Near-Lossless Coding of Plenoptic Camera Sensor Images for Archiving Light Field Array of Views

Emanuele Palma and Ioan Tabus
Information Technology and Communication Sciences
Tampere University
Tampere, Finland
name.surname@tuni.fi

***Abstract*—In this paper we propose a near-lossless encoder for sensor images acquired by plenoptic cameras, and we investigate its usage for encoding in an archive all information needed for reconstructing high quality versions of the light field (LF) array of views(AoV). The near-lossless encoding of the plenoptic camera sensor image is realized by a modified version of the recently published sparse relevant regressors and contexts (SRRC) encoder. The lossy reconstruction is obtained in two nested loops: the outer one operates over the sensor image patches (each patch corresponding to a microlens image), and the inner loop operates over the pixels in the patch. In the latter, we enforce the SRRC predictors to use the already reconstructed lossy version of the sensor image. Then, we examine the usage of the near-lossless SRRC (NL-SRRC) codec as a building block for an archiving scheme including all information needed for running the plenoptic processing pipeline and obtaining the LF-AoV. Finally, we replace in the archiving scheme the NL-SRRC codec with other state of the art lossy codecs and compare the results, which show that NL-SRRC based archiving scheme achieves better performance for the range of high bitrates.**

***Index Terms*—light field, compression, image processing, image compression, light field compression**

I. Introduction

The research area of light field image compression has been under active developments in the recent years, as part of the interest raised by the immersive media technologies.

Light Field (LF) images are conveying information about the color attribute of each point in a scene, as seen from several angular directions. The plenoptic cameras represent a fast mean of acquiring this information in a single shot, by using an array of microlenses in front of the image sensor. The acquired sensor image can be then processed by a plenoptic processing chain [1], to obtain in the end a $K_r \times K_c$ array of views (AoV), each view being an $n_r \times n_c$ color image, which represents the scene viewed from a certain angular direction. The LF-AoV is the final representation of the light field.

The lossy compression of the array of views produced from a plenoptic camera shot was studied extensively, see e.g., [2] for a comprehensive survey. A few competitive solutions have emerged [3], [4] leading to the finalized published standard [5]. The lossless compression of AoV was also investigated thoroughly, e.g., see references in [6]. Only a limited number of papers have studied the topic of near-lossless schemes, where the bitrates are above 1 bpp, e.g., [7], [8].

The lossless compression of plenoptic camera sensor images was studied in [9], which introduced "sparse relevant regressors and contexts" (SRRC) codec. Then the SRRC encoder was used in [10] as a building block for an archiver, that stored all information for reconstructing in a lossless way the LF-AoV, called "compress image and metadata for reconstruction of light field array of views" (CIMLF). The methods SRRC and CIMLF are modified in this paper so that they will not operate in a lossless regime, but instead they will provide *lossy versions* of the camera sensor image and of the LF-AoV, respectively, still at a very high quality.

A. Organization of the Paper

This contribution is structured as follows: in Section II the main structure of the NL-SRRC encoder is presented introducing modifications to the SRRC encoder [9]. In Section III a method for archiving the array of views via a new version of CIMLF [10] is illustrated and discussed. Section IV presents the experimental results obtained with NL-SRRC and with the lossy version of CIMLF, while Section V draws conclusions.

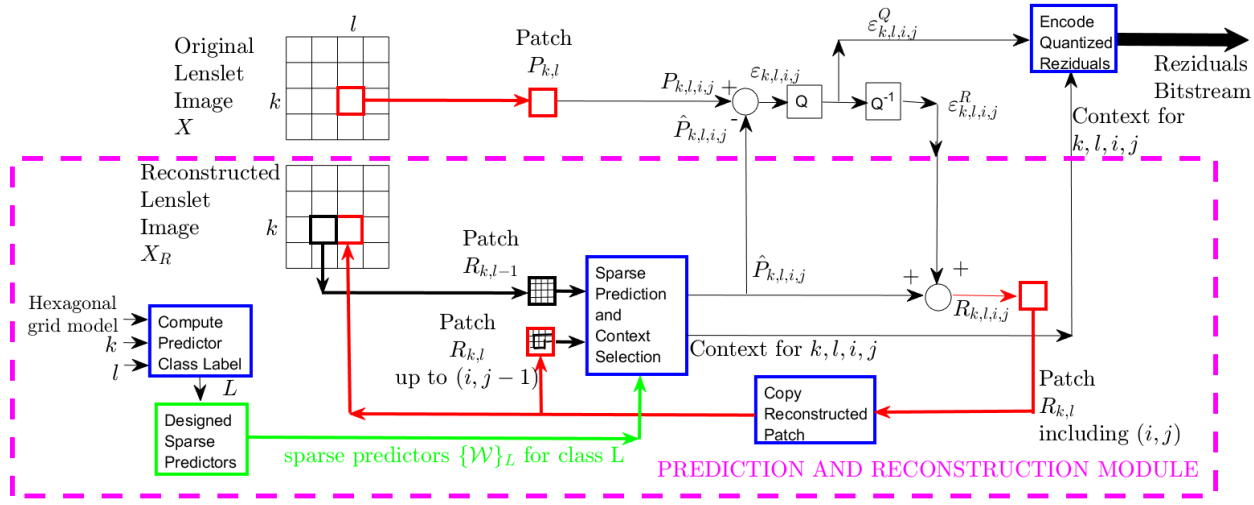
II. Near-Lossless Sparse Relevant Regressors and Contexts (NL-SRRC) Codec

