Quantization

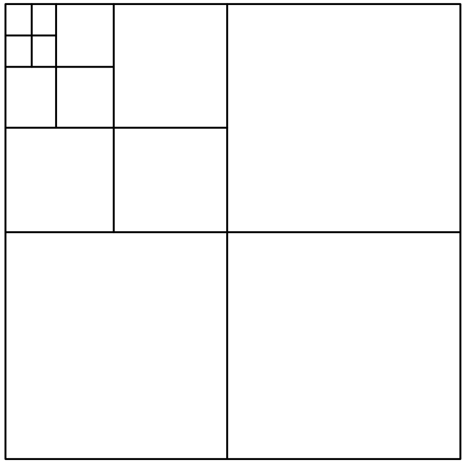
- Scalar dead zone quantization
- TCQ

JPEG2000: capabilities supported

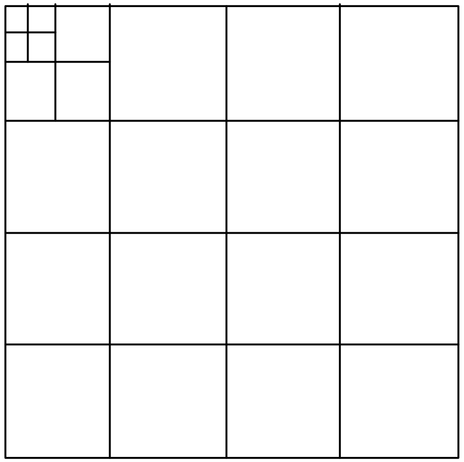
- Arbitrary wavelet decompositions
- Arbitrary wavelet kernels
  - can differ for each direction and for each resolution level
- Arbitrary bit-depth images
  - anywhere from 1 to about 30 bits per sample
  - signed and unsigned

Decompositions supported

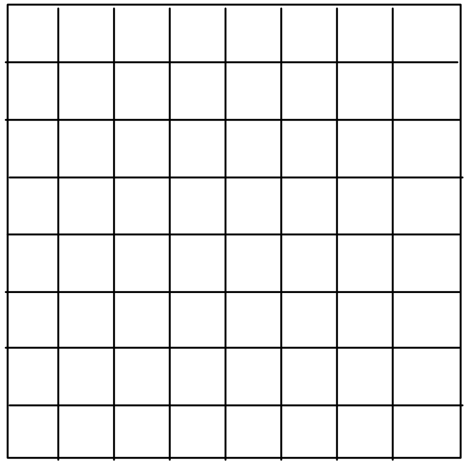
mallat



spacl



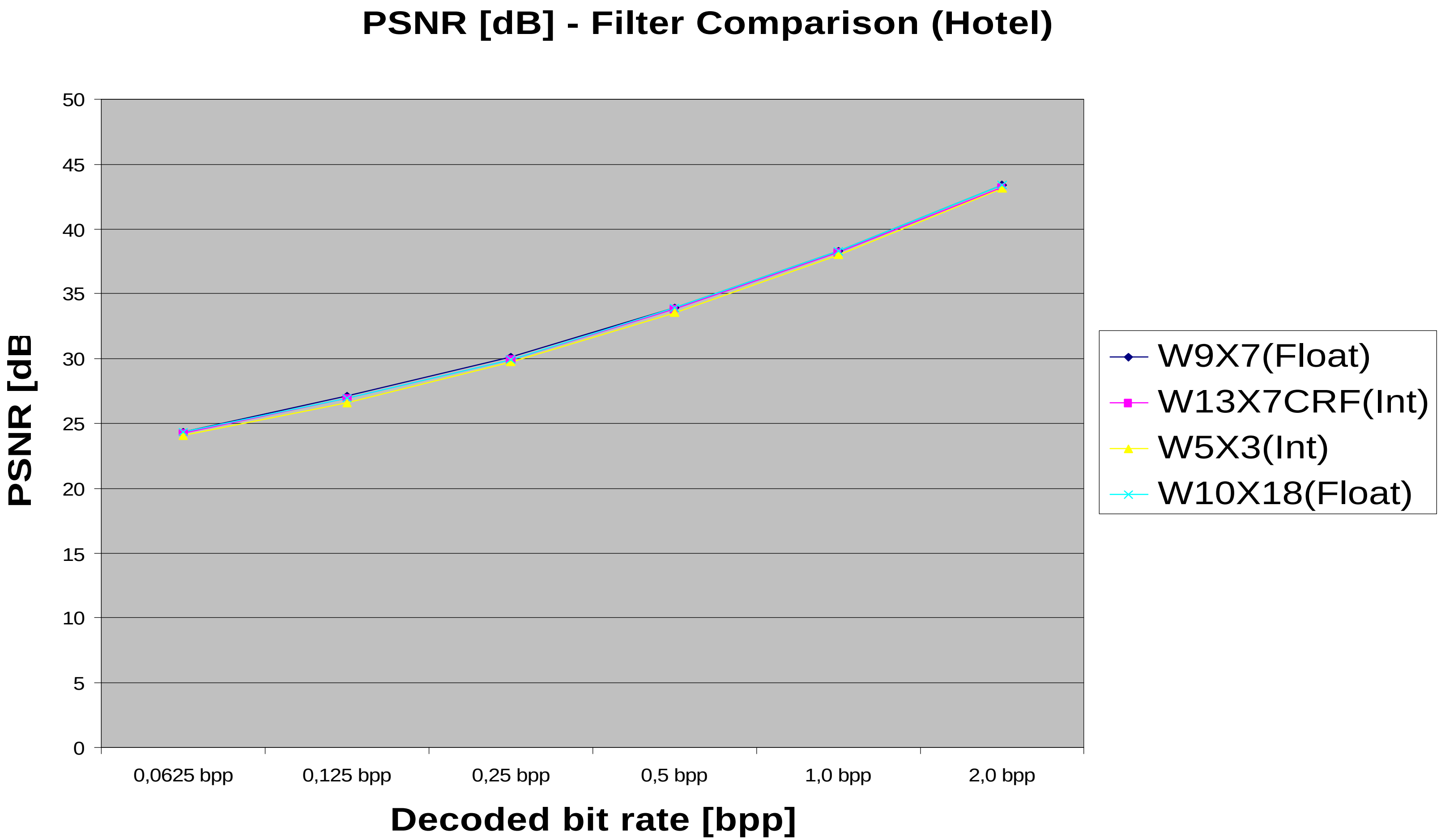
packet



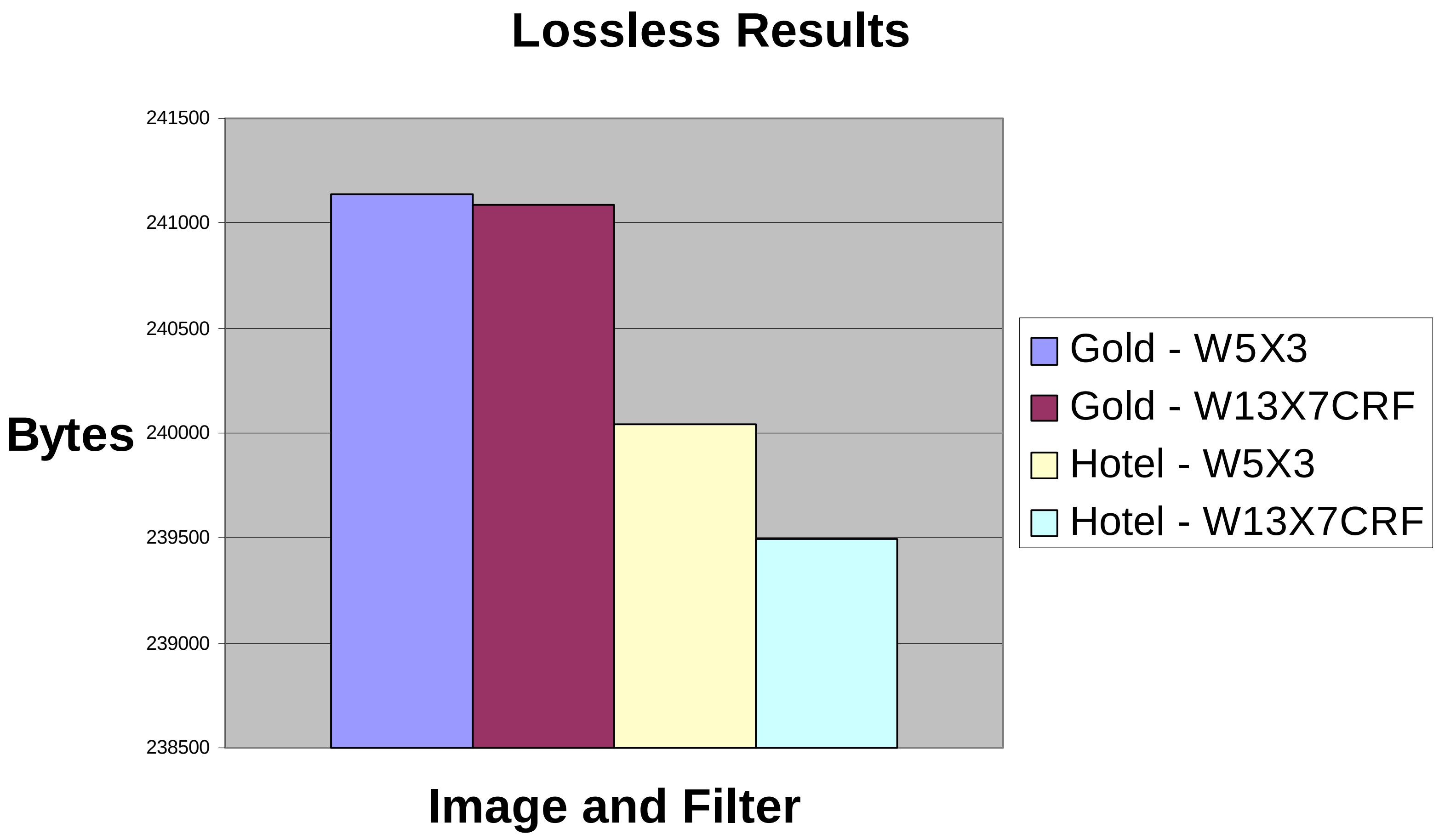
JPEG2000: Filters supported

- **Floating point wavelets:** (9,7), (10,18)
- **Integer**: (13,7), CRF(13,7), (5,3), (2,10), ...
- **Default integer** (for lossy coding): CRF(13,7)
- **Default integer** (for lossless coding): (5,3)
- **User defined filters**

