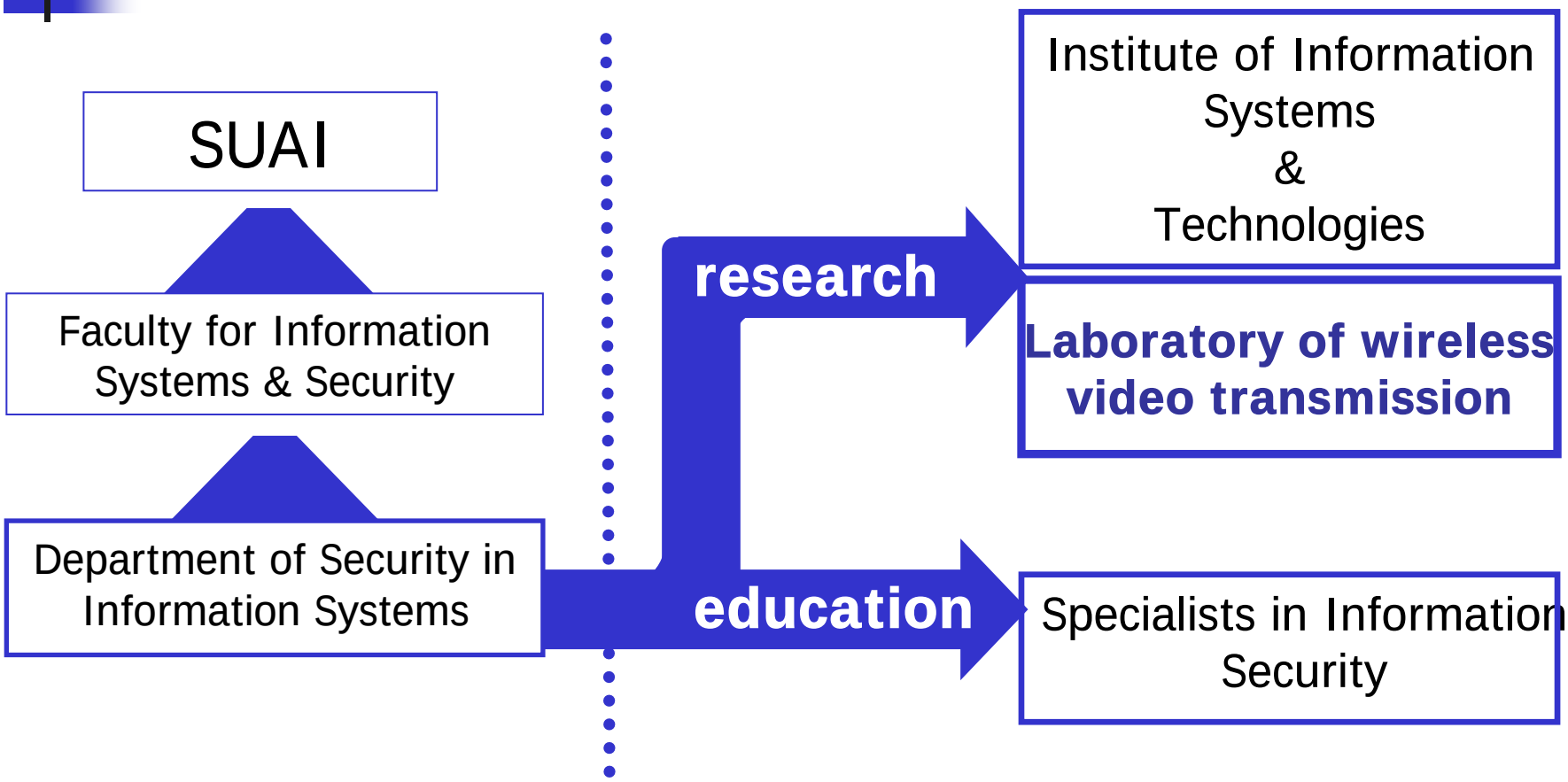
# Use case oriented video compression algorithms enhancement

Under the supervision of Dr. Andrey M. Turlikov

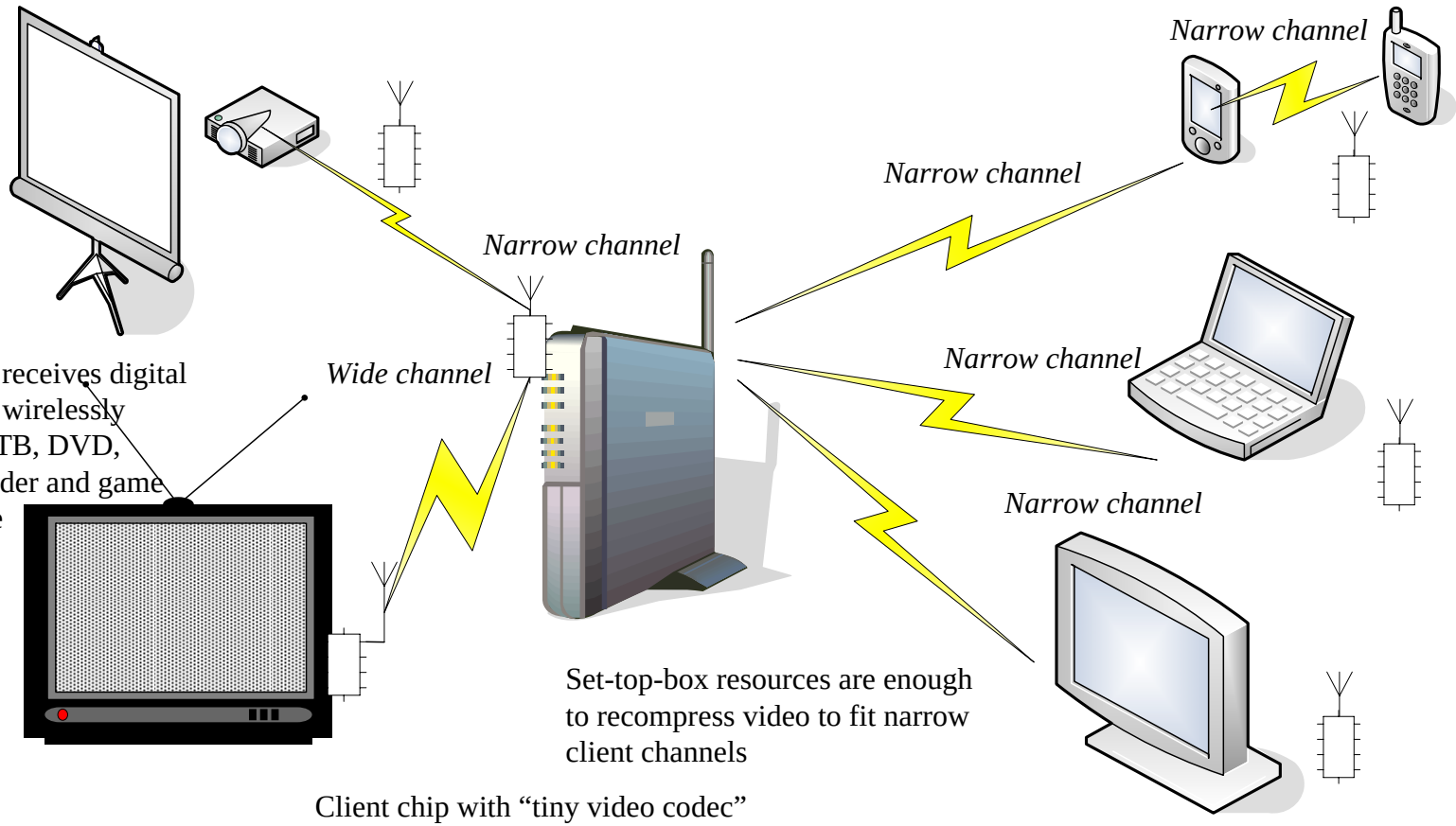
Ann Ukhanova
St. Petersburg University of Aerospace Instrumentation

# What is the laboratory of wireless video transmission?



# Wireless video transmission: use cases

**Variety of multimedia applications forms individual requirements**

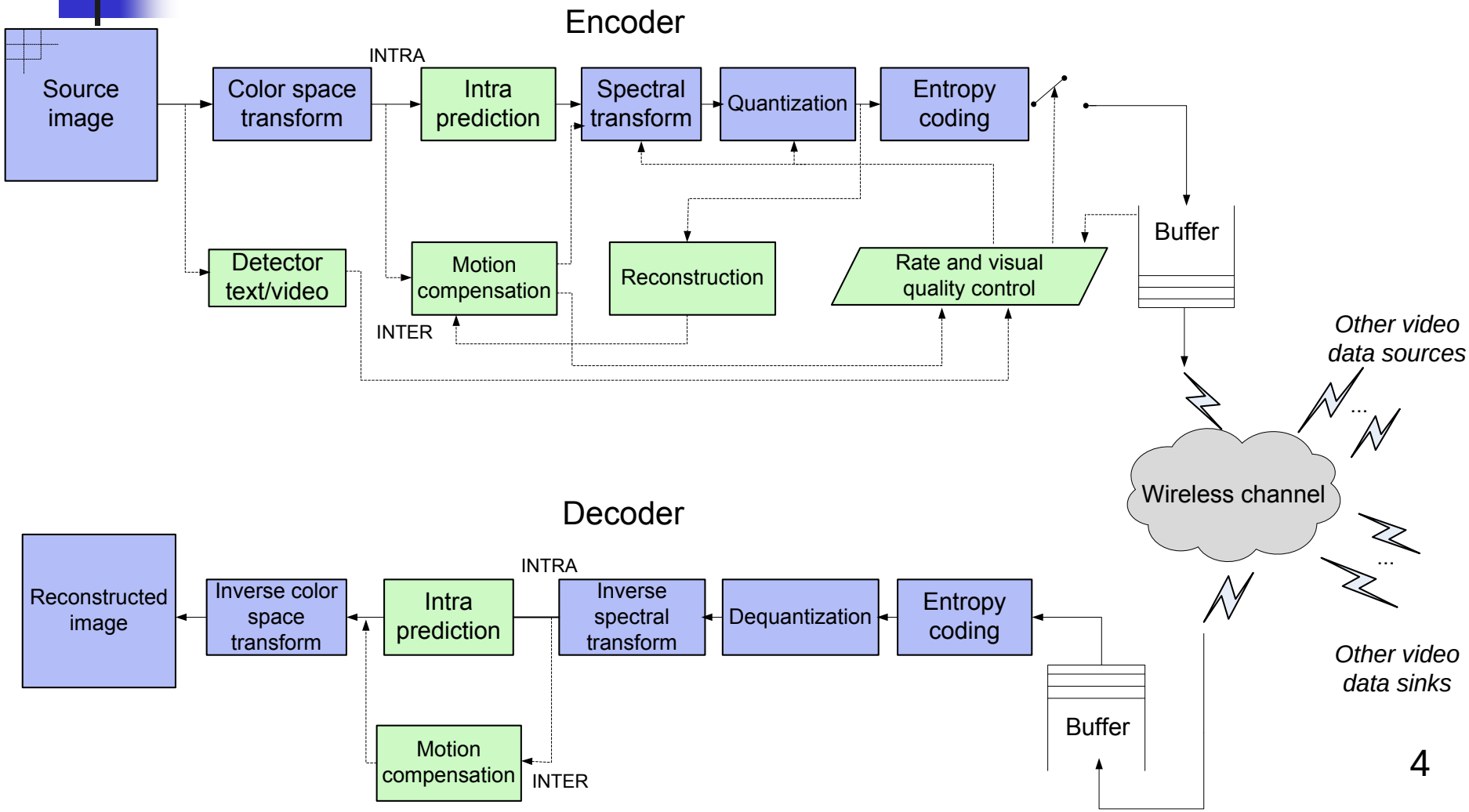


# Compression algorithms characteristics

What are the primary characteristics for the compression algorithm?

