Real-time per-pixel focusing method for light field rendering

T. Chlubna¹ (✉), T. Milet¹, and P. Zemčík¹

© The Author(s) 2021.

Abstract Light field rendering is an image-based rendering method that does not use 3D models but only images of the scene as input to render new views. Light field approximation, represented as a set of images, suffers from so-called refocusing artifacts due to different depth values of the pixels in the scene. Without information about depths in the scene, proper focusing of the light field scene is limited to a single focusing distance. The correct focusing method is addressed in this work and a real-time solution is proposed for focusing of light field scenes, based on statistical analysis of the pixel values contributing to the final image. Unlike existing techniques, this method does not need precomputed or acquired depth information. Memory requirements and streaming bandwidth are reduced and real-time rendering is possible even for high resolution light field data, yielding visually satisfactory results. Experimental evaluation of the proposed method, implemented on a GPU, is presented in this paper.

Keywords image-based rendering; light field; plenoptic function; computational photography

1 Introduction

A 3D scene can be represented using a set of objects described by their material attributes, geometry, and applied transformations. Such a geometric representation of the scene can be rendered using various methods, such as rasterization or ray-tracing. Complexity of the scene, however, considerably affects the time taken by the rendering process. Image-based rendering is an alternative way of producing

new views of the scene, where, instead of geometric representation, visual information about the scene is used. This representation usually consists of a set of images of the scene taken from different positions and angles. Performance of such image-based rendering methods does not depend on the scene's content.

The field of light in a scene can be ideally described by a function representing light information for each point in space and each direction through this point. The scene then can be visually reconstructed from an arbitrary camera position and orientation. Computationally, such a continuous function is impossible to represent. Therefore, a discrete structure that consists of images of the scene is usually used as a so-called 4D light field approximation; see Fig. 1. The input images sample the scene from determined viewing angles, which provides as much visual information about the scene as possible. The more images are available, the better quality the final rendering can achieve. Storing more images, however, increases memory requirements. The goal of light field rendering methods is to use only a sparse set of image samples while achieving high visual quality of the rendered result. In practice, light fields can be viewed as an extension of classic photography, allowing the user to focus on different parts of the scene or even change the camera position in post-processing.

Lack of information about the 3D geometry of the scene leads to problems in novel view image reconstruction. Our work focuses on eliminating so-called out-of-focus areas in light fields without additional information about depths or 3D models of the scene. When using the approximate light field from a set of images, pixel values that are combined together in the final rendering have to be taken from the same spot in the scene. This spatial information for each pixel has to be estimated and

¹ Faculty of Information Technology, Brno University of Technology, 612 00 Brno, Czech Republic. E-mail: T. Chlubna, ichlubna@fit.vut.cz (✉); T. Milet, imilet@fit.vut.cz; P. Zemčík, zemcik@fit.vut.cz.

Manuscript received: 2020-11-10; accepted: 2021-01-12

Fig. 1 Scene captured by a grid of cameras. A light field approximation consisting of the images from this grid can be used to reconstruct a novel view from any camera position outside the bounding volume of the scene. Three cameras are highlighted with rays coming through a pixel with the same coordinates on the viewing plane, each providing light information about a different part of the scene.

used to achieve the correct focusing for the final image, as shown in Fig. 2. Our proposed method is based on statistical analysis of pixel values that are eventually combined into one pixel color in the resulting image; we use a shift-sum algorithm [1]. The method iterates over a range of focusing distances

and stores the best distance for each pixel from a focus map. The best distance is chosen according to the minimal variance of those pixels contributing to the interpolation process. A weighted shift-sum algorithm is used for interpolation of the final image and for pixel analysis. A novel view is synthesized using the generated focus map. With this method, visually acceptable all-in-focus light field scenes can be rendered from the input set of images without further knowledge of the original scene.

2 Related work

Light transport in space can be described by a 7D plenoptic function $L = P(x, y, z, \theta, \varphi, t, \lambda)$ [2]. In terms of geometric optics, this function returns the light intensity (L) of a ray incoming from direction (θ, φ) at a point (x, y, z) in 3D space. This value can change over time (t) and vary for each wavelength (λ). In practical usage, this function can be approximated by a 4D representation which is commonly referred as a light field [3]. Let us assume that a scene is located between two parallel planes with a virtual camera outside them. Rays coming from its center of projection intersect these two planes, producing one intersection point per plane. The intersection coordinates with the camera st plane are then used to decide which images from the input are to be used for the final pixel interpolation and the coordinates from the image uv plane are used to determine the

