

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

J.J. Sánchez Hernández, J.P. García-Ortiz, V. González-Ruiz

Department of Informatics, University of Almería, Spain

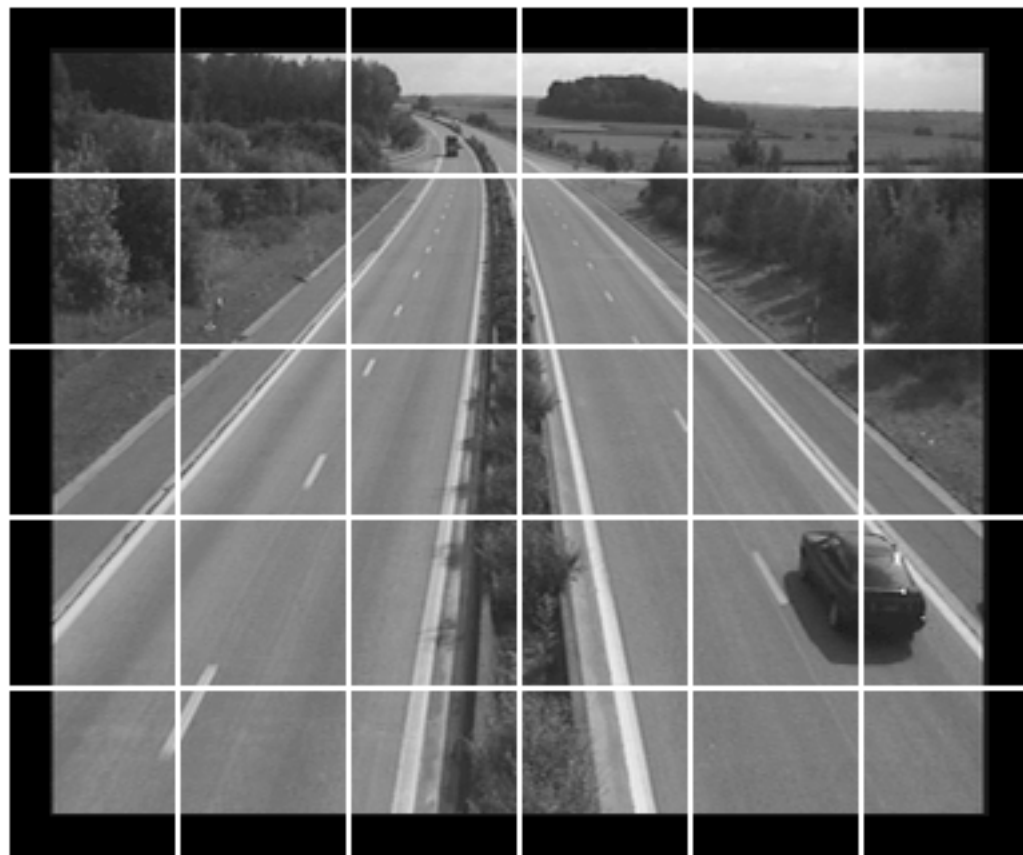
Work in progress

Image Requirements

The images have to be compressed with the **same number of precincts for each resolution level.**

For example: *Speedway* sequence (384x320)

```
CLEVELS=2  
CPREINCTS="{ 64, 64 }, { 32, 32 }, { 16, 16 }"  
CBLK="{ 16, 16 }"
```