Some results for different filters



Some results for lossless coding



JPEG2000: capabilities supported

- Arbitrary number of resolution levels
  - 0 to 16 currently allowed
  - image dimensions down to 1x1 for all decompositions
- Multi-component imagery
  - up to 256 components
  - arbitrary dimensions/bitdepths for each component
  - reversible & non-reversible component colour transforms

JPEG2000: capabilities supported

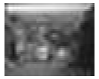
- Explicit and implicit scalar deadzone quantization
  - implicit mode degenerates to no quantizer at all
  - fixed visual weighting capability
- rate control
  - post-compression rate-distortion optimization
  - target one bit-rate or an arbitrary set of specified bit-rates

JPEG2000: capabilities supported

- Resolution scalability
- SNR scalability
  - selection of the degree of SNR scalability
  - progressive visual weighting
- “Random access” capability

Example:
Progressive by resolution

- Image: gold
- Resolution levels: 5



Example:
Progressive by resolution

- Image: hotel
- Resolution levels: 5

Plaza Laguna

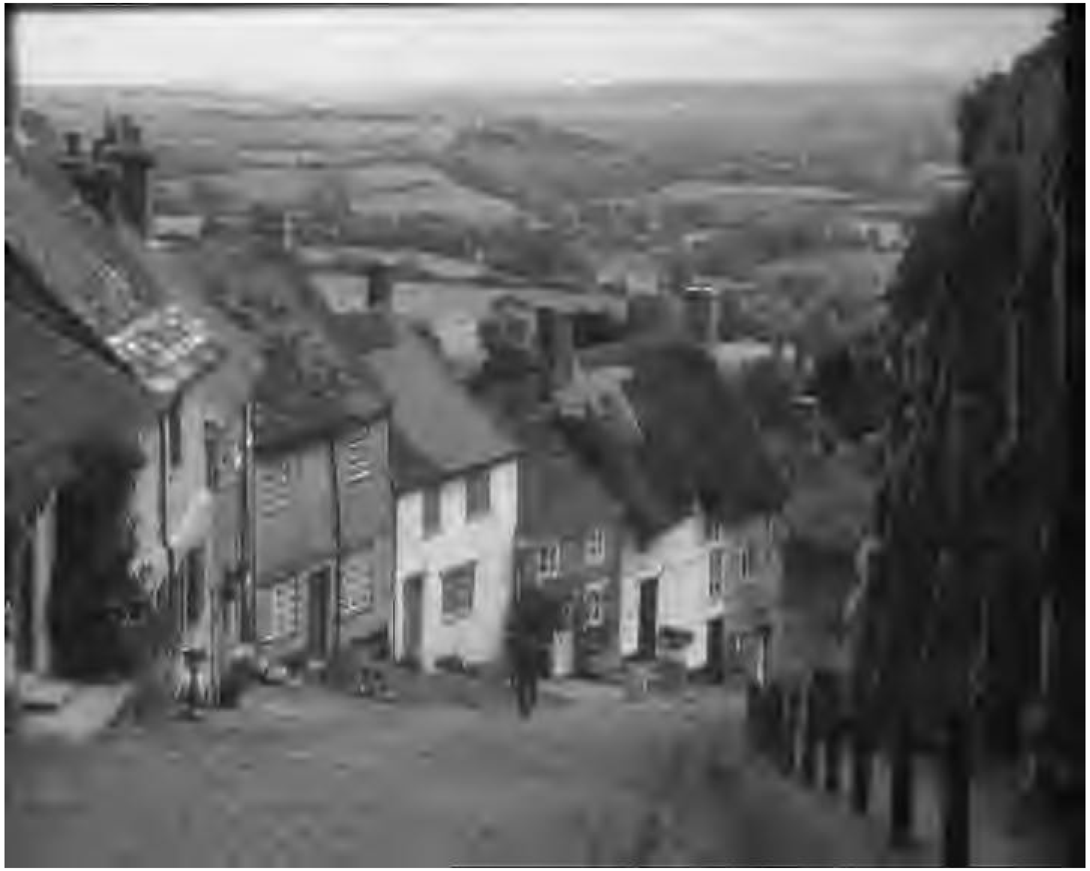
Hotel Lotus-Lila
Plava Laguna
Porec

Hotel Lotus-Lila
Plava Laguna
Porec

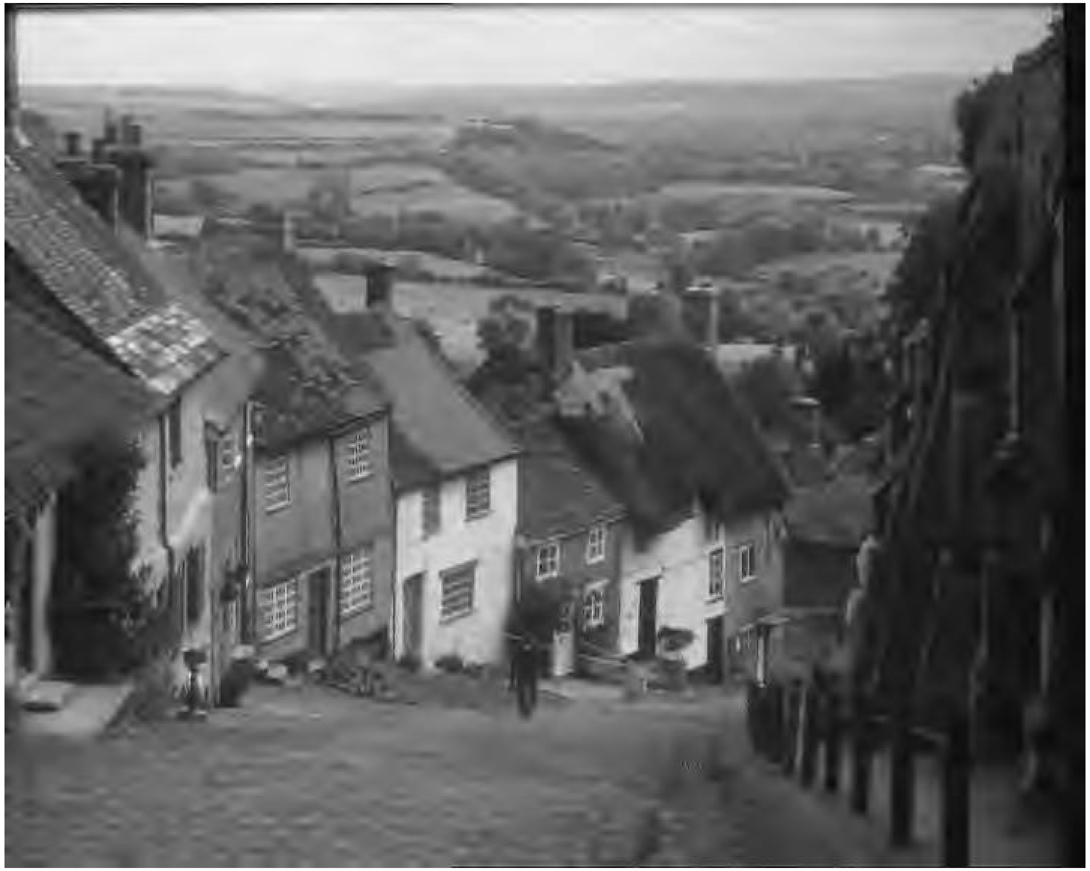
Example:
Progressive by quality

• Image:	gold
• Bitrates:	0.0625 bpp
	0.125 bpp
	0.25 bpp
	0.5 bpp
	1.0 bpp
	2.0 bpp