| Characteristics \ Algorithms | JPEG-LS | JPEG | JPEG2000 | H.264/AVC |
|---|---|---|---|---|
| **Complexity** | Bad | **Good** | Very bad | **Very good** |
| **Memory consumption** | **Good** | **Good** | Bad | **Very good** |
| **Compression ratio** | Very bad | **Good** | **Very Good** | **Very Good** |
| **Rate control** | No | No | **Yes** | **Yes** (in proposals) |
| **Progressiveness** | No | **Yes** | **Yes** | No |
| **Packetization** | No | No | **Yes** | **Yes** |

# Video codec typical scheme

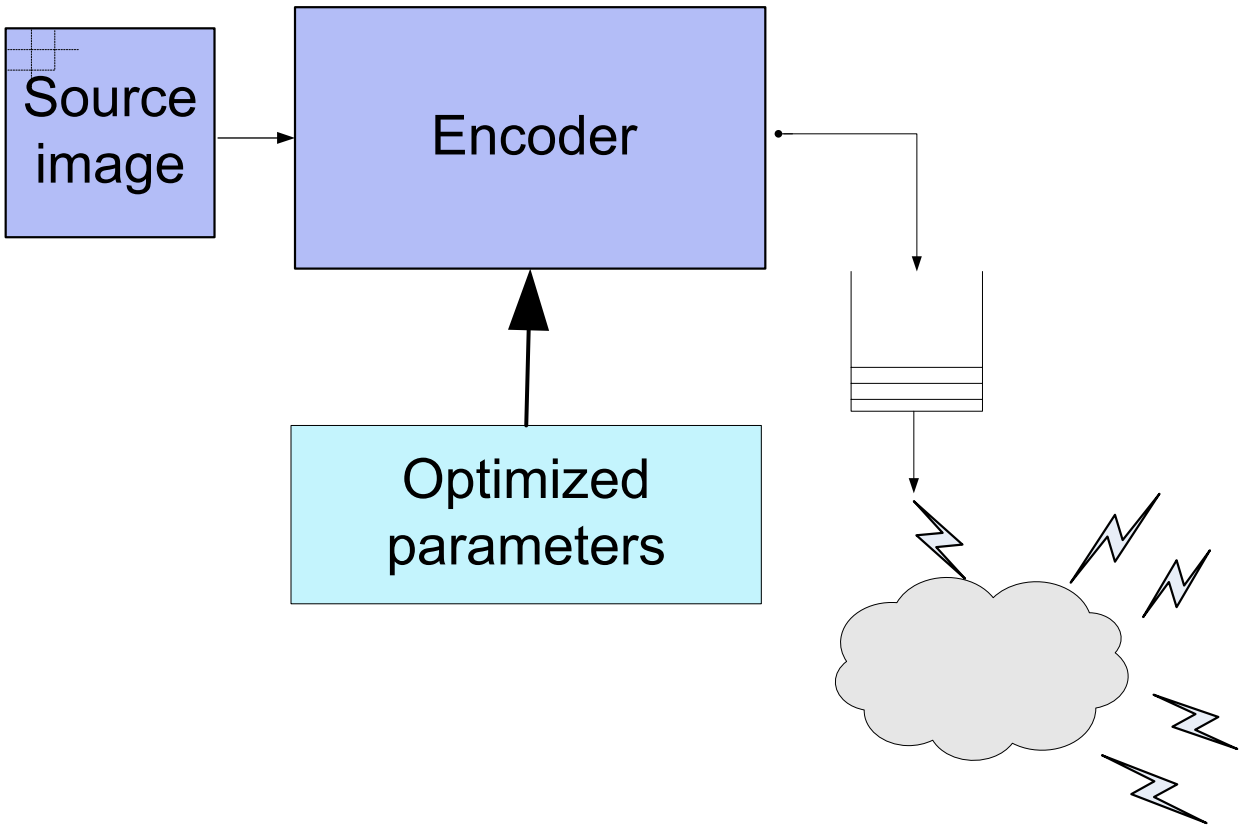


# Our approaches

- **No changes to encoder**
  - Parameters adjustment
- **External blocks**
  - Alternative color space transform
  - Detector “text”/“video”
  - Rate and quality control
- **Changes in the coding algorithm**
  - Progressiveness
  - Entropy coding

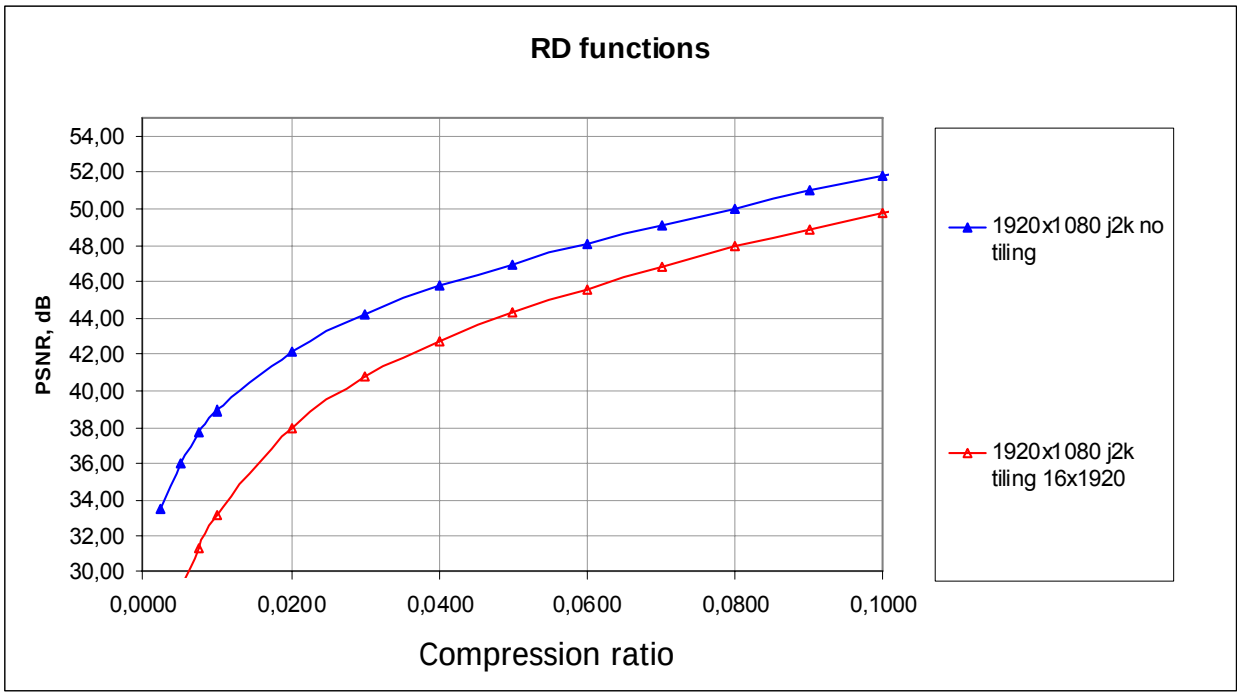
# Parameters adjustment

**No changes to encoder**



# Tiling option for memory consumption

**Goal:** decrease memory consumptions at coder/decoder (only for one tile)





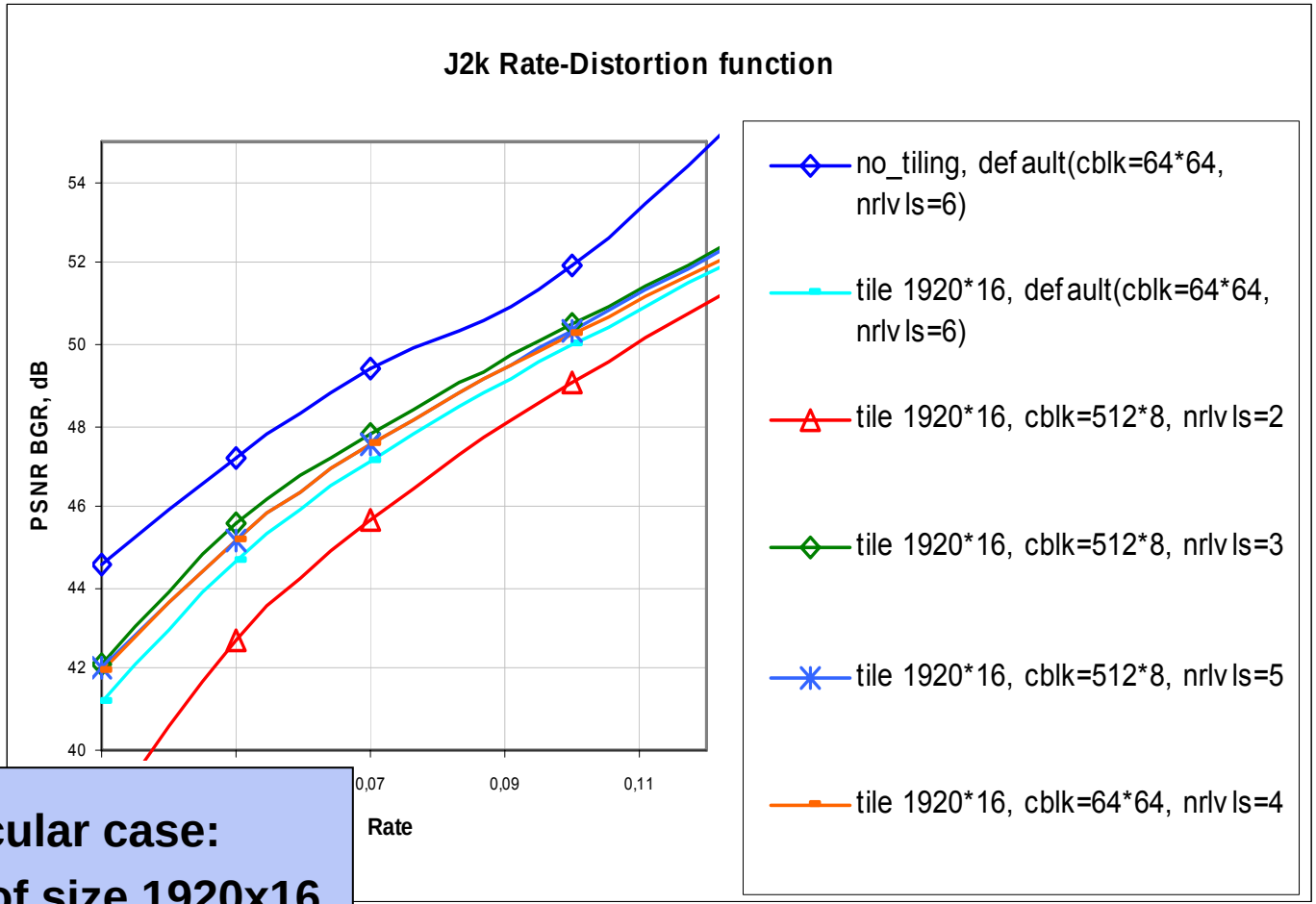
tiling provides image partitioning into rectangular and non-overlapping tiles

