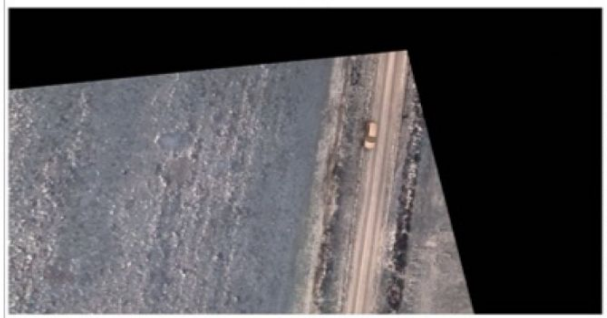
Second number → High-pass filter length

Wavelet Subbands – Level 1

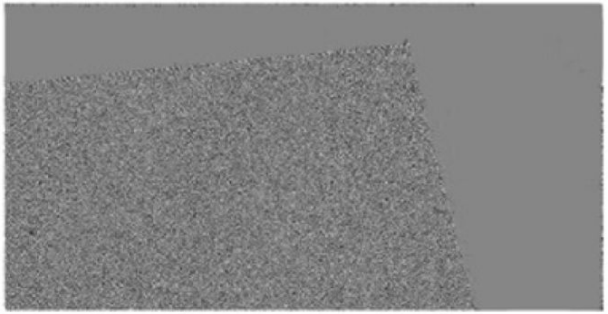
- LL: Low-Low (approximation)
- LH: Low-High (horizontal details)
- HL: High-Low (vertical details)
- HH: High-High (diagonal details)

(1) (2)

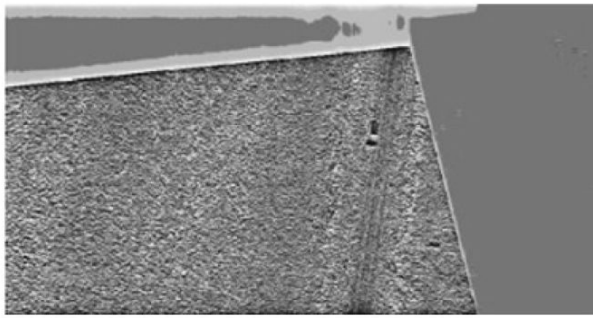
An example of Level-1 Wavelet transform using DWT to the cameraman image (1) and the corresponding transformed image (2) The transform, (2), produces 4 sub-bands. The top left sub-band is a low resolution version of the original sub-sampled in both horizontal and vertical directions, while the horizontal, h, vertical, v, and diagonal, d, details represent the down-sampled residual versions of the original image.