A. A brief summary of the SRRC Codec

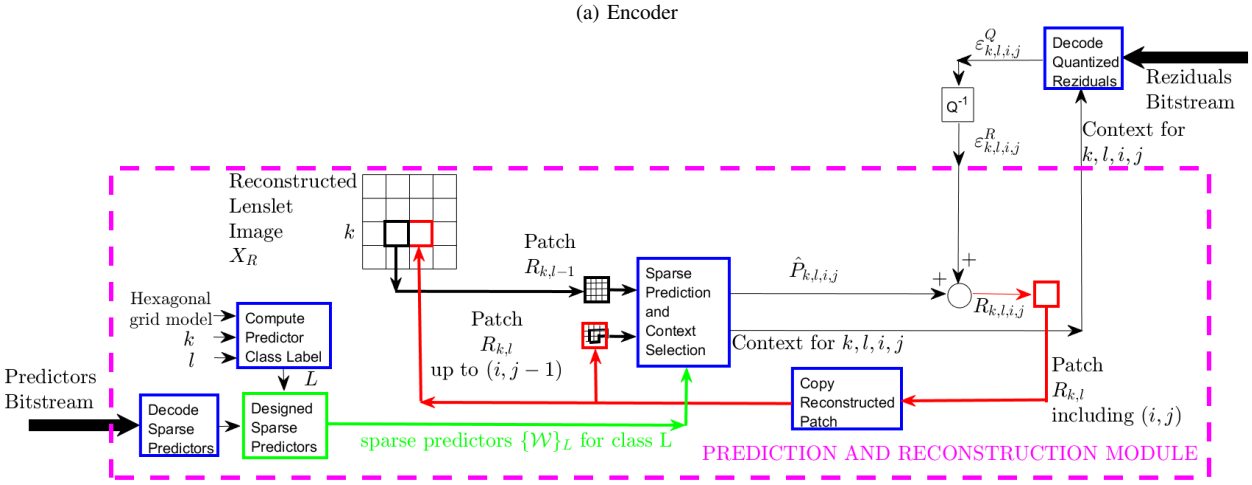
The plenoptic camera sensor images are acquired at a large $N_r \times N_c$ image sensor, in front of which there is a color filter array (CFA) for implementing the Bayer pattern sampling, in front of which there is an array of microlenses, having the centers located at the nodes of a hexagonal lattice. The images captured by neighbor microlenses are correlated, and the Bayer patterns induce also correlation patterns for the pixels of the sensor image. The encoding scheme SRRC [9] utilizes these dependencies very efficiently.

The SRRC encoder is a predictive scheme, in which each pixel value is predicted based on the values of the pixels in a causal predictive template, including all the causal pixels in the current patch $P_{k,l}$ and all pixels in the left neighbor patch,$P_{k,l-1}$. The sparse predictor design ensures that only the most relevant pixels from a large candidate template are selected for performing the prediction. In order to use efficient predictors, they are optimally designed for each triplet (i, j, L)

978-1-6654-6964-7/22/$31.00 ©2022 IEEE



(a) Encoder



(b) Decoder

Fig. 1: The encoding and decoding schemes of NL-SRRC

where (i,j) is the pixel index inside each 13×15 patch, and L is the patch label, derived from the patch alignment to the Bayer pattern [9].