JPEG2000 algorithm without tiling shows much better results than version with tiling because the first one operates with the whole image

# Optimal parameters for tiling

**Goal:** coordinate coding parameters with each other

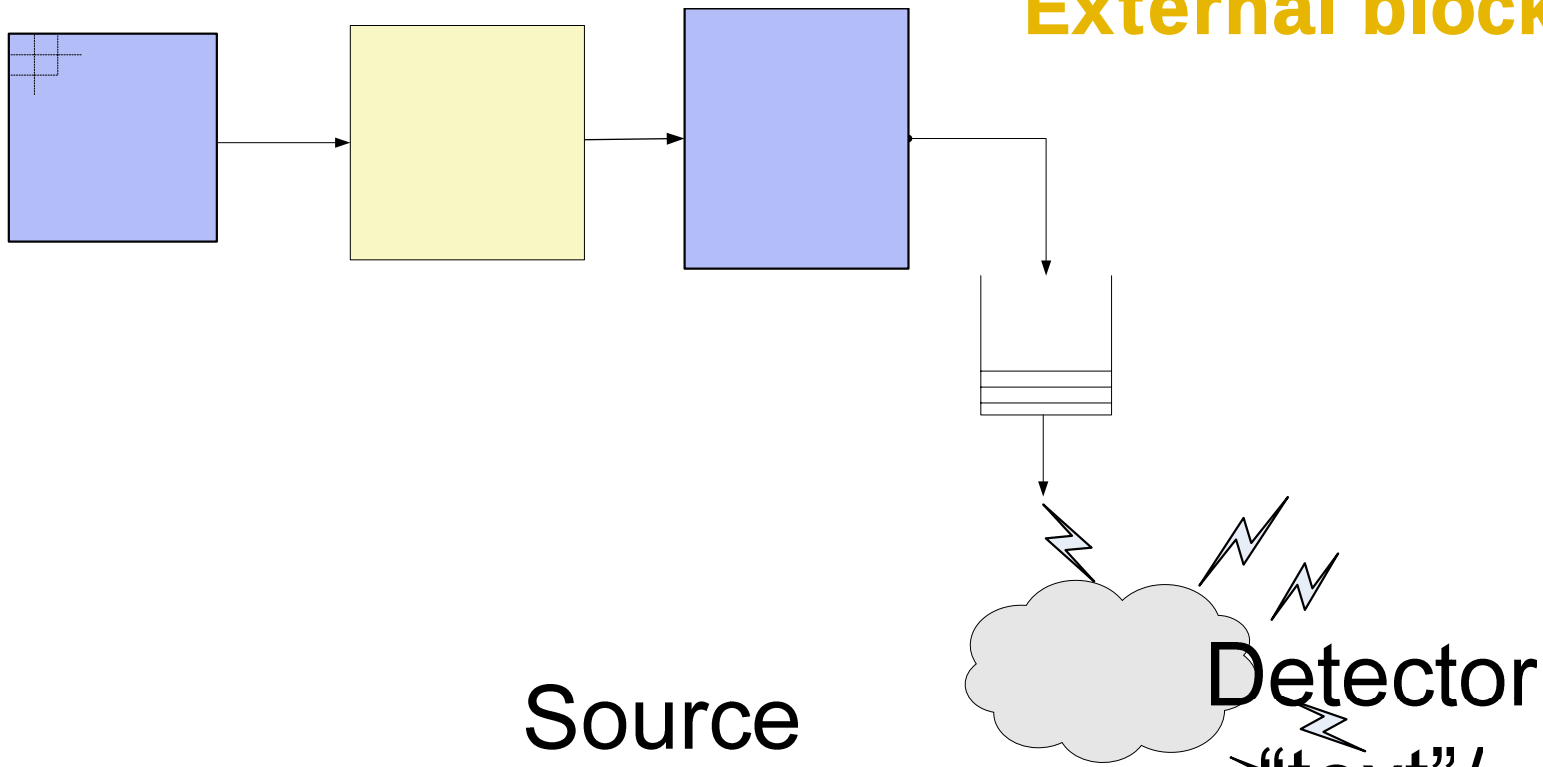
- **Coding parameters:**
- tilewidth
- tileheight
- codeblock width
- codeblock height
- number of resolution levels etc.




