Accelerated Shearlet-Domain Light Field Reconstruction

Suren Vagharshakyan, *Member, IEEE,* Robert Bregovic, *Member, IEEE,* and Atanas Gotchev, *Member, IEEE*

***Abstract*—We consider the problem of reconstructing densely sampled light field (DSLF) from sparse camera views. In our previous work, the DSLF has been reconstructed by processing epipolar-plane images (EPI) employing sparse regularization in shearlet transform domain. With the aim to avoid redundant processing and reduce the overall reconstruction time, in this article we propose algorithm modifications in three directions. First, we modify the basic algorithm by offering a faster and more stable iterative procedure. Second, we elaborate on the proper use of color redundancy by studying the effect of reconstruction of an average intensity channel and its use as a guiding mode for colorizing the three color channels. Third, we explore similarities between EPIs by their grouping and joint processing or by effective decorrelation to get an initial estimate for the basic iterative procedure. We are specifically interested in GPU-based computations allowing an efficient implementation of the shearlet transform. We quantify our three main approaches to accelerated processing over a wide collection of horizontal- as well as full-parallax datasets.**

***Index Terms*—Light field reconstruction, Graphics processing units, Densely sampled light field**

I. Introduction

3D visual scenes are completely represented by the light field they emanate. Given that the light field is a continuous function, its capture and consequent reconstruction is an important task, especially for visualization applications, which require multiple perspective views (e.g. super multiview displays [1]) or dense parallax (e.g. digitally printed holograms [2]). Many other light field image processing applications, such as depth estimation, compression, synthetic aperture imaging would benefit from accurately reconstructed light field [3]. A typical way of capturing light fields from real world scenes is to use a set of identical parallel cameras which are uniformly positioned on a plane. In order to support continuous parallax, such capturing setup requires that the cameras are densely positioned [4]. To overcome the demand for synchronously controlled high amount of cameras, the approach is to use a coarse set of cameras and to devise a consecutive light field reconstruction method, which can deliver densely sampled views from the coarse set of captured ones.

The approaches for reconstructing dense intermediate views from a given sparse set of views can be categorized into two categories. First, those are methods aimed at extracting geometry information about the scene in the of form of high quality depth maps collocated with the given input images, which can be used for depth-based view rendering [5] or unstructured lumigraph rendering [6]. Such methods utilize correspondences between images, found by block matching [7], and employ some global optimization of cost functions usually formed by data and smoothness terms [8], [9]. Apart from having problems with occlusions, when using sparse views, these methods result in over-smoothed depth estimates, and for finding finest details aligned with object boundaries they still need relatively densely positioned cameras [10]. Second category includes methods operating directly on the light field and aimed at employing some sparsity priors for this data. For example, the work [11] exploits sparse representation of full parallax 4D light field in continuous Fourier domain using a small number of 1D viewpoint trajectories.

For both categories, reconstructing a dense set of images is a computationally demanding problem. Global optimization methods aimed at obtaining multiple high quality depth maps do not scale well with the number of images and their resolution. The method in [10] targeted processing of 50 views with overall of 21-megapixels and accounted of about 50 mins of processing time. The runtime of the method in [11] ranged from 2 to 3 hours using a cluster of machines.

Previously, we have proposed a method for light field reconstruction which utilizes sparsification in shearlet transform domain [12], [13]. The method reconstructs DSLF from an undersampled light field captured by a small number of wide-baseline cameras. It demonstrated superior performance while compared with Motion Picture Experts Group's depth estimation reference software (DERS) [14] and view synthesis reference software (VSRS) [15], and with the state of the art in depth-from-stereo scene geometry reconstruction [16]. The method handles both horizontal and full-parallax capture settings and is highly successful when reconstructing non-Lambertian scenes formed by semi-transparent objects [13]. In this article, we further develop the method by proposing computational acceleration approaches based on inherent similarities in the assumed data representation and further algorithm tuning. Our aim is to decrease the necessary computational time while keeping or even increasing the reconstruction quality for a large set of test data.

The article is structured as follows: the light field parameterization and a summary of light field reconstruction algorithm from [13] are presented in Section II. Different acceleration approaches are proposed in Section III. Computing and evaluation setup, algorithm implementation, experimental results and discussions are presented in Section IV.

II. Reconstruction of Densely Sampled Light Field

4D light field is parameterized by the so-called two-plane parameterization $L(u, v, s, t)$ (Fig. 1), where (s, t) and (u, v) cor-

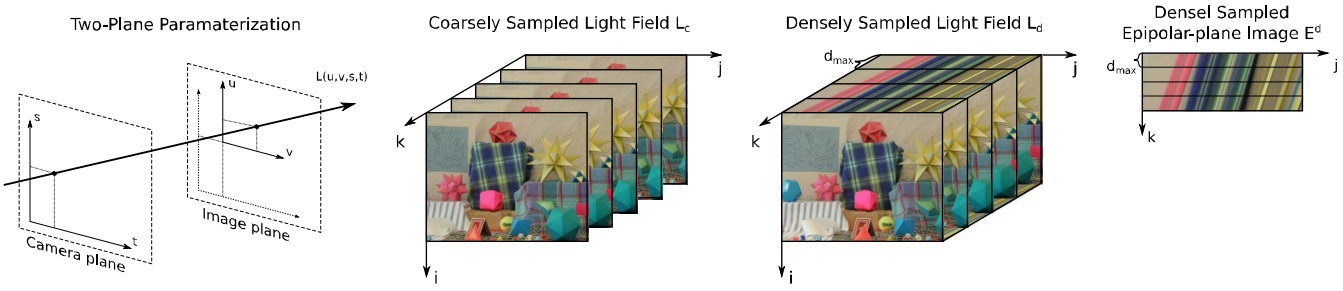


Fig. 1. Light field two-plane parameterization and corresponding discrete coarsely and densely sampled light field parameterizations.

respond to the camera plane and the image plane respectively [17]. This parameterization allows to conveniently describe and denote both the set of images (views) captured by a multi-camera setup and the required dense set of images fully representing the 3D scene of interest. For the sake of simpler notations and easier illustrations, hereafter we consider the case of horizontal parallax and explain the generalisations to full parallax when needed. Fixing the camera motion to horizontal direction only implies that the parameter $s = s_0$ corresponding to the vertical parallax can be omitted.

$$L(u, v, t) = L(u, v, s_0, t).$$

We aim at reconstructing continuous light field $L(u, s, t)$ from sampled views of 3D scenes. For such scenes and corresponding light fields, we define the densely sampled light field (DSLF), denoted by $L^d(i, j, k)$, as the light field having a maximal disparity between adjacent views less than 1 pixel. The k-th captured image $I^d_k(i, j) = L^d(i, j, k)$ corresponds to an image of the L continuous light field sampled at $t_k = k\Delta t_d$. The necessary parameter Δt_d for capturing DSLF can be calculated based on camera intrinsic parameters and specifying the minimum depth of the scene from the camera (capturing) plane. The continuous light field can be reconstructed from DSLF by linear interpolation [4]. Therefore, DSLF is a convenient representation and is in the core of many LF-based algorithms, such as refocusing, free view-point rendering and smooth-parallax visualization [3]. However, a direct capture of views providing one-pixel disparity is impractical.