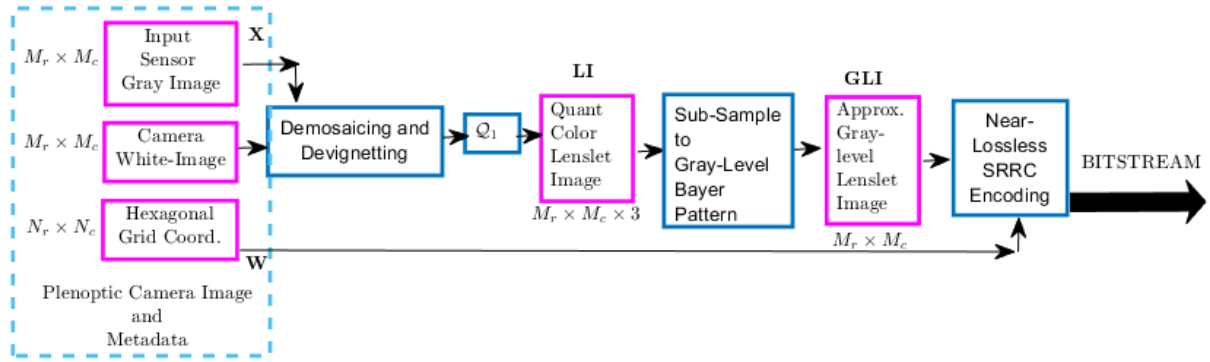
B. The NL-SRRC Codec

In the NL-SRRC codec the same predictive scheme is used as in SRRC, with dedicated sparse predictors for each triplet (i,j,L). The stage of predictor design is performed at the encoder exactly as in the SRRC codec. Briefly, first the needed statistics in the form of regression matrices are collected for each distinct (i,j,L) as presented in [9], resulting in the sparse parameters, specifying which regressors from the candidate template are used, and which values of the linear prediction coefficients are non-zero. We use these steps without any change in our near lossless SRRC scheme, and we refer to [9] for details.

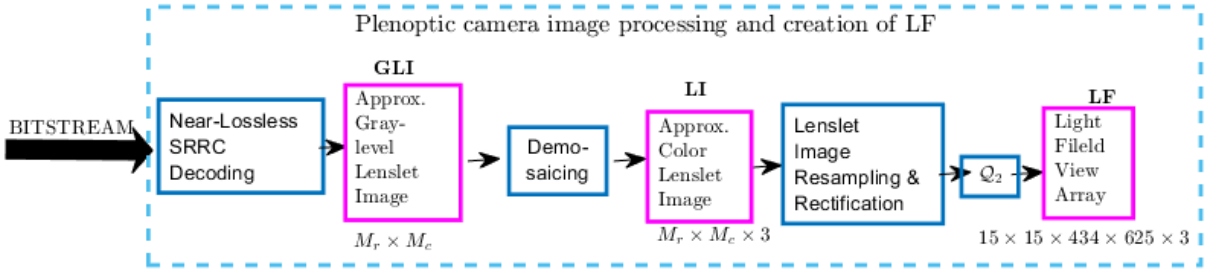
However, the following stage of computing the predicted values of the pixels, needs to be different from the one in SRRC, because the decoder has available only the lossy version of the image, which it reconstructs sequentially in the double loop over patches and pixels inside the patch. For getting the same prediction at both encoder and decoder, the encoder needs to maintain and update the current available lossy image version which will be available to the decoder, at the current patch (k,l) and pixel (i,j). Hence, the "Prediction and reconstruction module" emphasized in the violet dotted block in Fig. 1a will be organized in such a way that the encoder and decoder are using the same module.

At the encoder, the predicted value $\hat{P}_{k,l,i,j}$ computed by the sparse predictor is subtracted from the true pixel value $P_{k,l,i,j}$, to obtain the prediction error $\varepsilon_{k,l,i,j}$, then this prediction error is quantized, obtaining a quantized prediction error $\varepsilon^Q_{k,l,i,j}$ (which is the one encoded in the bitstream and transmitted to the decoder). The quantizer is applied to the three color components separately. For example, considering a color component $c \in \{R,G,B\}$ and denoting $p = P_{k,l,i.,j,c}$, $\hat{p} = \hat{P}_{k,l,i.,j,c}$, $\epsilon = \varepsilon_{k,l,i,j,c}$, and using the typical function for near-lossless schemes [11]:

$$\epsilon^Q = Q(\epsilon) = sign(\epsilon) \left\lfloor \frac{|\epsilon| + \delta}{2\delta + 1} \right\rfloor$$



(a) Archiving Scheme



(b) De-archiving Scheme

Fig. 2: Lossy CIMLF scheme. In the version dubbed Lossy CIMLF (NL-SRRC) the encoding and the decoding of the GLI image is done using NL-SRRC.

where δ controls the stepsize, and constitue the maximum absolute error allowed in the reconstruction, the encoder performs the inverse quantization to obtain a reconstructed quantized prediction error (the same way as the decoder will do) by using the function:

$$\epsilon^R = Q^{-1}(\epsilon^Q) = (2\delta + 1)\epsilon^Q$$

and then adds it to the predicted value $\hat{p}$, to get the reconstructed (lossy) pixel value $p^R = P^R_{k,l,i.,j,c} = \hat{p} + \epsilon^R$, which is stored into the local reconstructed image inside the module.

