Light Field Compression of HDCA Images Combining Linear Prediction and JPEG 2000

Pekka Astola and Ioan Tabus
Tampere University of Technology
Laboratory of Signal Processing
P.O. Box 553, FI-33101, Tampere, Finland

***Abstract*—We have proposed under JPEG Pleno standardization activities a scheme for lenslet image compression, where the regularities and similarities existing between neighbor angular views were successfully exploited, achieving competitive results in the JPEG Pleno core experiments using lenslet data. This paper proposes improvements on our previous scheme of light field compression, making our approach more suitable for compression of light fields acquired with dense camera arrays, where the disparities between farthest views can reach several hundreds of pixels. We review the functional blocks of the compression algorithm, replacing and modifying some of the functionality with more advanced and efficient solutions. Based on our submission to the JPEG Pleno core experiments, we present and discuss our results obtained on the Fraunhofer HDCA dataset. Additionally, we present a new view merging algorithm which substantially increases the PSNR at all bit rates.**

I. Introduction

Light field and plenoptic imaging are emerging imaging technologies which are used in a wide range of computer vision applications, such as virtual reality systems, medical imaging and robotics. These technologies provide an extension to the already well established stereo imaging which has for long been used in inferring scene geometry and depth in computer vision.

Light fields can also be acquired by using a dense array of cameras, which are often implemented with a high precision robot, sampling the view at adjacent locations along both vertical and horizontal directions [1]. We refer to this type of setup as high density camera array (HDCA). Compared to consumer grade light field devices such as Lytro Illum, HDCA systems provide several orders of magnitude more data and offer a larger change in perspective along the views. Nonetheless, the high quality of the individual views and the precise actuation of the imaging rig provides very high redundancy between adjacent views.

Recent interest in light field imaging has led JPEG to initiate the standardization of JPEG Pleno [2][3][4][5]. Based on our previous work on lossless light field coding [6][7], we have proposed a similar scheme in JPEG Pleno for lossy light field compression [8]. By successfully exploiting the similarities between adjacent angular views we have already achieved competitive results in JPEG Pleno core experiments for the lenslet datasets using the scheme from [8].

In this paper we implement several modifications to our previous light field coding scheme intended to handle both the geometrical and color redundancy among adjacent views more efficiently. We study the efficiency of our compression scheme for encoding the HDCA dataset [1], used in the JPEG Pleno standardization efforts.

II. Light field Coding Scheme

HDCA data consists of an array $N_h \times N_v$ of adjacent views with identical dimensions $n_r \times n_c$. The full set covering the whole array of views has the indices denoted as a set Γ. In our scheme the user has to choose first a (very sparse) set of reference views, with indices denoted as Γ_{ref}, to be used as references for encoding the rest of views. We encode the views at Γ_{ref} using already existing image coding tools, namely JPEG 2000. Additionally, we encode a quantized version of scene depth at each view selected by Γ_{ref} using [9]. The rest of the views, having indices in the set denoted $\Gamma_{side} = \Gamma \setminus \Gamma_{ref}$, are decoded by exploiting the redundancies between the views in sets Γ_{ref} and Γ_{side}.

A high quality depth estimation at Γ_{ref} is necessary in order to efficiently reconstruct the side views. The importance of the accuracy of the inferred scene depth increases as the distance between the reference and side views increases. In our experiments we use the depth provided in [10][11] for JPEG Pleno core experiments [12]. Our scheme greatly benefits from the use of [9] in efficient encoding of the depth.

First we describe the underlying principle for the encoding scheme. We have used in [8] sparse prediction as a tool to identify the relevant regressor elements when predicting one view based on its neighbor views, which is feasible and very efficient in the case of lenslet images. The success in lenslet case is due to the small disparities between adjacent views, of at most one-two pixels. In such a situation, a well designed template can explore the close pixels and the close views (in the whole 4D light field domain) to find the relevant regressors for prediction, with no need of further aligning the views before prediction. However, in the HDCA case, a similar approach will require to consider huge candidate regressor templates. There is one more impediment for our earlier scheme. In our practical implementation of [8], we have chosen a template of prediction based on those neighbor views of the current view, that were already encoded. All views were encoded by advancing in a certain scanning order of the array of views (a spiral way from center to boundary). However this results in a limited random access capability: one has to