In our previous work we have presented a method for DSLF reconstruction from coarsely sampled views [12]. A coarsely sampled light field is assumed to be a decimated version of DSLF, where the decimation factor is denoted by d_{max}

$$L^c(i, j, k) = L^d(i, j, d_{max}k).$$

Subsequently, the maximal disparity between adjacent coarsely sampled views $I^c_k(i, j) = L^c(i, j, k)$ is no more than d_{max}. In our previous work we presented an iterative algorithm which reconstructs L^d from L^c for $d_{max} \leq 32$. The algorithm works in EPI domain. More specifically, DSLF L^d is reconstructed by reconstructing every densely sampled epipolar-plane image (DSEPI) defined as

$$E^d_i(k, j) = L^d(i, j, k)$$

from the given decimated samples E^c_i such that $E^c_i(k, j) = E^d_i(d_{max}k, j)$. An example of coarsely-sampled and densely-sampled light fields is given in Fig. 1. Note how rows with step size d_{max} form E^c out of E^d. The specific value of $d_{max} \leq 32$ is related with image resolution and has been selected for practical reasons. The method can be applied for higher disparity ranges too, however this would impose processing images with higher resolution, which in turn would significantly increase the required amount of memory [13]. Therefore, in this article we consider the limit case of $d_{max} \leq 32$.

Below, we summarize the algorithm for DSEPI reconstruction [13]. To simplify the notations, we denote the unknown DSEPI matrix by $f \in \mathbb{R}^{N\times N}$. The decimated EPI $g \in \mathbb{R}^{N\times N}$ has the same dimension and contains sensed values at each d_{max}-th row while the other rows are set to 0. The relation between the two EPIs is formalized by setting a binary measurement matrix $M \in \mathbb{R}^{N\times N}$ which has zero values elsewhere than $M(kd_{max}, j) = 1$. Then, $g = M \odot f$, where $\odot$ is element-wise matrix multiplication. The direct and inverse shearlet transforms are denoted by $S : \mathbb{R}^{N\times N} \to \mathbb{R}^{\eta\times N\times N}$ and $S^* : \mathbb{R}^{\eta\times N\times N} \to \mathbb{R}^{N\times N}$, respectively, where η is the number of all shears in all scales of the shearlet transform. More details about the shearlet transform construction can be found in [13]. The reconstruction of unknown rows of the matrix g is formulated under the prior condition for having sparse solution in the shearlet domain, i.e.

$$\min_{f\in\mathbb{R}^{N\times N}} \|S(f)\|_0, \text{ subject to } g = M \odot f$$

which can be efficiently solved through the following iterative thresholding algorithm [13]:

$$f_{n+1} = S^*(T_{\lambda_n}(S(f_n + \alpha_n(g - M \odot f_n)))), \quad (1)$$

where the acceleration parameter α_n is chosen as follows

$$\alpha_n = \frac{\|\beta_n\|_2^2}{\|M \odot S^*(\beta_n)\|_2^2}, \quad (2)$$
$$\beta_n = S_{\Gamma_n}(y - M \odot f_n), \Gamma_n = \text{supp}(f_n),$$

and $(T_\lambda f)(k) = \begin{cases} f(k), & |x(k)| \geq \lambda \\ 0, & |x(k)| < \lambda \end{cases}$ is a hard thresholding operator. The initial value can be set to $f_0 = S^*_0(S_0(g))$, where S_0 and S^*_0 are direct and inverse transform using only low-pass element in the shearlet transform. The thresholding parameter λ_n is set to decrease with the iteration number n. In our case we apply a linear decrease from λ_{max} to λ_{min}, for L iterations such as $n = 0, \ldots, L$.

It is important to mention that one has to set a few parameters while running the algorithm. These are the number

of iterations L which directly influences the computational time; the initial estimation f_0 which can reduce the necessary number of iterations; and the threshold range $[\lambda_{max}, \lambda_{min}]$.

The generalization to full parallax is straightforward by successively implementing the basic algorithm along the horizontal and vertical camera axes. A more computationally-efficient alternative, referred to as hierarchical reconstruction [13], implements the reconstructions in a specific order, aimed at reducing the maximum disparity between input views after each iteration, thus reducing the required number of shearlet transform scales and the related processing time.

III. Accelerated Processing

The method in Section II is applicable for any EPI. A preferable solution would use the same set of parameters for every EPI and run the reconstructions independently. One can select optimal thresholding parameters for the reconstruction algorithm as $[\lambda_{max}^{opt}, \lambda_{min}^{opt}]$ and fix a common number of iterations N for all EPIs. In this case, the computation time linearly depends on the number of EPIs and the fixed number of iterations. By distributing the required computations equally between multiple GPUs, one can achieve the fastest computational time for this independent processing. Further acceleration can be achieved by speeding up the iterative algorithm itself and by utilizing similarities between EPIs.

A. Faster convergence by double overrelaxation

In this section we propose a modification of the main iterative algorithm aimed at its faster convergence. As presented in (2), the convergence is controlled by the parameter α_n, which was originally designed to provide stability for varying content for the price of increased computations [13]. As a computationally less expensive alternative, here we propose another update mechanism, based on the so-called double overrelaxation (DORE), similar to the one presented in [18]. Assume the light field EPI matrices are reordered in column vector form and assume the parameter $\alpha_n = \alpha$ is fixed. The thresholding operation

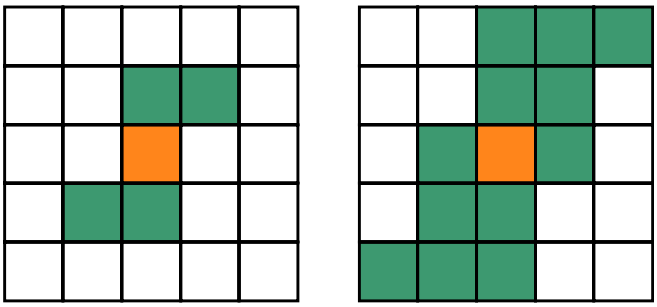
$$\hat{f}_n = S^*(T_{\lambda_n}(S(f_n + \alpha(g - M \odot f_n)))) \tag{3}$$

is followed by a two-step overrelaxation

$$\begin{aligned} \tilde{f}_n &= \hat{f}_n + \beta_1(\hat{f}_n - f_{n-1}) \\ \beta_1 &= \frac{(g - \hat{f}_n)^\top H(\hat{f}_n - f_{n-1})}{(\hat{f}_n - f_{n-1})^\top H(\hat{f}_n - f_{n-1})}, \end{aligned} \tag{4}$$

$$\begin{aligned} f_{n+1} &= \tilde{f}_n + \beta_2(\tilde{f}_n - f_{n-2}) \\ \beta_2 &= \frac{(g - \tilde{f}_n)^\top H(\tilde{f}_n - f_{n-2})}{(\tilde{f}_n - f_{n-2})^\top H(\tilde{f}_n - f_{n-2})}, \end{aligned} \tag{5}$$

where $H \in R^{N^2 \times N^2}$ is a diagonal matrix containing the elements of the measuring matrix M along its main diagonal. For additional stability we clamp the values of $\beta_1, \beta_2 \in [0, 1]$. The role of the double over-relaxation is to tackle potential instabilities by keeping the next iteration anchored to the previous iterations. Eq. (4) and (5) provide closed-form solutions for the respective line search problems $\beta_1 = \underset{\beta}{\operatorname{argmin}} ||H(g - (\hat{f}_n + \beta(\hat{f}_n - f_{n-1}))||^2$ and $\beta_2 = \underset{\beta}{\operatorname{argmin}} ||H(g - (\tilde{f}_n + \beta(\tilde{f}_n - f_{n-2}))||^2$. This leads to finding an optimal linear combination between consecutive solutions such that the error is minimized over the given samples defined by the matrix H.