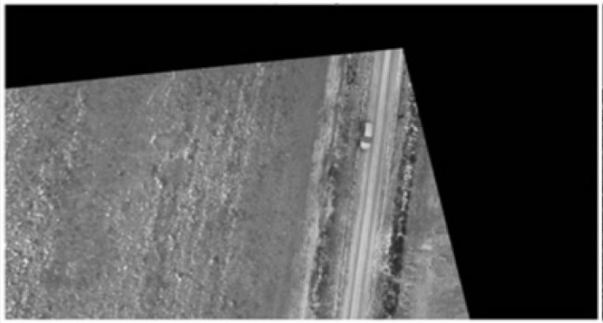
RGB Image



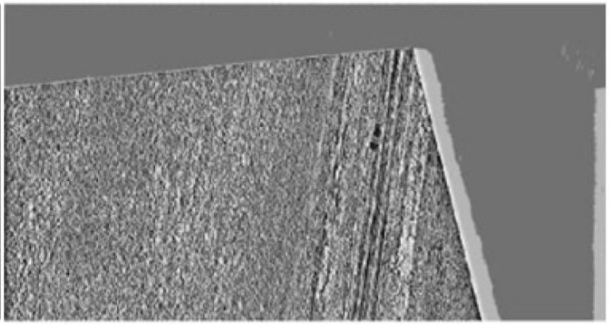
HH



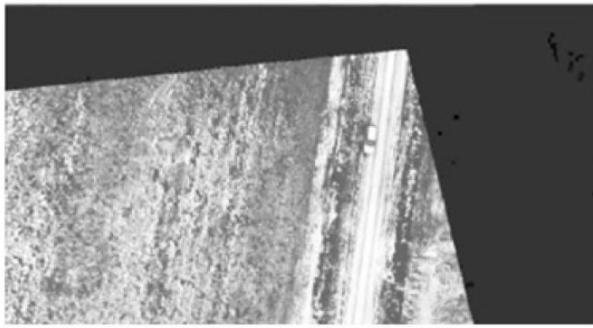
HL



Grey Image



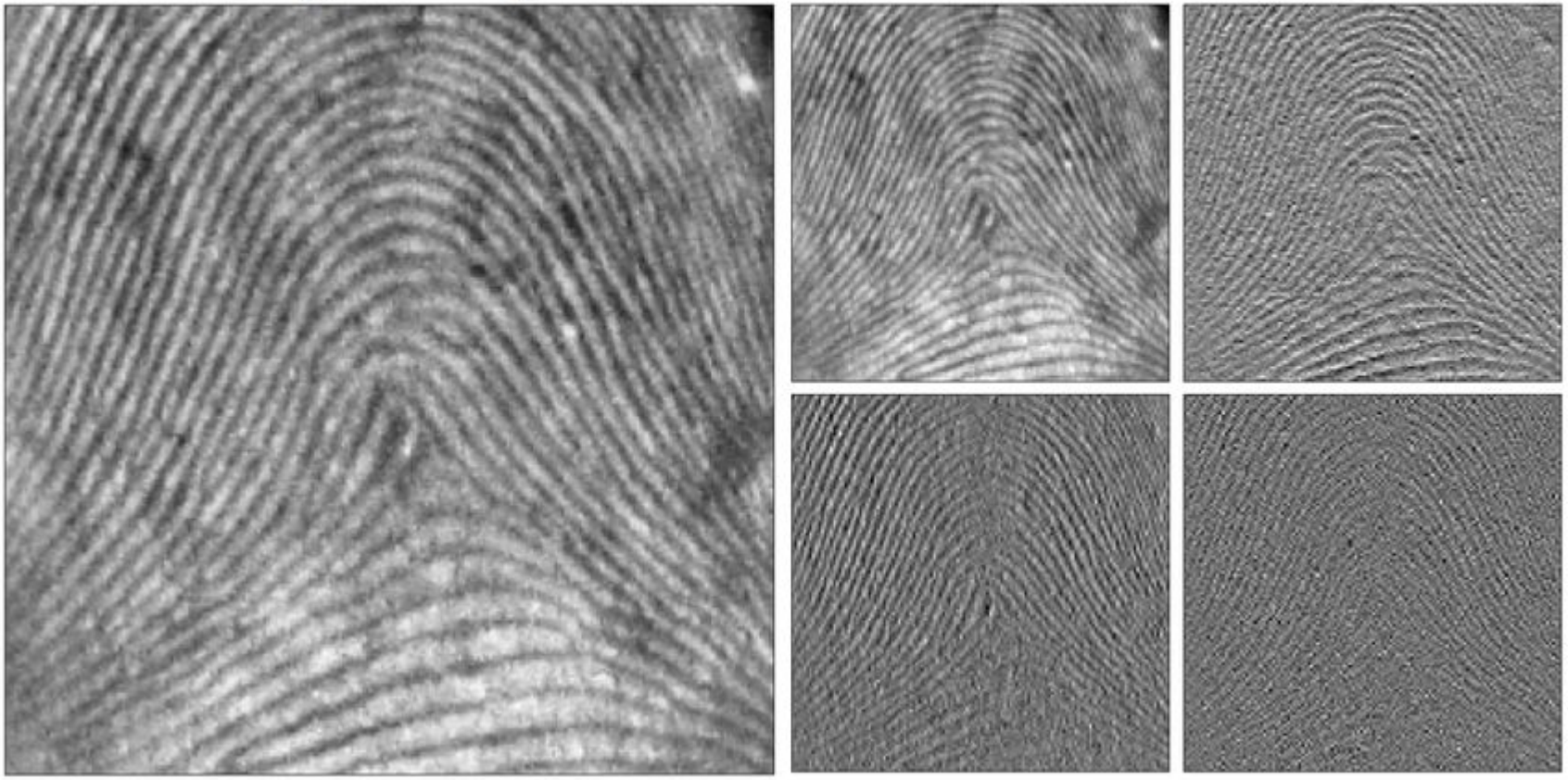
LH



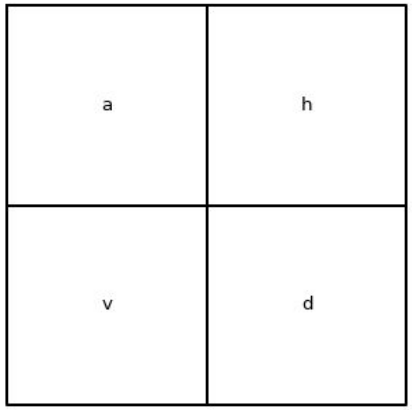
LL

Original Image

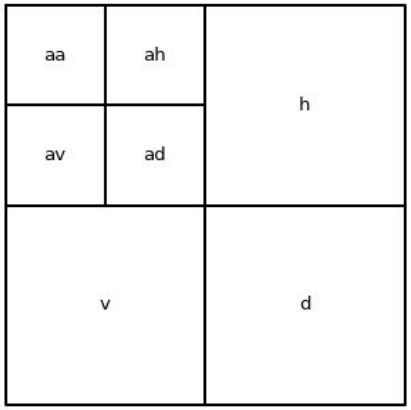