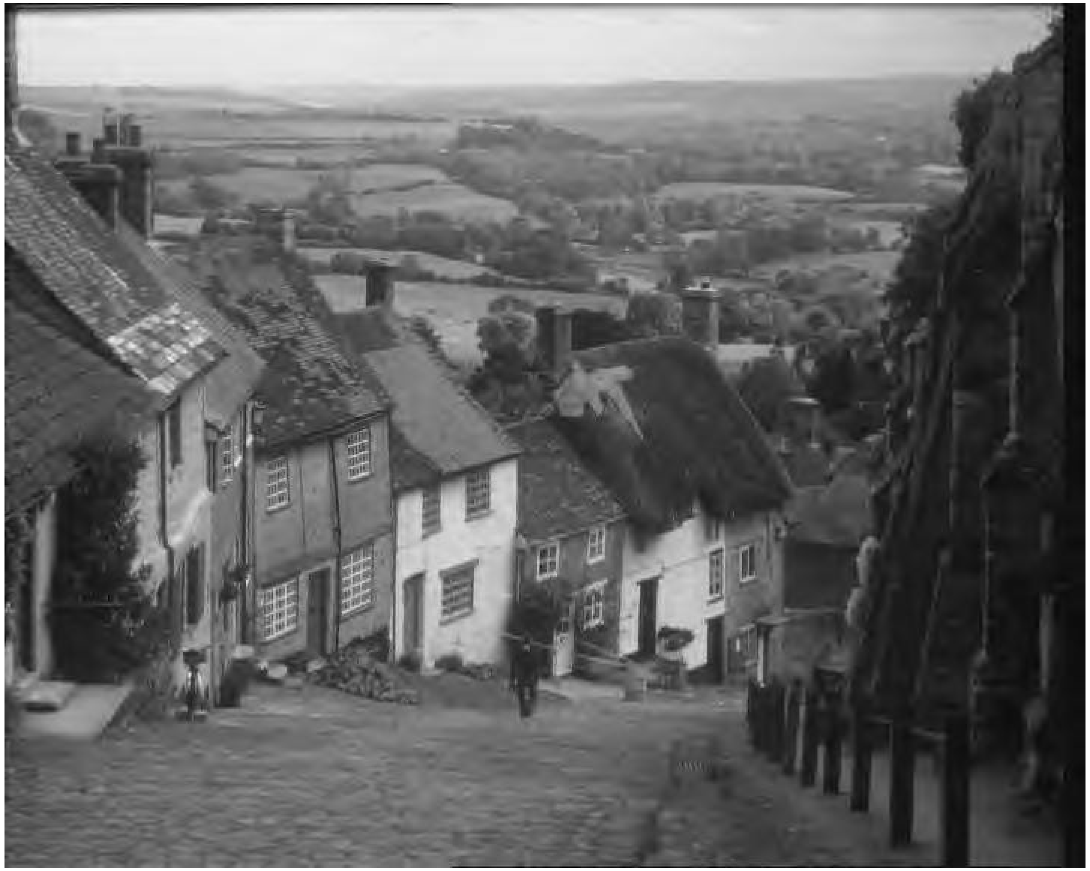
0.0625 bpp



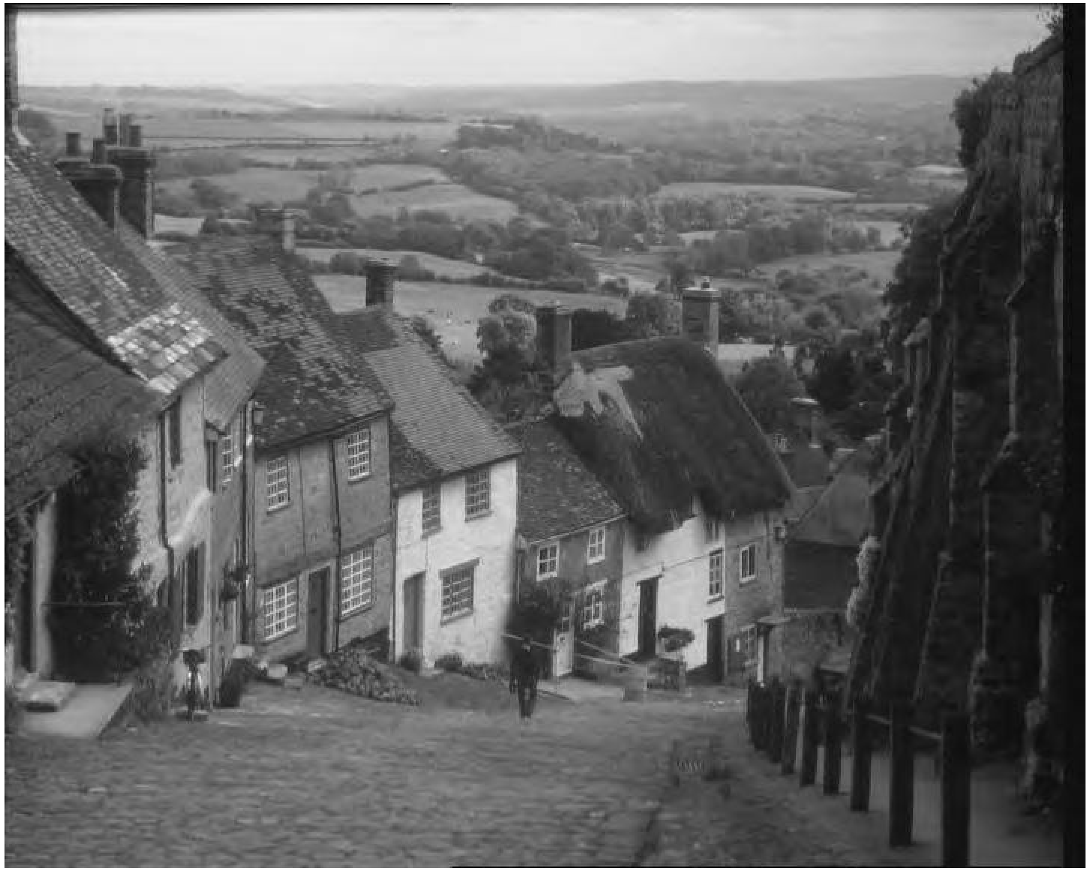
0.125 bpp



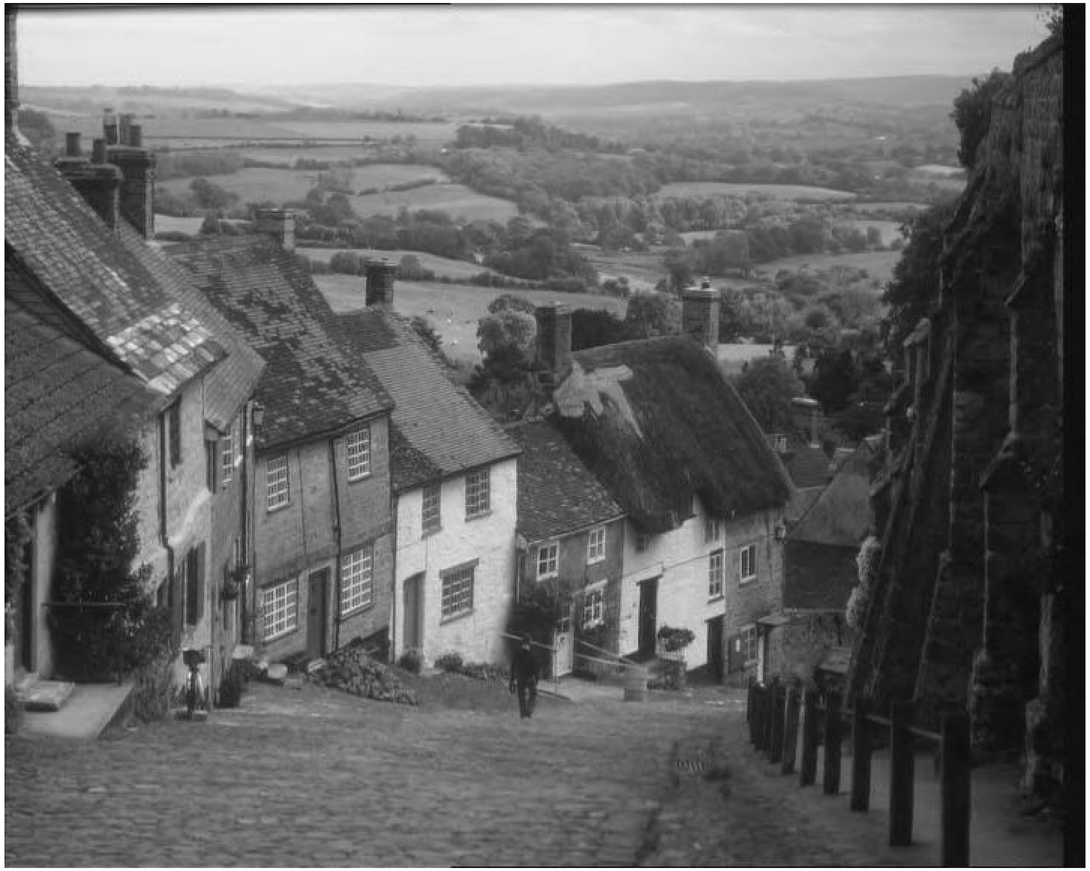
0.25 bpp



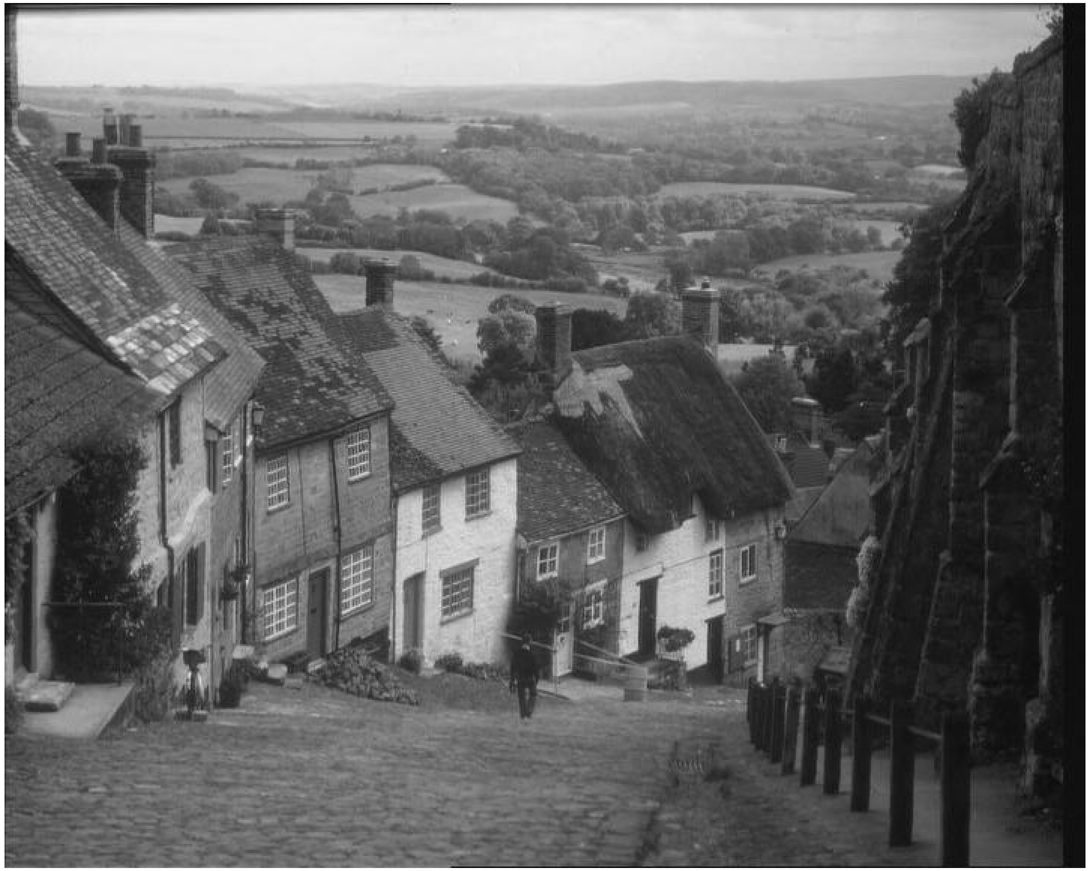
0.5 bpp



1.0 bpp