(a) (b)

Fig. 2. (a) Proposed window (green) for modelling guidance with respect to reference pixel (orange). (b) Neighbourhood (green) for forming Laplacian matrix entry with respect to reference pixel (orange).

B. Color spaces and guided colorization

A trivial approach is to convert the RGB color channels into YUV colour space and process EPIs there, while expecting significantly less energy in the U and V colour channels. Specifically, we apply reversible color transform (RCT [19]) without any quantization of values, i.e.

$$\begin{cases} \text{Y} = (\text{R} + 2*\text{G} + \text{B})/4 \\ \text{U} = \text{B} - \text{G} \\ \text{V} = \text{R} - \text{G} \end{cases}.$$

Usually, the spatial information in U and V channels is highly redundant for natural images. Therefore, in the case of processing in YUV colour space with given N number of iterations, we reconstruct Y channel with N iterations and U, V channels with $N/2$ iterations. Compared with the reconstruction in RGB colour space, the overall number of iterations is reduced from $3N$ to $2N$.

Furthermore, we investigate the possibility of applying the fully reconstructed Y-channel EPI as a guide in reconstructing R, G and B color channels from their decimated EPI versions. This type of problem can be solved by methods previously developed for image colorization [20], [21]. It has been also shown that colorization can be considered as a particular case of the more general problem of alpha matting [22]. Specifically, we adopt the so called closed-form alpha matting algorithm proposed in [23] and modify it for the purpose of reconstructing color EPIs.

Following the notations in Section II, we denote the targeted EPI color channel and its decimated version by f and g respectively. Let us denote also the reconstructed Y-channel EPI by E. Then, the targeted color channel pixels f_i are modelled as a linear function of the known (i.e. guiding) image pixels E_i, within a small window w

$$f_i \approx aE_i + b, \forall i \in w.$$

For natural images, typically, the small window has been assumed to be a square window of 3×3 pixels around the reference pixel. For the case of EPI, we propose to use a different window to leverage the directional information presented in EPI. We show the proposed shape in Fig. 2(a).

The cost function minimization problem can be formulated as follows

$$J(f, a, b) = \sum_{j=1}^{N^2} \left(\sum_{i \in w_j} (f_i - a_j E_i - b_j)^2 + \varepsilon a_j^2 \right),$$

where the regularisation term εa_j^2 is added for numerical stability. In [23], it has bee shown that an equivalent minimization problem can be formulated using *matting Laplacian* matrix Λ, which removes the need to identify a and b

$$J(f) = \min_{a,b} J(f, a, b) \sim J(f) = f^T \Lambda f,$$

where the entries of the matrix $\Lambda \in R^{N^2 \times N^2}$ are calculated as follows

$$\Lambda(i, j) = \sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{\aleph_k} \left(1 + \frac{1}{\frac{\varepsilon}{\aleph_k} + \sigma_k^2} (E_i - \mu_k)(E_j - \mu_k) \right) \right).$$

In the above equation, δ_{ij} denotes the Kronecker delta, and $\aleph_k$, μ_k, σ_k^2 denote the cardinality, the mean and the variance of the window w_k respectively. For the entry $L(i, j)$, the summation is done over all windows w_k which contain pixels with indices i and j. For a reference pixel at position i and for our choice of window shape, the pixel positions j are shown in Fig. 2(b). Given the true colors at the decimated EPI g, the problem is reformulated as

$$\text{minimize } f^T \Lambda f, \text{ s.t. } Hf = g,$$

where H is the diagonally-arranged measurement matrix M. The so-formulated problem is solved using the conjugated gradient method.

C. Group Processing of Similar EPIs

Previously, we have presented an attempt to accelerate the basic algorithm utilizing similarities between EPIs [24]. The method suggested constructing a tree, which defines the order of processing depending on similarity between EPIs. In the constructed tree, each node corresponds to an EPI. The tree is constructed by comparing EPIs for their similarity in terms of l^2 norm and consecutively connecting the most similar pairs of EPIs. Iterating over all EPIs, one obtains a connected graph. Then, the processing is performed from top to bottom, and the EPI being processed uses the reconstructed EPI at its parent node as an initial estimate. Our hypothesis was that the reconstruction over the graph would allow for adaptively choosing the number of iterations for each EPI depending on the similarity to its initial estimate. This approach heavily depends on the threshold defining similarity between EPIs and setting the same reconstruction parameters for different datasets is problematic. Therefore, in this article we adopt a more systematic approach toward exploring the EPI similarities, which would allow an easier tuning of algorithm parameters. First, we consider grouping of similar EPIs, done by comparing l^2 distances between EPIs against a predefined threshold t_s. Having the EPIs organized in groups, we fully reconstruct the average EPI over each group and use it as a guidance map to reconstruct the other EPIs in the group by the approach proposed in Subsection III-B.

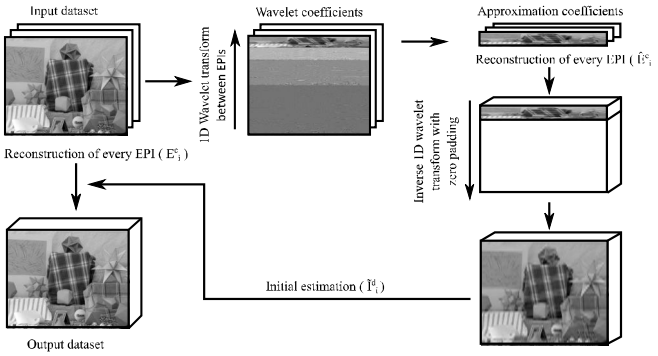


Fig. 3. Reconstruction flowchart using wavelet transform approximation coefficient as an initial estimation.

