The JPEG2000 Image Compression Standard

Part I: Image Compression

Compression techniques are used to reduce the redundant information in the image data in order to facilitate the storage, transmission and distribution of images (e.g. GIF, TIFF, PNG, JPEG)

Limitations of JPEG Standard

- **Low bit-rate compression**: JPEG offers an excellent quality at high and mid bit-rates. However, the quality is unacceptable at low bit-rates (e.g. below 0.25 bpp)
- **Lossless and lossy compression**: JPEG cannot provide a superior performance at lossless and lossy compression in a single code-stream.
- **Transmission in noisy environments**: the current JPEG standard provides some resynchronization markers, but the quality still degrades when bit-errors are encountered.
- **Different types of still images**: JPEG was optimized for natural images. Its performance on computer generated images and bi-level (text) images is poor.

Part II: The JPEG2000 Image Compression Standard

What is JPEG2000?

• JPEG2000 is a new compression standard for still images intended to overcome the shortcomings of the existing JPEG standard.

• The standardization process is coordinated by the Joint Technical Committee on Information technology of the International Organization for Standardization (ISO)/ International Electrotechnical Commission (IEC).

• JPEG2000 makes use of the wavelet and sub-band technologies. Some of the markets targeted by the JPEG2000 standard are Internet, printing, digital photography, remote sensing, mobile, digital libraries and E-commerce.

• The core compression algorithm is primarily based on the Embedded Block Coding with Optimized Truncation (EBCOT) of the bit-stream. The EBCOT algorithm provides a superior compression performance and produces a bit-stream with features such as resolution and SNR scalability and random access.

Features of JPEG2000

- **Lossless and lossy compression**: the standard provides lossy compression with a superior performance at low bit-rates. It also provides lossless compression with progressive decoding. Applications such as digital libraries/databases and medical imagery can benefit from this feature.

- **Protective image security**: the open architecture of the JPEG2000 standard makes easy the use of protection techniques of digital images such as watermarking, labeling, stamping or encryption

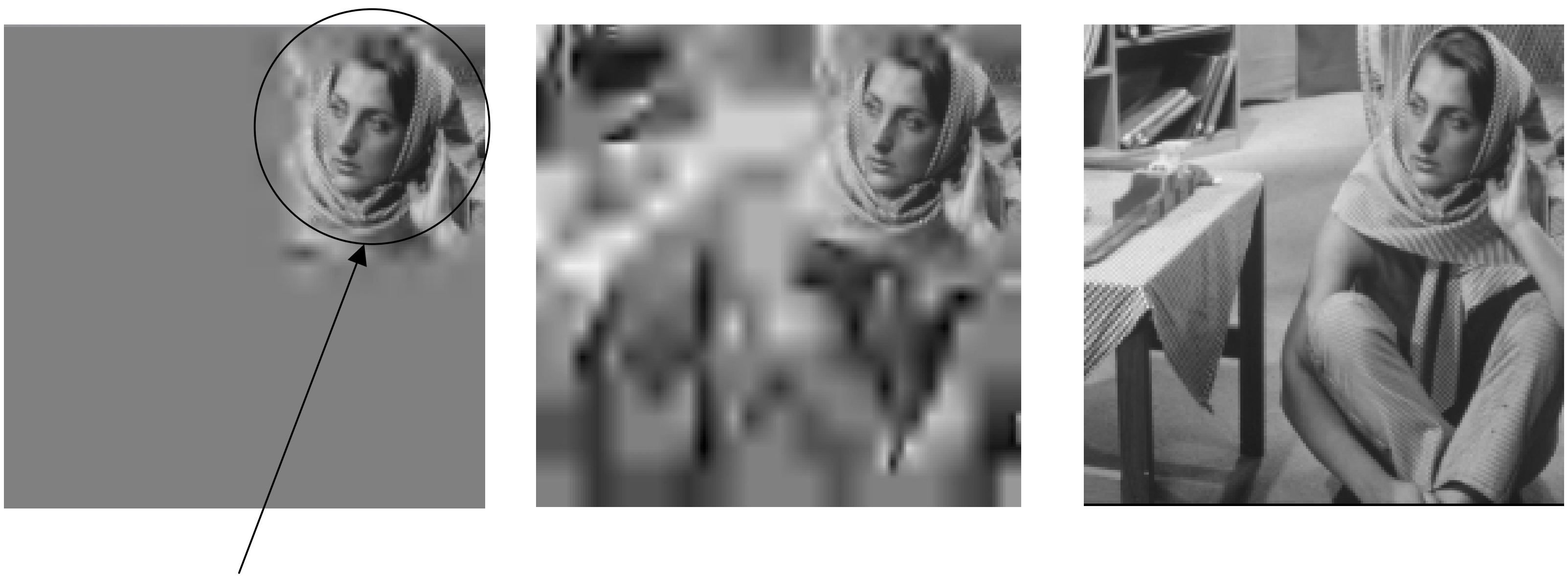
- **Region-of-interest coding**: in this mode, regions of interest (ROI’s) can be defined. These ROI’s can be encoded and transmitted with better quality than the rest of the image .

- **Robustness to bit errors**: the standard incorporate a set of error resilient tools to make the bit-stream more robust to transmission errors.

Features of JPEG2000 (cont'd)