**particular case:**
**tiles of size 1920x16**

# Detection “text”/“video”

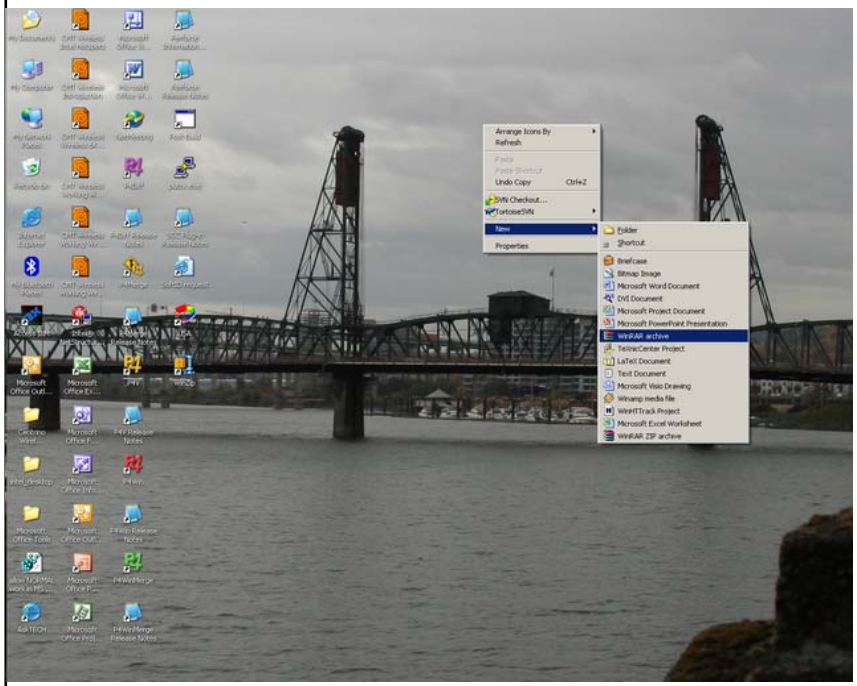
**External blocks**

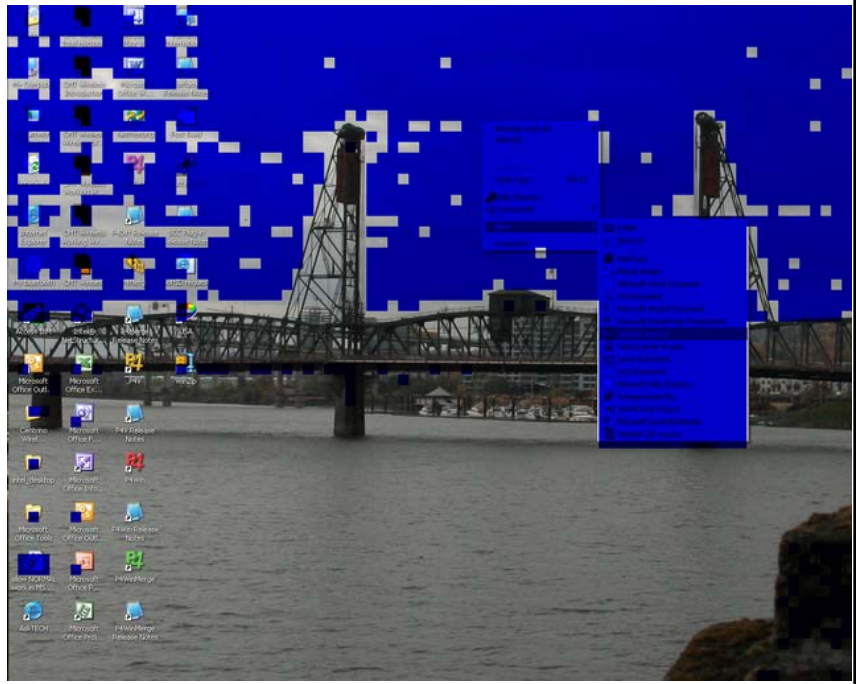


# “Text”/“video” detector - 1

Original image

Image after “text”/“video” detection

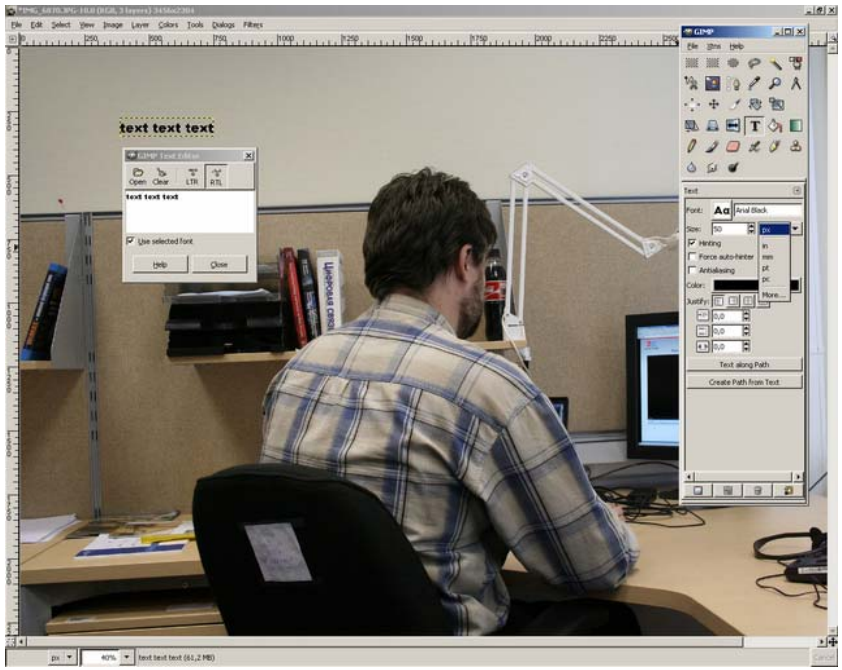


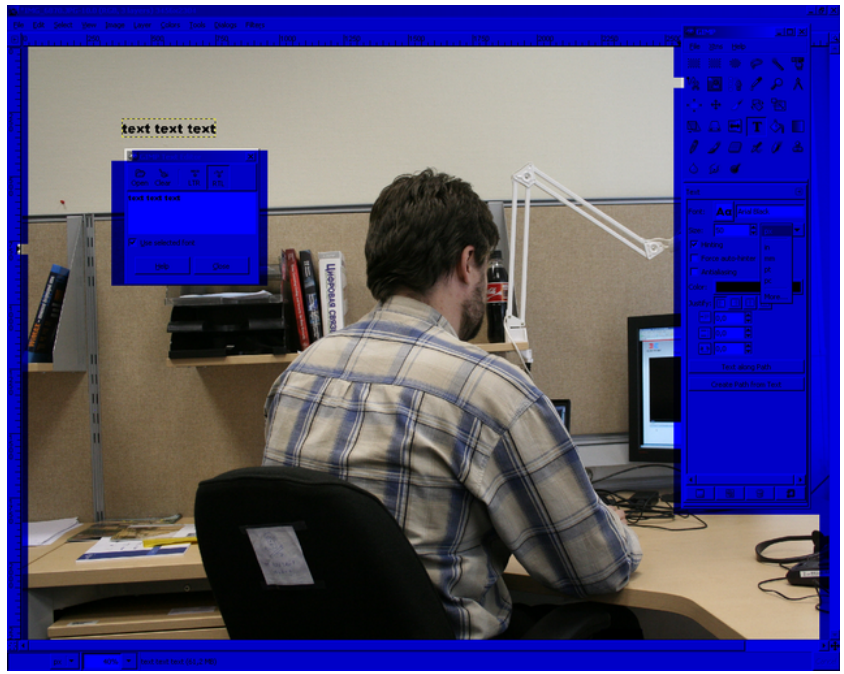


# “Text”/“video” detector - 2

Original image

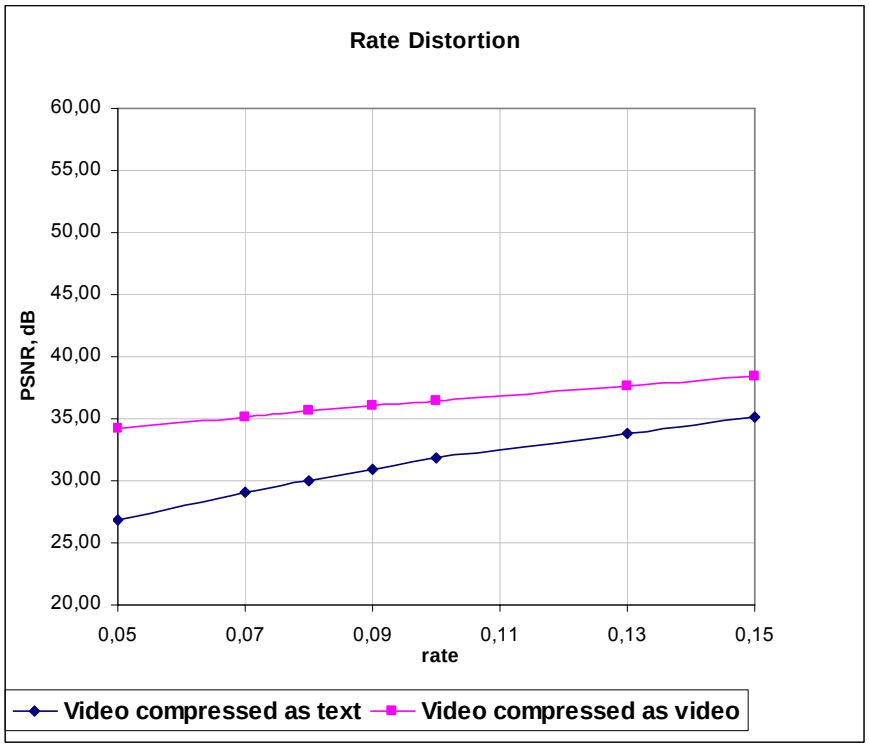
Image after “text”/“video” detection

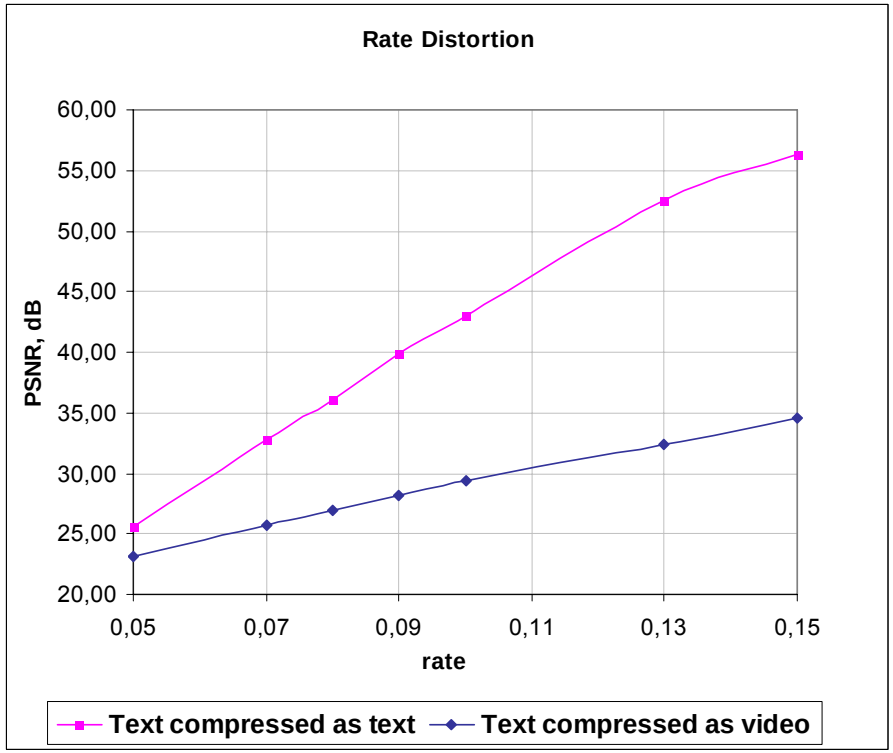


# Why should we discern “text” and “video”?

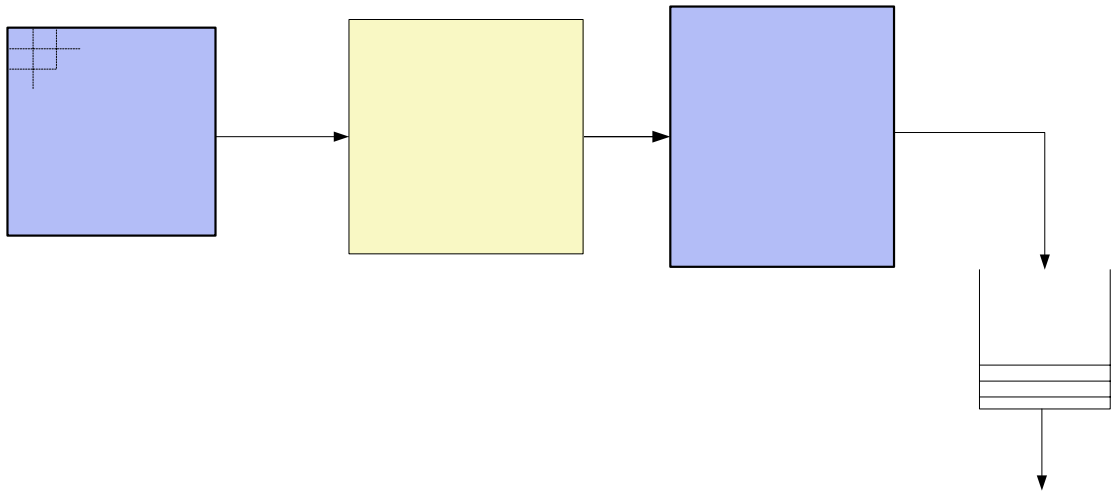
We should compress “text” and “video” with different parameters