ISBN 978-90-827970-1-5 © EURASIP 2018

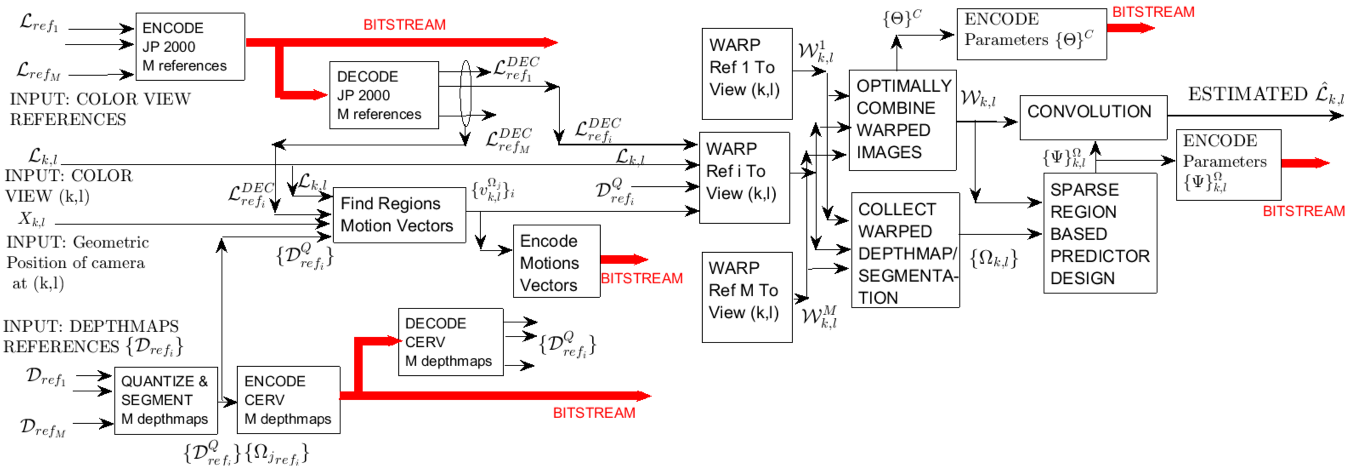
$\mathcal{L}_{ref_1}$
$\mathcal{L}_{ref_M}$
ENCODE JP 2000 M references
INPUT: COLOR VIEW REFERENCES
BITSTREAM
DECODE JP 2000 M references
$\mathcal{L}^{DEC}_{ref_1}$
$\mathcal{L}^{DEC}_{ref_M}$
$\mathcal{L}^{DEC}_{ref_i}$
$\mathcal{L}_{k,l}$
INPUT: COLOR VIEW (k,l)
$X_{k,l}$
INPUT: Geometric Position of camera at (k,l)
$\mathcal{L}^{DEC}_{ref_i}$
$\mathcal{L}_{k,l}$
Find Regions Motion Vectors
$\{v^{\Omega_j}_{k,l}\}_i$
$\{\mathcal{D}^Q_{ref_i}\}$
Encode Motions Vectors
BITSTREAM
$\mathcal{D}^Q_{ref_i}$
INPUT: DEPTHMAPS REFERENCES $\{\mathcal{D}_{ref_i}\}$
$\mathcal{D}_{ref_1}$
$\mathcal{D}_{ref_M}$
QUANTIZE & SEGMENT M depthmaps
ENCODE CERV M depthmaps
DECODE CERV M depthmaps
$\{\mathcal{D}^Q_{ref_i}\}$
BITSTREAM
$\{\mathcal{D}^Q_{ref_i}\}\{\Omega_{j_{ref_i}}\}$
WARP Ref 1 To View (k,l)
WARP Ref i To View (k,l)
WARP Ref M To View (k,l)
$\mathcal{W}^1_{k,l}$
$\mathcal{W}^M_{k,l}$
$\{\Theta\}^C$
ENCODE Parameters $\{\Theta\}^C$
BITSTREAM
OPTIMALLY COMBINE WARPED IMAGES
$\mathcal{W}_{k,l}$
CONVOLUTION
ESTIMATED $\hat{\mathcal{L}}_{k,l}$
COLLECT WARPED DEPTHMAP/ SEGMENTA- TION
$\{\Omega_{k,l}\}$
$\{\Psi\}^\Omega_{k,l}$
SPARSE REGION BASED PREDICTOR DESIGN
ENCODE Parameters $\{\Psi\}^\Omega_{k,l}$
BITSTREAM

Fig. 1. Block diagram of the encoder.

decode all prior views along the spiral, before being able to decode the view of interest in a random access application.