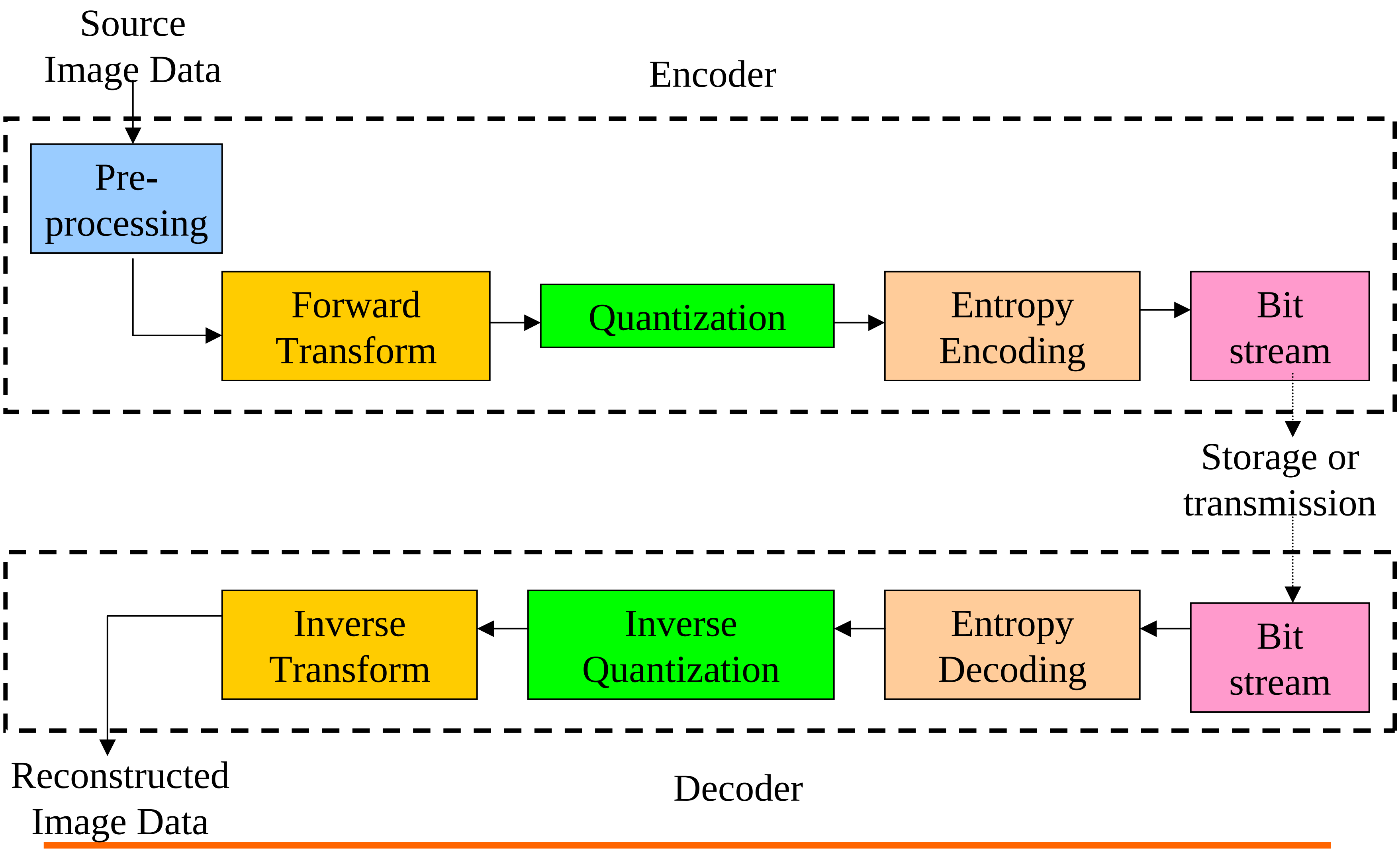
Example of region of interest coding.

A **region of interest** in the *Barbara* image is reconstructed with quality scalability. The region of interest is decoded first before any background information.



Region of interest

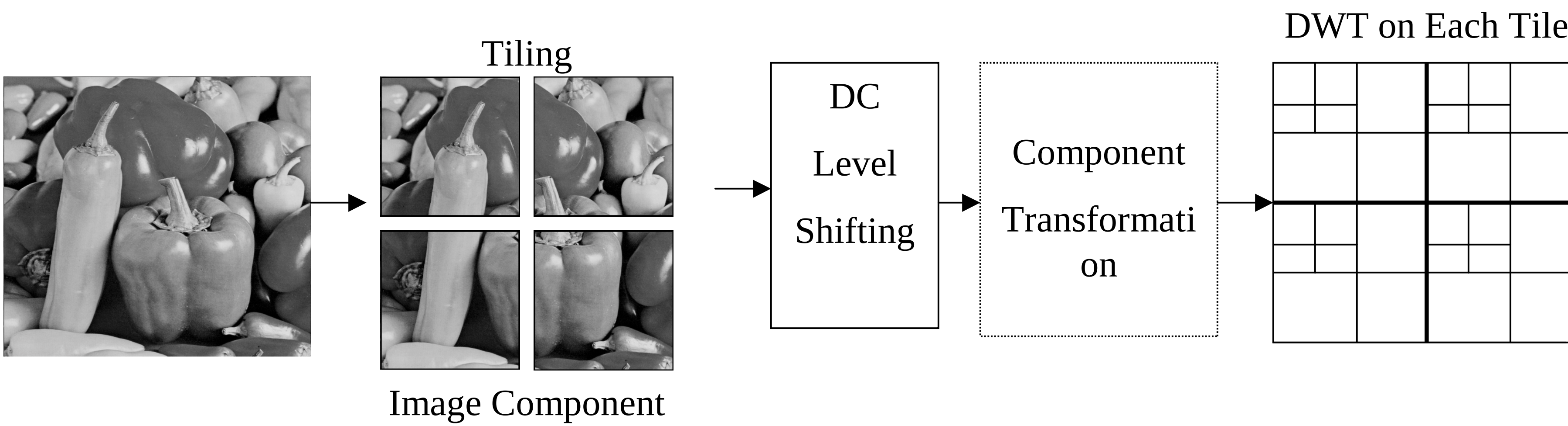
The JPEG 2000 Compression Engine



Pre-processing

Three steps:

1. Image tiling (optional)→ for each image component.



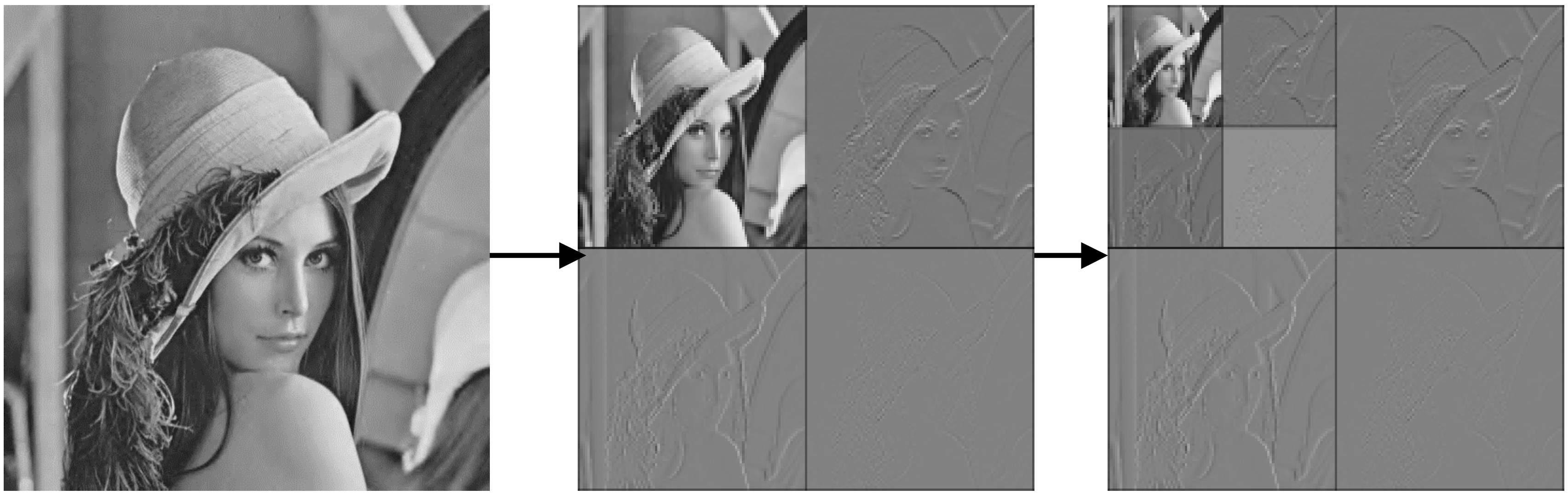
• DC level shifting → samples of each tile are subtracted the same quantity (i.e. component depth).

• Color transformation (optional) → from RGB to Y Cb Cr

Forward Transform

• Discrete Wavelet Transform (DWT) is used to decompose each tile component into different sub-bands.

• The transform is in the form of dyadic decomposition and use bi-orthogonal wavelets.



Forward Transform (cont'd)

- DWT can be irreversible or reversible.

    - Irreversible transform →Daubechies 9-tap/7-tap filter
    - Reversible transform → Le Gall 5-tap/3-tap filter

- Two filtering modes are supported:

    - Convolution based
    - Lifting based

Analysis and Synthesis Filter Coefficients

Le Gall 5/3

Analysis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	6/8	1
+/- 1	2/8	-1/2
+/- 2	-1/8	
Synthesis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	1	6/8
+/- 1	1/2	-2/8
+/- 2		-1/8

Analysis and Synthesis Filter Coefficients

Daubchines 9/7

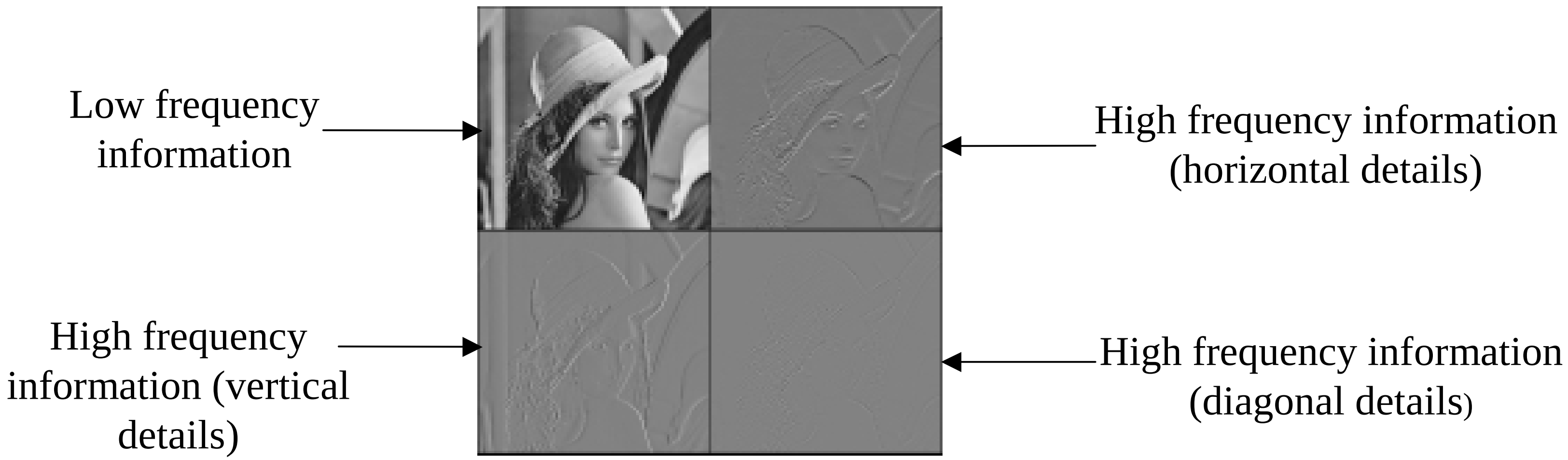
Analysis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	0.6029490182363579	1.115087052456994
+/- 1	0.2668641184428723	-0.5912717631142470
+/- 2	-0.07822326652898785	-0.05754352622849957
+/- 3	-0.01686411844287495	0.09127176311424948
+/- 4	0.02674875741080976	
Synthesis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	1.115087052456994	0.6029490182363579
+/- 1	0.5912717631142470	-0.2668641184428723
+/- 2	-0.05754352622849957	-0.07822326652898785
+/- 3	-0.09127176311424948	0.01686411844287495
+/- 4		0.02674875741080976

2D-Forward Transform

• 1-D sets of samples are decomposed into low-pass and high-pass samples.

• Low-pass samples represent a down-sampled, low resolution version of the original set.

• High pass samples represent a down-sampled residual version of the original set (details).