The inverse quantization is not a reversible operation, so in general $\epsilon^R \neq \epsilon$, but we can show that $|\epsilon^R - \epsilon| \leq \delta$, which in turn ensures the absolute value of the reconstruction error in the image of maximum δ, since the true image value p is the predicted value plus error ϵ, i.e., $p = \hat{p} + \epsilon$ and the reconstructed value is $p^R = \hat{p} + \epsilon^R$, so $|p^R - p| = |\epsilon^R - \epsilon| \leq \delta$.

To prove that $|\epsilon^R - \epsilon| \leq \delta$, since ϵ and δ are integer values, we denote $n = |\epsilon| + \delta$ and consider the (positive) integers division operation $\frac{n}{2\delta+1} = q + \frac{\gamma}{2\delta+1} = \lfloor \frac{n}{2\delta+1} \rfloor + \frac{\gamma}{2\delta+1}$ with the quotient $q = \lfloor \frac{n}{2\delta+1} \rfloor$ and the remainder $\gamma \in \{0, 1, \ldots, 2\delta\}$. From this, $\left\lfloor \frac{|\epsilon|+\delta}{2\delta+1} \right\rfloor = \frac{|\epsilon|+\delta}{2\delta+1} - \frac{\gamma}{2\delta+1} = \frac{|\epsilon|+\delta-\gamma}{2\delta+1}$ and,

$$\epsilon^R = sign(\epsilon)\,(|\epsilon| + \delta - \gamma) = \epsilon + sign(\epsilon)\,(\delta - \gamma)$$

from which $|\epsilon^R - \epsilon| = |\delta - \gamma| \leq \delta$, since $(\delta - \gamma) \in \{-\delta, \ldots, \delta\}$.

At the decoder, the "Prediction and reconstruction module" is connected to the incoming bitstream of quantized prediction errors, which are then inverse quantized and fed to the module for adding the predicted value and reconstructing the lossy version of the image.

The optimal sparse predictors designed at the encoder are forming a catalog of sparse predictors, which is encoded as presented in [9]. The decoder reconstructs this catalog of sparse predictors from the bitstream, to be used for the prediction in the double loop.

III. Archiving the Array of Views via Lossy CIMLF

CIMLF, or *Compress Image and Metadata for Reconstruction of Light Field Array of Views*, is a lossless archiving scheme built on top of the SRRC encoder, that we proposed in [10], which saves in an archive all elements necessary for reconstructing the LF-AoV, when the input lenslet image $\mathbb{X}$ representing the scene, the white image $\mathbb{W}$ (specific to each plenoptic camera) and all the necessary additional side information (*meta-data*) needed to losslessly reconstruct the LF-AoV are available.

The most efficient CIMLF archiver scheme called in [10] CIMLF[1] was mimicking the process by which a given LF-AoV was obtained, starting from the sensor images $\mathbb{X}$ and $\mathbb{W}$. CIMLF[1] selected one color lenslet intermediate image LI within the plenoptic chain, obtained after a devignetting and demosaicing operation (see left handside of Figure 2.a). To LI a sub-sampling similar to a color filter array is applied, in order to get a graylevel image denoted GLI. The GLI image is very similar to a plenoptic camera sensor image, having the same type of regularities, due to microlens image similarities and

Bayer pattern regularities, and can be encoded very efficiently by SRRC. However, by demosaicing the lossless version of GLI one obtains $\widehat{LI}$, which is only an approximate version of LI, and for that reason the CIMLF[1] archiver had to include a second coding operations, of $LI - \widehat{LI}$, in order to get a perfect lossless reconstruction of LF-AoV, resulting in a complex two branch encoder [10].

Here, we introduce the Lossy CIMLF archiver, which does not need to include the branch for encoding $LI - \widehat{LI}$, since removing that branch makes the LF-AoV lossy, but it still achieves a very high PSNR. The Lossy CIMLF operates with the much simpler scheme at the encoder presented in Figure 2.a), which encodes in the bitstream the image GLI in a lossy way using the NL-SRRC. Then it also encodes additional meta-information, mainly the parameters of the hexagonal lattice of the microarray and the parameters used by the matlab plenoptic pipeline [1], which require a very small size compared to the NL-SRRC bitstream size.