# Color space transform

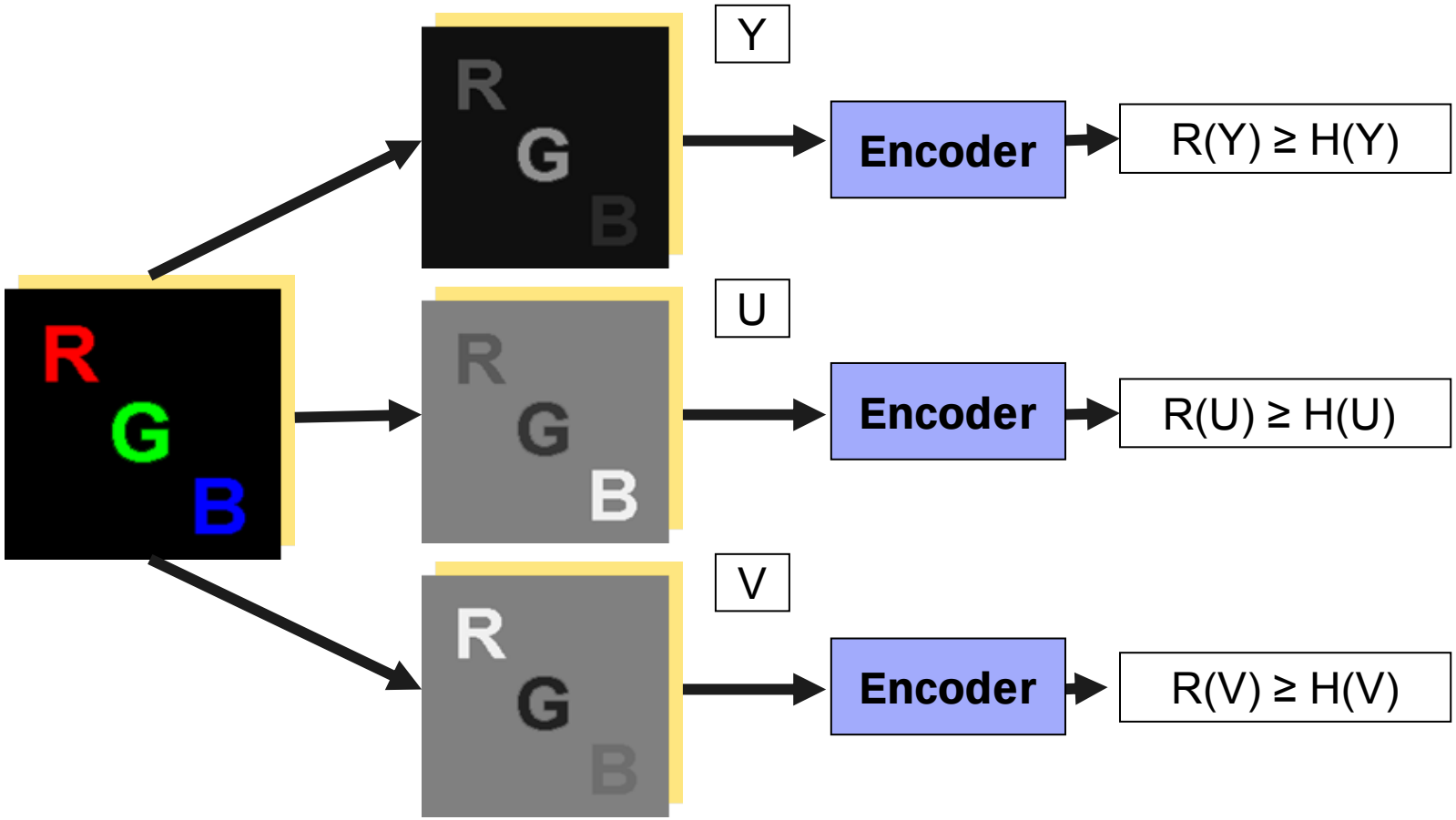
**External blocks**



Source image

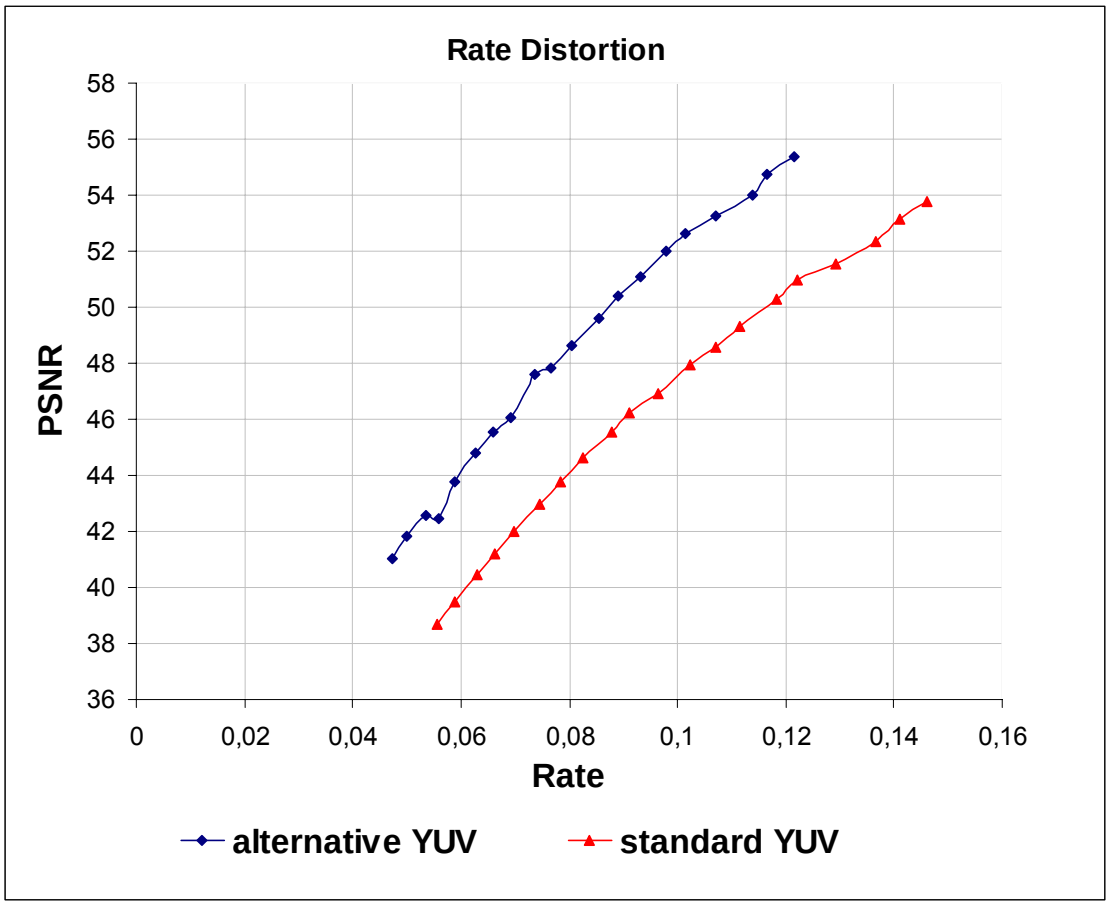
Color space transform

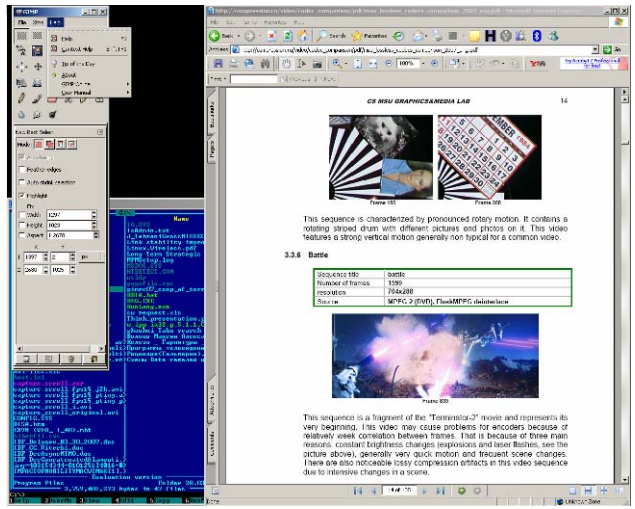
# Standard color YUV transform




R ≥ H(V) + H(U) + H(Y)

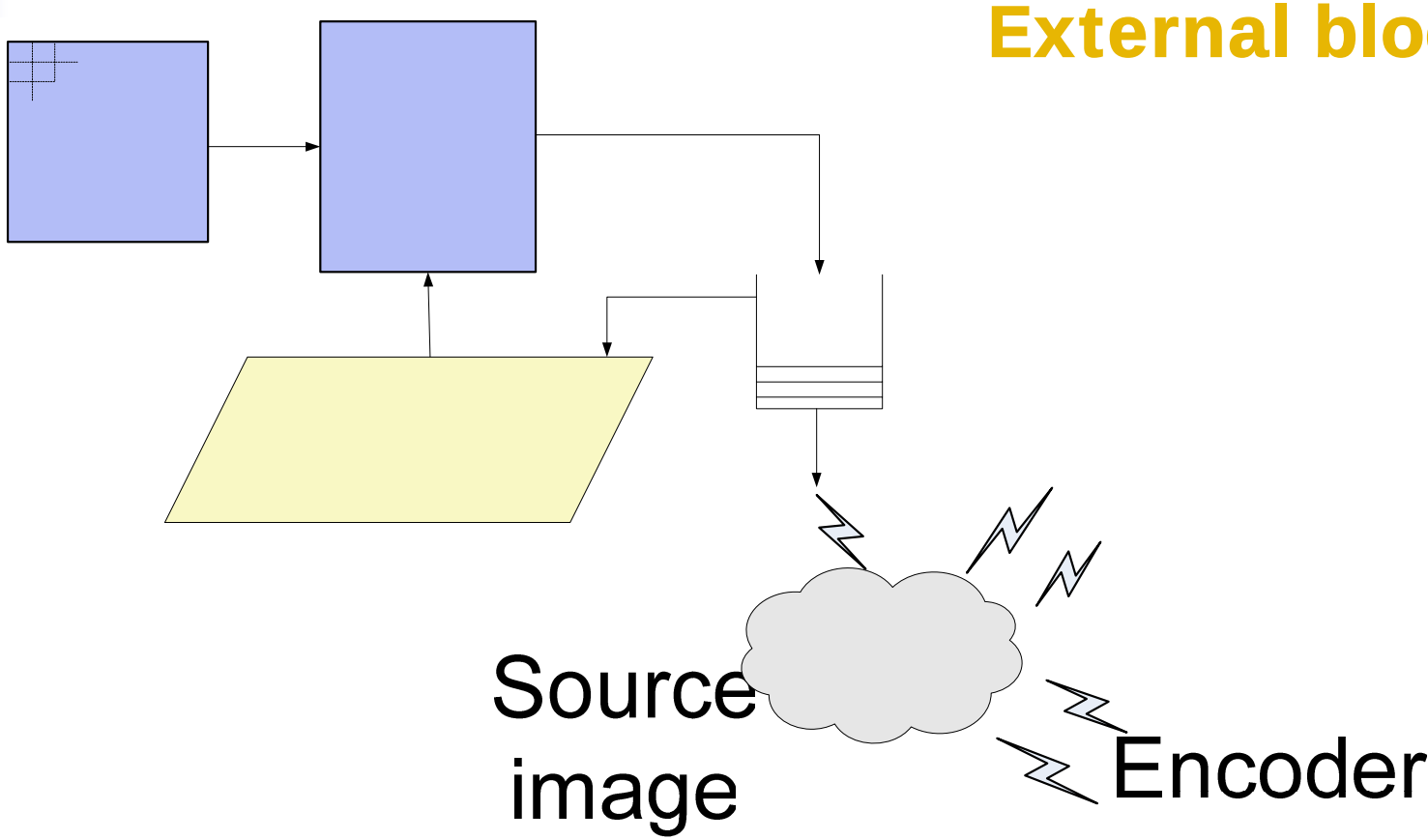
# Alternative vs. standard YUV transform



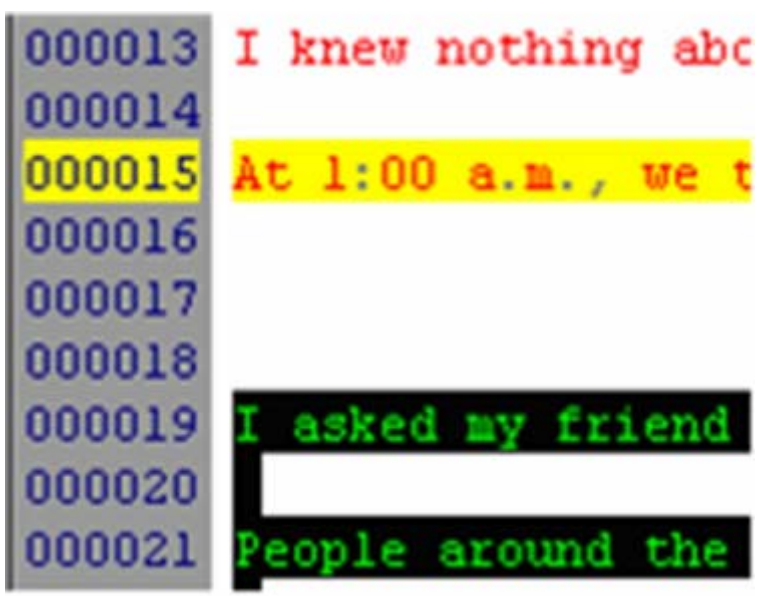
# Rate and visual quality control

**External blocks**



# Disadvantages in Rate Control in JPEG2000 with tiling

When statistical properties of each tile vary a lot...