D. Initialization by Wavelet Transform

The redundancy between EPIs can be regarded as redundancy in the vertical direction in the given multi-perspective images. Instead of local grouping of similar EPIs as in Subsection III-C, we consider the alternative of decorrelating the vertical image lines by a fixed transform, e.g. a wavelet transform. Namely, a wavelet transform is performed on $E_i^c(\cdot, \cdot)$ along the i axis which is equivalent to performing 1D wavelet transform vertically on every input image $I_k^c(i, \cdot)$ along i axis. By performing L level of 1D wavelet transform between EPIs $E_i^c, i = 1, ..., p$, we expect to split them into EPIs with small-magnitude detail coefficients and $\tilde{E}_i^c, i = 1, ..., p/2^L$ EPIs with higher-magnitude approximation coefficients. The approximation coefficients gather most of the information of the original set of EPIs. The reconstruction is then applied directly on EPIs formed by wavelet transform approximation coefficients. The obtained set of densely sampled EPIs $\tilde{E}_i^d, i = 1, ..., p/2^L$ contain a good amount of directional structures (more global ones), however, they still require further processing to obtain desirable quality of reconstruction (add details from the original set of EPIs). Therefore, the inverse wavelet transform can be applied on the reconstructed EPIs of the approximation coefficients with an appropriate padding with zeros corresponding to detail coefficients. The obtained set of EPIs $\tilde{I}_i^d, i = 1, ..., p$ is used as an initial estimate for reconstruction of the original input E_i^c EPIs by performing additional processing by the modified basic algorithm. The processing times for the two steps can be set independently. The flowchart of the approach is shown in Fig. 3.

IV. Experimental Evaluation

A. Algorithm Implementation

We have implemented the core reconstruction algorithms, on a GPU using CUDA Toolkit [25]. Since the shearlet transform is a translation invariant transform, it can be efficiently

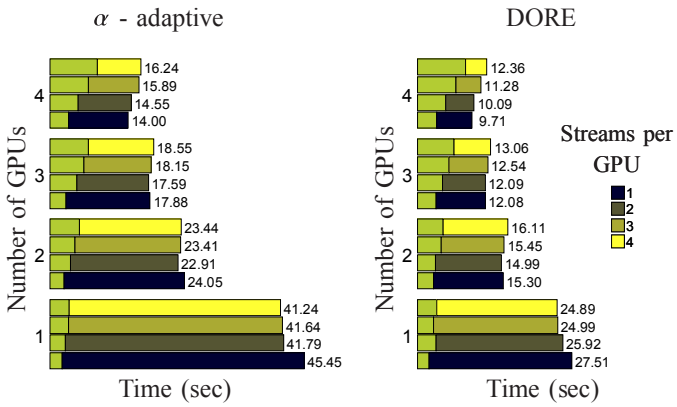


Fig. 4. Computational time required to perform 50 iterations of α-adaptive algorithm (left) and $DORE$ (right) on 100 EPIs using different parallelization between GPUs. Light green color represents the necessary time for initialization of the algorithm. For resolution 256×512 using more than one stream per GPU doesn't provide acceleration.

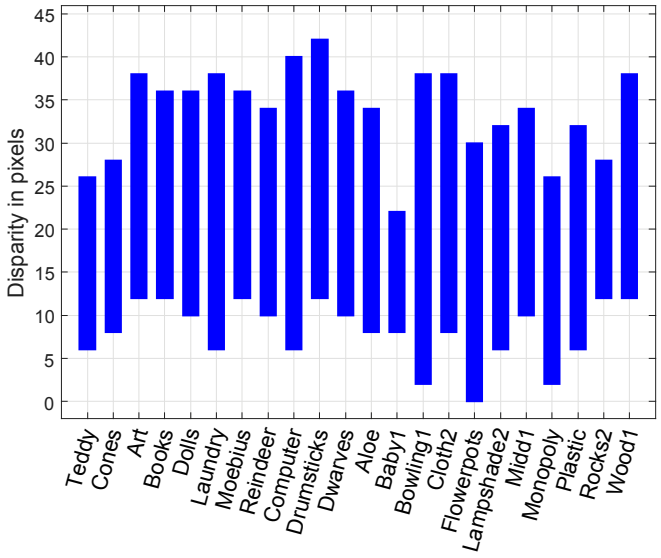


Fig. 5. Illustration of the disparity ranges $[d_{min}, d_{max}]$ between adjacent views for input dataset used in this paper (obtained from ground true disparity maps).

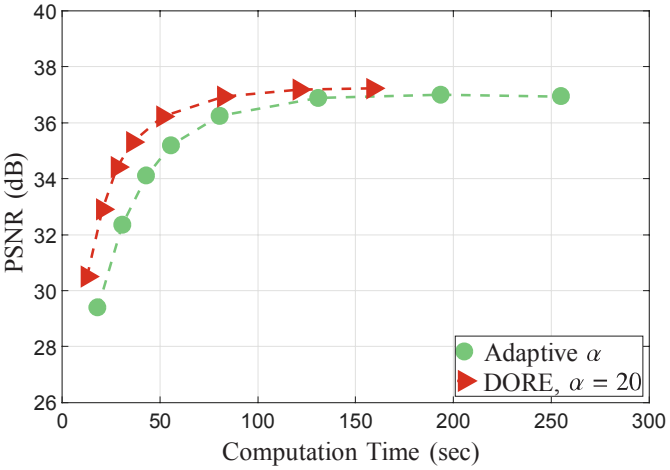


Fig. 6. Comparison of average performance between basic algorithm with adaptive selection of the parameter α and DORE with fixed α = 20 for the horizontal-parallax datasets.

computed using the Fast Fourier Transform (FFT). In our case, we used the cuFFT library to get an FFT implementation on the GPU [26]. For generating our experimental results we have used a system consisting of four Nvidia GeForce GTX Titan X GPUs. The computational time for reconstructing an EPI mainly depends on the number of overall iterations that have to be performed. In order to achieve fastest computation for a given set of input EPIs with a corresponding number of iterations, the set has been distributed between GPUs such that the overall number of iterations that has to be performed are approximately equal for each GPUs. On the level of one GPU, the whole iterative processing is performed independently from other GPUs. Depending on the size of the processed EPI, we get different occupation of GPU kernels at a time. We consider EPIs with the size of 256×512 processed with the shearlet transform at 5 scales using algorithm presented in Sections II, III-A. In both cases, 100 EPIs are processed. The computational times are presented in Fig. 4. Note that reconstructing in the case of 256×512 with S^5 transform, only one process per GPU is sufficient for both algorithms, while DORE is significantly faster.

B. Evaluation

In our comparative tests, we have used datasets presented in [27], [28] for horizontal parallax and in [29] for full parallax datasets. In overall, 22 horizontal-parallax datasets and two full-parallax datasets of various depth and spatial content have been used. In all experiments, the input data is formed by every second view of the test dataset. The other views form the reference. The algorithm performance has been evaluated by comparing the difference between the reconstructed and the reference views in terms of PSNR (dB). $J = 5$ scale levels have been considered for the shearlet transform (S^5) which corresponds to an intermediate view reconstruction, where the maximum disparity between adjacent views is in the range of $[0, 32]$ pixels. The exact disparity ranges of each scene as obtained from the ground truth disparity maps are shown in Fig. 5. In order to shift the available (existing) disparity ranges to $[0, d_{max} - d_{min}]$, for each input dataset we perform first a horizontal shearing by $-d_{min}$ on all EPIs. After reconstructing $d_{max} - d_{min} - 1$ intermediate views, a shearing by $d_{min}/(d_{max} - d_{min})$ is applied to return the imagery to the original disparity range. In general, we evaluate algorithms presented in Section III for different number of iterations in order to compare trend of convergence speed of different algorithms in average for all datasets.