Level 1 Wavelet Decomposition



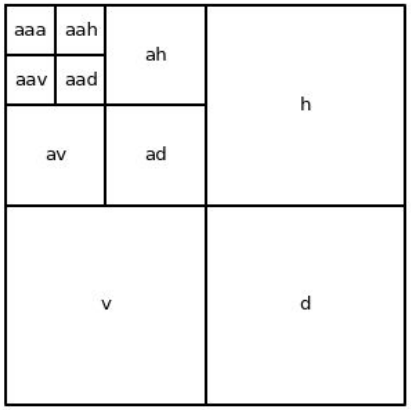
1 level
decomposition



2 level
decomposition

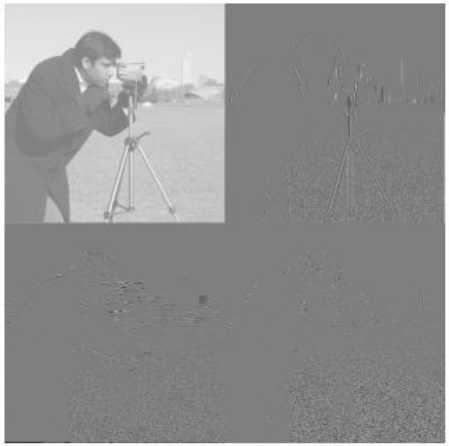


3 level
decomposition

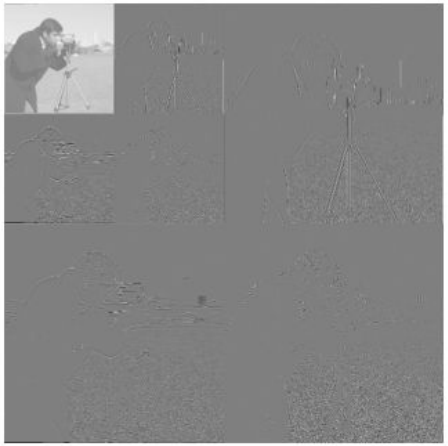


Image

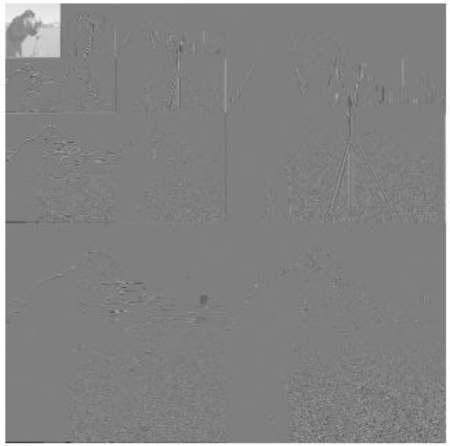
Coefficients
(1 level)