The decoder illustrated in Fig. 2 b) reconstructs the lossy version of GLI, which is then demosaiced and sent through the processing pipeline [1], which needs to use also the meta-information (specifying the matlab function parameters and the hexagonal lattice parameters) saved by the encoder.

We name the presented version Lossy CIMLF (NL-SRRC) and we consider for the experiments a version Lossy CIMLF (FLIF) where the NL-SRRC codec is replaced with the FLIF codec [12], and Lossy CIMLF (JPEG2000) where the NL-SRRC codec is replaced with the JPEG2000 codec.

IV. Experimental Results and Discussion

We first illustrate the performance of the NL-SRRC codec and then the performance of the Lossy CIMLF archiver on the light field datasets *Bikes, Danger de Mort, Stone and Pillars Outside and Fountain & Vincent 2* used in [13], which are part of the datasets introduced in [14], including data about the input sensor images $\mathbb{X}$ and $\mathbb{W}$, about the metdata information used in the plenoptic chain, and about the final LF-AoV.

Since in the Lossy CIMLF archiver we are using the NL-SRRC codec for lossy encoding of the GLI images (see Fig. 2a), derived from the input sensor images $\mathbb{X}$ and $\mathbb{W}$, we are illustrating next the performance of NL-SRRC over the GLI images corresponding to the four datasets.

A. Performance Evaluation of NL-SRRC

We compare in this section the performance of the lossy codec NL-SRRC with the performance of FLIF codec in lossy regime, and the performance of JPEG2000 codec. The results are shown in Fig.3 and Fig.4.

We consider the encoding of the images GLI for each of the four lightfield datasets. The image GLI is a graylevel image having a size $N_r \times N_c = 5368 \times 7728$ and 10 bits per pixel, and by construction its pixels are representing the values of the grbg colors in a Bayer pattern format, making it well suitable to be compressed by both SRRC and NL-SRRC.

We compute the PSNR of the lossy reconstructions of GLI for the plenoptic scene Bikes, by all three codecs: NL-SRRC, FLIF, and JPEG 2000 and show the rate distortion plots in Fig. 3. The best performance is achieved by NL-SRRC, followed by FLIF, which unfortunately could be run at its lowest quality factor only until 2 bpp, and then followed by JPEG 2000, at a quite big PSNR gap (reaching about 8dB at the highest bitrates).

It is interesting to notice the distribution of the reconstruction errors shown in Fig. 4, for the same Bikes GLI image: while the errors of NL-SRRC are bounded to be in the set of integers $\{-\delta, \ldots, \delta\}$ (exemplified in Figure 4 for $\delta = 1$ and $\delta = 7$), the other codecs have a wider dynamic range of the errors.

B. Performance Evaluation of the variants of Lossy CIMLF

We have run the archiver Lossy CIMLF in three configurations, where the lossy archiver for the GLI image in Figure 2 is NL-SRRC, FLIF, and JPEG 2000, respectively, for all the four light field datasets.

We then reconstructed from the archives the lossy versions of the LF-AoV, and compared them against the ground truth LF-AoV downloaded from [14]. We denote $LF_{k,l}$ the view (k, l), which is a RGB color image of size $(n_r \times n_c) = (434 \times 625)$, where $LF_{k,l}(i, j, c)$ represents the color component c ($c = 1, 2, 3$ for R,G,B, respectively) of the pixel (i, j) in view $LF_{k,l}$. We have then computed the PSNR$_{YUV}$ values according to [13] and we evaluated the rate as the size of the archive on the disk.

We show in Fig.6, leftmost panel, the rate-distortion curves with PSNR$_{YUV}$ values against the bitrate for *I01 Bikes*. As we can see, Lossy CIMLF(NL-SRRC) achieves better PSNR values in the high bitrate regime (bpp>2), e.g., for *I01 Bikes* at 4.2 bits per pixel (bpp), the encoder shows an increase of 2% in performance with 0.1 less bits per pixel allocation (PSNR$_{YUV}$ of 55.8636 dB, with Lossy CIMLF(FLIF) at 4.3 bpp reaching a peak value of 54.6 dB).

In order to better illustrate the performance of the three variants of Lossy CIMLF, we choose the angular view $LF_{12,12}$ from the LF-AoV of Bikes, and for each of the three Lossy CILMF versions, we compute the image of reconstruction errors $e_{k,l} = LF_{k,l} - \widehat{LF}_{k,l}$. For each image of errors we compute the average error image $\overline{e}_{k,l}(i, j) = \sum_{c=1}^{3} e_{k,l}(i, j, c)/3$.