Source image

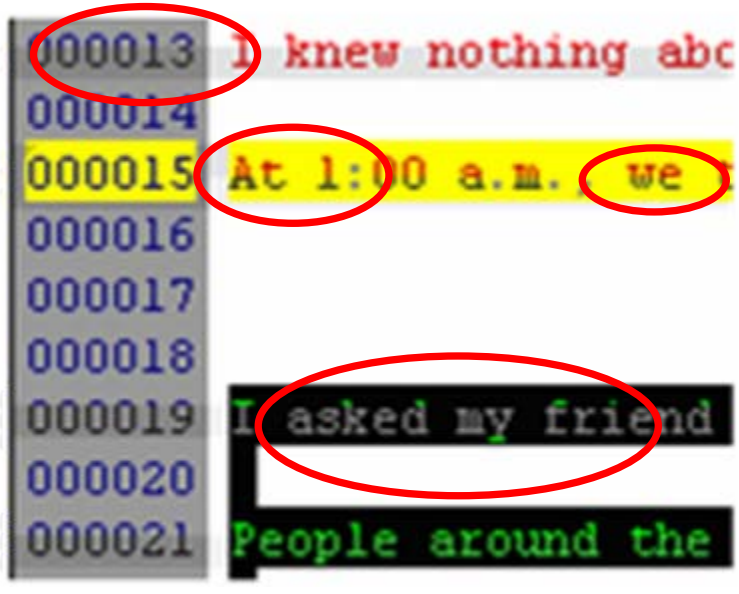



Image compressed with JPEG2000

# Proposed rate and visual quality control

For each tile

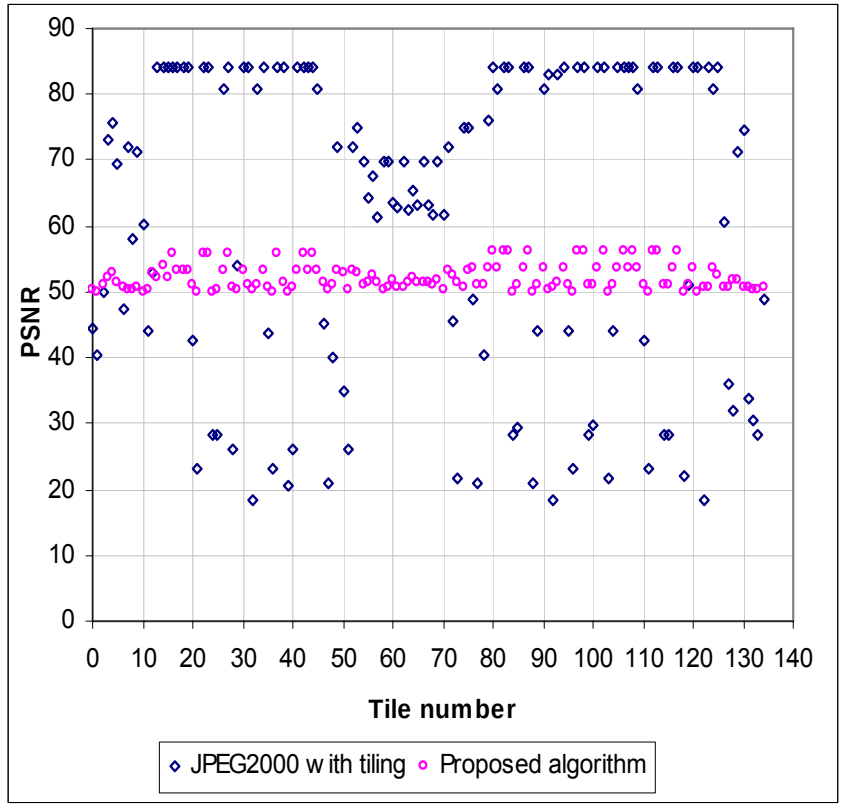
find the truncation vector $\mathbf{x}_t$, so that:

$$\begin{cases} r(\mathbf{x}_t) = \min\limits_{\mathbf{n}\in\mathbf{N}_t} r(\mathbf{n}) \\ d(\mathbf{x}_t) \le d \end{cases}$$

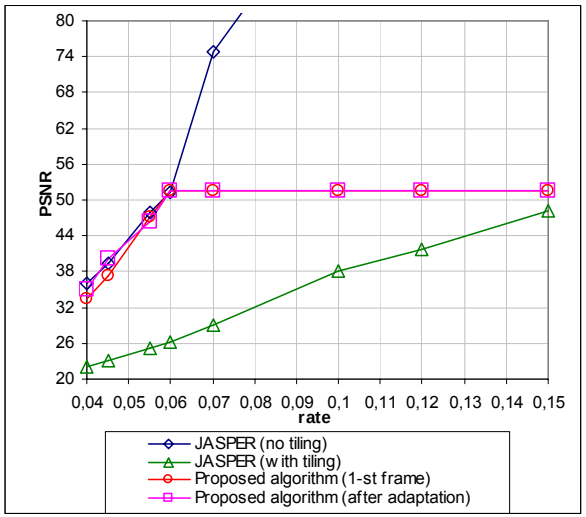
Optimization task (for all tiles):

$$\begin{cases} minimize \max\limits_{t} d(\mathbf{x}_t) \\ b(t) \le B_0 \end{cases},$$

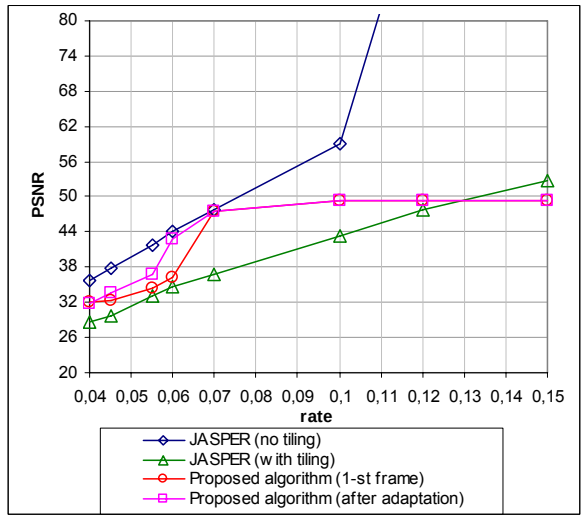
where $b(t) = \max\{0, b(t-1) - r\} + r(\mathbf{x}_t),$