R₀



R₁

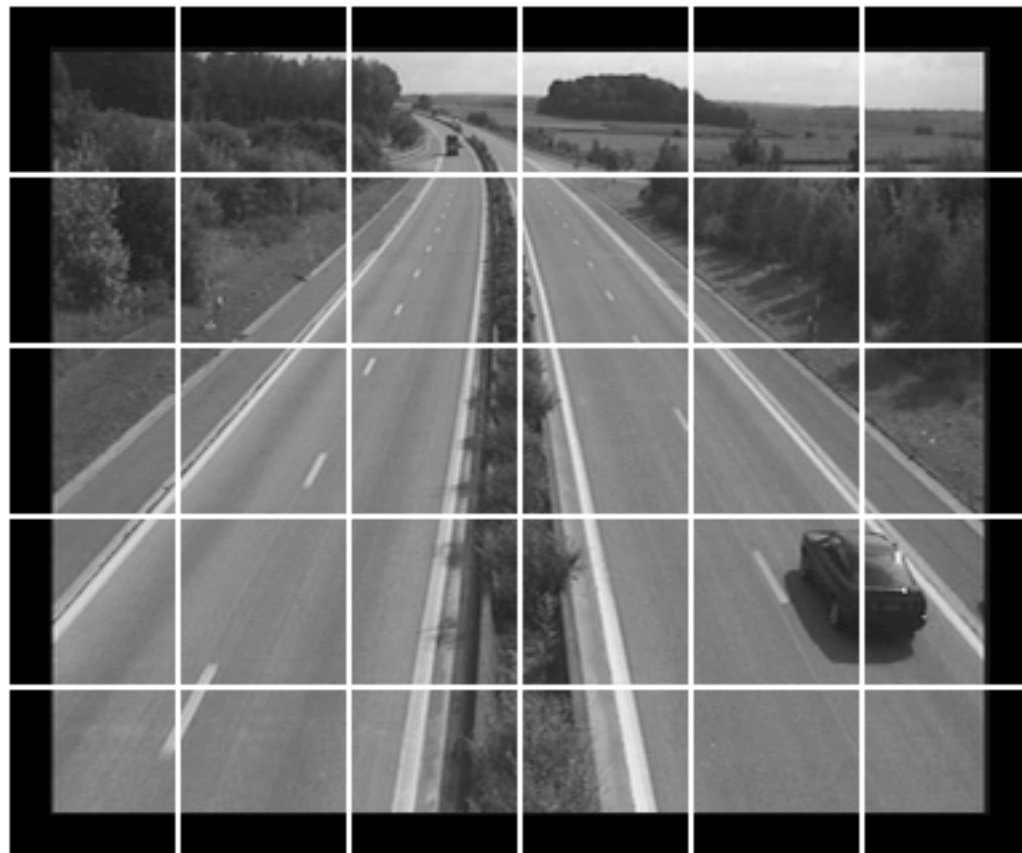


R₂

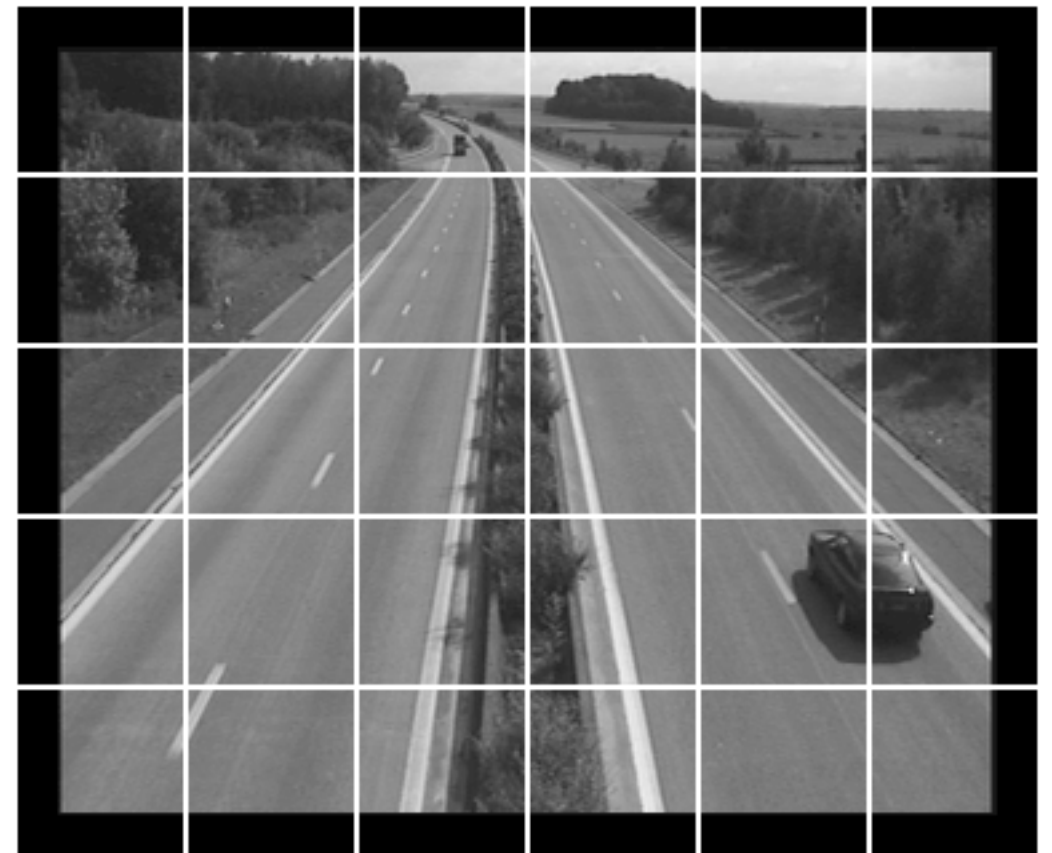
Algorithm

1. **Download two consecutive images** of the sequence, I_i and I_{i+1} , at maximum quality and maximum resolution.