In a different configuration, one can decide to encode and decode most of the views based on a small set of reference views, which are needed to be decoded prior to decoding any other view. This configuration, which we call for short random access configuration, is introducing constraints in the choice of what type of redundancy can be used: in the spiral case the encoding is very efficiently done based on neighboring views, which are extremely similar. In the random access configuration some of the views will be encoded based on more distant views, where the correlations are typically weaker than between neighboring views (the more the viewing angle is changed, the more the obtained images differ).

In the HDCA case, the corresponding pixel in a reference view that corresponds to a pixel in the current view is further apart, and one has to know this disparity for being able to make an efficient prediction. Hence now the preliminary stage of warping the reference view for aligning with the current view is an essential step, which we undertake before applying the prediction, in a similar way to the sparse prediction of stereo color images we described in [13]. When several references are available, each has to be warped to the location of the target view before combining their warped versions into a prediction.

III. Description of the Functional Blocks

Fig. 1 illustrates the proposed encoding scheme as a block diagram. Next, we describe the functional blocks, and for those that have a correspondent in [8] we briefly make note of similarities and differences.

The proposed compression scheme is based on our previous work [8], which divided the encoding process into nine functional blocks. In this section we review the modifications to the existing block scheme.

Accessing the rectified light field structure: In the case of lenslet data considered in [8], the data was accessible in several forms: lenslet image, non-rectified aperture images, and rectified aperture images. The HDCA case corresponds to the last form, and is the easiest, since it does not require any pre-processing of data before encoding.

In the current scheme, the available views are split into M reference views, denoted $\mathcal{L}_{ref_1}, \dots, \mathcal{L}_{ref_M}$, having indices in the view-array specified in the M elements of the index set Γ_{ref}.

Estimating and quantizing disparity maps: The Block 2 in [8] performed the disparity estimation and quantization at the center view only. Here we propose a more flexible approach by handling an arbitrary number of reference disparity maps. We consider for exemplification mainly the case of $M = 5$ color references and disparity maps, which has a great random access capability: after decoding five references, the access to any particular view is ensured. In the experiments we have used the reciprocal depthmaps $\{\mathcal{D}_{ref_m}\}, m = 1 : 5$ and the estimates of the camera positions $\{\mathbf{X}_{k,l}\}$ provided by [10][11] for JPEG Pleno core experiments.

We first median filter the reciprocal depth data to enforce a smoother data surface, which is useful for better depth image compression, for which we use crack-edge region value (CERV) encoder [9]. Quantization of this disparity is the next processing step towards compression, to ensure a flexible rate distortion of the decoded disparity. Before encoding the depth at each of the references in Γ_{ref} we apply Lloyd quantization using n_Q levels. This reduces the initial 16-bit representation of the depth to a representation with n_Q levels distributed optimally according to the histogram of the disparity image. By choosing a smaller n_Q, the quantization becomes coarser, and the disparity map can have a more efficient lossless encoding by [9], but the distortions introduced are higher. Fig. 2 illustrates the performance of this approach for HDCA dataset S_2, with a varying number of quantization levels. The PSNR reflects the average over Γ_{ref} and the bit rate is reported as bits per pixel.

Disparity and motion vectors estimation for a generic view: We now describe the process for finding the disparity needed when predicting a generic view, with indices denoted (k, l).