Coefficients
(2 level)



Coefficients
(3 level)



2D-Discrete Wavelet Transformation and its applications in Digital Image Processing

Wavelet based Denoising of Images

https://medium.com/@koushikc2000/2d-discrete-wavelet-transformation-and-its-applications-in-digital-image-processing-using-matlab-1f5c68672de3

Multi-level Wavelet Decomposition

- LL band recursively decomposed
- Produces multiple resolution levels
- Enables scalability and progressive decoding

Quantization

- Reduces precision of wavelet coefficients
- Scalar quantization
- Step size controls compression level

EBCOT – Bit-plane Coding

- Embedded Block Coding with Optimized Truncation
  - **Embedded** → bitstream can be truncated at any point and still decode
  - **Block Coding** → operates on small blocks independently
  - **Optimized Truncation** → encoder decides the best place to cut bits for a given bitrate

- After wavelet transform and quantization, JPEG2000 has:
  - Lots of numbers (coefficients)
  - Mostly small values and many zeros
  - Strong correlation between neighbors

Goal of EBCOT: Convert these numbers into a compressed bitstream such that

- important information is sent first
- the stream can be cut anywhere
- compression is very efficient

EBCOT – Bit-plane Coding

- Each wavelet subband is divided into code-blocks
- Typical size: 64 × 64 or 32 × 32, Why blocks?
  - Independent decoding
  - Error resilience
  - Easy rate control
  - Enables spatial scalability

Represent coefficients in bit-planes

- After quantization, coefficients are integers.
- Example code-block (small for teaching):

5 1 0

-3 0 2

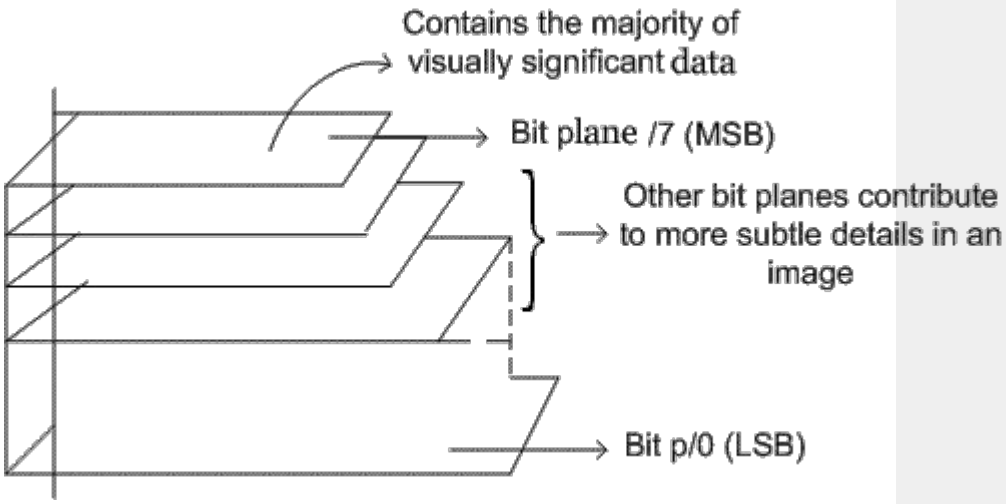
0 0 -1

Highest bit-plane = bit-plane 2

Bit-plane 2 → Bit-plane 1 → Bit-plane 0

Magnitudes in binary:

Value	Binary
5	101
3	011
2	010
1	001


Contains the majority of visually significant data
Bit plane /7 (MSB)
Other bit planes contribute to more subtle details in an image
Bit p/0 (LSB)