JPIP Server



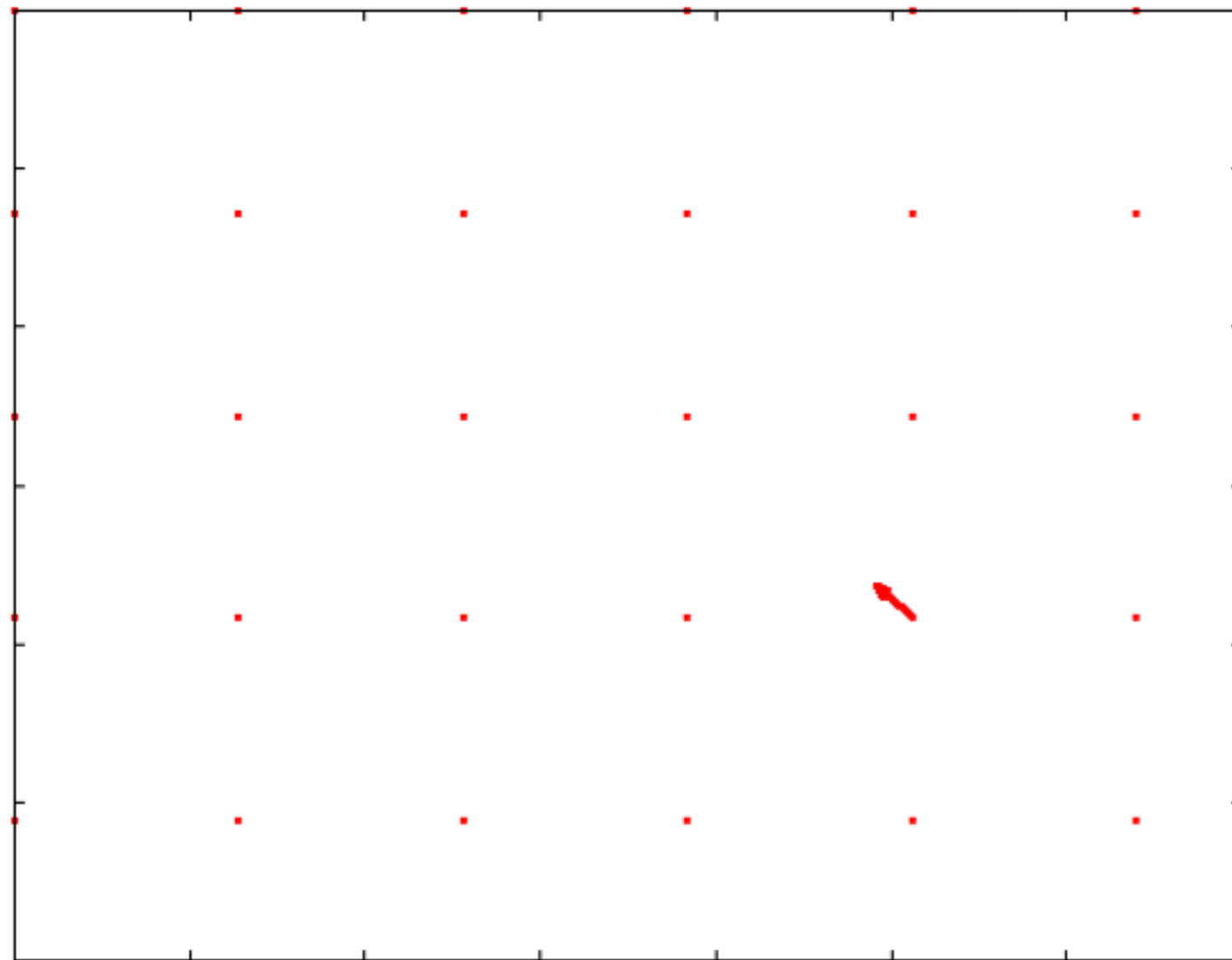
I_i



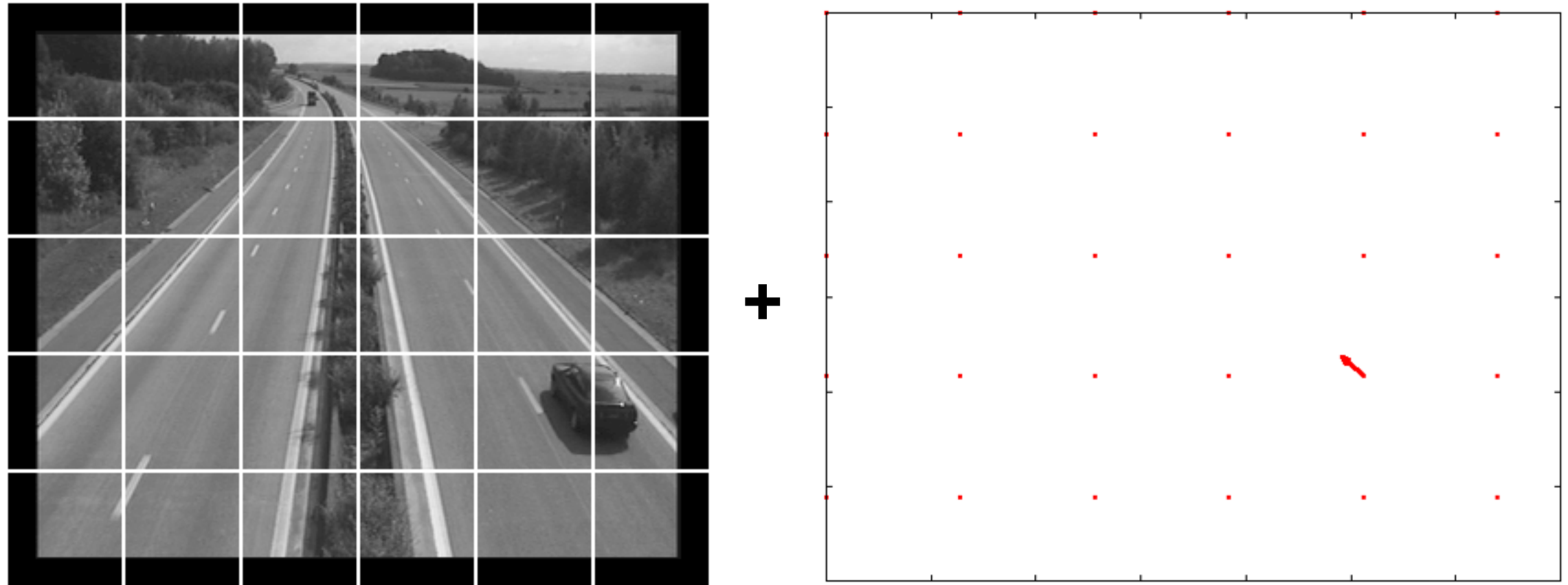
I_{i+1}

2. Calculate the motion vectors between the two images, I_i and I_{i+1} , at maximum quality and maximum resolution.

The block size used in the motion estimation matches with the precinct size.



- 3. Generate a prediction image, \mathbf{Pl}_{i+2} from