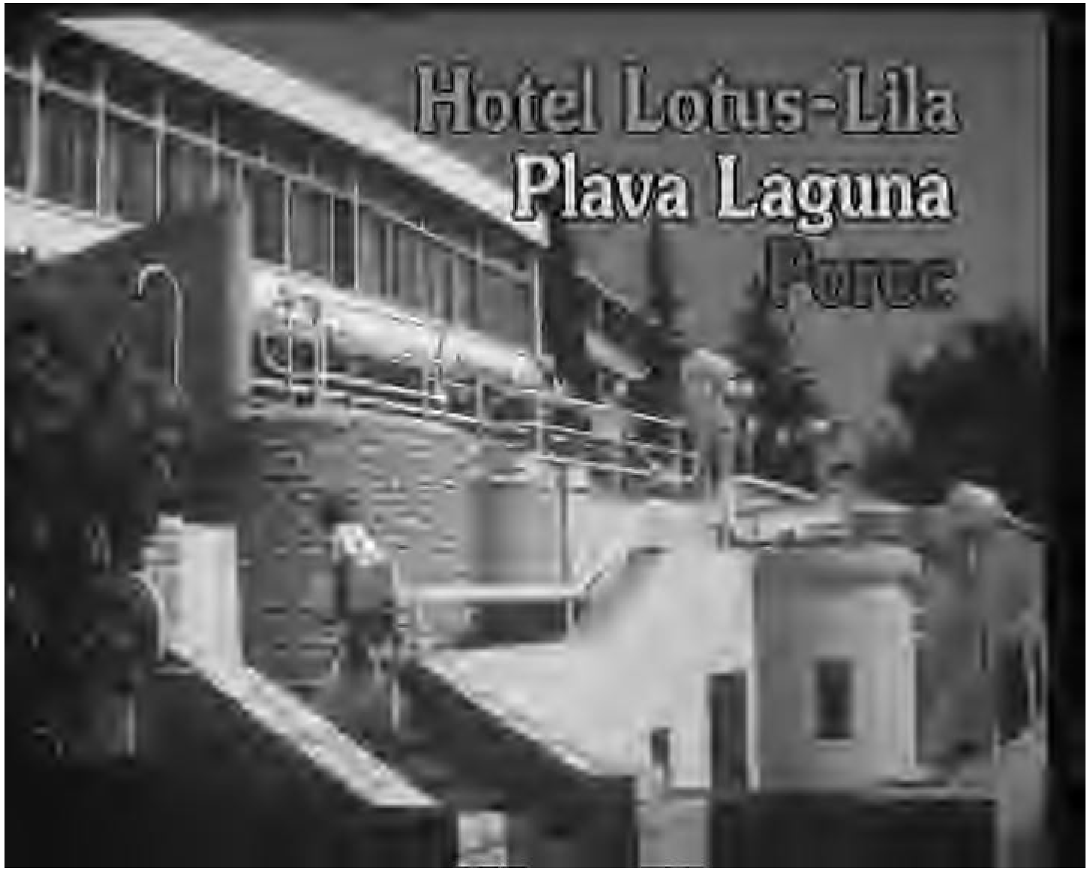
2.0 bpp



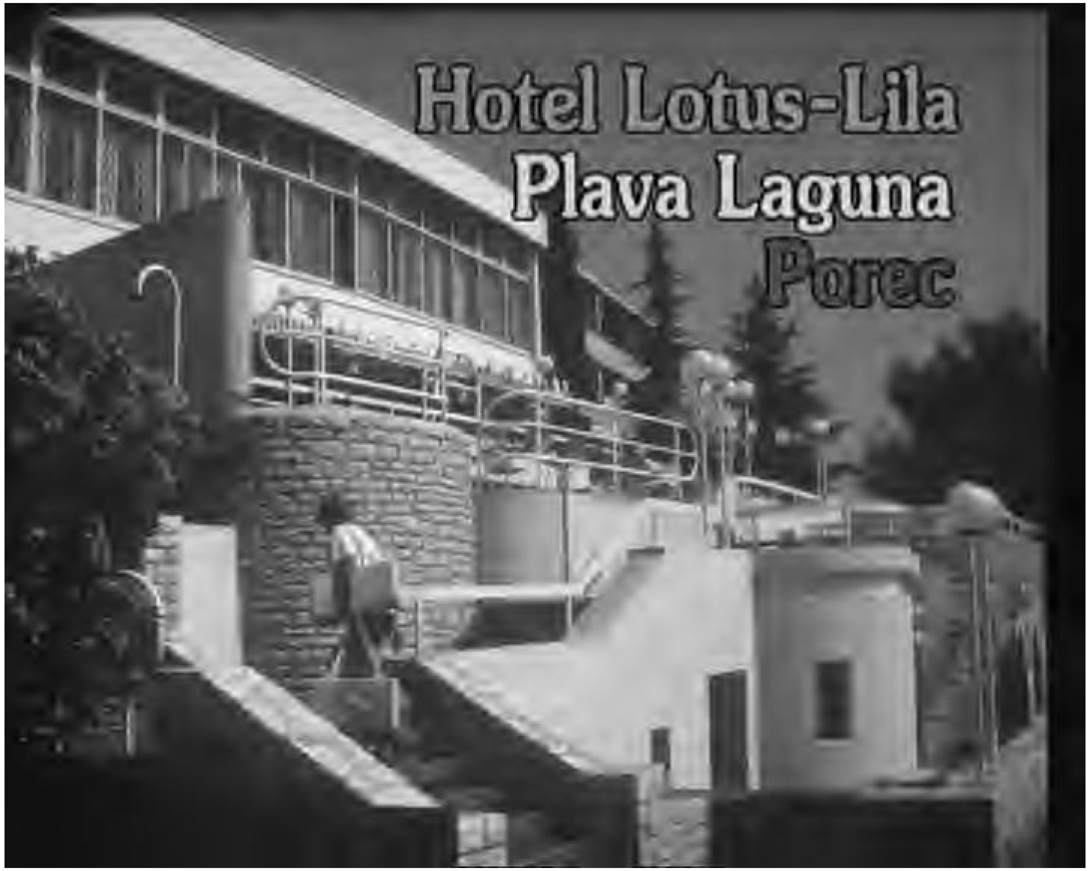
Example:
Progressive by quality

- Image: hotel
- Bitrates: 0.0625 bpp
  0.125 bpp
  0.25 bpp
  0.5 bpp
  1.0 bpp
  2.0 bpp

