Client-driven Transmission of JPEG2000 Image Sequences using Motion Compensated Conditional Replenishment

J.J. Sánchez-Hernández[*], V. González-Ruiz[*], J.P. García-Ortiz[*], and D. Müller[†]

[*]University of Almería – Ctra. Sacramento, s/n – Almería, 04120, Spain

[†]European Space Agency – ESTEC, PO Box 299 – 2200 AG Noordwijk, The Netherlands

Abstract

This is a work focused on remote browsing of JPEG2000 image sequences which takes advantage of the spatial scalability of JPEG2000 to determine which precincts of a subsequent image should be transmitted, and which precincts should be reused from a previously reconstructed image. The results of our experiments demonstrate that the quality of the reconstructed images can be significantly increased by using motion compensation and conditional replenishment on the client side. While this is not a novel idea, our proposal is the first to be 100% compatible with any JPIP server. In fact, our proposal only requires the clients to have some degree of spatial random access to the images stored in the server.

MCCR (Motion Compensated Conditional Replenishment)

MCCR is a fully JPIP-compliant proposal for client-driven transmission of JPEG2000 image sequences using motion compensation and conditional replenishment of JPEG2000 precincts. The MCCR logic is exclusively implemented on the client side.

Using JPIP, clients request WOIs from servers, which reply with one or more data-bins (portions of a JPEG2000 file which contain precinct-based data, tile-based data, headers or metadata). It is important to note that: (1) clients request WOIs, not packets or data-bins; and (2) since two or more requested WOIs may overlap, clients store the received packets in a cache, and servers track the content of those caches in order to avoid resending the same data to the same client in the same JPIP session.

Description

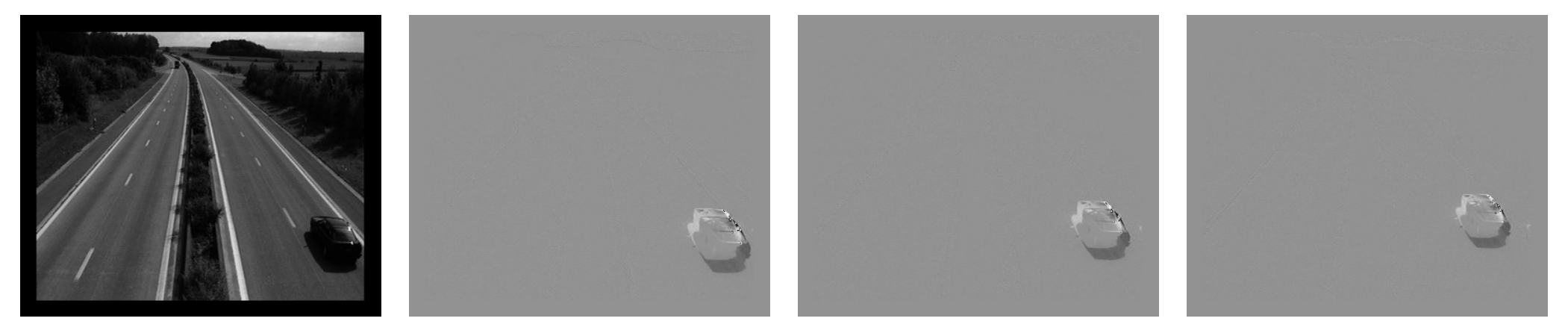
1. At maximum quality and maximum resolution, download the first image of the retrieved sequence, I_i. i denotes the iteration of the algorithm.
2. At maximum quality and maximum resolution, download the next image of the retrieved sequence, I_{i+1}.
3. The two first rendered images that will be displayed are $\tilde{I}_i = I_i$ and $\tilde{I}_{i+1} = I_{i+1}$.
4. Estimate the motion between $\tilde{I}_i$ and $\tilde{I}_{i+1}$ using subpixel accuracy A and a search area of $S \times S$ (integer) pixels. In the case of using a block-based estimator, for each WOI of size $B \times B$ pixels, calculate a one-direction forward motion vectors field $\overrightarrow{V}$.
5. Generate a prediction image $\hat{I}_{i+2}$, by projecting the image $\tilde{I}_{i+1}$ using $\overrightarrow{V}$. In other words,
$$\hat{I}_{i+2} = \overrightarrow{V}(\tilde{I}_{i+1}). \tag{1}$$
Notice that in doing so, we are supposing a constant movement of the block's content among the images of the sequence.
6. Generate a low resolution representation of $\hat{I}_{i+2}$, $\texttt{Thumbnail}(\hat{I}_{i+2})$, using the same analysis filters that the DWT used during the compression of the images. The number of reduction operations must produce a thumbnail of the size of the image to be retrieved in the following step.
7. Retrieve (from the server) the thumbnail (a low-resolution representation) of I_{i+2}, $\texttt{Thumbnail}(I_{i+2})$.
8. Calculate the pixel differences
$$E = \texttt{Thumbnail}(I_{i+2}) - \texttt{Thumbnail}(\hat{I}_{i+2}) \tag{2}$$
between the thumbnail prediction image and the thumbnail for the next image of the sequence.
9. Compute the average of each WOI of E and sort the averages in descending order. This results in a list L of WOIs that are sorted by distortion.
10. Depending on the available bandwidth, request a number of WOIs (with all the quality layers) for the image I_{i+2}. The selected WOIs are the first entries in L.
11. For all those precincts of I_{i+2} that have not been received, use the precincts of $\hat{I}_{i+2}$. The resulting image $\tilde{I}_{i+2}$ is the rendered image, resulting from merging the precincts of both images (conditional replenishment step).
12. $i \leftarrow i+1$. Go to step 4.