In our first comparative test, we present the average reconstruction quality for the algorithm modification based on DORE with $\alpha = 20$ and the original α adaptive algorithm. The comparison is done for $5, 10, 15, 20, 30, 50, 75, 100$ iterations per EPI. The results for the horizontal parallax datasets are shown in Fig. 6, while the reconstruction results for the full-parallax datasets are shown in Fig. 7. The DORE algorithm provides faster convergence for all datasets and results in

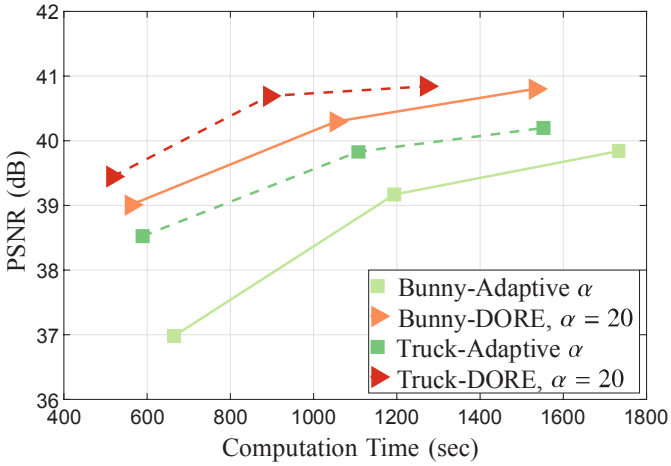


Fig. 7. Comparison of performance between basic algorithm with adaptive selection of the parameter α and DORE with fixed $\alpha = 20$ for the full-parallax datasets.

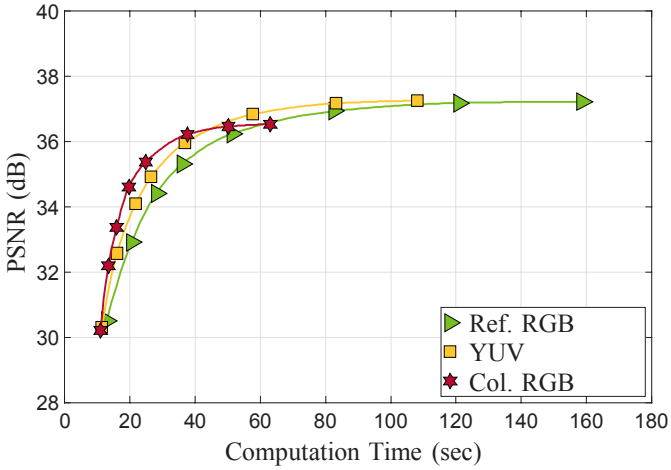


Fig. 8. Comparison between reference algorithm (RGB), YUV, and colorization of RGB for the horizontal-parallax datasets.

better quality enjoying also faster processing. In all subsequent experiments, we use the DORE algorithm with $\alpha = 20$, referring to it as the reference algorithm.

Next, we aim at quantifying the performance of the colorization algorithm. For the horizontal-parallax datasets, we present the trend in reconstruction quality for different number of iterations, see Fig. 8. The reference algorithm reconstructs the three color channels, R, G, and B in an equal number of iterations, while in YUV, priority is given to the Y channel, which is processed twice longer than the U and V channels. In the case of colorization, an average intensity channel is formed as $Y = (1/3) * (R + G + B)$ and fully reconstructed in varying number of iterations, then each of the color channels is reconstructed by colorization using the reconstructed Y channel as a guidance. As can be seen in the figure, the prioritized processing brings some improvement over the reference algorithm and the algorithm based on colorization is significantly faster as it processes a single channel only. All three algorithms saturate in performance, which means that after some number of iterations, no quality improvement is

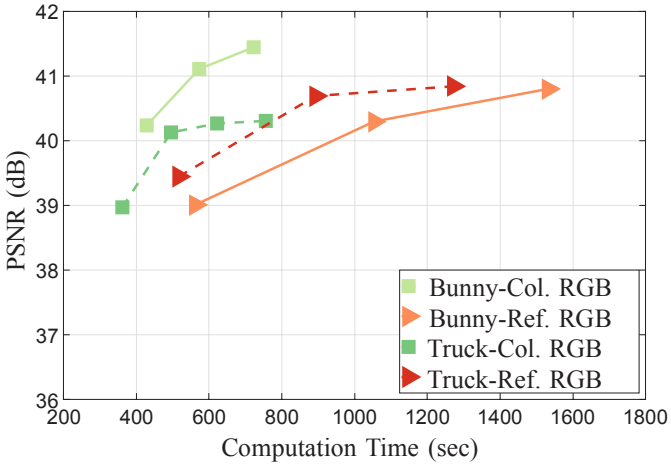


Fig. 9. Comparison between reference algorithm (RGB), and colorization of RGB for the full-parallax datasets.

achieved. The colorization algorithm saturates at lower level, which indicates that the structural differences in the three color channels have not been fully reconstructed in the averaged intensity channel. The reached values after 100 iterations for each of the three algorithms are the following: $RGB-37.22dB$, $YUV-37.24dB$, $Col.RGB-36.22dB$. These results suggest that the colorization algorithm is preferable in case of limited computation time, since it converges faster, while for the case of better computing resources, the best quality is achieved by the YUV color space processing.

Fig. 9 presents the results for the full-parallax datasets. For the *Bunny* dataset, the colorization shows a significant improvement both in terms of time and quality, while for the *Truck* dataset, the results are in agreement with the average result over the horizontal-parallax datasets.

Fig. 8 presents the average results over the whole group of test scenes. The result in terms of rate of convergence vary substantially for the individual test scenes. In order to further analyse the algorithm based on colorization, we look at the saturation points for the reference algorithm and the colorization algorithm for each individual dataset. The two algorithms are run for increasing number of iterations and saturation points are estimated at the iteration where further improvement is negligible. Denote by $E(k), T(k), k = 1, \ldots$ the quality level (e.g. PSNR), and the corresponding time in seconds for the running iteration k. We define $k_{\max} = \arg\max_k E(k)$ and define the saturation point as $k_{sat} = \arg\max_k \left(\frac{E(k+1)-E(k)}{E(k_{\max})} < 0.0015 \right)$. The idea is illustrated in Fig. 10 for the Teddy dataset, where the saturation points are given by the circles around the corresponding iterations.