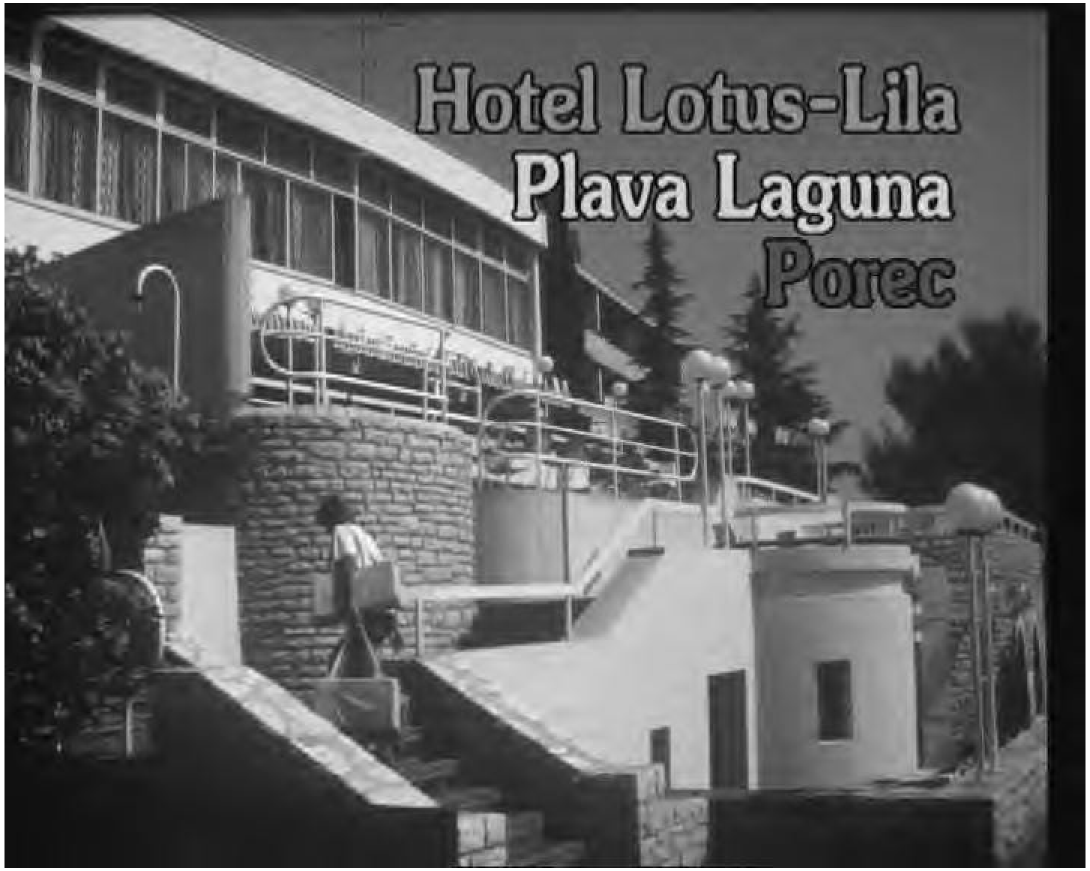
0.0625 bpp



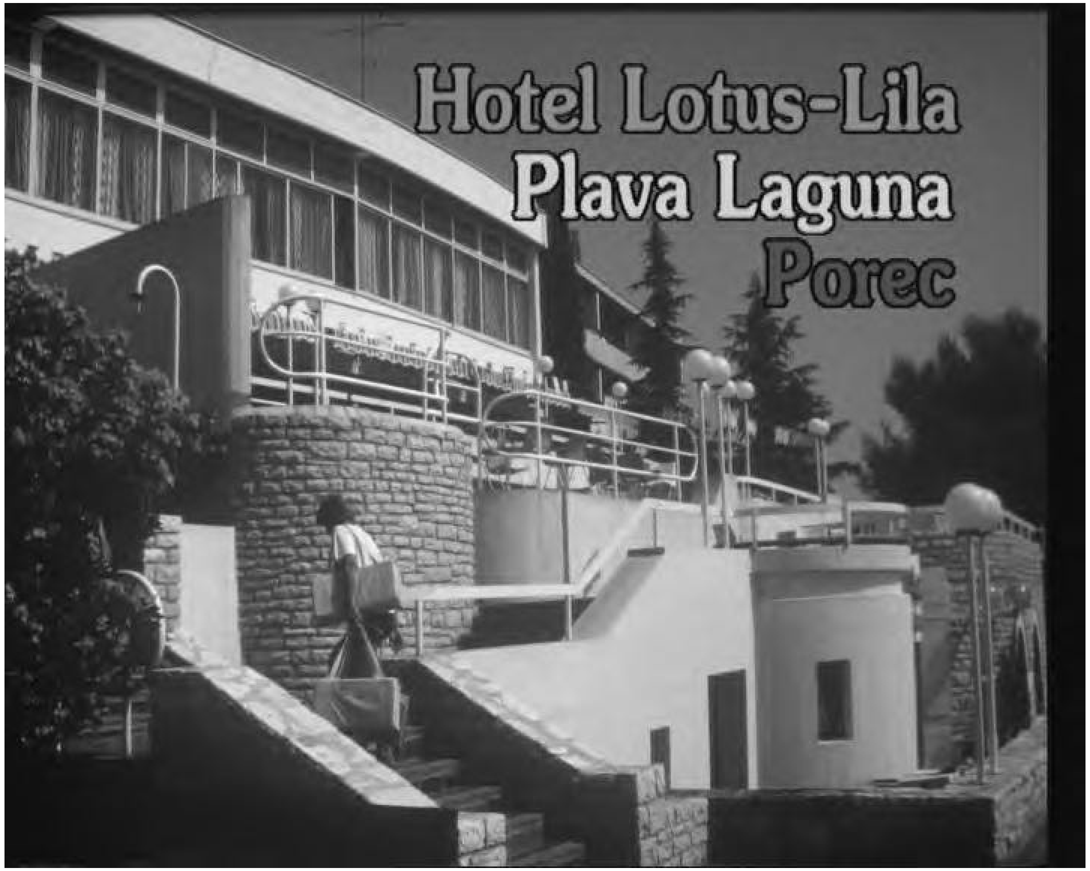
0.125 bpp



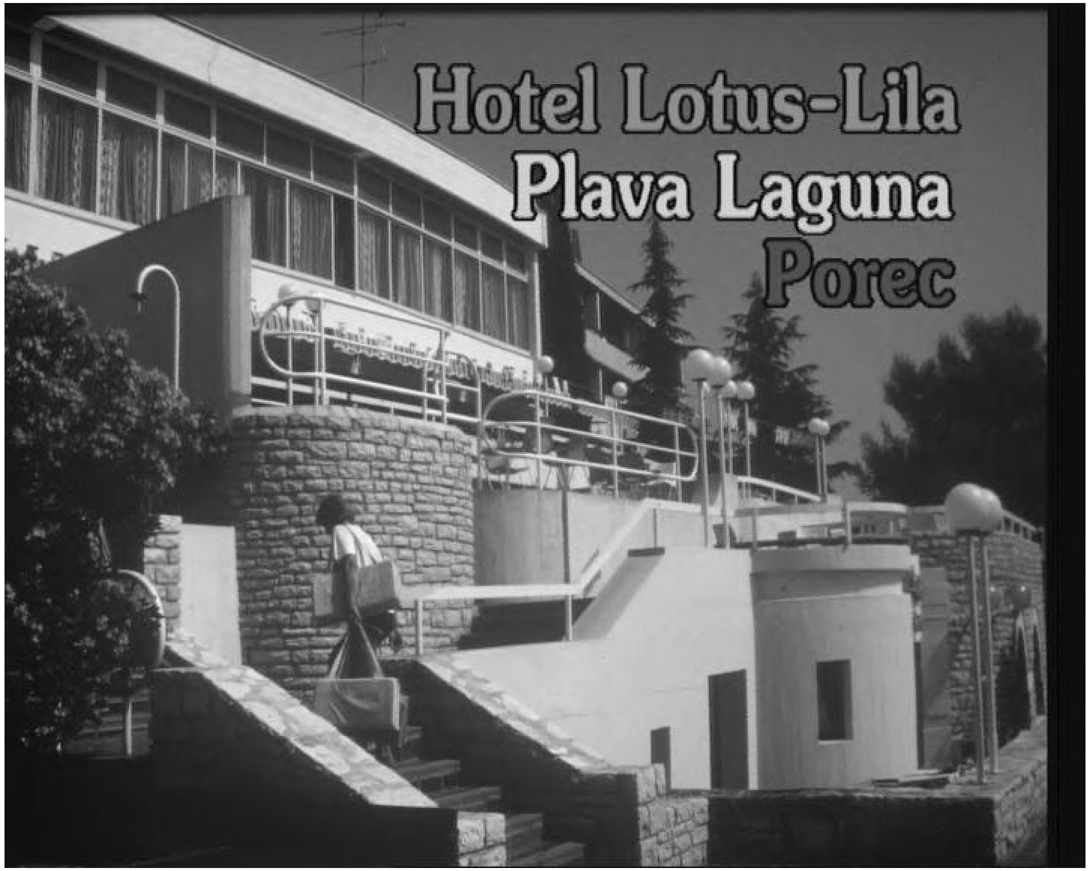
0.25 bpp



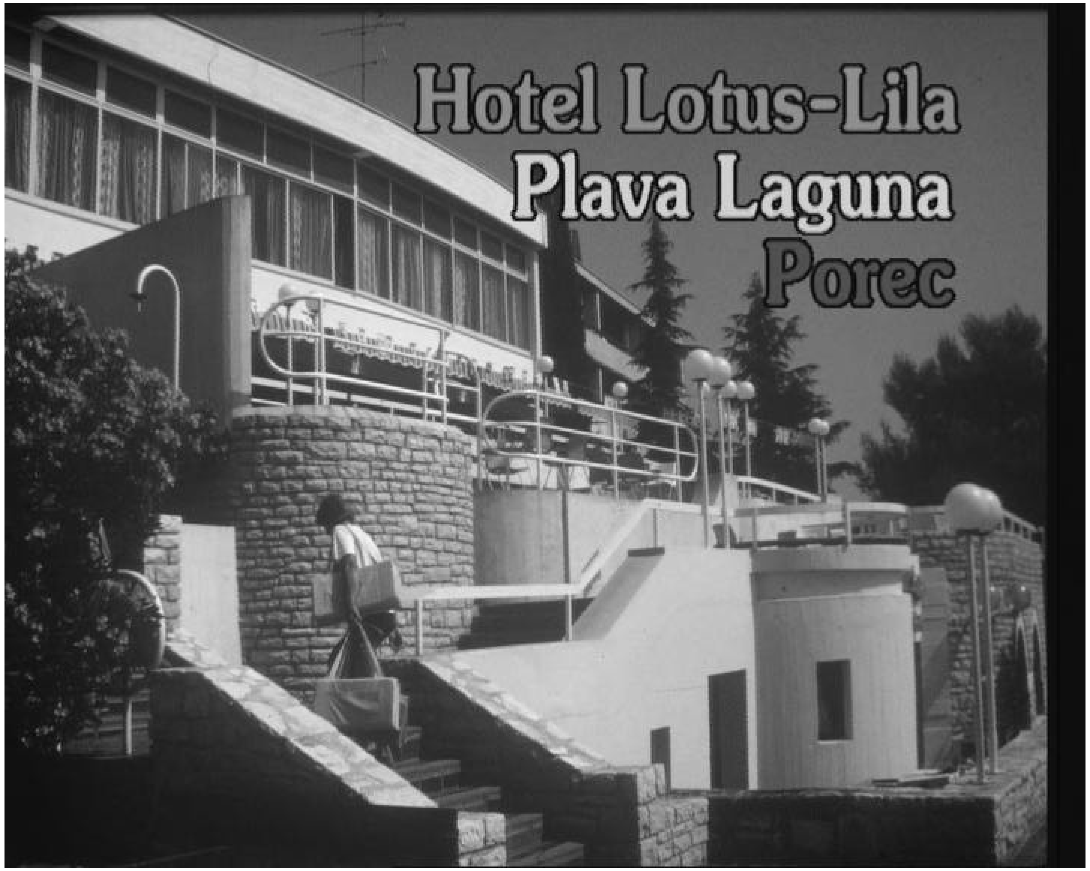
0.5 bpp



1.0 bpp



2.0 bpp



JPEG2000 supported capabilities

- Region-of-Interest coding
  - scaling method
  - max-shift method for implicit region identification
  - arbitrary region shapes
  - any number of regions
  - block aligned mode for non-ROI capable decoders
- Error resilience
  - resync marker option
  - propagation of most errors limited to block boundaries

JPEG2000 supported capabilities (cont'd)

- Tiling
  - any number of tiles
  - rate-control performed jointly over all tiles

- Frames (SSWT)
  - similar to tiles
  - coder operates independently in frames

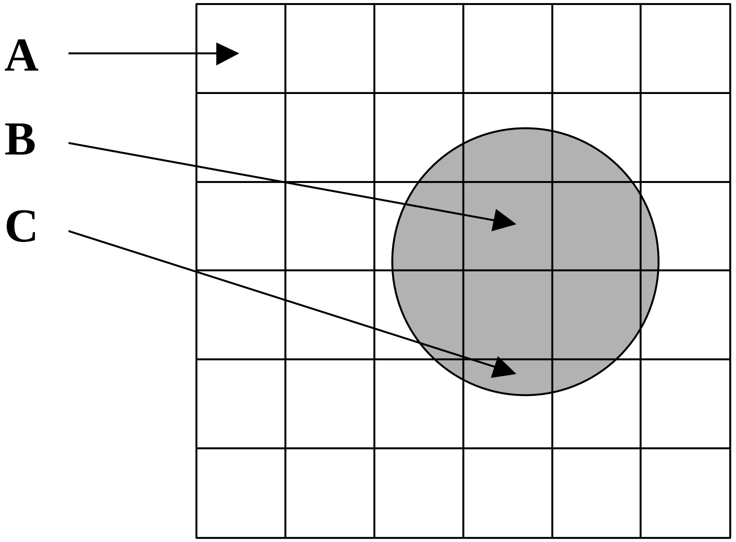
JPEG2000 supported capabilities (cont'd)

