Motion JPEG-2000 Interactive Region-of-Interest Coding on Mobile Devices

Glenn Van Wallendael, Sammy Lievens, Tim Dams and Peter Schelkens

Hogeschool Antwerpen
Department of Applied Engineering

Vrije Universiteit Brussel
Department of Electronics and Informatics (ETRO)

Abstract: Motion JPEG2000 (**MJP2K**) has several mechanisms that can be used to introduce a Region-of-interest **(ROI)** in a videostream. These methods are based on coefficient scaling, code-block selection and on tiles (using the SOP, PLT or TLM markers). We developed a proof-of-concept of interactive ROI coding within Motion JPEG2000 in order to compare these methods (both on a regular client pc as on PDA). Results show that every method has or a significant impact on the PSNR, or on the computational overhead. For real-time applications the tile-based methods perform better than the other methods, even though their visual results are less appealing.

Problem

Heterogeneous networks and end-user devices:

•*Network variables: available* bandwidth, error characteristics, etc.

•*End-user devices:* computational power, screen resolution, etc.

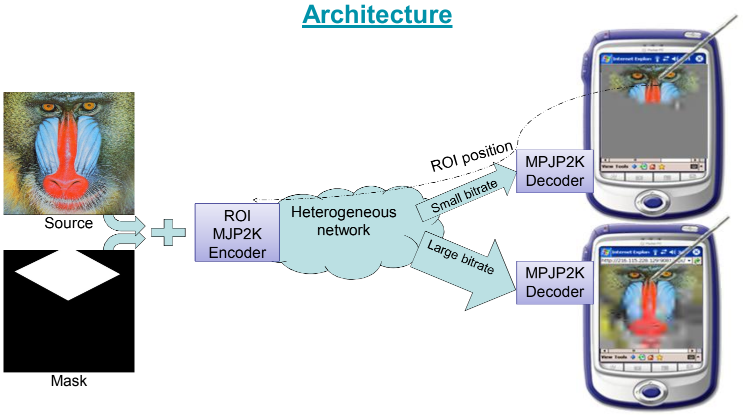
Classical way of overcoming this problem is by transmitting several independently encoded versions of the same video material specifically targeted to the different terminal and connection characteristics.

Two possible solutions:

- *Scalable video coding*
- *ROI-based video coding*

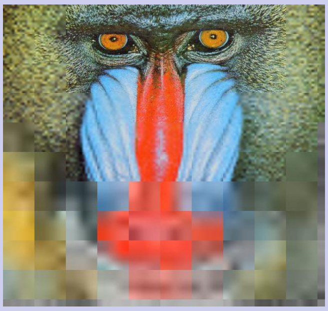
ROI-based video coding has the advantage over scalability that the version of the material for devices with a smaller screen size, connect via a low bandwidth connection, does not have to be a lower-quality dyadically downscaled version of the original, but can be a cut-out of the video focussing on the most important part of the scene, without loss of resolution or quality.

Architecture

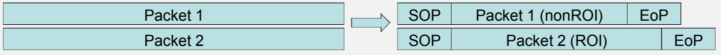


ROI Mechanisms in MJP2K

Mechanism 1: Tiling and markers



SOP markers (Start of Packet)



Packets (of a tile) contain more data in the ROI, due to the SOP marker, since all data in front of the SOP is cut off. Flexibility dependent of amount of packets. (overhead vs flexbility)

TLM markers (Tile-part Lengths: Main header)

TLM (optional) allows random accessing of tiles => resorting algorithm can be used to move ROI-tiles to front of bitstream
Extreme form of ROI: dependent of available bandwidth, only cutoff up to tiles.

PLT markers (Packet Lengths: Tile-part header)

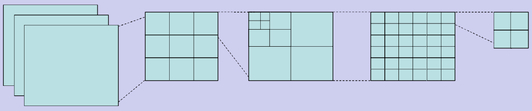
PLT (optional) allows random accessing of packets within the codestream => same mechanism as with TLM, but now on packets.

PLM markers (Packet Lengths: Main header)

Collection of each PLT markersegment of all tile-parts => same mechanism possible.
Note: PLT+PLM is possible, but lengths need to be consistent

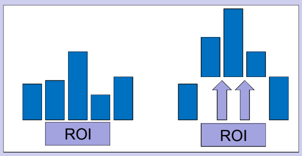
Mechanism 2: Code-block selection

JPEG2000 Data Hierarchy :
Components 》Tiles 》Subbands 》Precincts 》Codeblocks



Reorder packets (which contain entropy encoded code-blocks).
Result:ROI moved up in coded bitstream and thus transmitted more rapidly.

Mechanism 3: Coefficient scaling

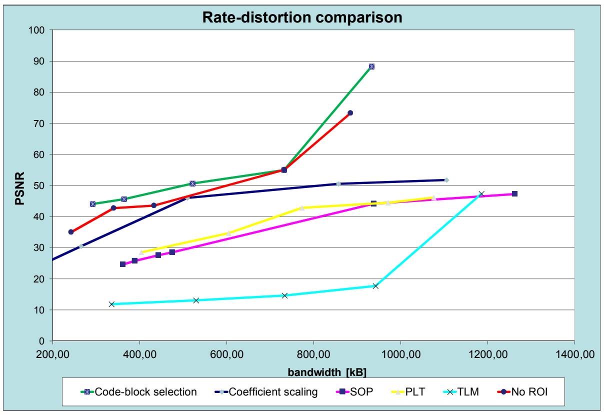


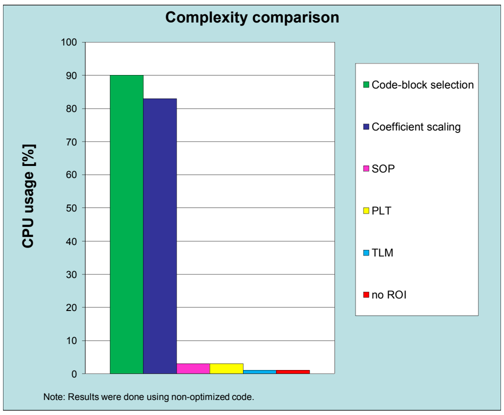
Upshift in the MAXSHIFT method

1. ROI mask creation: 1-bit mask that indicates which wavelet coefficients belong to the ROI.
2. Upshift: Upshift corresponding pixel values with an amount equal to the maximum number of (coefficient) bits in the original picture.
3. Compression: Compress the result to a new MJP2K-file.

Result: ROI moved up in coded bitstream and thus transmitted more rapidly.

Test Results





Note: Results were done using non-optimized code.