Quantization

- After transformation, all coefficients are quantized using scalar quatization.
- Quantization reduces coefficients in precision. The operation is lossy unless the quatization step is 1 and the coefficients integers (e.g. reversible integer 5/3 wavelet).
- The process follows the formula:

$$q_b(u,v) = sign(a_b(u,v)) \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor$$

Largest integer not exceeding a_b

Quantized value

Transform coefficient of sub-band b

Quantization step

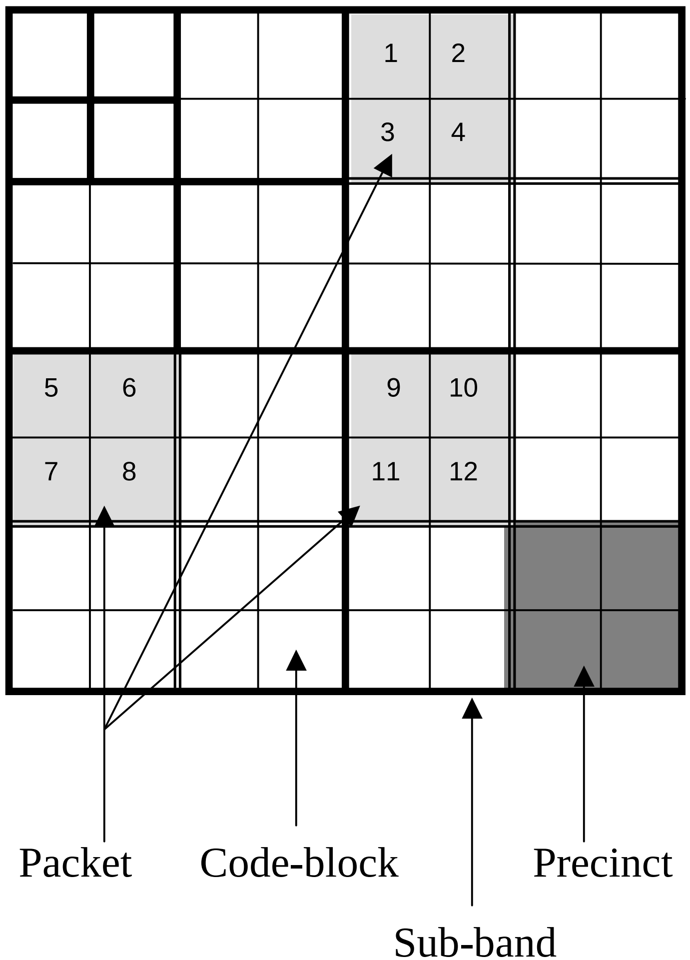
Modes of Quantization

• Two modes of operation:

➢ **Integer mode→** integer-to-integer transforms are employed. Quantization step are fixed to one. Lossy coding is still achieved by discarding bit-planes.

➢ **Real mode→** real-to-real transforms are employed. Quantization steps are chosen in conjunction with rate control. In this mode, lossy compression is achieved by discarding bi-planes or changing the size of the quantization step or both.

Code-blocks, precincts and packets



Precinct: each sub-band is divided into rectangular blocks called precincts.

Packets: three spatially consistent rectangles comprise a packet.

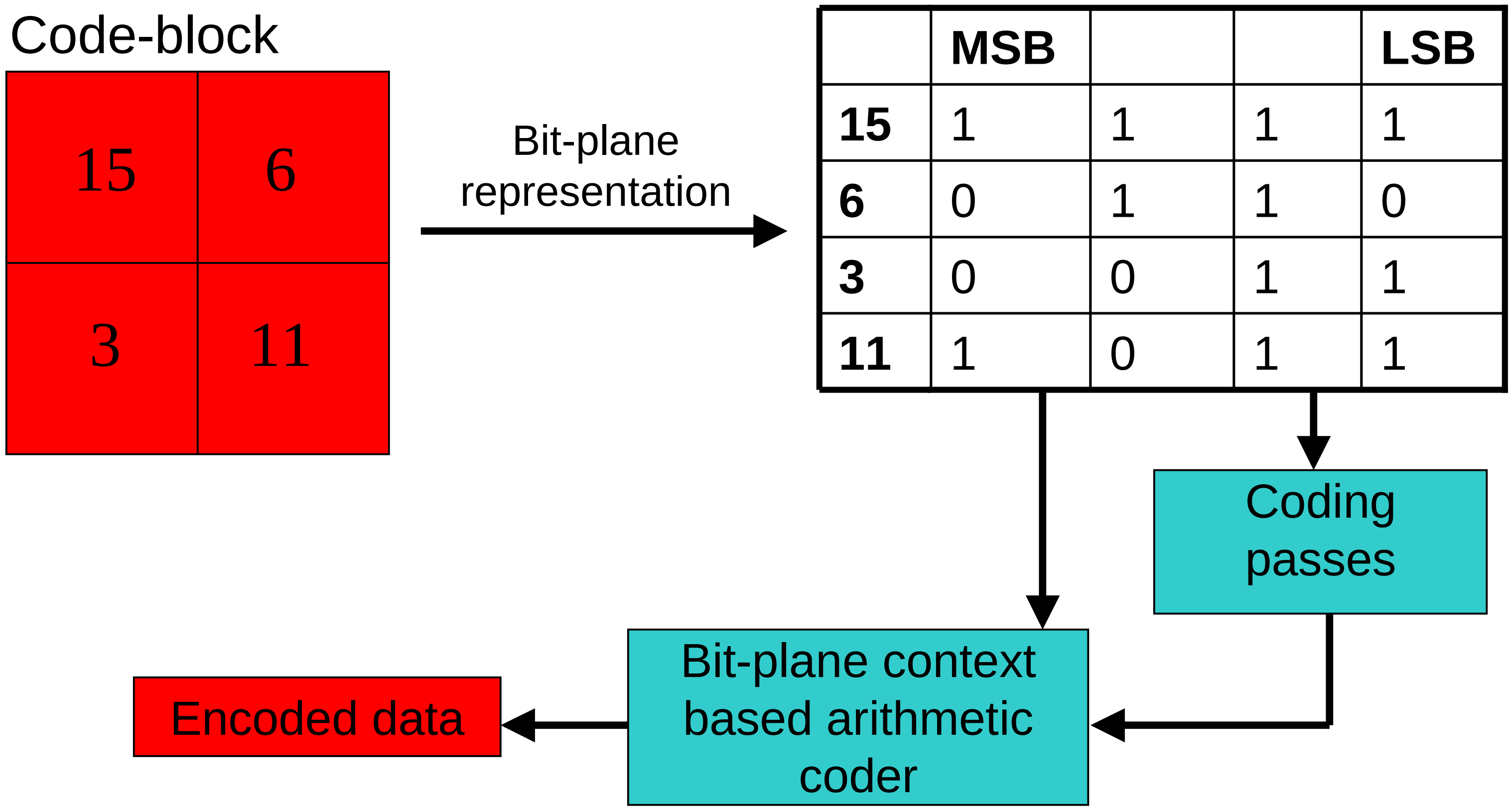
Code-block: each precinct is further divided into non-overlapping rectangles called code-blocks.