image \mathbf{l}_{i+1} and the motion vectors calculated in the previous step.**

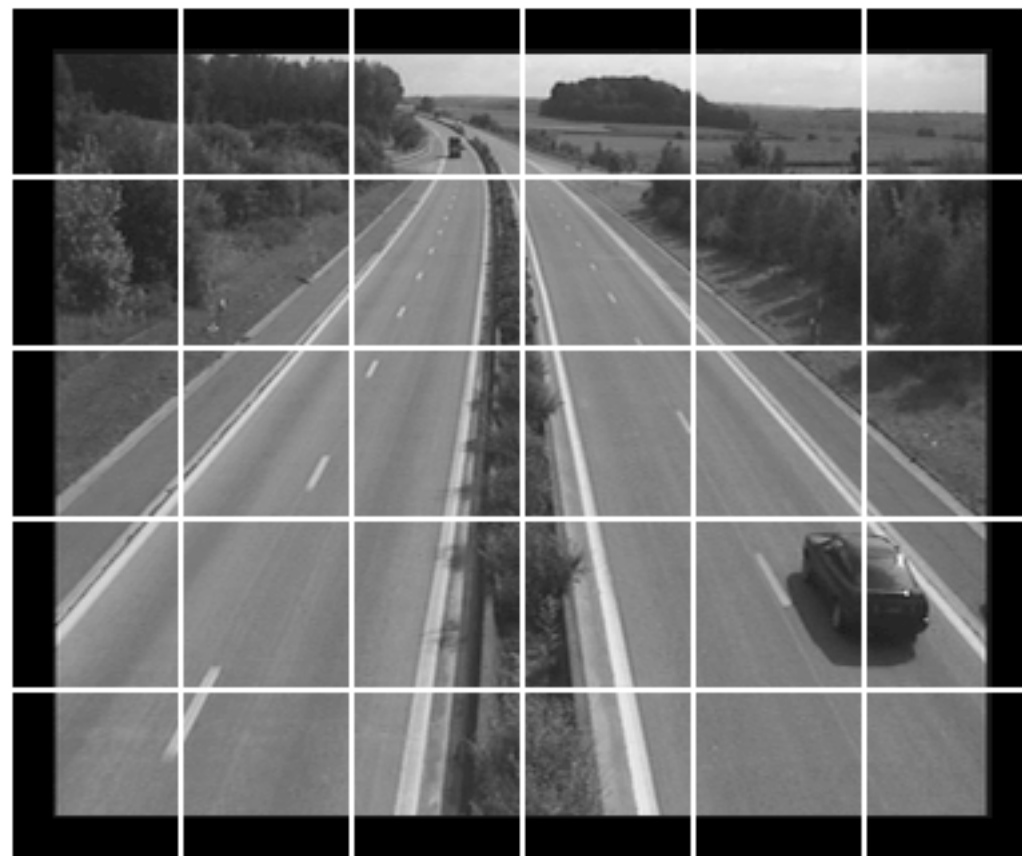


\mathbf{l}_{i+1}

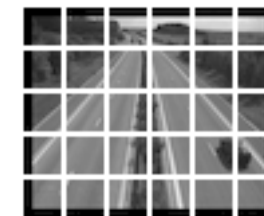
motion vectors

\mathbf{Pl}_{i+2}

4. Generate the thumbnail (using a lower resolution level) of the prediction image \mathbf{Pl}_{i+2} , $\text{Thumbnail}(\mathbf{Pl}_{i+2})$.