Image compression requirements

To use MCCR, it is necessary: (1) to have spatial random access to each of the WOIs of the compressed images, and (2) the images must be compressed defining a precinct partition that fulfills this condition. In other words, the images have to be compressed with the same number of precincts in each resolution level of the DWT domain. Moreover, the number D of DWT levels must be chosen according to the resolution of the thumbnails.

Experimental results

Speedway is a low-motion surveillance 384×320 grayscale video where a fixed camera at 30 frames/s shows cars moving on a motorway. The first 4 pictures of the sequence (original and next 3 differences) are shown below.



The first 50 images of Speedway were compressed with JPEG2000, generating 3 spatial resolution levels and 8 quality layers. The code-block size is 16×16 coefficients. The precinct partition generates a WOI that is 64×64 pixels in size ($B = 64$), at the maximum resolution level. For the motion estimation, $A = 1$ (half subpixel accuracy) and $S = 4$ were selected.

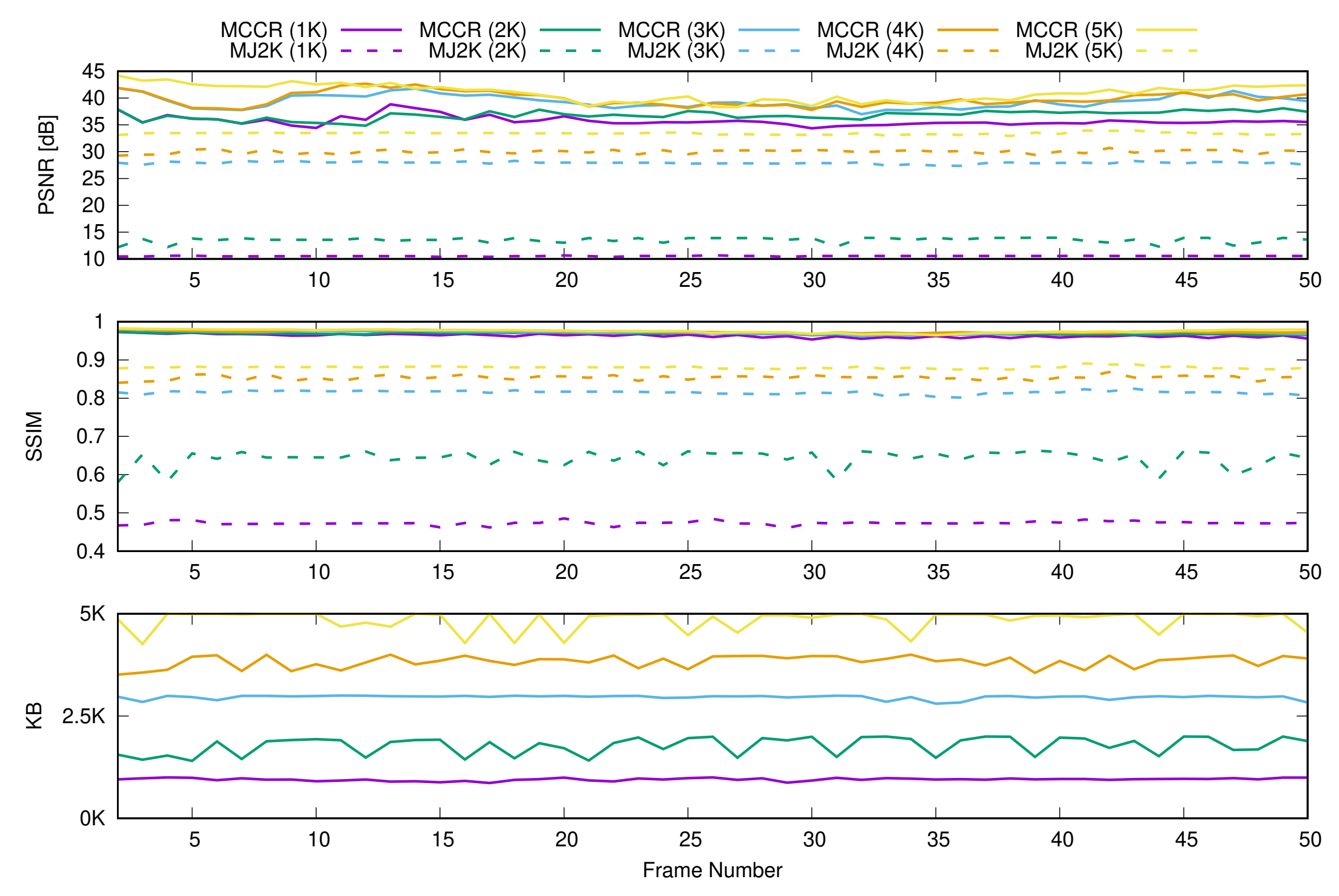
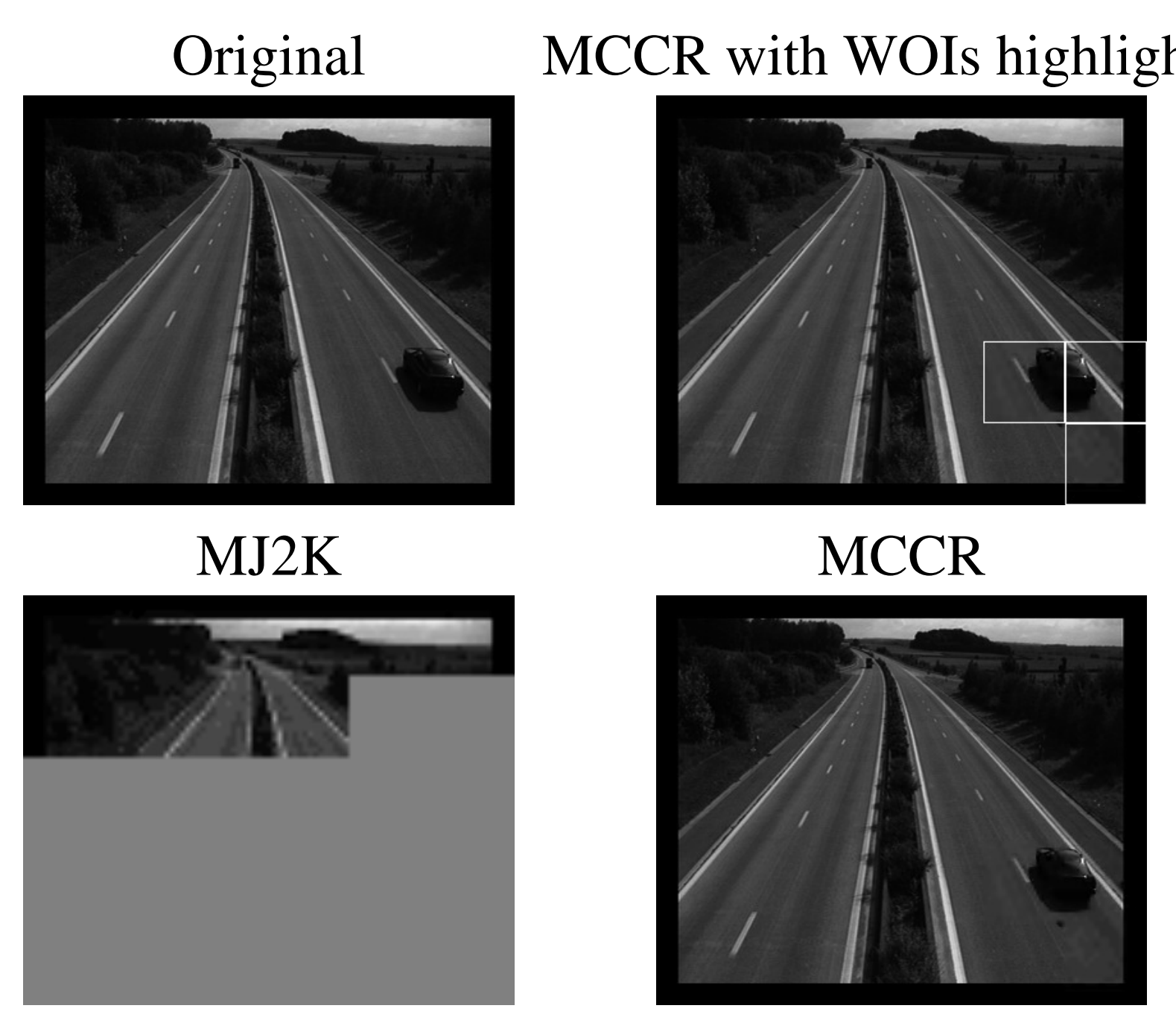
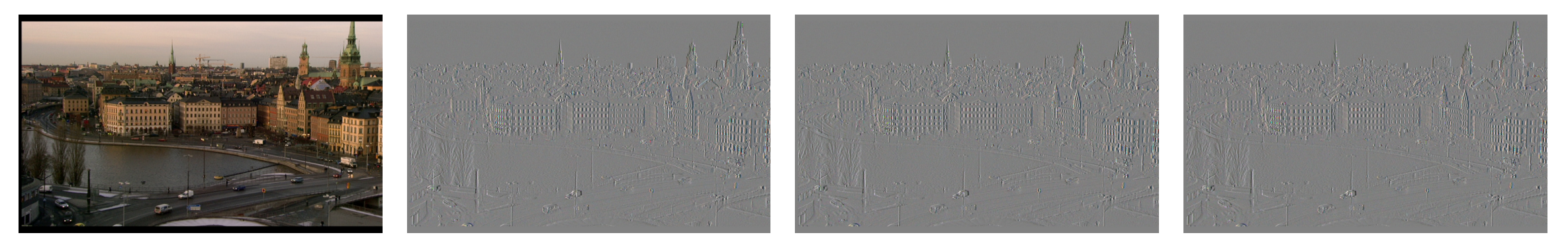