Each code-block forms the input to the entropy encoder and is encoded independently.

Within a packet, code-blocks are visited in raster order.

Entropy Coding: Bit-planes

• The coefficients in a code block are separated into bit-planes. The individual bit-planes are coded in 1-3 coding passes.



	MSB			**LSB**
15	1	1	1	1
6	0	1	1	0
3	0	0	1	1
11	1	0	1	1

Entropy Coding: Coding Passes

• Each of these coding passes collects contextual information about the bit-plane data. The contextual information along with the bit-planes are used by the arithmetic encoder to generate the compressed bit-stream.

• The coding passes are:

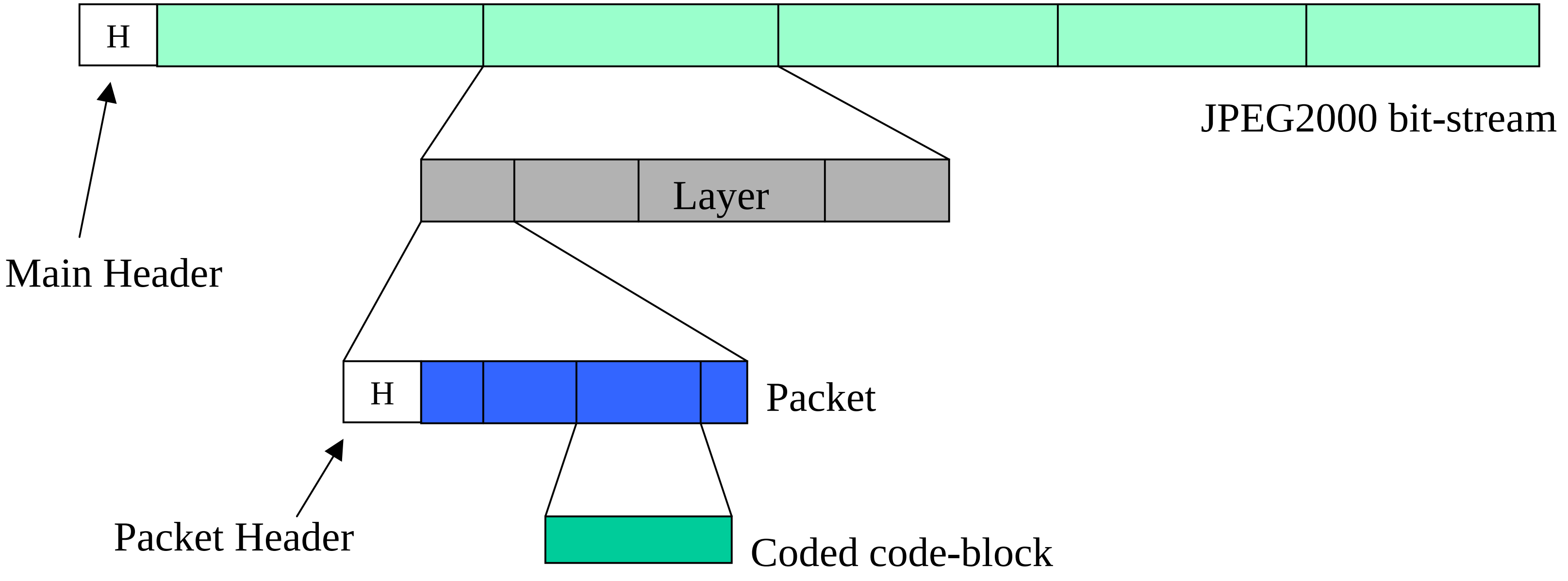
- **Significance propagation pass →** coefficients that are insignificant and have a certain preferred neighborhood are coded.
- **Magnitude refinement pass →** the current bits of significant coefficients are coded.
- **Clean-up pass →** the remaining insignificant coefficients for which no information has yet been coded are coded.

JPEG2000 Bit-stream

• For each code-block, a separate bit-stream is generated.

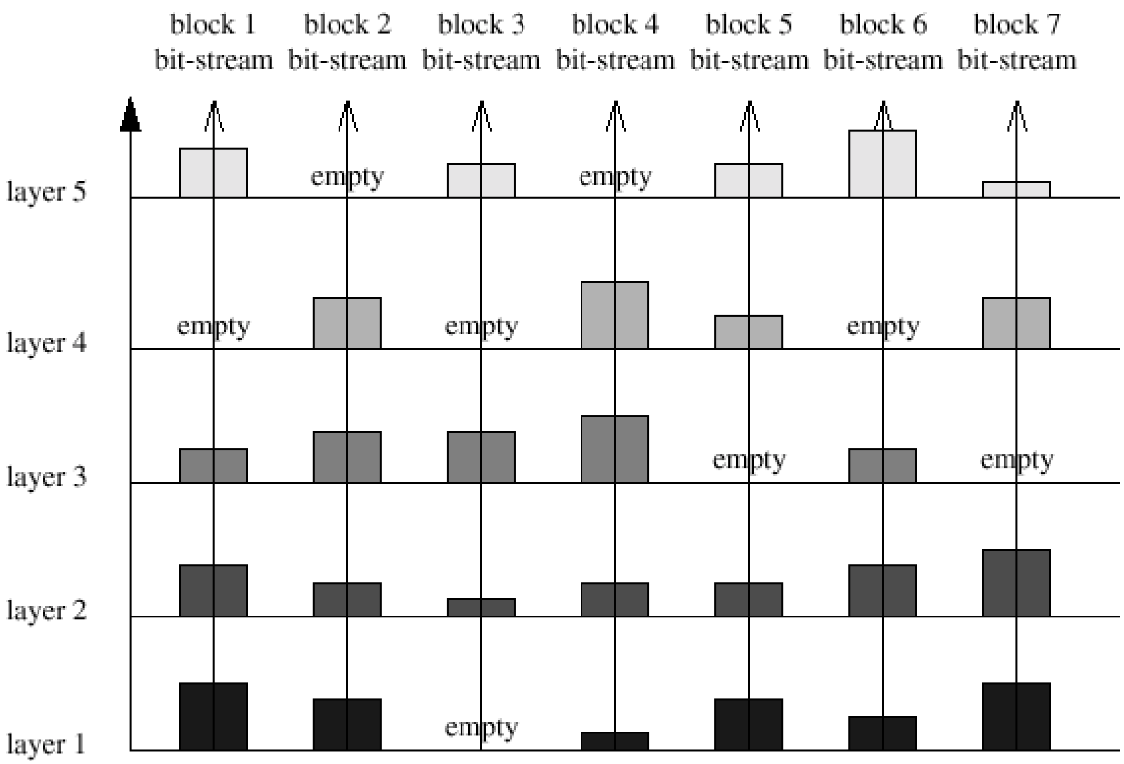
• The coded data of each code-block is included in a packet.