\mathbf{Pl}_{i+2}



$\text{Thumbnail}(\mathbf{Pl}_{i+2})$

- 5. Download the thumbnail for the next image of the sequence, $\text{Thumbnail}(I_{i+2})$.**

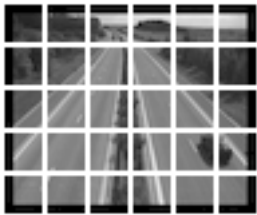
JPIP Server



$\text{Thumbnail}(I_{i+2})$

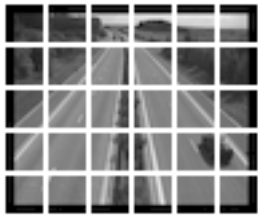
- 5. Calculate the difference between the thumbnail prediction image, $\text{Thumbnail}(P_{l_i+2})$ and the thumbnail of the next image of the sequence, $\text{Thumbnail}(I_{l_i+2})$.**

The result of this step give us a list of precincts sorted by distortion.



Thumbnail(P_{l_i+2})

-

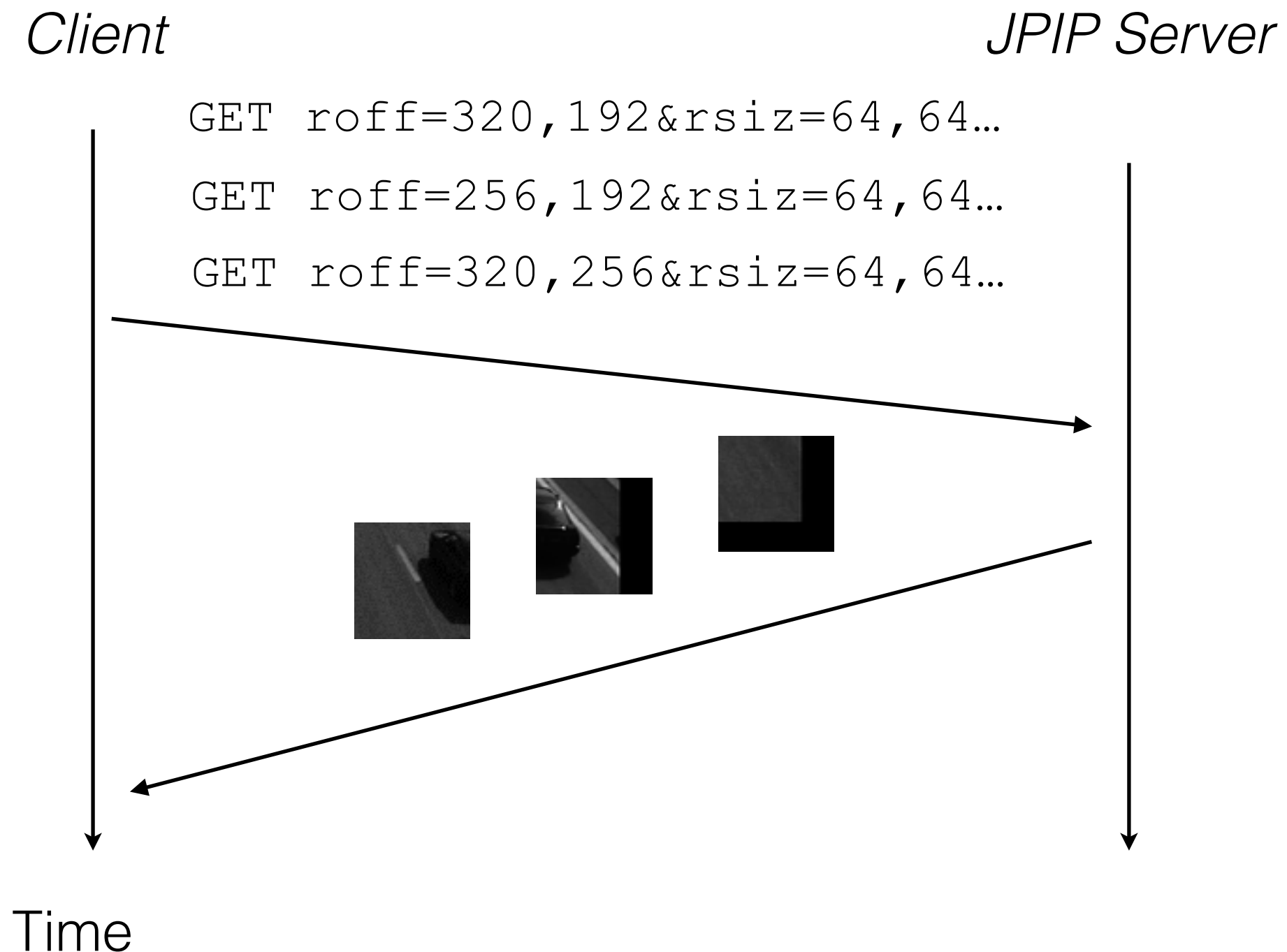


Thumbnail(I_{l_i+2})

=

x	y	MSE/block
320	192	9.078125
256	192	5.820312
320	256	1.378906

- 7. Retrieve from the server the previous list of WOIs for the next image, I_{i+2} .** Note that 1 WOI = 1 precinct. The total number of sent WOIs depends on the channel capacity.



8. Merge the precincts received for the next image I_{i+2} with the prediction image PI_{i+2} .



PI_{i+2}

+



Precincts received from
the next image I_{i+2}

9. Update the index of the images and go to step 2.

Experiments

Speedway : sequence of 50 images of **384x320** pixels.