- All modes and capabilities work together whenever this is meaningful (almost always)
  - e.g. ROI with frames, overlapping in some levels only, with a packet transform, using convolution horizontally and lifting vertically, and tiled multi-component images.

ROI Coding Using Blocks/Tiles

⌧When encoding an image using tiles/SSWT, individual tiles can be either:

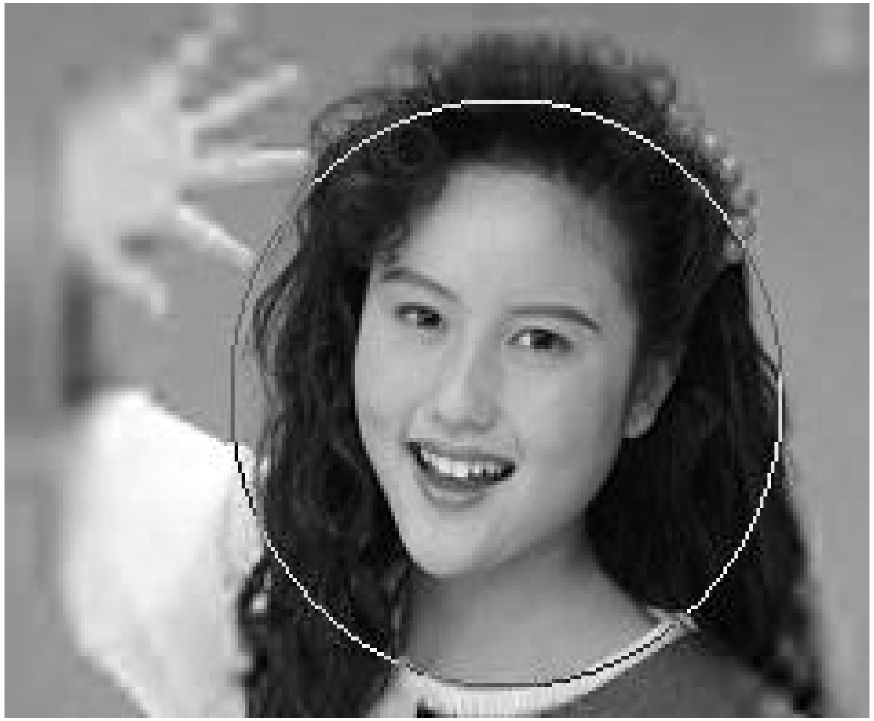
- Outside the ROI (A)
- Inside the ROI (B)
- Partly inside the ROI (C)



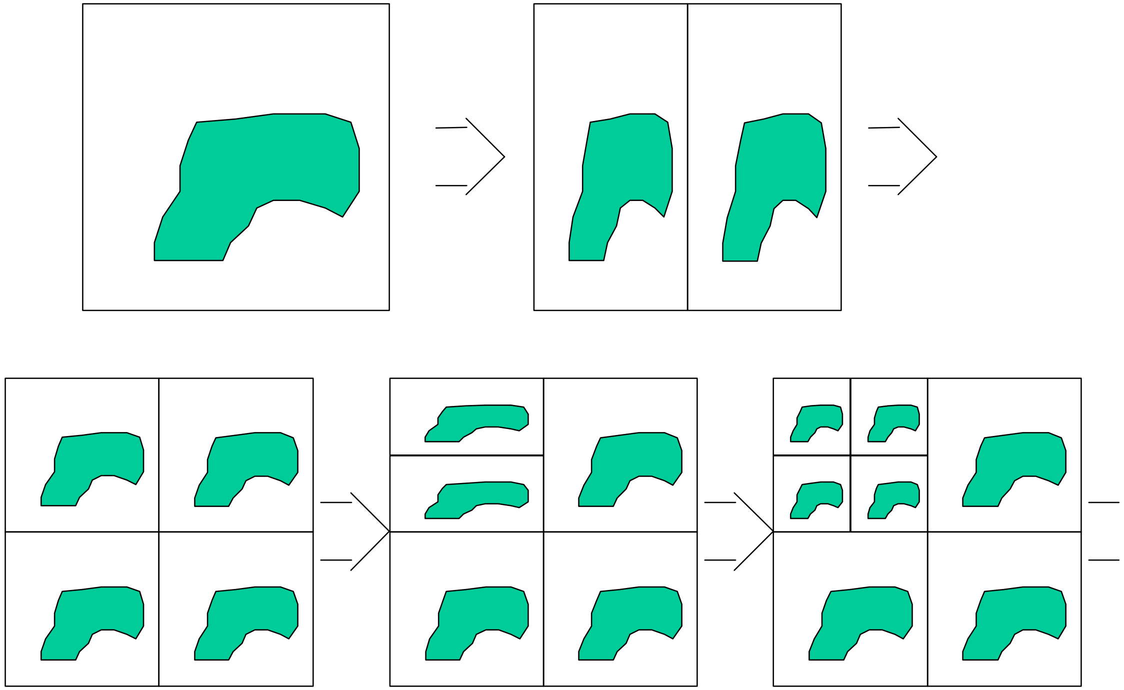
Region Of Interest coding

- Allows certain parts of image to be coded in better quality
  - **BASIC IDEA**:
    - Calculate wavelet transform of whole image/time
    - calculate ROI mask == set of coefficients that are needed for up to lossless ROI coding
    - Encoding is progressive by accuracy and resolution
  - **NOTE**: ROI mask need NOT be transmitted to decoder (location and shape of ROI needs however)