As a first step towards finding the disparity we have used

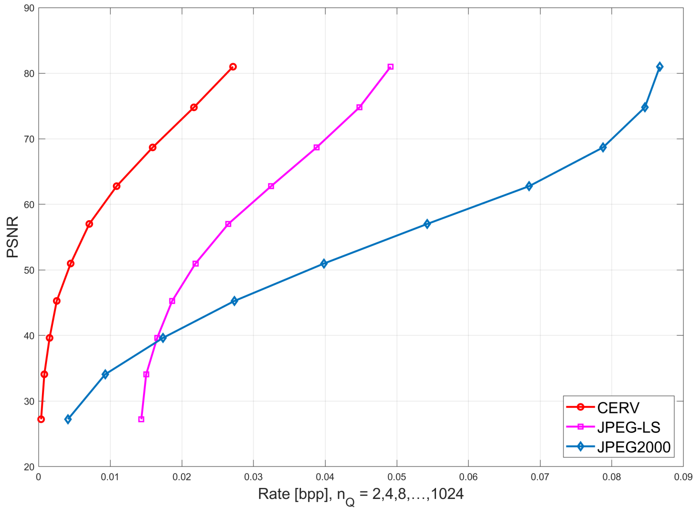

Fig. 2. The average Rate Distortion performance when encoding the HDCA disparities dataset S_2 [10]. For each reference disparity view 10 nonuniform quantizers were designed having the number of levels $n_Q = 2^1, 2^2, \ldots, 2^{10}$. The average PSNR and rate obtained by CERV, JPEG 2000, and JPEG-LS at each n_Q are shown in the plot. The R-D point obtained for $n_Q = 2$ is the left-bottom point and n_Q increases in order along the curve.

the simple mechanism presented in [10] for converting the reciprocal depthmaps for each of the M references to disparity maps between a reference and a given view. In this process, the geometrical coordinates of the camera at the reference, $\mathbf{X}_{ref_m}$, and at the given view $\mathbf{X}_{k,l}$ are used, being estimated by the method in [11]. This disparity data is anchored at the reference and we perform a disparity based warping process to get the disparity data anchored at the view k, l. Since the disparity data is quantized to n_Q levels, we obtain implicitly a segmentation of the view k, l in n_Q non-connected regions. For each region, the horizontal and vertical displacements of the region from the color warped image to the true view k, l are estimated by the block "Find Regions, Motion Vectors" in Fig. 1. The final warped version $\mathbf{W}^m_{k,l}$ obtained by warping the color light field reference view $\mathcal{L}_{ref_m}$ to the location of the view (k, l) is obtained by warping with the combined motion vector and pixel disparity, for each pixel of the reference. Due to occlusions, not all pixels at the view (k, l) receive a value. However, since we use M distinct references, the pixels that are missing in all the views $\mathbf{W}^m_{k,l}$ will be very few. They will be filled by successive median filtering of the combined warped image $\mathbf{W}_{k,l}$, obtained as presented next.

The Block 3 in [8] covers for the estimation of horizontal and vertical disparities between the center and the side views. Now we use an equivalent of Block 3 as a disparity refinement step. We note that the functionality of Block 5 from [8], that of generating segmentations for each view is covered now by the described process of warping of disparity information, and is marked in Fig. 1 as the block "Collect Warped Depthmap/Segmentation".

We stress that by warping we mean simultaneously creating a warped color image and a warped disparity image, by using the same movements of the pixels.

Encoding blocks: All the blocks in Fig. 1 that perform encoding have their output marked in red, signifying that their outputs are sent to the final bitstream, which is obtained by multiplexing all the block bitstreams. The final bitstream will contain also additional metadata.

All the encoding blocks from [8], namely encoding disparity refinement motion vectors, encoding of depth for reference views, encoding the predictor parameters are very similar to the ones that we use in the current scheme.

Reconstructing a side view from available references: In [8], the Block 8 used to perform the encoding of a side view using a sparse linear filter (dedicated to each region and view), which used as regressors at most 8 neighboring views. Having five references, and their disparities, we replace this with an approach similar to conventional depth-image-based rendering for warping each side view conditional on the reference views, explained in the next section. The main difference to [8] is the necessity of warping the views prior to prediction. That introduces also the need of merging various warped views.

IV. Predicting and Encoding a Side View Based on the Reference Views

The main functionality of the codec is to predict one view based on the information from the reference views. We represent in Fig. 1 in more detail this function, which encompasses the whole right half of Fig. 1, including the blocks of Warping, Optimally Combine Warped Images, Collect Warped Depthmap/Segmentation, Sparse Region Based Predictor Design.

A. Warping of disparity

We assume that reciprocal depthmap is encoded only for views in Γ_{ref}. For each reference, we obtain its corresponding disparities, both horizontal and vertical, anchored at the side view by warping its reciprocal depth. Inverse depth is a floating point quantity which during warping has to be multiplied by the horizontal and vertical spacing between the reference and side view. Rounding the resulting values to integers produces the integer precision disparity estimate between the views, which we use for warping.