To visualize the errors and compare their spatial distributions, we dispaly in Fig. 5 the grayscale images $\overline{e}_{12,12}$ of size $(n_r \times n_c) = (434 \times 625)$, using the colormap shown in the colorbar next to each panel, so that small errors are shown in blue and high errors in red. In column 1 of Fig. 5 we have the three panels representing the $\overline{e}_{12,12}$ for Lossy CIMLF(NL-SRRC) at three values of δ: 1,3, and 7, while in each row of Fig. 5 we have corresponding panels for Lossy CIMLF(FLIF) and Lossy CIMLF(JPEG2000), when the bitrates used by these two coders where as close as possible (but slightly higher) than the rate used by Lossy CIMLF(NL-SRRC) shown on the same row. Indeed, comparing the images over each row shows that for all values of rates considered, the Lossy CIMLF(NL-SRRC) has a more favorable distribution of the errors, as compared to the other two. Furthermore, in the fourth column

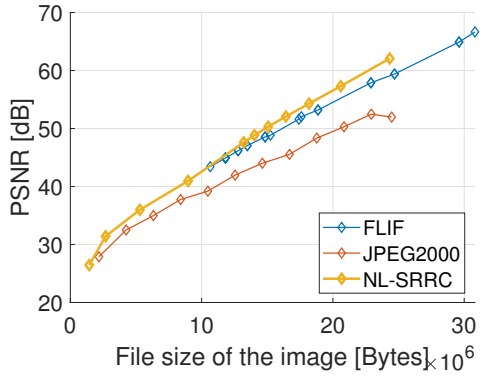


Fig. 3: Rate-Distorsion performance curves for *Bikes* gray-level image (GLI) of the NL-SRRC encoding scheme (PSNR-File size)

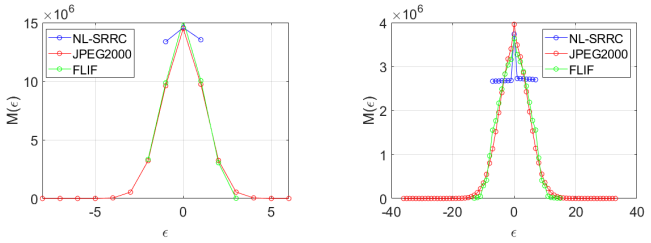


Fig. 4: Histogram $M(\epsilon)$ of the reconstruction errors for $\delta = 1$ (left) and $\delta = 7$ (right).

of Fig. 5 we show the histogram of $e_{12,12}(i, j, c)$, collected for all pixels (i, j) and all colors $c \in \{1, 2, 3\}$. Again, here it is visible that the distribution of errors from Lossy CIMLF(NL-SRRC) is narrower than that from the other two coders. The quantitative values of $PSNR_{YUV}$, and $SSIM$ listed under each panel concur to the conclusion that Lossy CIMLF(NL-SRRC) is superior, under this range of bitrates.

We have furthermore uploaded a version of our pipeline as a Matlab Code with all the needed input files, stored as a $.zip$ archive, one for each plenoptic scene. The readme file will help in the installation and usage of the code. All the files and the installation instructions can be found at [15]. This was done to allow the reproducibility of the results and to guarantee the clearness of the content.

C. Comparison with the state of the art

Finally, we compare the performance of the Lossy CIMLF archivers with a state-of-the-art coding algorithm for LF-AoV, namely the High Efficiency Video Coding (HEVC) [16], which is used in the majority of papers dealing with lossy LF compression. The LF encoding is done arranging all views in a pseudo-video sequence (PVS), as presented in [13]. At very low bitrates the PSNR_{YUV} achieved by HEVC is better than that of any presented encoding scheme: at 0.75 the PSNR is above 40 dB, where instead Lossy CIMLF(NL-SRRC) achieves lower levels. In the 0 to 1 bpp regime then, HEVC shows its true efficiency, setting back after the 2 bpp threshold. From this bitrate value onwards, Lossy CIMLF(NL-SRRC) performs better than HEVC, improving the PSNR value by almost 2 dB already at around 2.5 bpp.

V. Conclusions

This contribution introduces an efficient Near-Lossless plenoptic camera sensor image encoder, and presents its usage for lossy archiving a LF-AoV, exploiting efficiently the redundancies within the sensor image. The Lossy CIMLF archiving based on NL-SRRC proves to be very efficient, surpassing its versions based on FLIF and JPEG 2000, and also surpassing HEVC codec for the range of rates higher than 2bpp, where high quality LF-AoV are obtained.

Acknowledgment

The work of the first author has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764951.

References

[1] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 1027–1034, IEEE Computer Society, 2013.

[2] C. Conti, L. D. Soares, and P. Nunes, "Dense light field coding: A survey," *IEEE Access*, vol. 8, pp. 49244–49284, 2020.