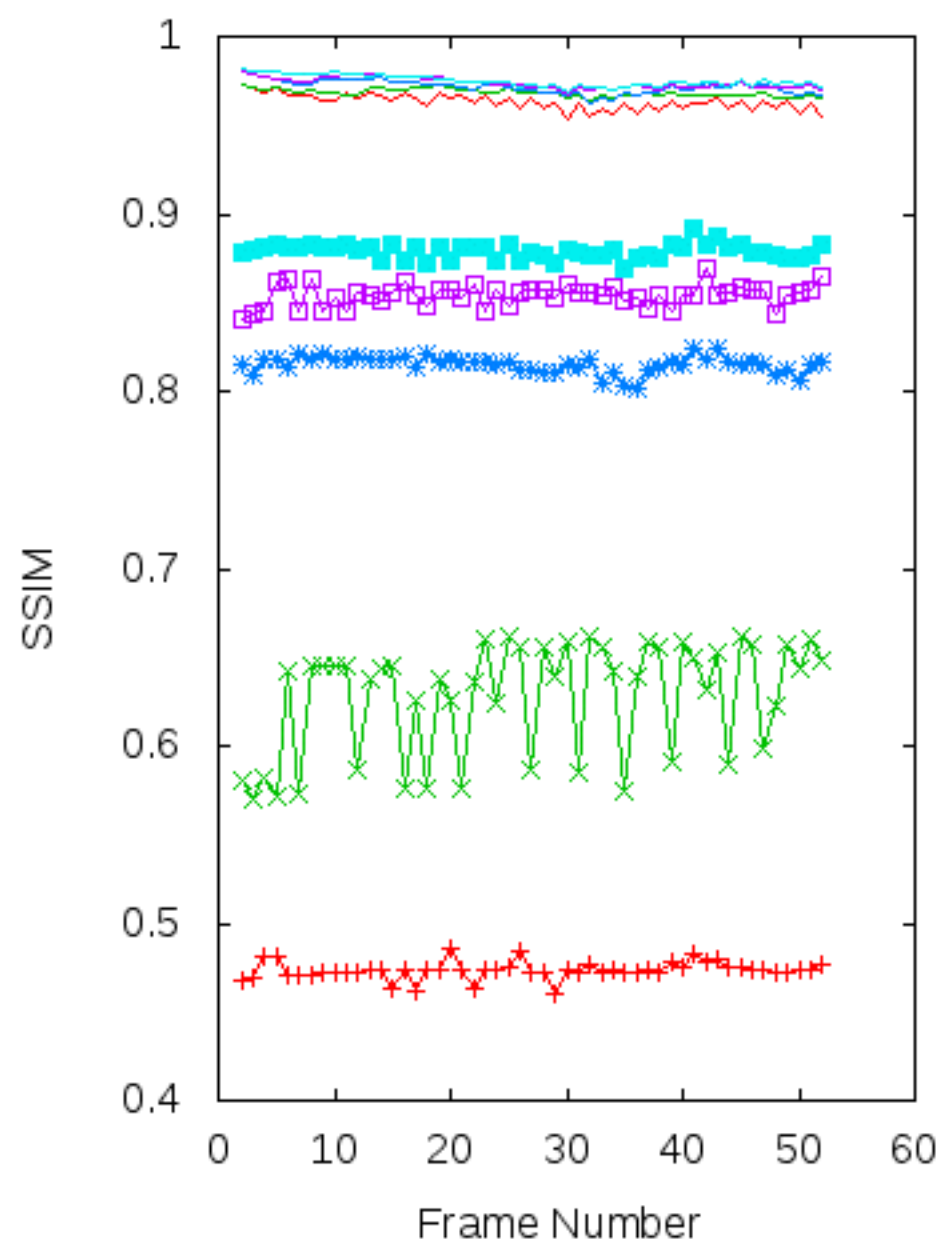
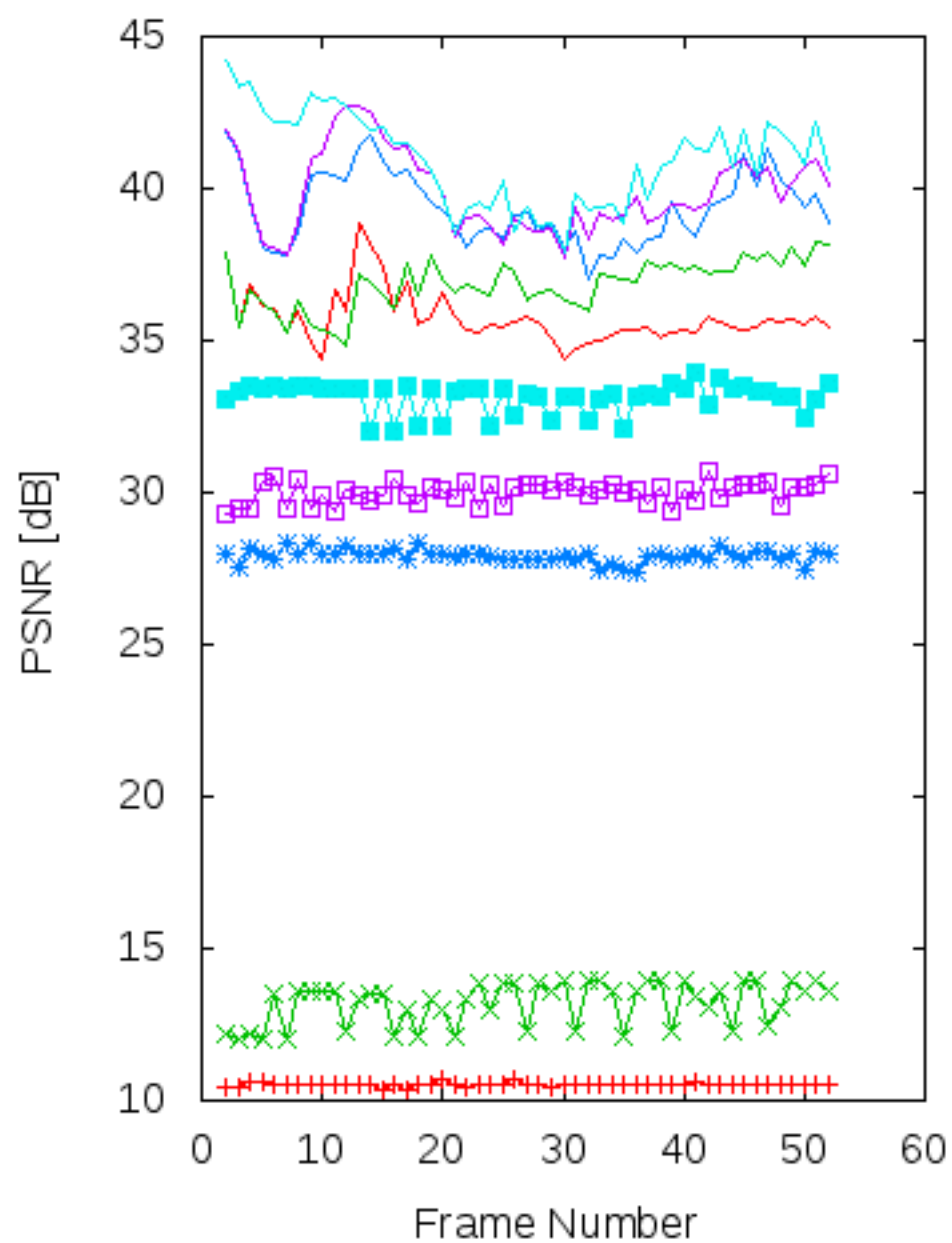
- Compression parameters used in **kdu_compress**

```
CLAYERS=8
CLEVELS=2
CPRECINCTS="{ 64, 64 }, { 32, 32 }, { 16, 16 }"
CBLK="{ 16, 16 }"
Corder=LRCP
```