Having found two saturation points per dataset, one for the reference and one for the colorization algorithm, one can compare them in terms of quality variation (ISNR) and time acceleration ratio. In other words, we compare the best achievable quality per dataset for the two algorithms versus the time acceleration ratio it brings. Fig. 11 presents this comparison over the horizontal-parallax datasets. In the figure, each dot represents one dataset, the $x-$ axis represents the relative time acceleration achieved by the colorization algorithm versus the

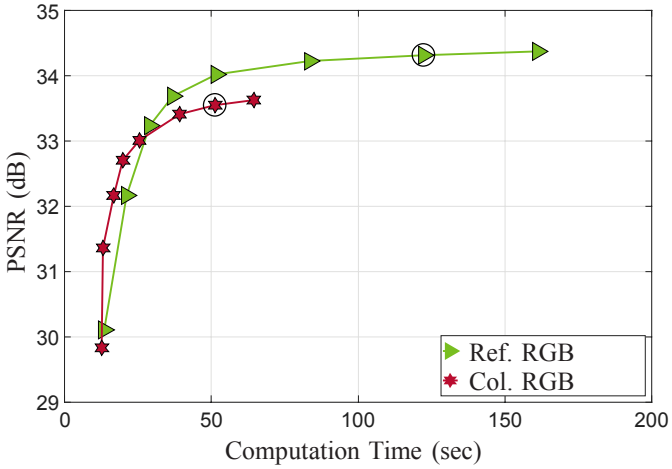


Fig. 10. Saturation points for the reference and colorization algorithms for the Teddy dataset. The obtained saturation points are illustrated by black circles.

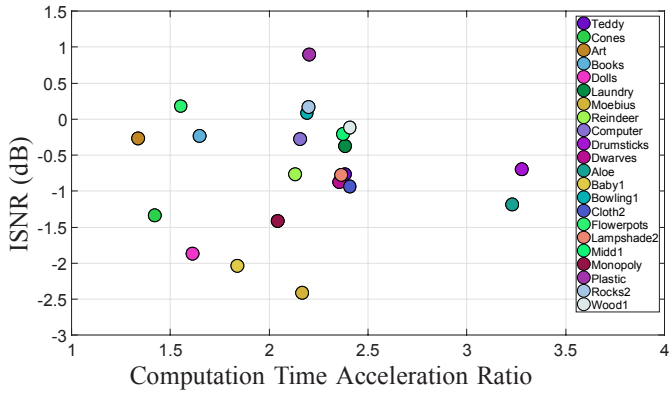


Fig. 11. Comparison of saturation points of the colorization and reference algorithms for the horizontal-parallax test datasets.

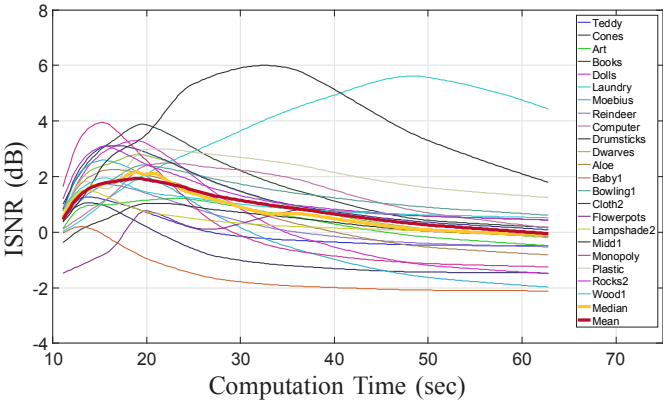


Fig. 12. ISNR of colorization versus reference algorithm for individual sequences.

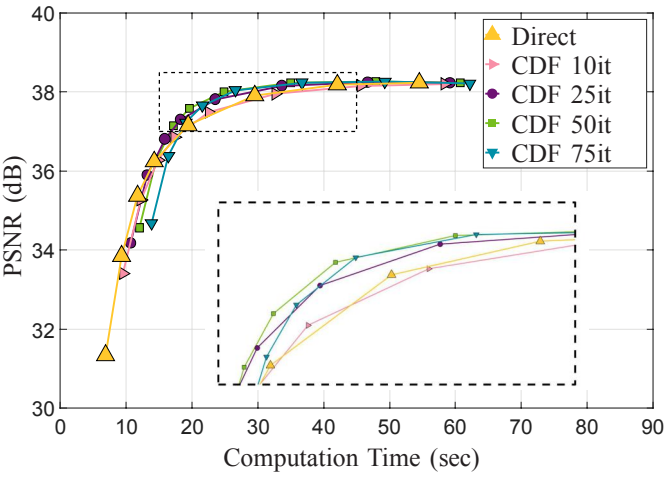


Fig. 13. Comparison of the reconstruction trends depending on number of iterations used for obtaining initial estimation in method utilizing wavelet transform. In the legend of the figure presented corresponding number of iterations used for processing initial estimations.

reference one, and the $y-$ axis represents the improvement in the signal-to-noise ratio ISNR (dB). As seen in the figure, there are a few sets, where the acceleration in time comes together with improved quality, while for the majority of datasets, the acceleration is achieved for the price of reduced quality.

Another way to illustrate the performance of the colorization algorithm is to show the ISNR in comparison to the reference algorithm for the same time, using interpolation between iteration points. Fig. 12 gathers the performance for each individual dataset, along with the mean and median values. For short processing times, colorization is to be preferred as most of the sequences show positive ISNR values. As the processing time gets longer, the values cluster around the zero ISNR line (as shown also by the mean and median curves), while there are a few sequences still enjoying better performance with the colorization method and other sequences showing worsening results. Apparently, among the former group these are datasets with relatively simpler color and depth distributions.

To simplify the next experiments, we limit the comparison of algorithms exploring the inter-EPI similarities and decorrelation to comparing the results for the Y channel only, assuming that the RGB color channels can be efficiently reconstructed by the Y channel.

For the wavelet transform based acceleration approach, presented in Section III-D, we perform $L = 3$ levels of the $CDF\ 9/7$ transform. Here, the issue is to find the best proportion between the processing time allocated for obtaining the initial estimate and the processing time allocated for refining the EPI reconstruction based on this initial estimate. In order to illustrate the trend in convergence, we perform experiments where the initial estimate is obtained by reconstructing the coarse wavelet coefficients with $10, 25, 50, 75$ iterations. The obtained initial estimate is then refined for the same number of iterations, as the direct (reference) algorithm. Fig. 13 depicts the trends. Naturally, the time needed for obtaining the initial estimates, shift the initial curve points to the right, e.g. the curve corresponding to 75 iterations allocated for getting the initial estimate is the rightmost in the figure. Then, the curves corresponding to wavelet-based initialization get better and saturate faster, with the case of 50 iterations for the initial estimate showing the best performance.

The algorithm based on grouping similar EPIs and processing them together as presented in Section III-C does not show consistent results for all datasets. This is to be attributed to the

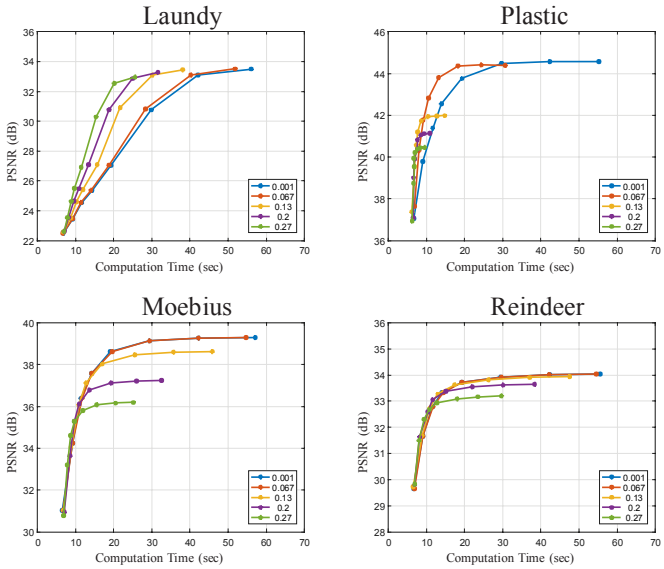


Fig. 14. Comparison of the reconstruction trends depending on different thresholding value in method utilizing groups formed based on similarity of the EPIs. In the legend of the figure presented corresponding thresholding values.