[3] P. Astola and I. Tabus, "Coding of light fields using disparity-based sparse prediction," *IEEE Access*, vol. 7, pp. 176820–176837, 2019.

[4] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4D DCT-Based lenslet light field codec," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 435–439, Oct 2018.

[5] ISO/IEC JTC1/SC29/WG1, "Information technology - Plenoptic image coding system (JPEG Pleno) - Part 2: Light field coding," in *ISO/IEC JTC 1/SC29/WG1, Doc. ISO/IEC IS-21794-2:2021*, 2021.

[6] M. Mukati and S. Forchhammer, "Epipolar plane image-based lossless and near-lossless light field compression," *IEEE Access*, vol. 9, pp. 1124–1136, 2021.

[7] Y.-H. Chao, G. Cheung, and A. Ortega, "Pre-demosaic light field image compression using graph lifting transform," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3240–3244, 2017.

[8] M. Rizkallah, T. Maugey, and C. Guillemot, "Prediction and sampling with local graph transforms for quasi-lossless light field compression," *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, 12 2019.

[9] I. Tabus and E. Palma, "Lossless compression of plenoptic camera sensor images," *IEEE Access*, vol. 9, pp. 31092–31103, 2021.

[10] I. Tabus and E. Palma, "Lossless archiving view arrays from plenoptic cameras when camera sensor images are available," in *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, 2021.

[11] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.

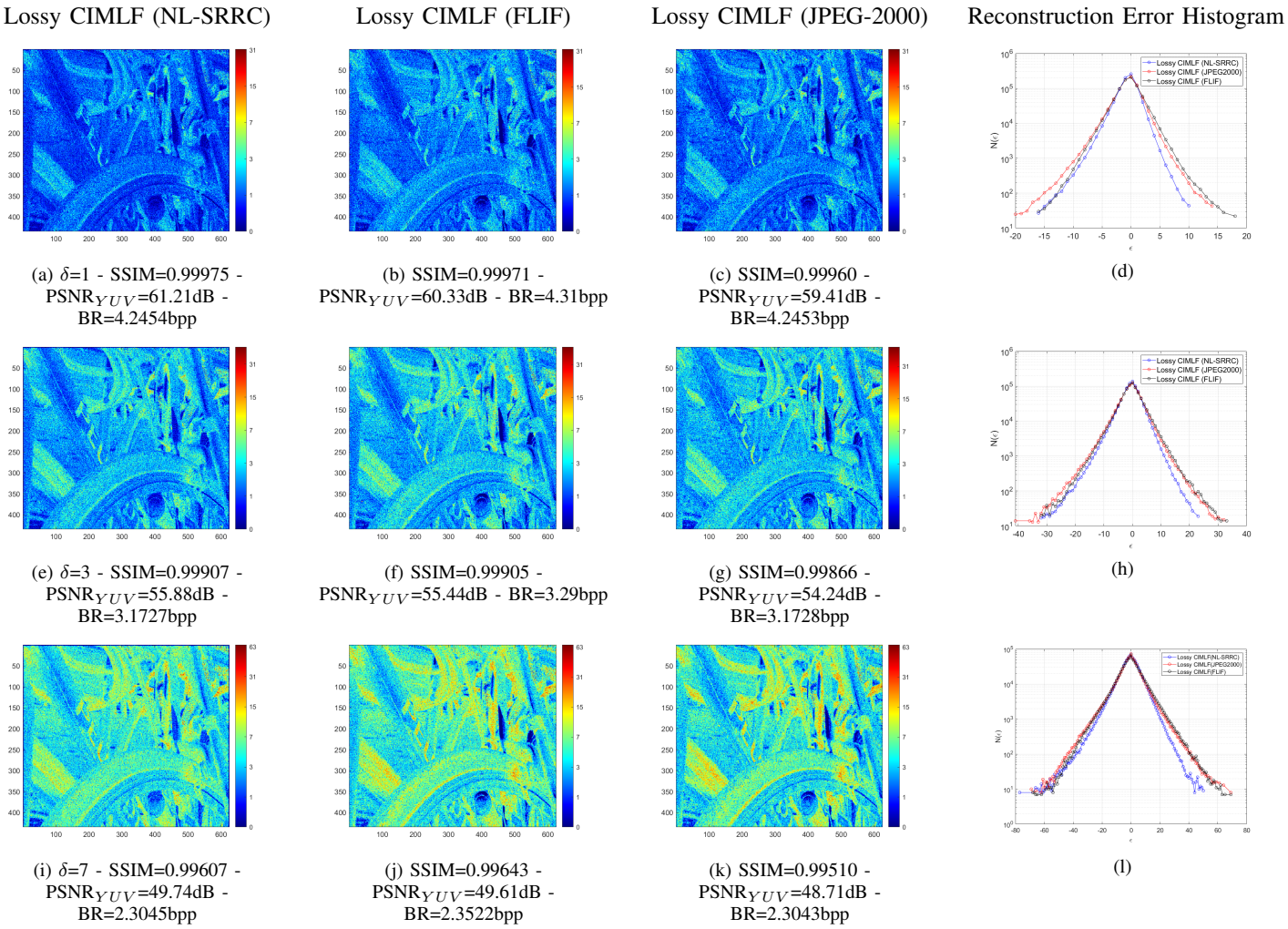
[12] J. Sneyers and P. Wuille, "FLIF: Free lossless image format based on maniac compression," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 66–70, 2016.