- Motion estimation parameters

```
B=64      # block size
A=0       # subpixel accuracy
D=0       # border size
S=4       # search range
V=2       # block overlapping
```

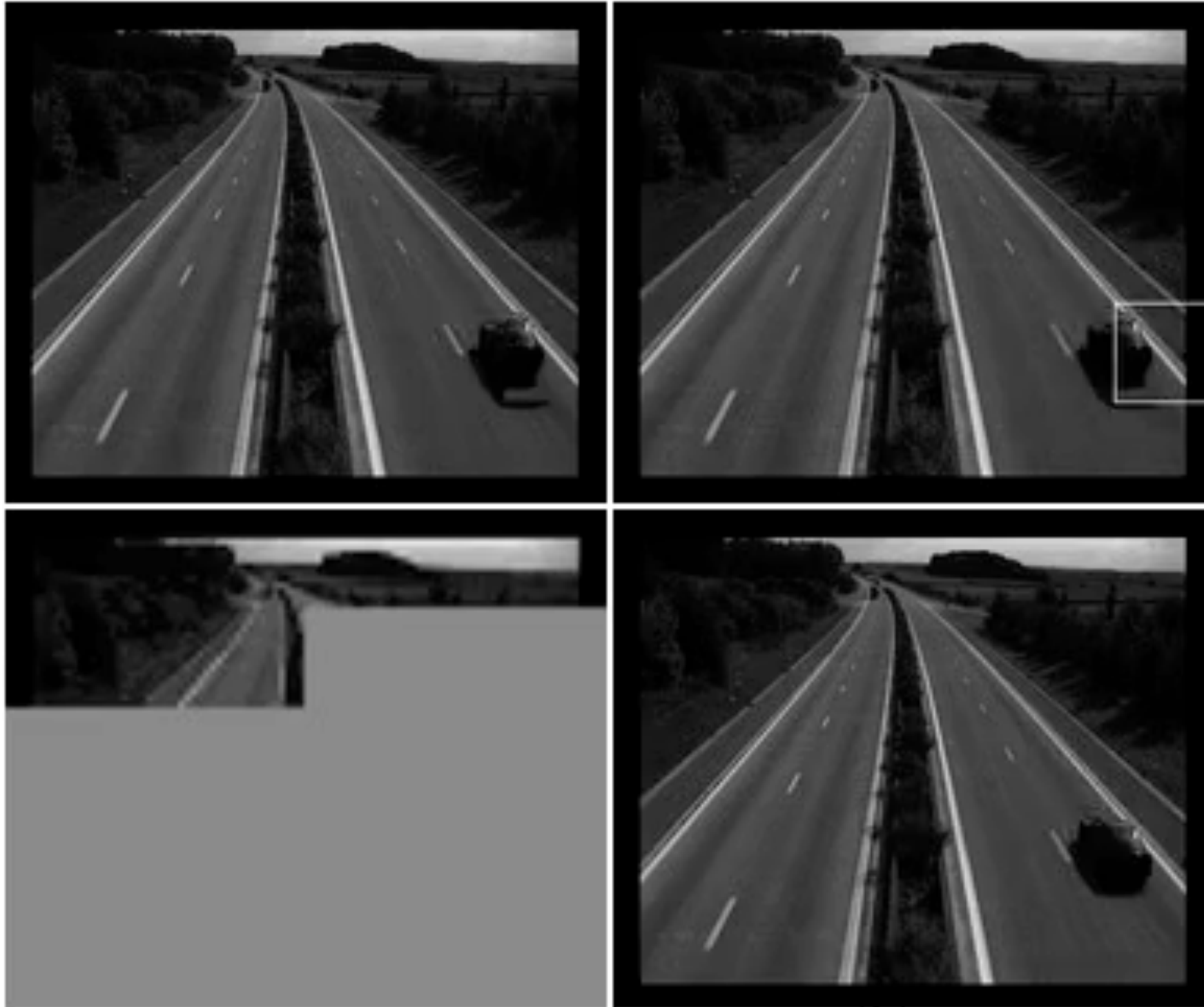
PSNR and SSIM



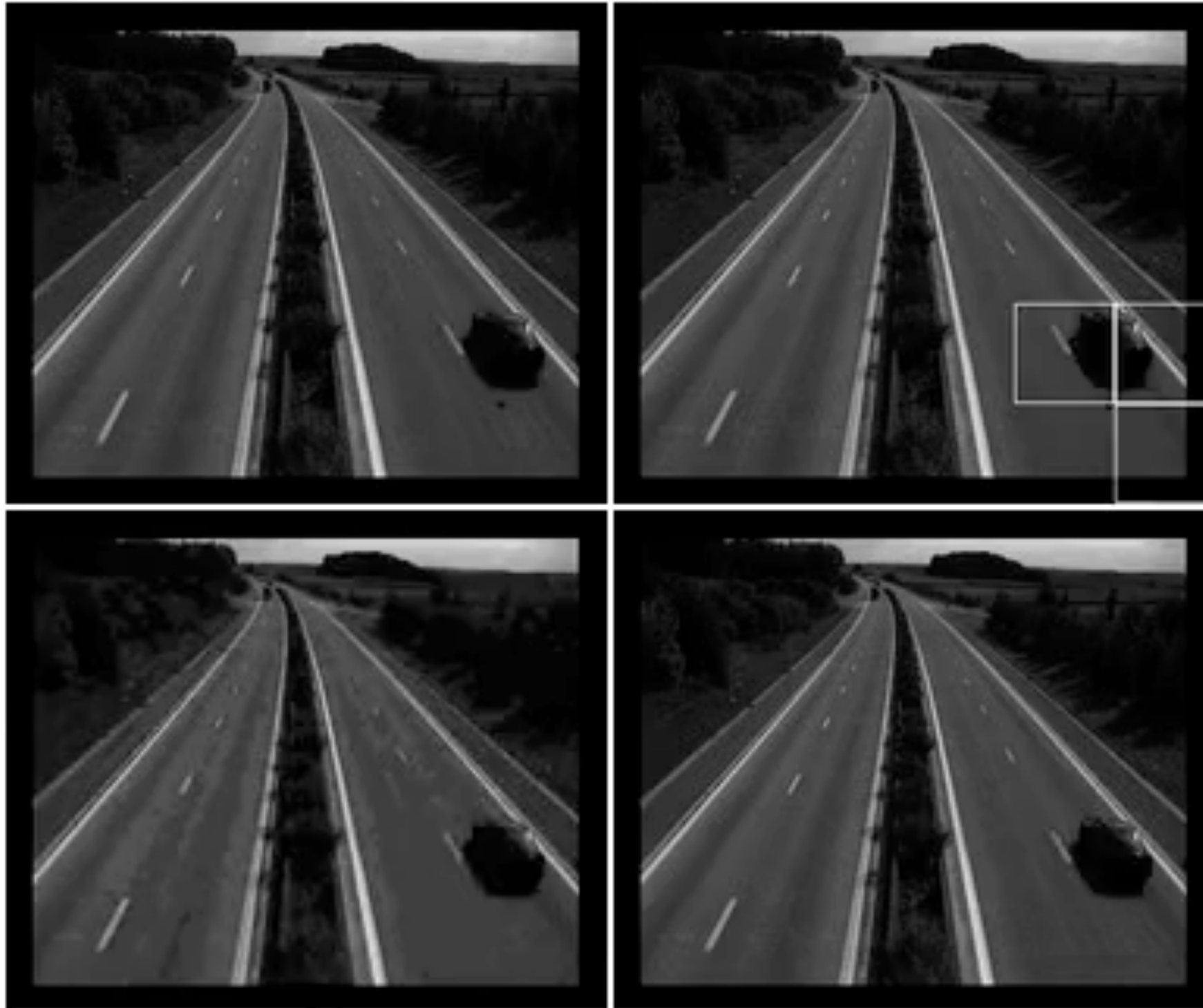
- cr + mc (1000 Bytes) —
- cr + mc (2000 Bytes) —
- cr + mc (3000 Bytes) —
- cr + mc (4000 Bytes) —
- cr + mc (5000 Bytes) —
- mj2 (1000 Bytes) —+
- mj2 (2000 Bytes) —x
- mj2 (3000 Bytes) —*
- mj2 (4000 Bytes) —□
- mj2 (5000 Bytes) —■

*Bytes per image

1000 Bytes per image



5000 Bytes per image



Experiments

Stockholm : sequence of 90 images of **1280x768** pixels.