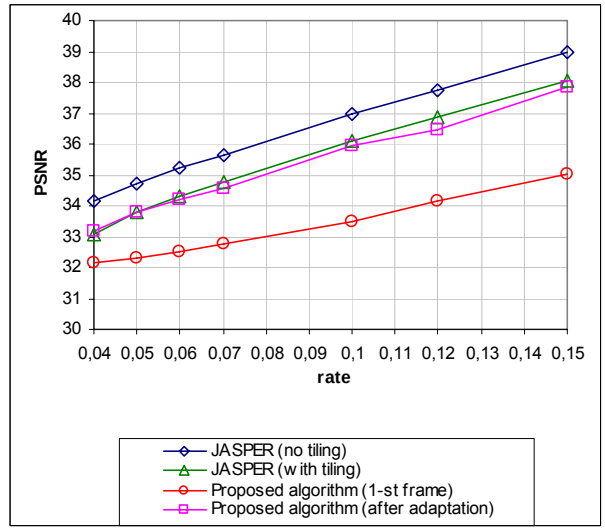
# Rate control: summary




Synthetic images



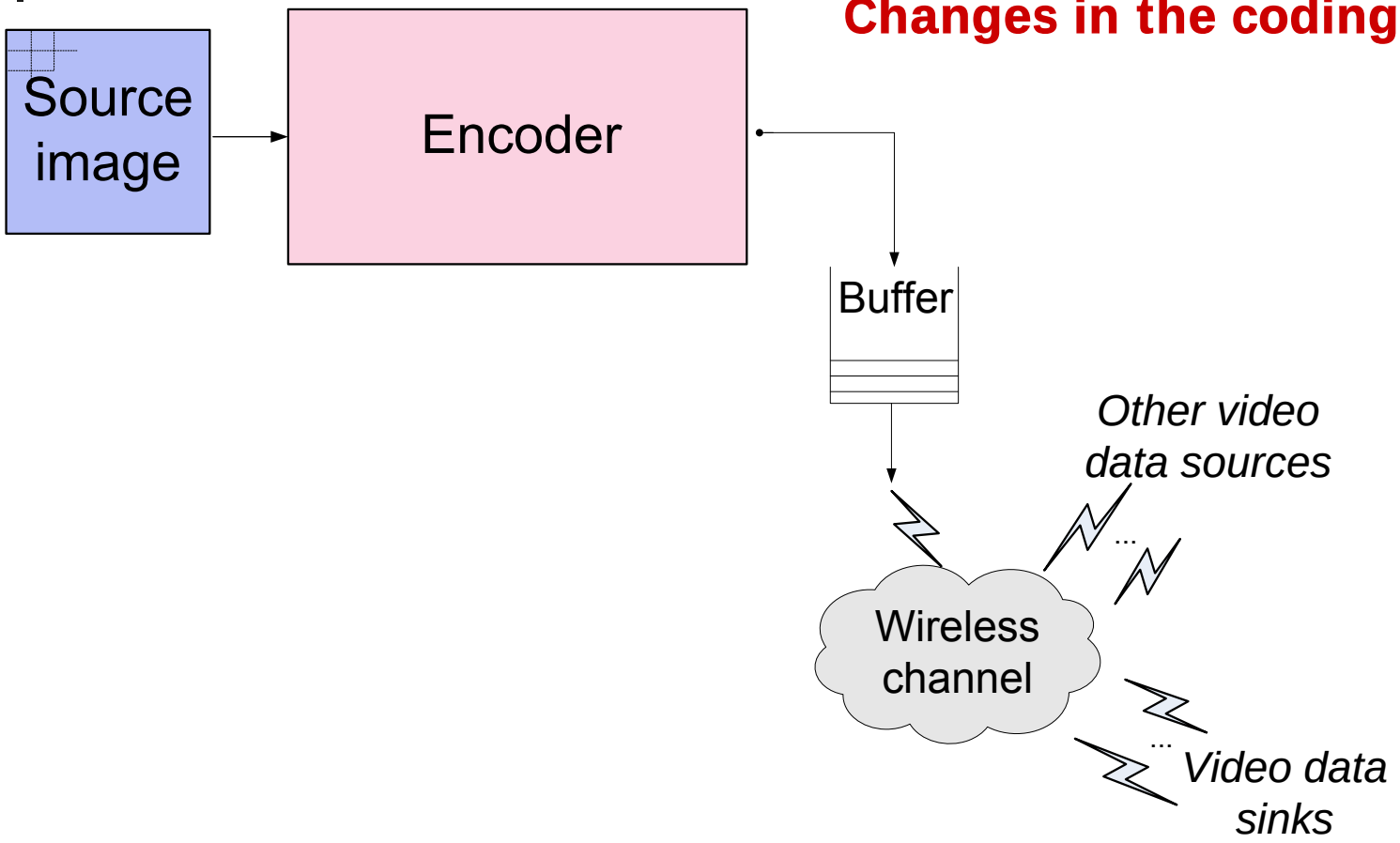

Images with natural and synthetic fragments



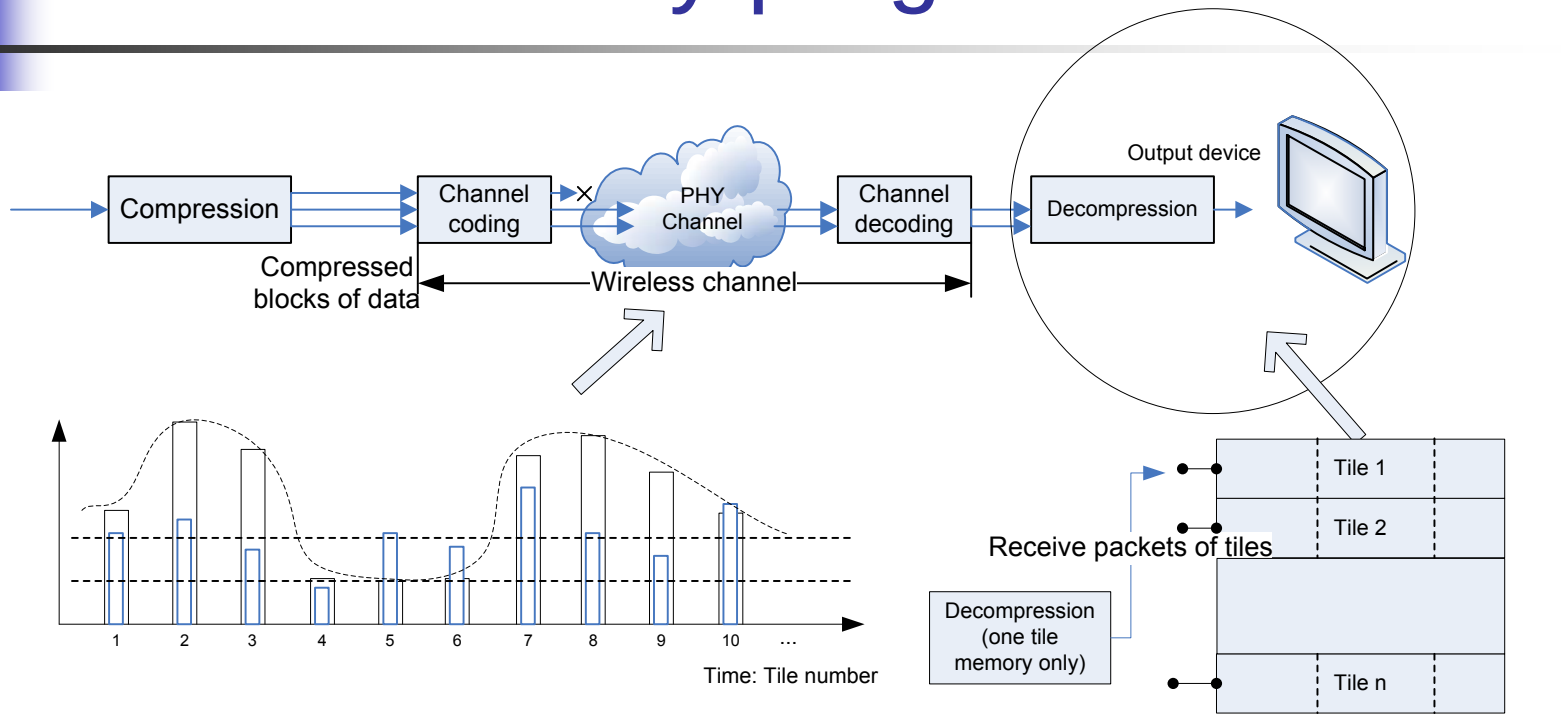

Natural images

# Progressiveness

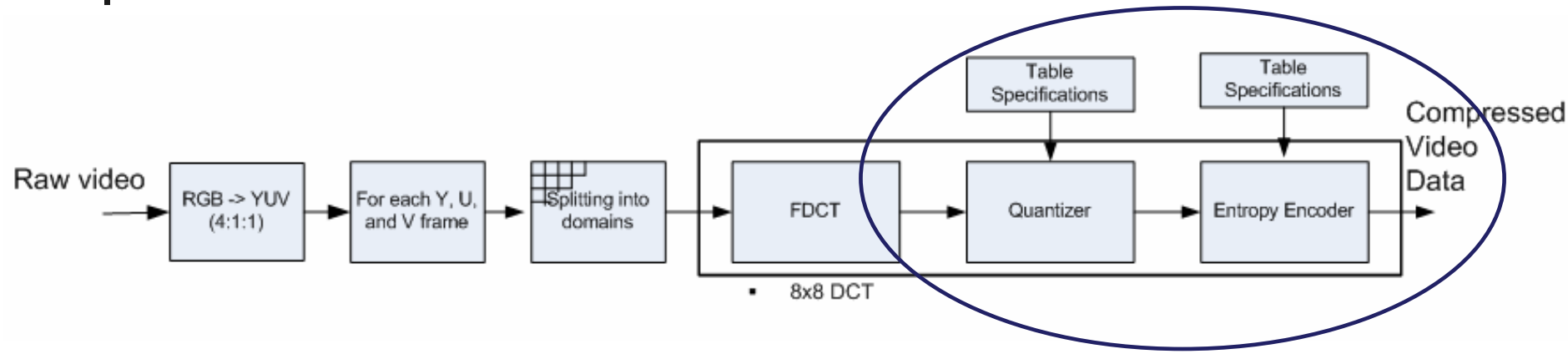
**Changes in the coding algorithm**

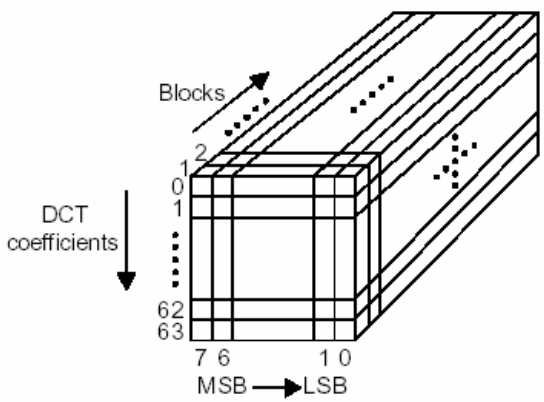


# What is actually progressiveness?



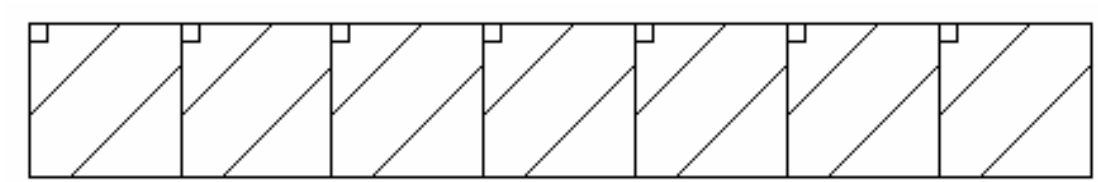

# General JPEG structure

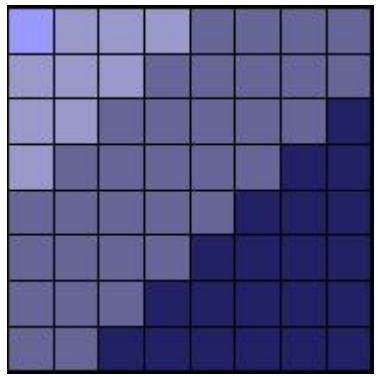


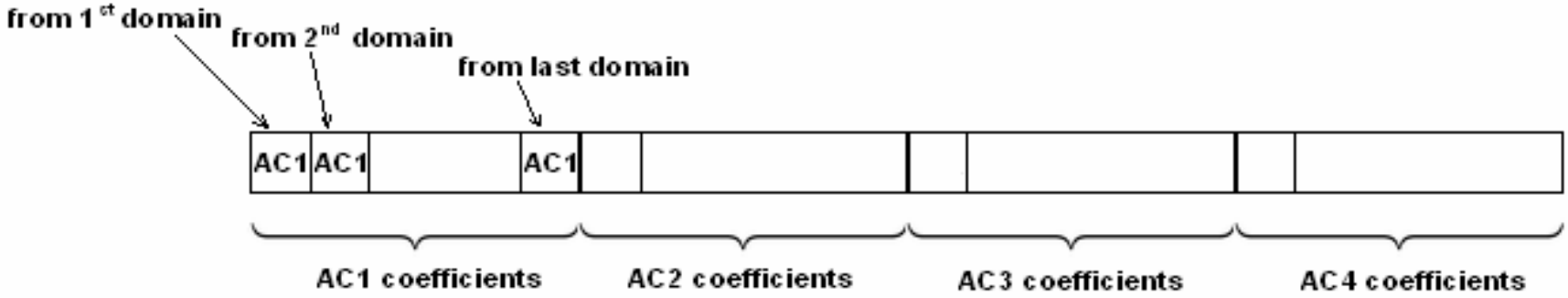
# Progressive encoding scheme

- Partitioning into slices

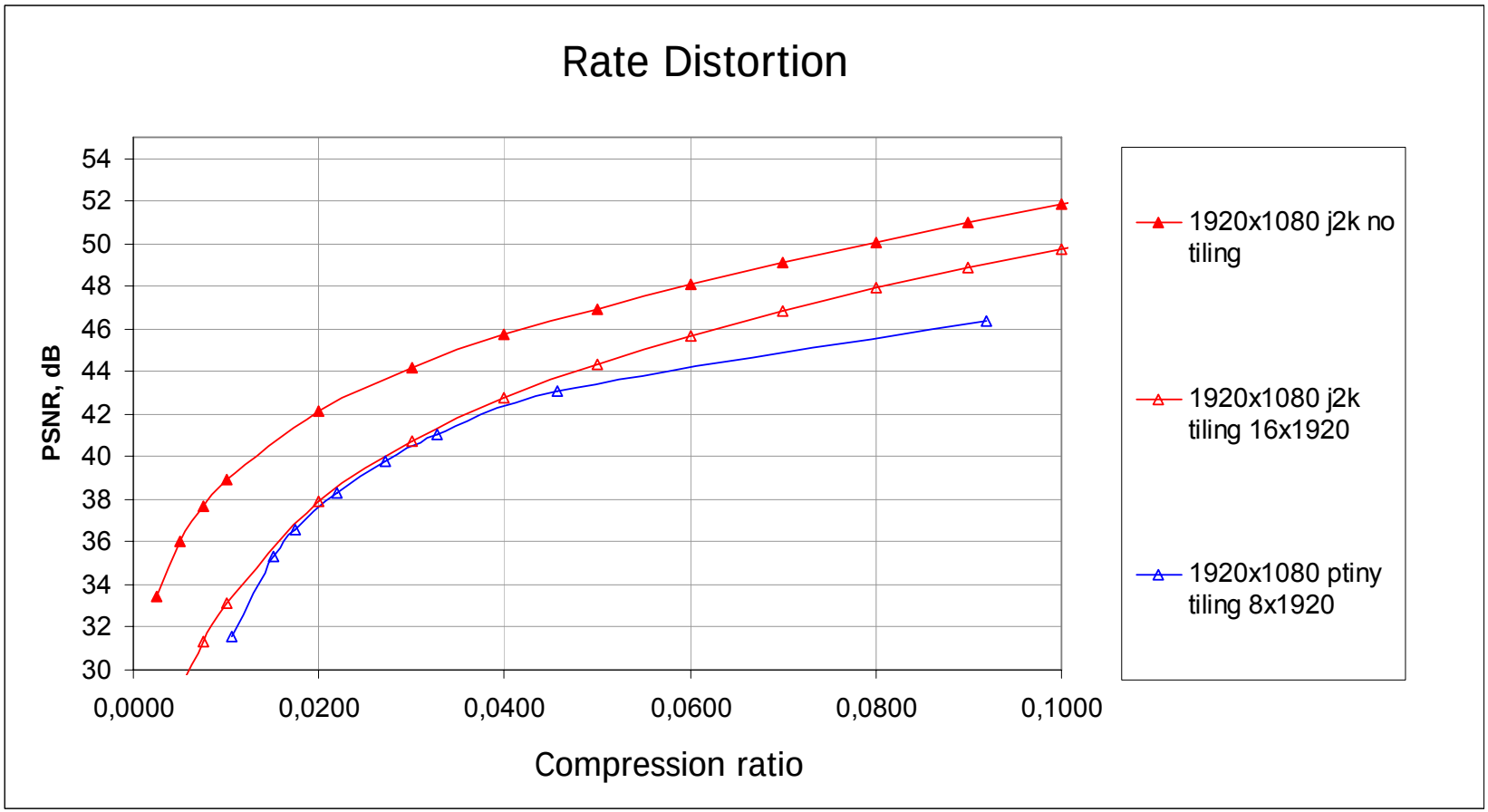




- Coefficients ordering in slices

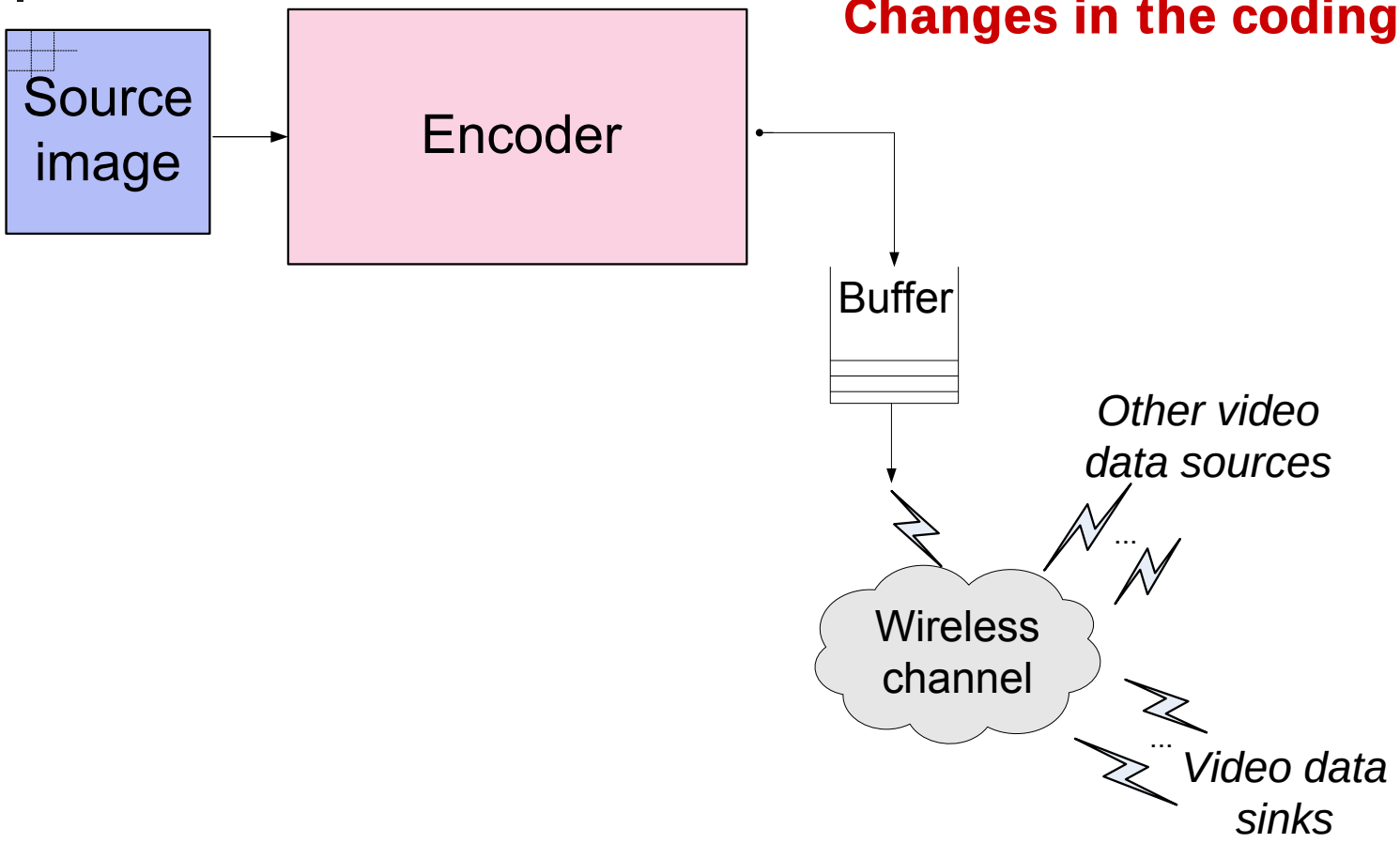


# Progressive Tiny

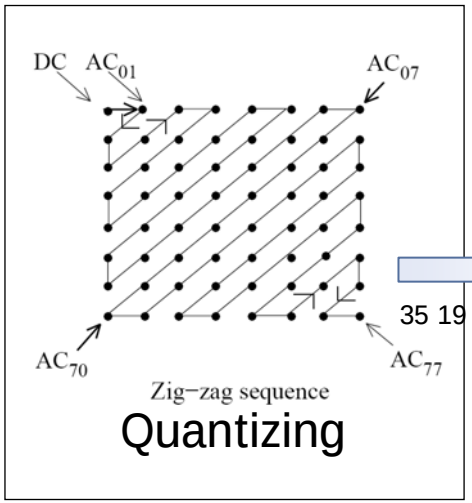


# Entropy coding

**Changes in the coding algorithm**



# Run Length Encoding (RLE) in JPEG




35 19 0 0 5 0 0 0 ... 0

OR

OR

1. Standard

Run Length Encoding → (0,6) (35) (0,5) (19) (2,3) (5) EOB → Huffman encoding → Run Table Huffman Table

2. Reduce size of Huffman tables

Run Length Encoding → (0)(6) (35) (0)(5) (19) (2)(3) (5) EOB → Huffman encoding with small table sizes → Run Table Huffman Table