After obtaining the warped disparity views $\mathbf{G}^m_{k,l}$ for each $m \in \Gamma_{ref}$ a proper merging of the results needs to be performed to obtain $\mathbf{G}_{k,l}$. The difficulty consists in the fact that due to occlusions the warping process is not assigning values to $\mathbf{G}^m_{k,l}$ at each pixel location.

A simple scheme is to arrange the references by their closeness, in the view array, to the current view (k, l). Then we pick the closest reference m_1 and fill $\mathbf{G}_{k,l}$ with all non-occluded values from $\mathbf{G}^{m_1}_{k,l}$; then proceed and fill only the missing values in $\mathbf{G}_{k,l}$ with the non-occluded values from next closest reference m_2, and continue the same way until the last reference. The last remaining missing pixels in $\mathbf{G}_{k,l}$ will be filled by successive median filtering.

ISBN 978-90-827970-1-5 © EURASIP 2018

B. Disparity refinement by motion vectors

The quality of the obtained disparity at the side view is dependent on the accuracy of the reciprocal depth at the reference and on the proper scaling with the vertical and horizontal baselines. These quantities are subject to estimation and quantization errors, and thus we perform a disparity refinement process at each side view for each reference in Γ_{ref}. The motion vectors are searched on a small search window, since the large disparities (values in the order of several hundred pixels) are approximately known from the reciprocal depthmap. The refinement process finds small adjustments specific to each local region at the side view (k, l).

Code length for the disparity refinement motion vectors is relative to the search radius and the number of unique levels in initial warped disparity. For one side view with search window of $w \times w$ code length becomes $CL_{MV} = n_Q M \log_2(w^2)$. The encoding of these vectors is very similar to the encoding of the displacements in Block 4 of [8].

C. Least-squares view merging

The fusion method presented in Section IVA is a "hard-decision" warping, based on the closeness of the reference views, and it does not make the best use of the fact that each non-occluded pixel in $\mathbf{W}^m_{k,l}$ contains a noisy estimate of the corresponding pixel's value in the side view. We introduce the selector $\delta^m_{k,l}(i, j)$ which equals 1 for a non-occluded pixel $\mathbf{W}^m_{k,l}(i, j)$, and 0 otherwise. We use the following algorithm to merge the warped reference views. Over all warped views in Γ_{ref}, obtain for each pixel a classification based on the number of reference views in which it was a non-occluded pixel. In our setting we have M reference views in Γ_{ref} so overall we have at most 2^M classes for each pixel. In Fig. 3 we show by different colors the resulting classes of the pixels, $c = 0, 1, \ldots, 2^M - 1$. For a pixel (i, j) belonging to class c, we evaluate the merged $\mathbf{W}_{k,l}(i, j)$ as:

$$\mathbf{W}_{k,l}(i,j) = \sum_{m=1}^{M} \mathbf{W}^m_{k,l}(i,j)\delta^m_{k,l}(i,j)\theta^c_m,$$

where the optimal parameters $\Theta^c = [\theta^c_1, \theta^c_2, \ldots, \theta^c_M]$ are obtained for each class by performing a least-squares design by minimizing the sum of residuals for every pixel (i, j) belonging to class c in the model:

$$\mathcal{L}_{k,l}(i,j) = \mathbf{W}_{k,l}(i,j) + \varepsilon(i,j).$$

D. Sparse filter design

It is the task of the last prediction stage to find which regressors are useful in a prediction template convolving the merged warped image $\mathbf{W}_{k,l}$ with a sparse predictor. We have used the candidate template as a square window with dimensions 7×7 pixels. The design procedure is the same as in [8], which was operating on the regions having the same disparity. The only big difference is that in [8] the convolution was used over several neighbour views, since they were already well aligned, while here we only perform the final convolution on $\mathbf{W}_{k,l}$ by a specific sparse predictor.