ROI: Some visual results



ROI coding: mask computation

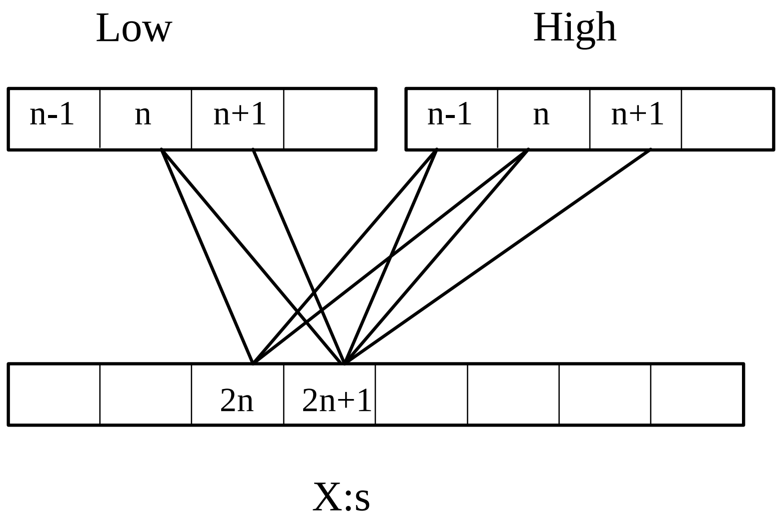


Creation of ROI mask

- The ROI masks are acquired by looking at the inverse transform

- For each pixel (X) that is in the ROI, the low and high frequency coefficients (L:s and H:s) that are needed to reconstruct the pixel, are included in the ROI mask

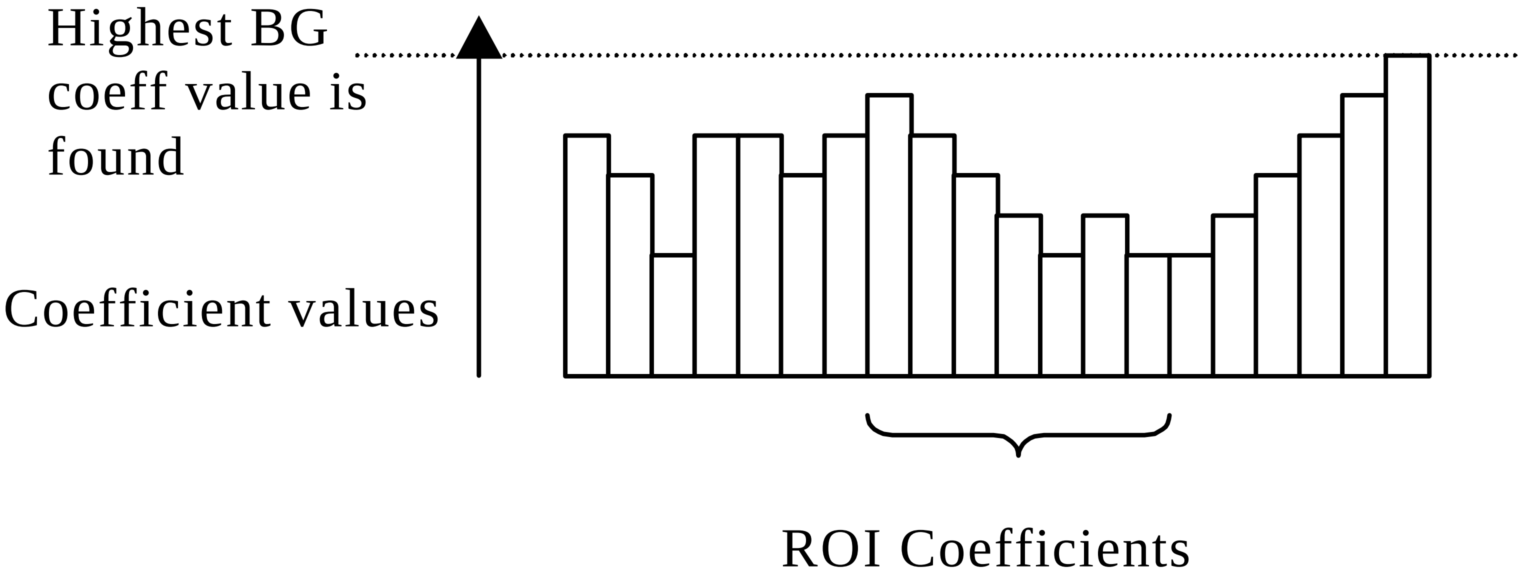
Inverse transform of the 5-3 filter



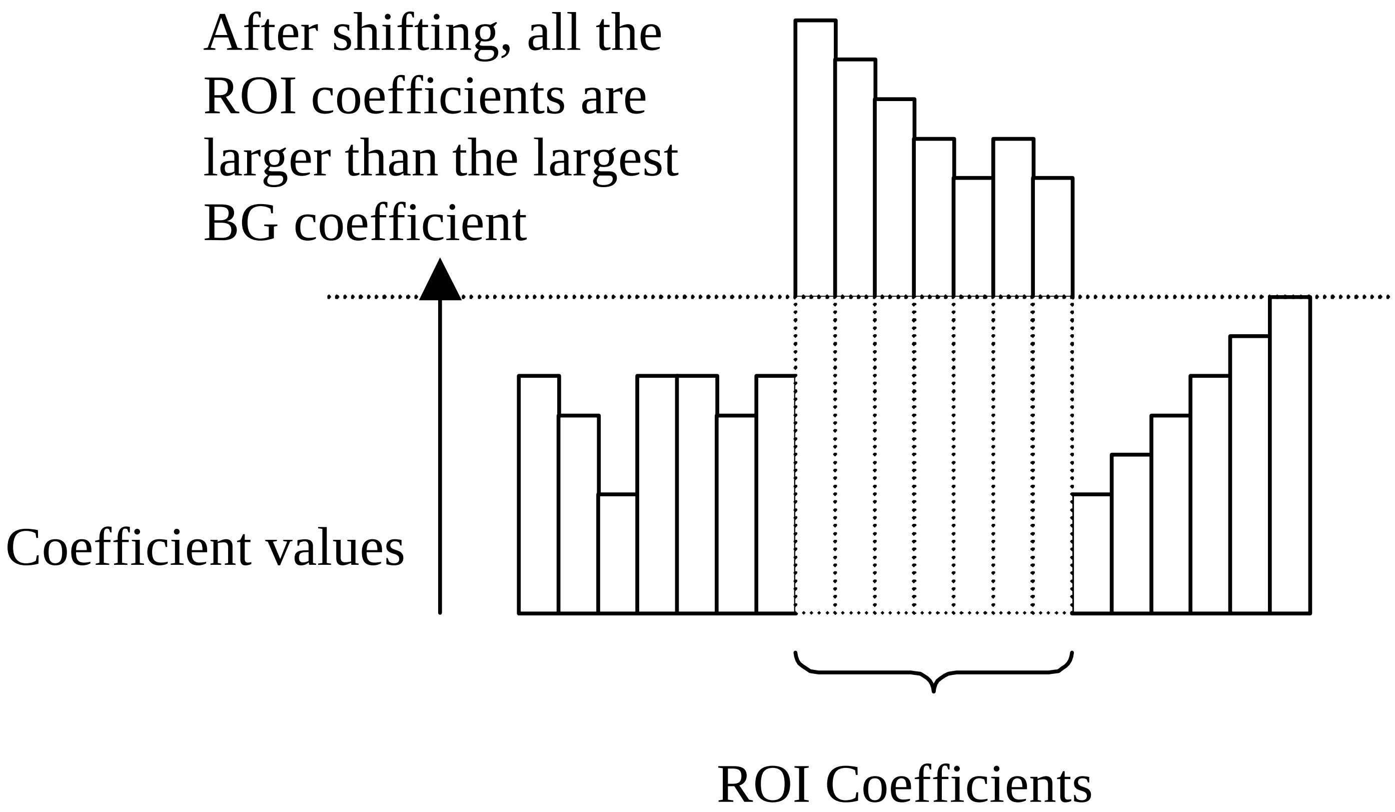
Modes for ROI coding

- **'Block' based mode**
  - Comes for free but not the optimal solution for static ROI
  - Useful in dynamic ROI coding
- **'Arbitrary shape scaling' based mode**
  - Scale ROI mask coefficients up (decoder scales down)
  - During encoding the ROI mask coefficients are found significant at early stages of the coding
  - ROI always coded with better quality than BG
- **MaxShift Method**

ROI Scaling based method



ROI MaxShift method


After shifting, all the ROI coefficients are larger than the largest BG coefficient
Coefficient values
ROI Coefficients

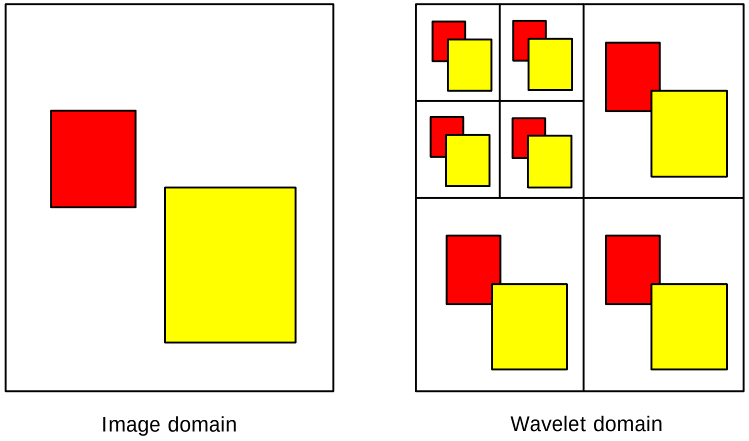
ROI Maxshift mode: what is the gain?