• If more that one layer is used to encode the image information, the code-block bit-streams are distributed across different packets corresponding to different layers.



Layers of the JPEG2000 Bit-stream

• Therefore, each layer consists of a number of consecutive bit-plane coding passes from each code-block in the tile, including all sub-bands of all components for that tile.



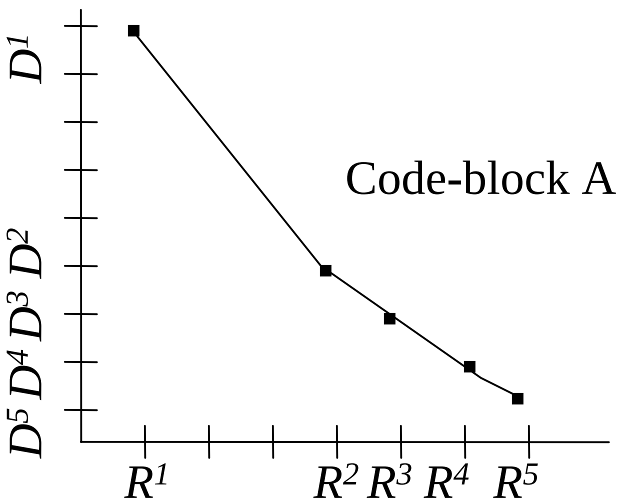
Layer Formation

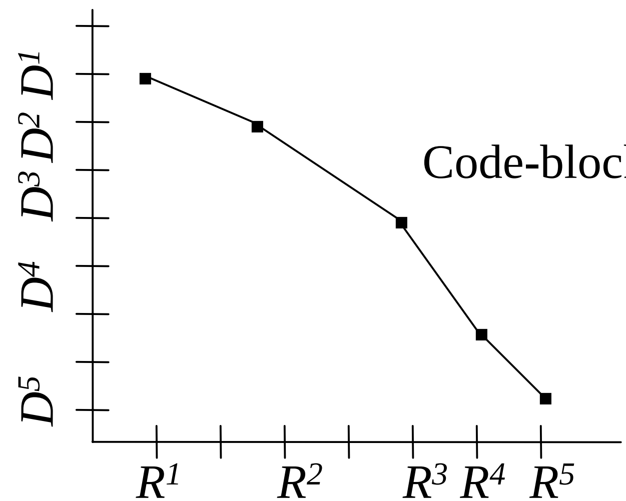
• The individual code-block streams have the property that they can be truncated to a variety of discrete lengths $R^1, R^2, R^3 \ldots R^n$.

• The distortion incurred when reconstructing from each of these truncated subsets is estimated and denoted by $D^1, D^2, D^3 \ldots D^n$. The Mean Squared Error distortion metric is generally used.

• The first, lowest quality layer, is formed from the optimally truncated code-block bit-streams.

Example: two individual code-block streams

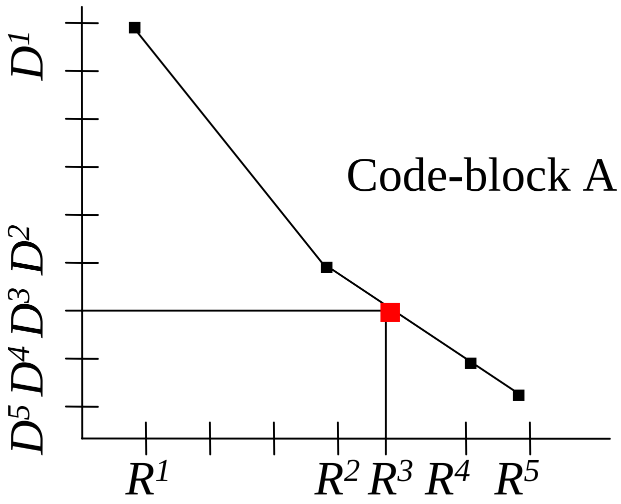


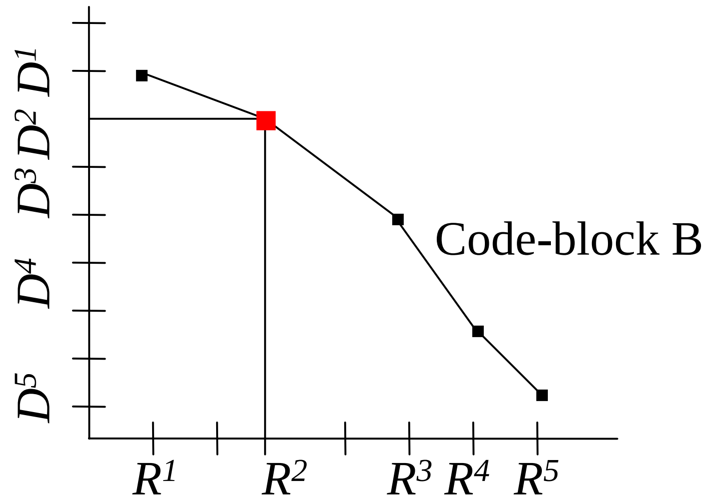


Code-block contributions

• Each subsequent layer is formed by optimally truncating the code-block bit-streams to achieve successively higher target bit-rates, distortion bounds or other quality metrics, as appropriate, and including the additional code words required to augment the information represented in previous layers to the new truncation points.