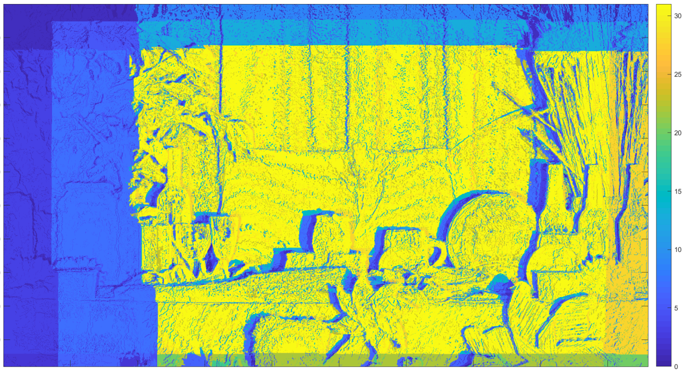

Fig. 3. Example of the 32 classes considered in view merging for view $(023, 004)$ in S_2.

V. Experimental Results

The original HDCA dataset for JPEG Pleno core experiments consists of four sets, denoted by S_2, S_6, S_9, S_{10} of 21×101 views at 2160×3840 resolution. In accordance to the JPEG Pleno core experiments, we consider only a subset of 11×33 views, where each view is cropped center-wise to Full HD size of 1080×1920. Bit rates reported are total code lengths divided by the amount of pixels in the subset. For the experiments the new codec, labeled here as TUT, was used for encoding the HDCA data set S_2.

A. Large set of references in checkerboard arrangement

In this experiment we have used a checkerboard arrangement of the reference views (starting from the top-left corner) with disparities quantized at $n_Q = 511$ levels. This configuration provides good results at high bit rates. The color references were encoded by JPEG 2000 and the reciprocal depthmap was compressed by CERV. Since in [10] are provided only 5 reciprocal depthmaps, the missing disparities for the rest of the references were obtained by warping from the five CERV compressed reciprocal depthmaps (by using the block "Collect Warped Depthmap/Segmentation"). Only the baseline of Subsection IVA was used in the codec (Subsections B, C, and D were not used). The results of $PSNR_{YUV}$ are shown in Fig. 4. A comparison with encoding all the views by JPEG 2000 is presented, showing a big improvement by the TUT method. This comparison is relevant for the use cases when one may want to use encoding tools only from the JPEG 2000 system, which is license free, and does not want to use more advanced video coding methodology, which require licensing. No comparison with existing light field compression methods can be shown since there are no public results in the literature for HDCA dataset.

B. Sparse set of references

In Fig. 5 we present results under same experimental conditions as above for the case of only 5 references (corners plus the center of the view array). We present results over the set S_2 for various quantization levels n_Q for disparity. Only the baseline of Subsections IVA was used in the codec

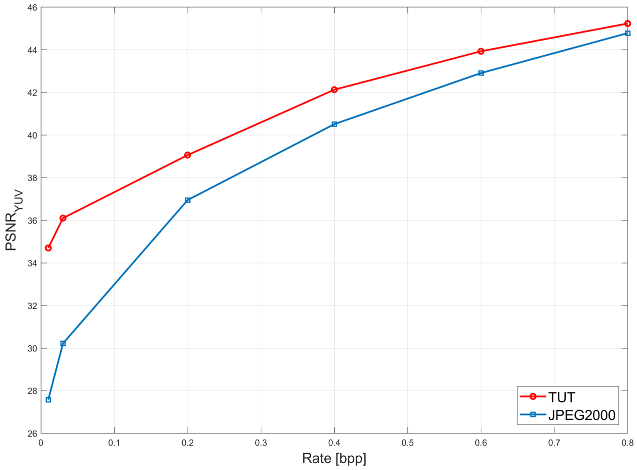

Fig. 4. Rate-distortion performance for a high-density set of references in checkerboard configuration for S_2.

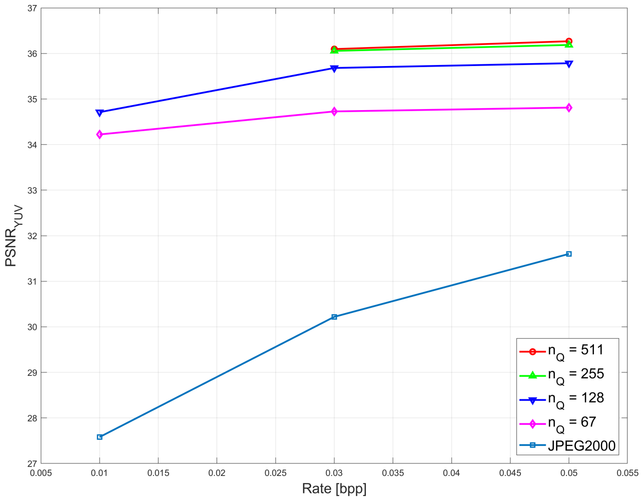

Fig. 5. Rate-distortion performance for a sparse set of only 5 references for S_2.