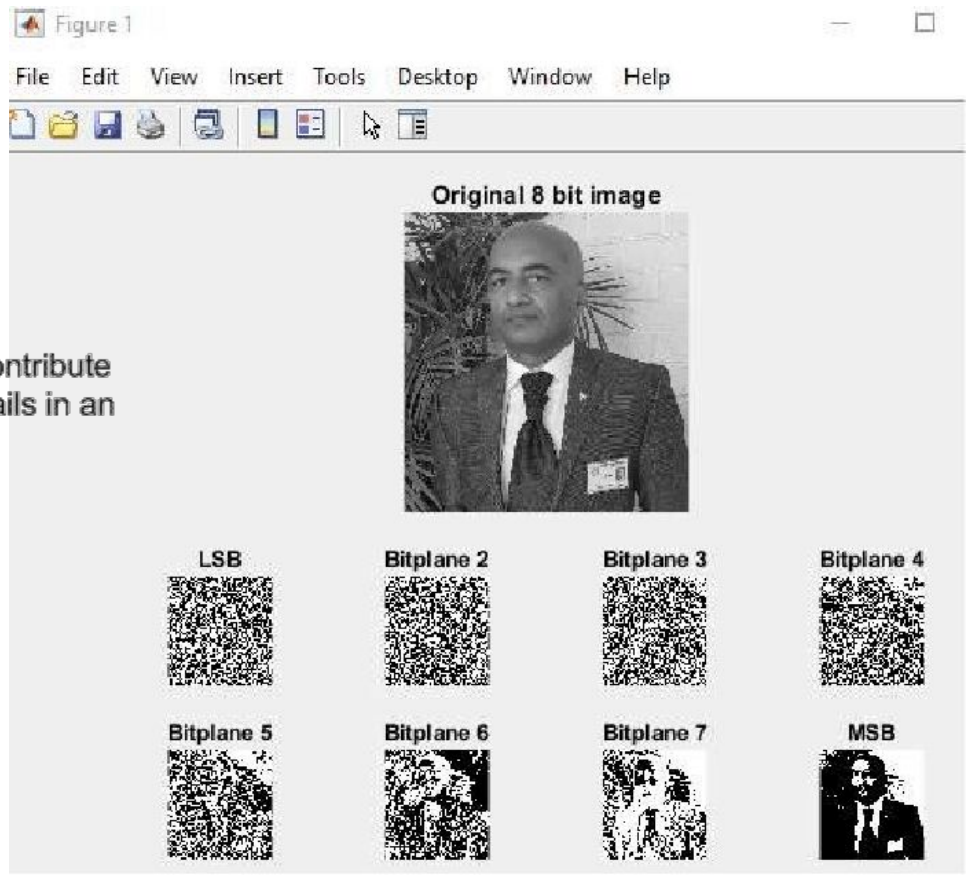


Figure 1
File Edit View Insert Tools Desktop Window Help
Original 8 bit image
LSB
Bitplane 2
Bitplane 3
Bitplane 4
Bitplane 5
Bitplane 6
Bitplane 7
MSB

Rate Control

- Rate Control in JPEG2000 is the mechanism used to control the final compressed bit-rate or file size while maintaining the best possible image quality.
- Selects optimal bitstream length
- Achieves target bit-rate
- Supports quality layers

Why Rate Control Is Needed

To meet a target bit-rate (e.g., 0.5 bpp, 1 bpp)

To achieve a specific file size

To balance compression vs quality

Essential for network transmission & storage limits

How

First compress fully, then truncate the bitstream optimally.

Wavelet Transform: Image is decomposed into sub-bands using DWT

Bit-plane Coding (EBCOT)

- Each code-block is encoded bit-plane by bit-plane
- Generates multiple coding passes
- Each pass has: Bit cost and Distortion reduction (quality gain)

Rate–Distortion Optimization

- For every coding pass:
  - Calculate **Distortion Reduction / Bits Used**
- This gives **RD slope**
- Higher slope = more quality gain per bit

Optimal Bitstream Truncation

- Sort coding passes by RD slope
- Keep the **most efficient passes**
- Discard less useful ones
- Stop when **target bit-rate / file size** is reached

Layer Formation

- Selected passes are grouped into **quality layers**
- Enables:
  - Progressive quality decoding
  - Scalable transmission

Example

Target bit-rate = **0.5 bits per pixel**

- JPEG2000 compresses image completely
- Rate control selects only the best coding passes
- Output file exactly matches 0.5 bpp
- Image quality is maximized at that rate

JPEG2000 File Structure

- JP2 file format
- Header + compressed codestream
- Supports metadata and region-of-interest

Advantages of JPEG2000

- • Higher compression efficiency
- • Progressive transmission
- • Error resilience
- • Lossless & lossy in one framework