[13] ISO/IEC JTC 1/SC 29/WG 1, "JPEG Pleno Light Field Coding Common Test Conditions," in *ISO/IEC JTC 1/SC29/WG1, Doc. N84027*, 2019.

[14] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX),*, 2016.

[15] E. Palma and I. Tabus, "NL-SRRC and Lossy CIMLF Schemes." https://homepages.tuni.fi/emanuele.palma/NL_SRRC_Lossy_CIMLF_Schemes.zip, 2022.

[16] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding {HEVC} standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.



(a) δ=1 - SSIM=0.99975 - PSNR$_{YUV}$=61.21dB - BR=4.2454bpp

(b) SSIM=0.99971 - PSNR$_{YUV}$=60.33dB - BR=4.31bpp

(c) SSIM=0.99960 - PSNR$_{YUV}$=59.41dB - BR=4.2453bpp

(d)

(e) δ=3 - SSIM=0.99907 - PSNR$_{YUV}$=55.88dB - BR=3.1727bpp

(f) SSIM=0.99905 - PSNR$_{YUV}$=55.44dB - BR=3.29bpp

(g) SSIM=0.99866 - PSNR$_{YUV}$=54.24dB - BR=3.1728bpp

(h)

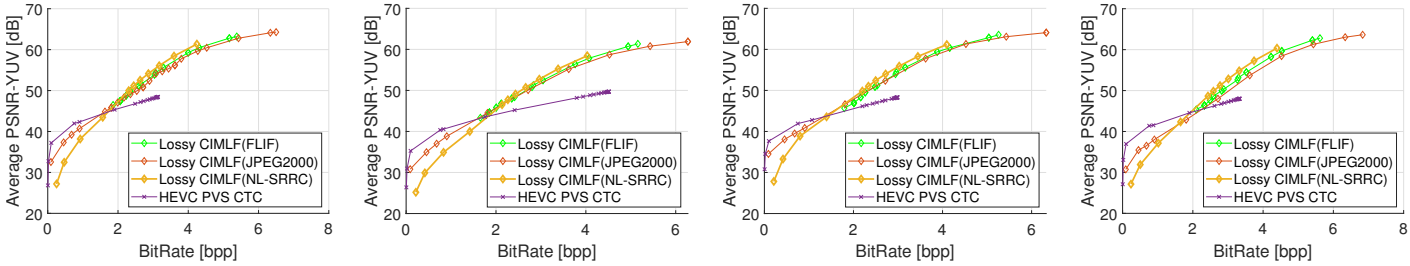
(i) δ=7 - SSIM=0.99607 - PSNR$_{YUV}$=49.74dB - BR=2.3045bpp

(j) SSIM=0.99643 - PSNR$_{YUV}$=49.61dB - BR=2.3522bpp

(k) SSIM=0.99510 - PSNR$_{YUV}$=48.71dB - BR=2.3043bpp

(l)

Fig. 5: Comparison of Reconstruction Error Images for the View (12,12) of *I01 Bikes*. In each row the images $\overline{e}_{12,12}$ should be visually comparable, because they were encoded with the same target bitrate, corresponding to the Lossy CIMLF (NL-SRRC) bitrate at the values of δ=1 (a,b,c), δ=3 (e,f,g), δ=7 (i,j,k).



(a) I01 Bikes

(b) I02 Danger de Mort

(c) I04 Stone and Pillars Outside

(d) I09 Fountain and Vincent 2

Fig. 6: Rate-Distorsion Performance Curves (PSNR-Bitrate) for *Bikes, Danger de Mort, Stone and Pillars Outside and Fountain & Vincent 2* of our presented archivers against the encoding done via HEVC, as presented in [13], dubbed HEVC PVS CTC.