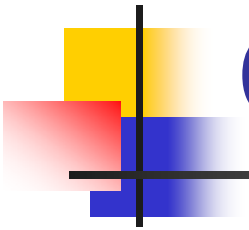
3. Remove RLE coding for non-tail zeroes

~~Run Length Encoding~~ → (35)(19)(0)(0) EOB → Huffman encoding with small table sizes → Huffman Table

**RLE variants comparison**

# Main achievements

- **Color space transform**. **Up to 8 dB** (H.264) gain in PSNR metrics in comparison with traditional YUV transform
- **Detector text/video**. Gain of **up to 10 dB** in PSNR (JPEG 2000)
- **Progressiveness**. Several versions of progressive encoding (JPEG) with sufficiently less complexity (**70 % memory consumption**) for the same level of quality
- **Rate and visual quality control**. Provides with the uniform high level of visual quality that could be realized with sufficiently small transmitter/receiver memory consumption (JPEG2000 and H.264/AVC). It gives the gain of about **10 dB** in PSNR metrics



# Conclusion

- There will be plenty of need for further work.
- There will be plenty of need for more bits.
- There will be plenty of need for higher compression ratio.
- There will be plenty of need for good ideas.
- And those good ideas will come.

*Dream no small dreams, for they have no power to move the hearts of men.*

- Goethe (1749-1842)