Figure 1: Quality of the reconstructions of the first 50 images of Speedway using MCCR and Motion JPEG2000 (MJ2K). Different bandwidth scenarios have been simulated using different amounts of transmitted bytes per image (from 1000 bytes to 5000 bytes).



Stockholm is a medium-motion 1280×720 color video where a panning camera at 50 frames/s shows houses, water and cars moving in a Stockholm city landscape. The first (not gray) 4 pictures of the sequence (original and next 3 differences) are shown below.



The first 100 images of Stockholm were compressed with JPEG2000 generating 3 spatial resolution levels, 8 quality layers. A code-block size of 32×32 coefficients was used. The precinct partition generates a WOI size of 128×128 pixels ($B = 64$), at the maximum resolution level. For the motion estimation, $A = 2$ (quarter subpixel accuracy) and $S = 4$ were selected.

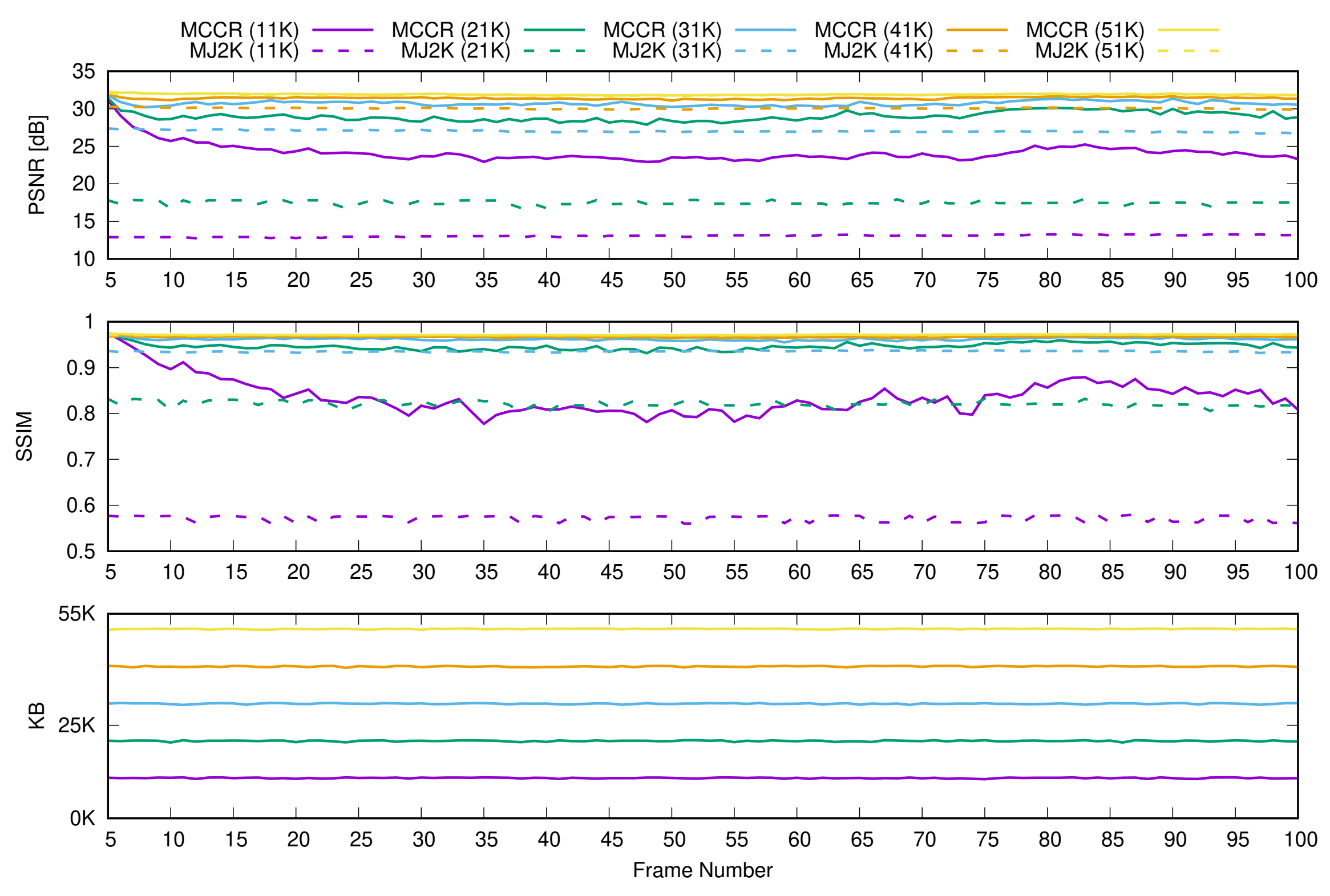
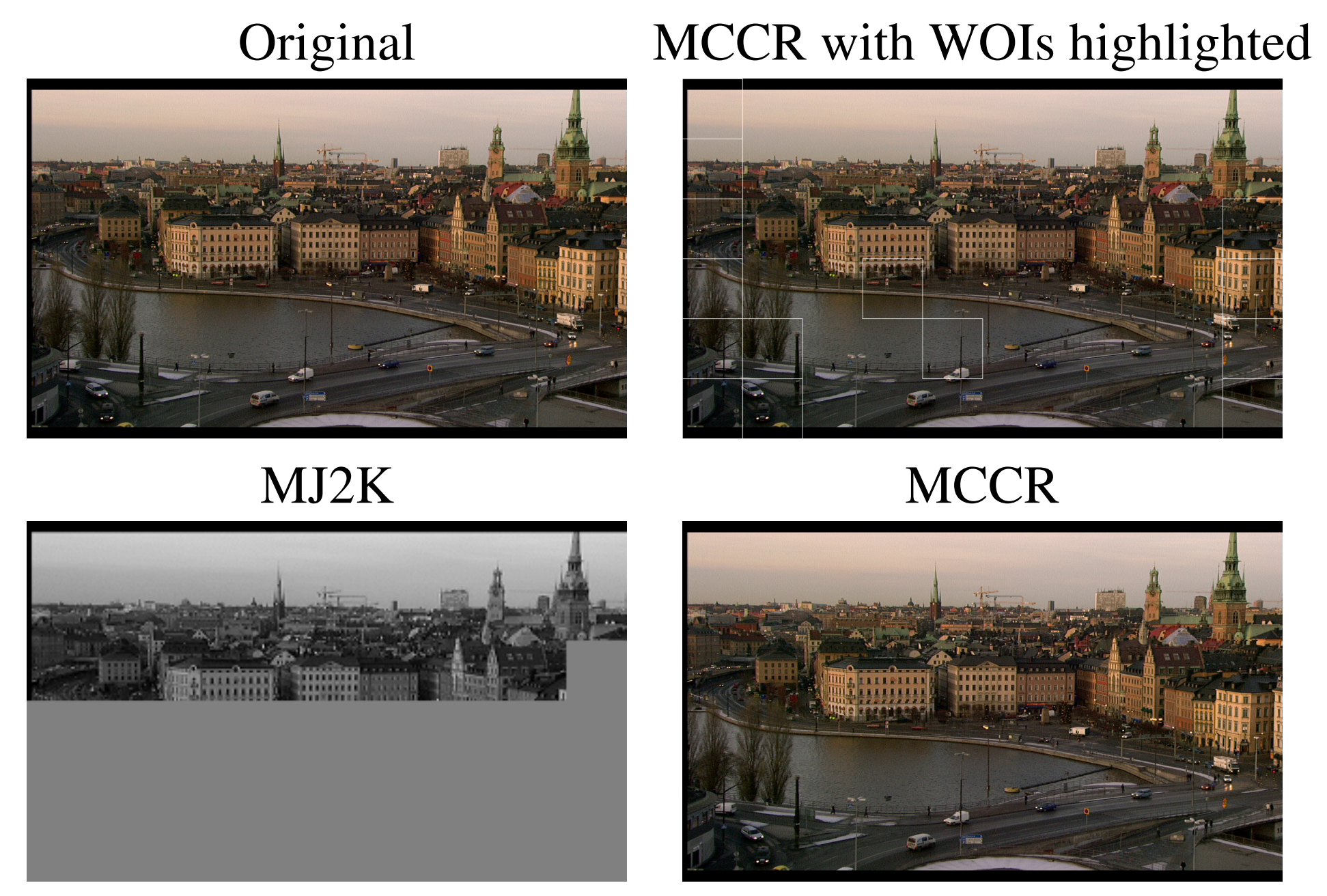


Figure 2: Quality of the reconstructions of the first 100 images of Stockholm using MCCR and Motion JPEG2000 (MJ2K). Different bandwidth scenarios have been simulated using different amounts of transmitted bytes per image (from 11000 bytes to 51000 bytes).



Figures 1 and 2 show the quality of the reconstruction for different amounts of transmitted bytes simulating different bandwidth scenarios. In the case of Stockholm, a sequence with more movement than Speedway, the results show the following expected behaviour:

1. The more complex the motion in the scene (the lower the temporal correlation), the lower the gain of MCCR. Notice that the gain of MCCR should be one if there is no temporal correlation.
2. If the transmission bit-rate is too small, the number of replenished WOIs is not enough to reconstruct the images of the sequence. This effect, that can be seen when 11000 bytes/image are retrieved, can be solved by monitoring the quality at the receiver and requesting an "intra" image when the quality drops below a given threshold controlled by the user.

Conclusions and future work

The research carried out in this work shows that MCCR is a more efficient alternative than Motion JPEG2000 when low-motion image sequences are interactively retrieved from JPIP servers. Unlike previous proposals, MCCR can be driven solely by the receivers, and therefore, it can be used without any modification on the server side.