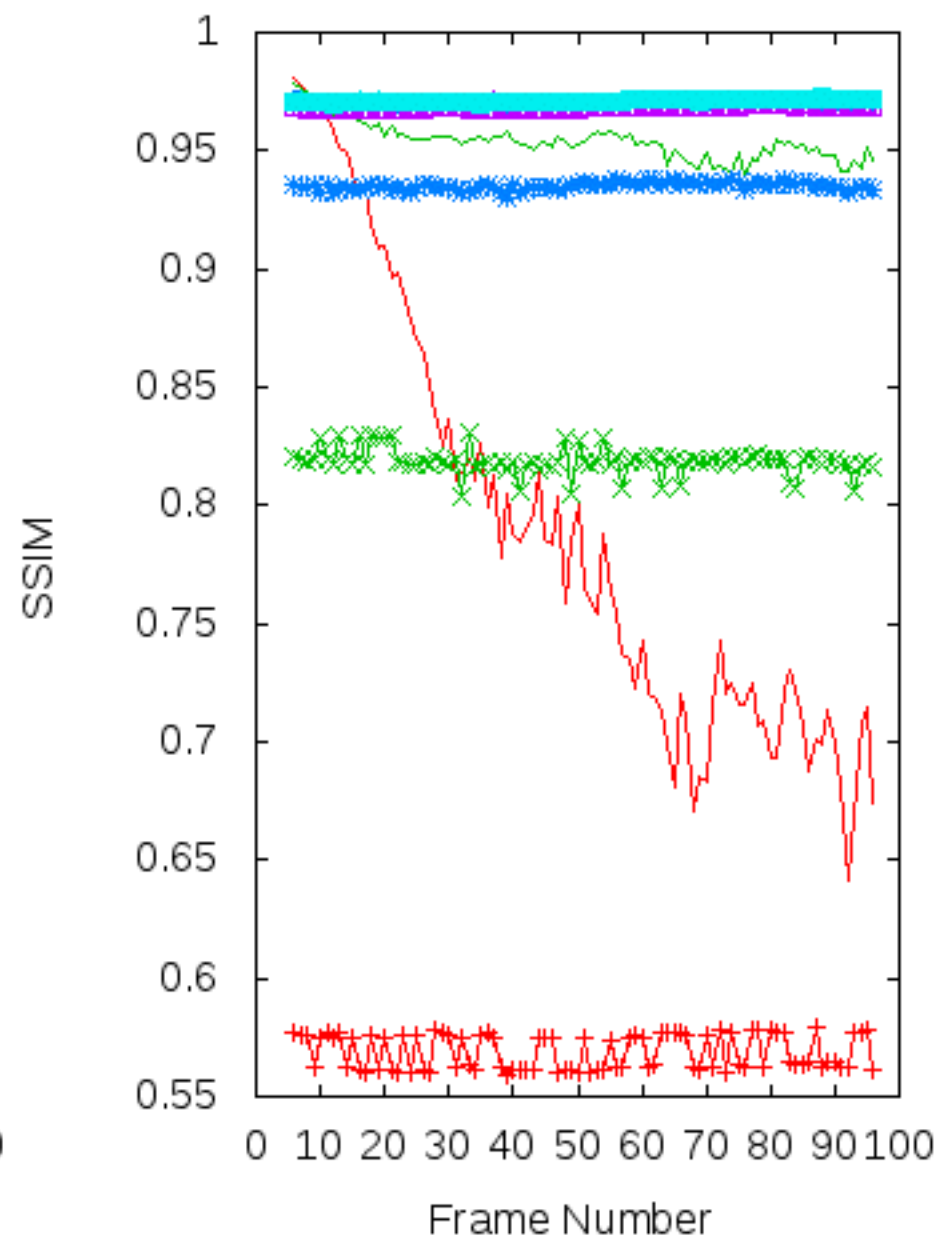
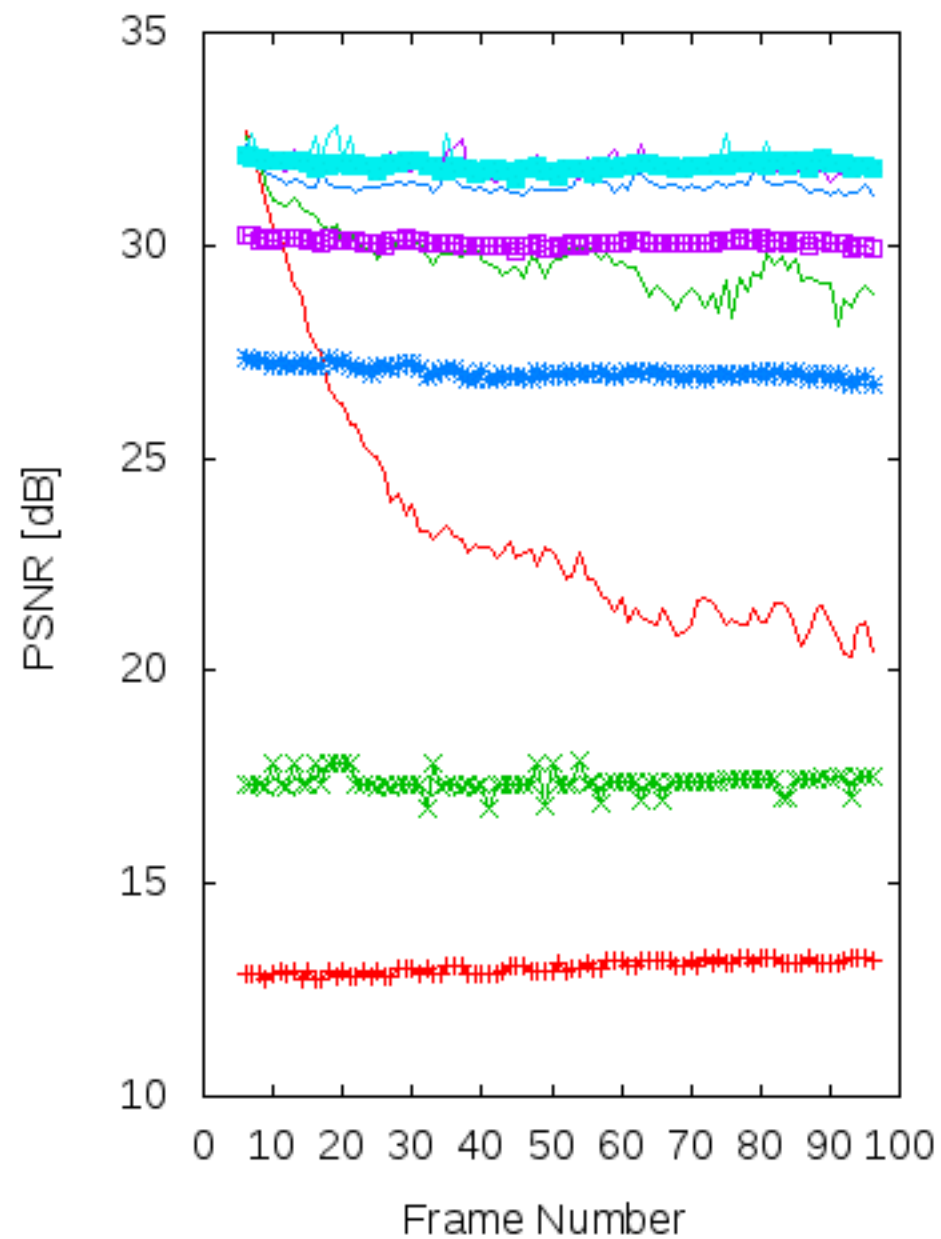
- Compression parameters used in **kdu_compress**

```
CLAYERS=8
CLEVELS=2
CPRECINCTS="{128,128},{64,64},{32,32}"
CBLK="{32,32}"
Corder=LRCP
```

- Motion estimation parameters

```
B=128    # block size
A=2      # subpixel accuracy
D=0      # border size
S=4      # search range
V=2      # block overlapping
```


PSNR and SSIM



cr + mc (11000 Bytes) —
cr + mc (21000 Bytes) —
cr + mc (31000 Bytes) —
cr + mc (41000 Bytes) —
cr + mc (51000 Bytes) —
mj2 (11000 Bytes) —+—
mj2 (21000 Bytes) —x—
mj2 (31000 Bytes) —*—
mj2 (41000 Bytes) —□—
mj2 (51000 Bytes) —■—

*Bytes per image

11000 Bytes per image



21000 Bytes per image



31000 Bytes per image



Experiments

Stockholm : sequence of 90 images of **1280x768** pixels.

- Compression parameters used in **kdu_compress**

```
CLAYERS=8
CLEVELS=2
CPRECINCTS="{ 32, 32 }, { 16, 16 }, { 8, 8 }"
CBLK="{ 8, 8 }"
Corder=LRCP
```

- Motion estimation parameters

```
B=128    # block size
A=2      # subpixel accuracy
D=0      # border size
S=4      # search range
V=2      # block overlapping
```

11000 Bytes per image



Thanks