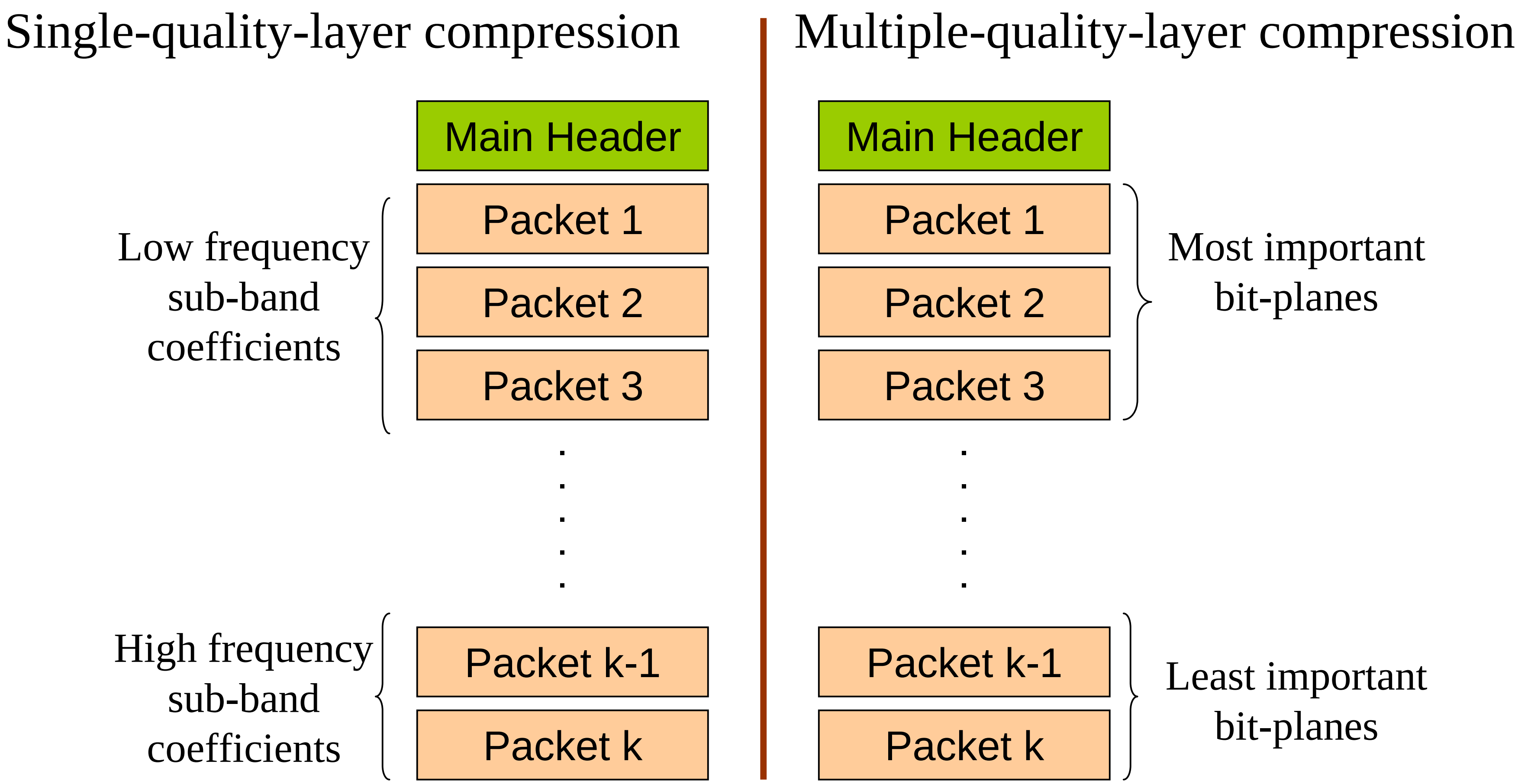
Example:





If target rate for layer 1 is R_{L1} and $R_A^3 + R_B^2 \approx R_{L1}$ with a minimum distortion; then, code-block A is truncated to R^3 and code-block B is truncated to R^2 for layer 1.

Hierarchical structure of the JPEG2000 Bit-stream



Quality Scalability

By interleaving the packets in different orders, four possible progression orders can be achieved in JPEG2000:

- Quality
- Resolution
- Spatial location
- Component

The *Barbara* image is decompressed at different **qualities**.

0.125 bpp

0.25 bpp

0.5 bpp

Resolution Scalability

The *Barbara* image is reconstructed at three **resolutions**.

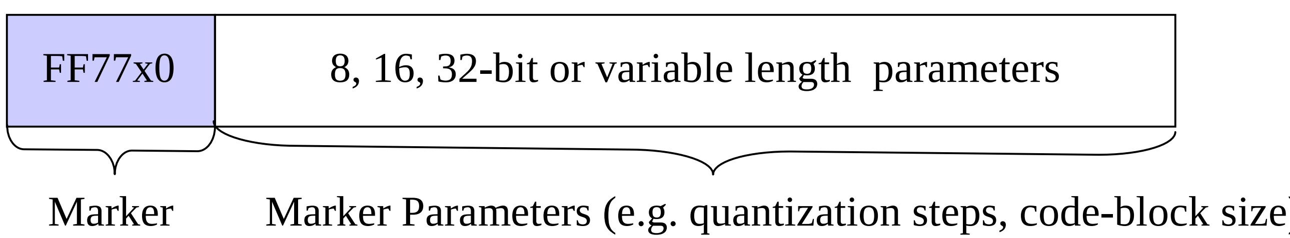
Markers and Markers Segments

• A main header is appended to the final bit stream. This header is different from the header found in front of every packet and every tile.

• The headers from all the packets and tiles can be placed on the main header → useful if all the header information is to be separated from the compressed data.

•The main header consist of markers and markers segments.

• A marker is a known word (16 bits) that identifies the information contained in the marker segment.