- Support for arbitrary shaped ROI’s with minimal complexity
- No need to send shape information
- No need for shape encoder and decoder
- No need for ROI mask at decoder side
- Decoder as simple as non-ROI capable decoder
- Progression by accuracy and by quality is supported

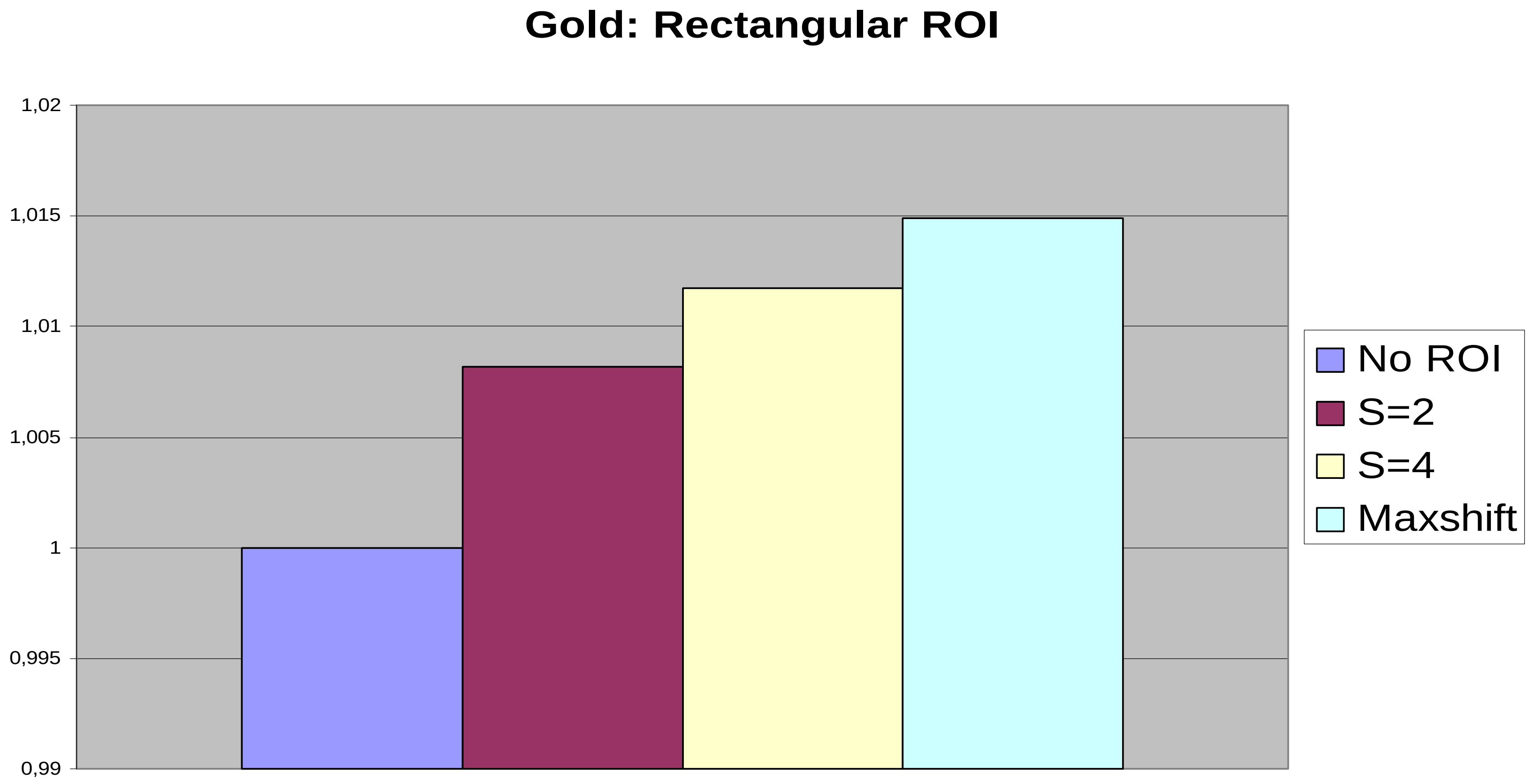
Multiple ROI coding

- ROI’s might overlap in the wavelet domain

- Coefficients in overlapped area are coded as belonging to the highest quality ROI

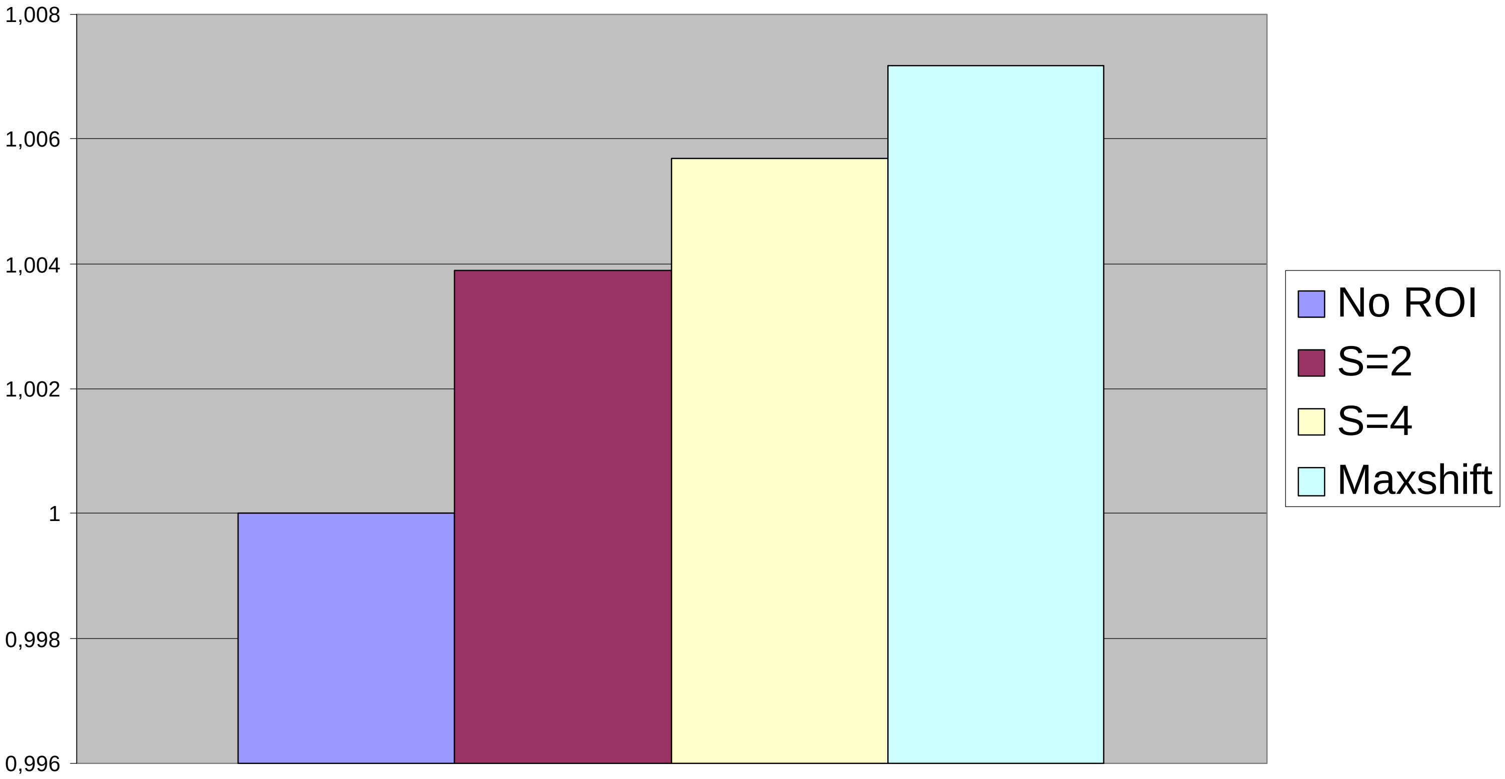


ROI coding: what do we pay?
Lossless image coding with ROI



ROI coding: what do we pay?

Lossless image coding with ROI



Woman: Circular ROI