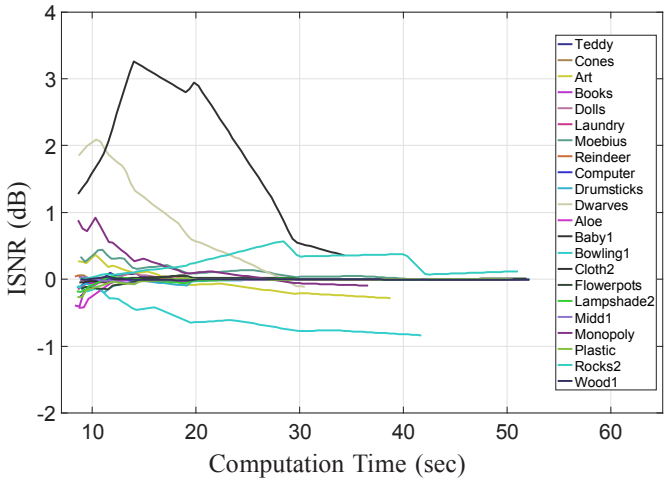


Fig. 15. Average reconstruction performance for the method utilizing grouping based on similarity between EPIs for different datasets for fixed thresholding value.

strong dependence on the threshold value, which determines which EPIs are sufficiently similar. In general, increasing the threshold value leads to increasing the size of formed groups and therefore decreases the computation time. However, the effect on quality is very much content-dependent. Results for several datasets with different threshold values are presented in Fig. 14. For the dataset *Laundry*, one can get significant acceleration, while e.g. for the dataset *Moebius*, the reconstruction quality is always inferior compared with the reference algorithm. Selecting one of the well-performing thresholds, i.e. the value of 0.067, one can get the performance for each individual dataset, as shown in Fig. 15.

Some examples of synthesized views are presented in Fig. 16, 17 with the corresponding quality in terms of PSNR. The DORE-based algorithm provides consistently better convergence compared to the original method [13]. The colorization-based algorithm achieves good reconstruction results when the Y channel manages to get the important structure of the scene (edge information) existing in R, G and B channels. For the particular scene *Laundry*, the algorithm based on wavelet transform provides better convergence compared to the reference RGB algorithm.

V. Conclusions

In this article, we have addressed the problem of accelerating the DSLF reconstruction algorithm, which originally uses sparse camera views and works on each EPI independently by employing regularized iterative reconstruction in shearlet transform domain. In order to speed up the algorithm, we proposed modifications in three categories. First, we aimed at improving the algorithm itself by using double relaxation in the iterative procedure. Second, we explored the similarities between color channels within the same EPI in the flavor of colorization based approaches. Third, we aimed at avoiding redundant processing and reducing the overall reconstruction time through exploiting similarities between EPIs. Furthermore, our implementation employed GPUs allowing for an efficient parallelized computation of the iterative procedure and the underlying shearlet transform.

We have generated experimental results on a wide set of test sequences and analyzed the performance of the considered approaches. The new reconstruction method based on double overrelaxation shows better convergence speed in comparison with the original algorithm. We favor the use of colorization as the approach catches well the color dependences in natural images. The benefit of using similarities between EPIs is very much content dependent. The wavelet based approach shows a marginal improvement in terms of convergence rate, which is still worth employing. As of the algorithm based on grouping of similar EPIs and group processing, it provides acceleration only for scenes where significant amount of EPIs are similar.

The modifications employ structured similarities within EPI and between EPIs were integrated within the DSLF reconstruction algorithm. However, they are perfectly applicable also in other LF image processing algorithms, where DSLF reconstruction is not the main goal. Such potential applications include LF depth estimation, compression, segmentation, and matting.

References

[1] N. S. Holliman, N. A. Dodgson, G. E. Favalora, and L. Pockett, "Three-dimensional displays: a review and applications analysis," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 362–371, 2011.

[2] B. Javidi and F. Okano, *Three-dimensional television, video, and display technologies*. Springer Science & Business Media, 2002.

[3] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," 2017.

[4] Z. Lin and H.-Y. Shum, "A geometric analysis of light field rendering," *Int'l J. of Computer Vision*, vol. 58, no. 2, pp. 121–138, 2004.

[5] C. Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," vol. 5291, 2004, pp. 93–104.

[6] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 425–432. [Online]. Available: http://doi.acm.org/10.1145/383259.383309

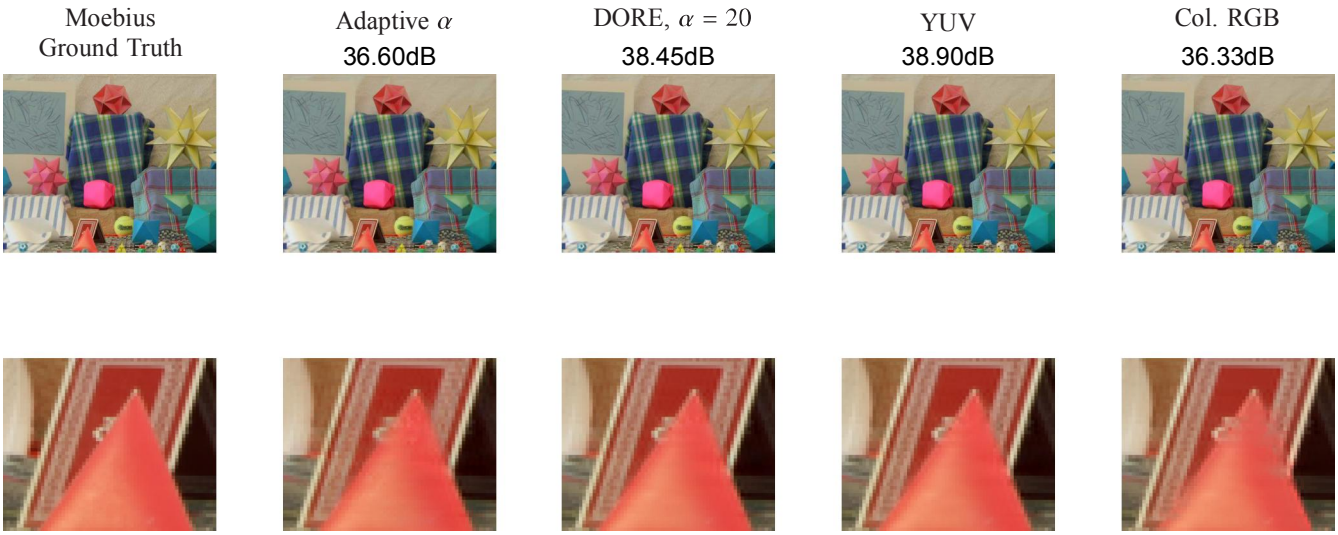


Fig. 16. Example of synthesized views for datasets $Moebius$ using different algorithms for approximately same computation time. Presented algorithms are used for accelerated color space reconstruction.