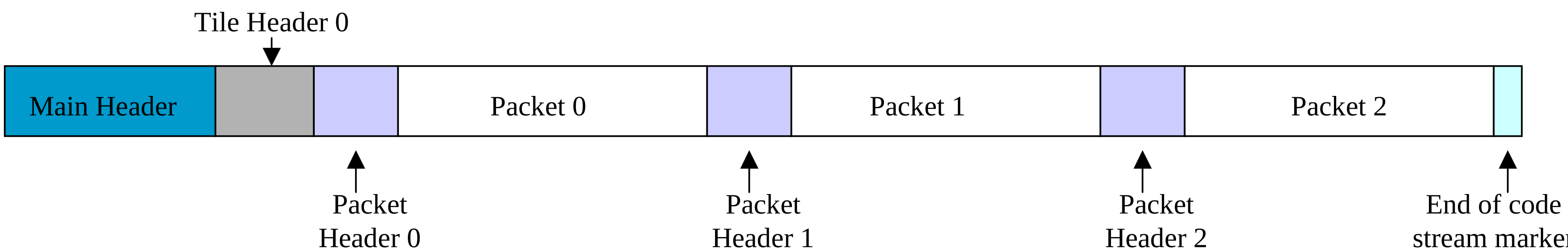
Marker Segment

Packet Headers

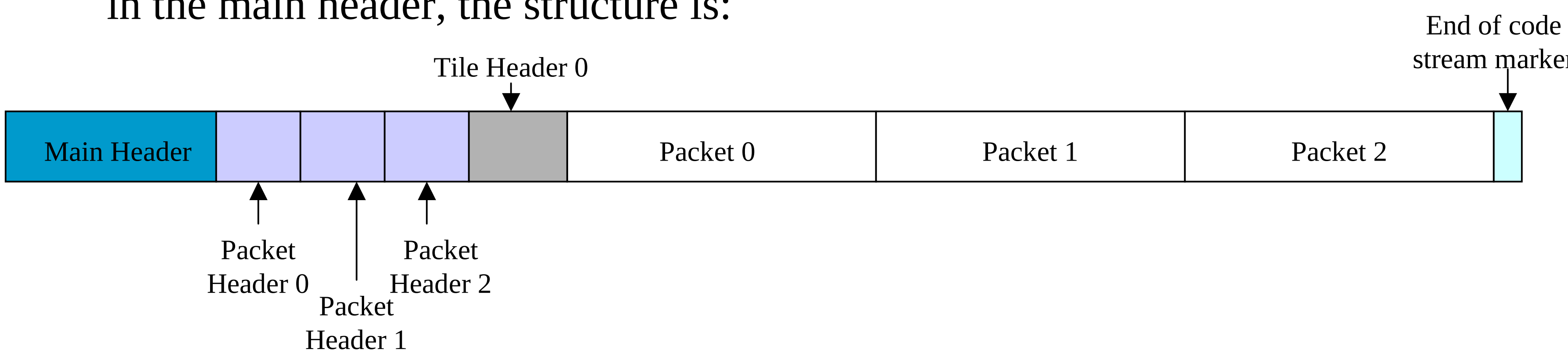
- A header is placed in front of every packet and contains the following information :
  - **Zero length packet** →indicates whether the packet has a length of zero or not.
  - **Code-block inclusion** →indicates whether a code block has been included in the packet or not.
  - **Zero-bit plane information** →if a code-block is included for the first time, the packet header contains information identifying the actual number of bit-planes used to represent coefficients from the code-block.
  - **Number of coding passes** →the number of coding passes included in this packet from each code-block.
  - **Length of data from a given code block** →the packet header identifies the number of bytes contributed by each included code-block.
- The information in the packet header is variable length encoded.

Sample JPEG2000 bit-stream

• For a 512 x 512 gray-level image compressed with one tile, code-block size 64 x 64, precinct size 512 x 512, 3 levels of decomposition and one layer, the general structure of the bit stream is as follows

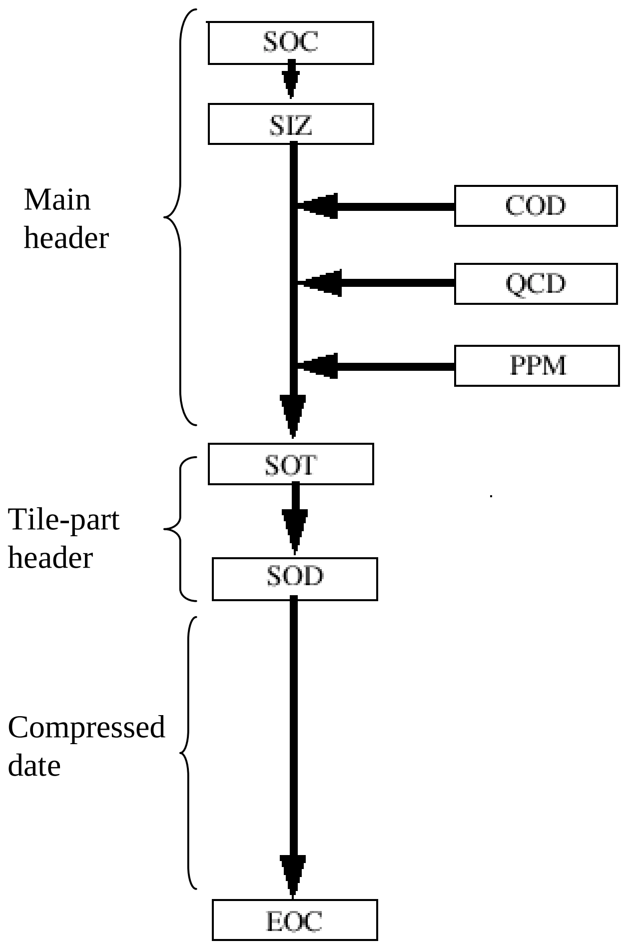


• If all the packet headers are grouped together in a single header and placed in the main header, the structure is:



Markers in a JPEG2000 bit-stream

• The main header would include the following markers and marker segments:



SOC: Start of Code Stream marker

SIZ: Image and Tile Size marker

The SIZ marker segment provides information about the size of the uncompressed image.

COD: Coding Style Default marker

The COD marker segment provides information about the coding style, decomposition and layering used for all the components of the image

QCD: Quantization Default marker

The QCD marker segment provides information about the quantization used for all the components of the image

PPM: Packed Packet Headers, main header

The PPM marker segment is the collection of packet headers.

SOT: Start of Tile-part marker

The SOT marker segment marks the beginning of a tile-part

SOD: Start of Data marker

EOC: End of Code Stream marker