(Subsections B, C, and D were not used). Fig. 5 shows the improvement in performance as n_Q increases from 67 to 255.

C. Improvements by each prediction stage

To illustrate the performance of stages IVA, IVB, and IVC from Section IV we report rate distortion at each of these stages, reported in Table I. Adding the stage IVB improves by 0.6 dB and adding the stage IVC improves by more than 2.3 dB at the tested bit rates 0.03, 0.05 bits per pixel. Adding the stage IVD to the chain IVA+IVB+IVC did not improve significantly yet, most likely due to implementation issues.

VI. Conclusions

The new codec presented was shown to obtain favorable results for the encoding of HDCA data when compared to the baseline JPEG 2000. More experiments, under various experimental conditions, are needed for fine tuning the various elements of the scheme. The scheme has the potential to improve when a full rate-distortion procedure will be implemented, while promising results were shown here using intuitive selections of the parameters.

TABLE I
Comparison of different stages for prediction presented in Section IV, for $n_q = 128$

	PSNR	
Method	BITRATE = 0.03	BITRATE = 0.05
IVA	35.68	35.79
IVA+IVB	36.30	36.42
IVA+IVB+IVC	38.63	39.00

References

[1] M. Ziegler, R. op het Veld, J. Keinert, F. Zilly, "Acquisition System for Dense Lightfield of Large Scenes," in *2017 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, June 2017, pp. 1–4.

[2] ISO/IEC JTC 1/SC29/WG1 JPEG, "JPEG Pleno Call for Proposals on Light Field Coding," in *ISO/IEC JTC 1/SC29/WG1 JPEG, Doc. N74014*, Jan 2017.

[3] T. Ebrahimi, S. Foessel, F. Pereira, P. Schelkens, "JPEG Pleno: Toward an Efficient Representation of Visual Reality," *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, Oct 2016.

[4] I. Viola and T. Ebrahimi, "Quality Assessment of Compression Solutions for ICIP 2017 Grand Challenge on Light Field Image Coding," in *9th Workshop on Hot Topics in 3D Multimedia (Hot3D)*, July 2018.

[5] S. Zhao and Z. Chen, "Light field image coding via linear approximation prior," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017, pp. 4562–4566.

[6] P. Helin, P. Astola, B. Rao, I. Tabus, "Minimum Description Length Sparse Modeling and Region Merging for Lossless Plenoptic Image Compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1146–1161, Oct 2017.

[7] P. Helin, P. Astola, B. Rao, I. Tabus, "Sparse Modelling and Predictive Coding of Subaperture Images for Lossless Plenoptic Image Compression," in *2016 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, July 2016, pp. 1–4.

[8] I. Tabus, P. Helin, P. Astola, "Lossy Compression of Lenslet Images from Plenoptic Cameras Combining Sparse Predictive Coding and JPEG 2000," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017, pp. 4567–4571.

[9] I. Tabus, I. Schiopu, J. Astola, "Context Coding of Depth Map Images Under the Piecewise-Constant Image Model Representation," *IEEE Transactions on Image Processing*, vol. 22, no. 11, pp. 4195–4210, Nov 2013.

[10] A. Naman, R. Mathew, D. Ruefenacht, D. Taubman, "UNSW Depth Reciprocal Fields for the HDCA Dataset," in *ISO/IEC JTC 1/SC29/WG1 JPEG, Doc. N78000*, Dec 2017.

[11] D. Ruefenacht, A. Naman, R. Mathew, D. Taubman, "Inter-View Compression Framework with Base Anchored Modeling and Inference," in *ISO/IEC JTC 1/SC29/WG1 JPEG, Doc. N78051*, Jan 2018.

[12] ISO/IEC JTC 1/SC29/WG1 JPEG, "Core Experiments Set 2 for JPEG Pleno," in *ISO/IEC JTC 1/SC29/WG1 JPEG, Doc. N77016*, Oct 2017.

[13] I. Tabus, P. Astola, "Sparse Prediction for Compression of Stereo Color Images Conditional on Constant Disparity Patches," in *2014 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, July 2014, pp. 1–4.