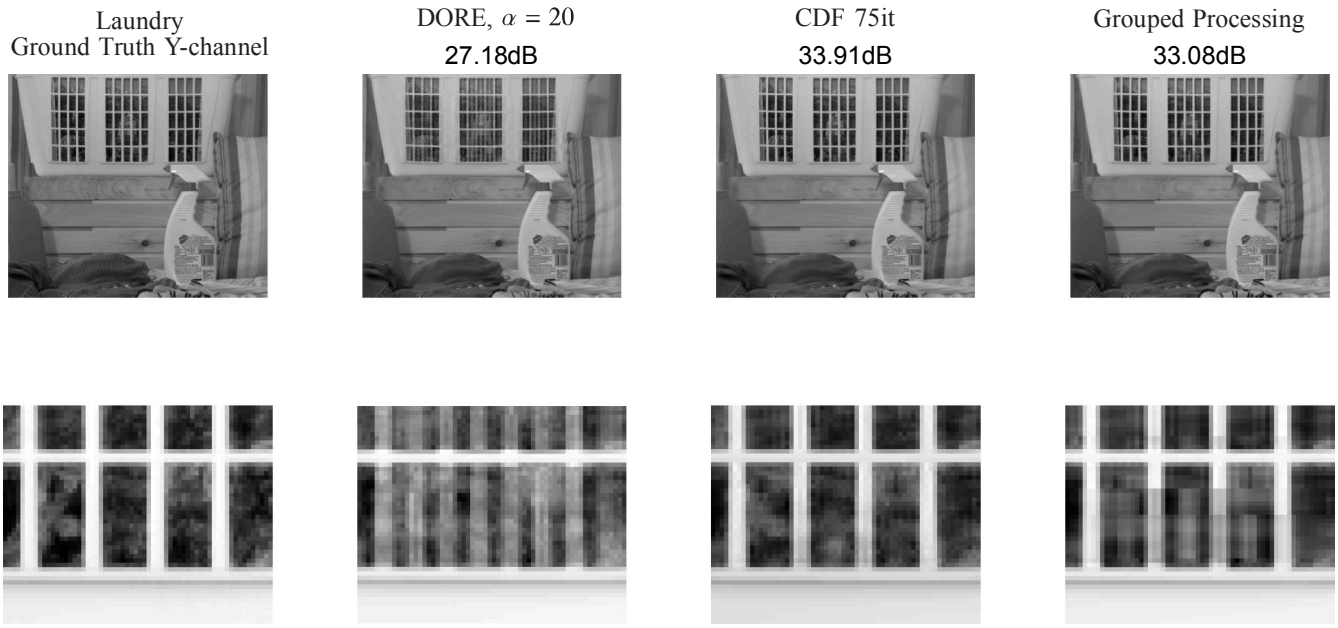


Fig. 17. Example of synthesized views for only Y-channel of dataset $Laundry$ using different decorrelation algorithms for approximately same computation time.

[7] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 519–528. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.19

[8] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, June 2008.

[9] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "Global solutions of variational models with convex regularization," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1122–1145, 2010.

[10] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 73:1–73:12, Jul. 2013.

[11] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous fourier domain," *ACM Trans. on Graphics (TOG)*, vol. 34, no. 1, p. 12, 2014.

[12] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Image based rendering technique via sparse representation in shearlet domain," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015, pp. 1379–1383.

[13] ——, "Light field reconstruction using shearlet transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[14] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, "Depth estimation reference software (ders) 5.0," *ISO/IEC JTC1/SC29/WG11 M*, vol. 16923, 2009.

[15] M. Tanimoto, T. Fujii, and K. Suzuki, "Reference software of depth estimation and view synthesis for ftv/3dv," *ISO/IEC JTC1/SC29/WG11 M*, vol. 15836, 2008.

[16] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[17] C.-K. Liang, Y.-C. Shih, and H. Chen, "Light field analysis for modeling image formation," *IEEE Trans. Image Processing*, vol. 20, no. 2, pp. 446–460, Feb 2011.
[18] K. Qiu and A. Dogandzic, "Double overrelaxation thresholding methods for sparse signal reconstruction," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, March 2010, pp. 1–6.
[19] Y. Chen and P. Hao, "Integer reversible transformation to make jpeg lossless," in *Signal Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on*, vol. 1. IEEE, 2004, pp. 835–838.
[20] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics (ToG)*, vol. 23, no. 3. ACM, 2004, pp. 689–694.
[21] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, ser. EGSR'07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 309–320. [Online]. Available: http://dx.doi.org/10.2312/EGWR/EGSR07/309-320
[22] J. Wang and M. F. Cohen, "Image and video matting: A survey," *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 2, pp. 97–175, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1561/0600000019
[23] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, Feb 2008.
[24] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Tree-structured algorithm for efficient shearlet-domain light field reconstruction," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 478–482.
[25] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," *Queue*, vol. 6, no. 2, pp. 40–53, 2008.
[26] V. Podlozhnyuk, "Fft-based 2d convolution," *NVIDIA white paper*, p. 32, 2007.
[27] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–195–I–202, June 2003.
[28] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, June 2007.
[29] V. Vaish and A. Adams, "The (new) stanford light field archive," *http://lightfield.stanford.edu*, 2008.

Suren Vagharshakyan Suren Vagharshakyan received the MSc in mathematics from Yerevan State University (2008). He is a PhD student at the Department of Signal Processing at Tampere University of Technology since 2013. His research interests are in the area of light field capture and reconstruction.

Robert Bregovic Robert BregoviÄĞ (Member, IEEE) received the Dipl. Ing. and MSc degrees in electrical engineering from University of Zagreb, Zagreb, Croatia, in 1994 and 1998, respectively, and the Dr. Sc. (Tech.) degree (with honors) in information technology from Tampere University of Technology, Tampere, Finland, in 2003. From 1994 to 1998, he was with the Department of Electronic Systems and Information Processing of the Faculty of Electrical Engineering and Computing, University of Zagreb. Since 1998, he is with the Laboratory of Signal Processing, Tampere University of Technology. His research interests include the design and implementation of digital filters and filterbanks, multirate signal processing, and topics related to acquisition, processing, modeling, and visualization of 3D content.

Atanas Gotchev Atanas Gotchev (Member, IEEE) received the M.Sc. degrees in radio and television engineering (1990) and applied mathematics (1992) and the Ph.D. degree in telecommunications (1996) from the Technical University of Sofia, and the D.Sc.(Tech.) degree in information technologies from the Tampere University of Technology (2003). He is a Professor at the Laboratory of Signal Processing and Director of the Centre for Immersive Visual Technologies at Tampere University of Technology. His research interests have been in sampling and interpolation theory, and spline and spectral methods with applications to multidimensional signal analysis. His recent work concentrates on algorithms for multisensor 3-D scene capture, transform-domain light-field reconstruction, and Fourier analysis of 3-D displays.