Fig. 2 Left: Comparison of fully focused light field image from the proposed method with a light field focused at a single distance. Right: From the input set of images taken by the camera grid, a new synthetic view of the scene is generated; every location in the scene is focused as if captured by a pinhole camera. The proposed method performs real-time per-pixel focusing, solving the task of light field focusing when a 3D model of the scene is unavailable. Focusing distance values for each pixel are estimated and stored in a focus map which resembles a disparity or depth map of the scene. This map is used to achieve correct focusing of each pixel.

correct pixels in the given images. Input images are typically captured in a regular grid that is mapped onto the camera plane in such a way that the location of the image in the grid corresponds to the location on the camera plane. The chosen image, according to the intersection coordinates, is then mapped on the image plane. The position of the image plane affects the focusing distance of the light field. Objects in the scene that are located at the focusing distance are in focus while the rest of the scene is blurred. To achieve a sharp image with all parts of the scene in focus, intersection points in the image plane need to be corrected, using the depth of the scene. A simplified geometry of the scene can be used [4] as a scene surface approximation to replace the planar image plane. Depth maps can also be used for ray intersection correction and to enhance photographic effects such as depth of field [5]. The generalized concept of the two-plane parameterization can be extended to support various light field shapes using two spheres, points, and directions, etc. [6]. Instead of plane intersection calculations, a shift-sum algorithm may be used to perform interpolation [1], using simple shifting of the input images and summation of the corresponding pixels. They also proposed possible depth-aided user definitions of the focusing plane. The shift-sum algorithm is used as a part of the proposed method in this paper, not only for refocusing but also for camera position change, using input image weights. The mutual orientation of the intersection point on the geometric proxy and the arbitrarily positioned input cameras can be used directly without a regular grid to determine which pixels contribute most to the result [7]. An alternative approach to the two-plane parameterization is to use view dependent texture mapping onto a simplified scene geometry in which each polygon is associated with a part of the texture acquired from the light field. This texture may change according to the viewing angle of the virtual camera [8]. Finally a simple way to generate synthetic views from two neighbouring light field images is to use optical flow aided interpolation [9].

Because most rendering methods rely on depth information, depth maps have to be estimated from the input images if they are not already available (using depth sensors placed at the capturing spot or obtaining them from synthetic scenes). A semi-

global matching method has been developed for dense disparity estimation from rectified stereo images by searching in predefined directions and search range for the most similar pixel blocks between images [10]. Those with lowest disparity are chosen. In this way, disparity maps can be obtained from the light field images and filtered, resulting in an approximate depth map [11]. Optical flow based depth estimation methods using feature matching in images also exist but they are generally very slow [12]. An optimized approach was proposed where four corner light field images are used to obtain disparity maps which are then aggregated using an energy minimization and warped into the resulting views [13]. A graph cut method for energy minimization for multi-camera scene reconstruction has also been proposed [14]. Spatial-aware edge-aware filters can be used to estimate dense depth maps from first sparse phase which is faster than a dense optical flow calculation [15]. For datasets acquired by plenoptic cameras, a depth-from-light-field technique exploiting symmetry of the focal stack has been proposed [16]. Another technique suitable for plenoptic camera datasets uses spatial variance after angular integration of the epipolar image to find defocus depth cues and angular variance for depth cue correspondence estimation [17]. Small radius matching windows can be used if there are many images in the light field datasets. Flat uniform regions which are unsuitable for such an approach can be analyzed at a lower resolution, leading to multi-resolution matching approaches [18]. Multi-resolution depth estimation can also be used when working with wide-baseline sparse datasets. A whole capturing and rendering pipeline using such an approach, with point cloud projection based final image synthesis, has already been proposed [19]. Using more than the 4×4 proposed grid of cameras to achieve better rendering results in this pipeline might, however, negatively affect performance. The extremely narrow baseline in lenslet light field camera datasets causes problems when estimating depth or disparity from such data. This problem can be solved by exploiting the phase-shift theorem in the Fourier domain to estimate sub-pixel shifts [20]. The performance of depth or disparity estimation methods is in most cases insufficient for real-time usage with rendering. Optimized methods for light field data also usually

work well with plenoptic camera data but not with large baseline datasets. Our proposed method uses only the necessary information from a subset of light field images and generates only one necessary map for the novel view, reducing memory access operations.

Light field images can also be analyzed in the spectral domain by using image transformations, analyzing frequencies present in the images. One depth independent reconstruction method exploits sparsity in the continuous Fourier domain to sample the light field effectively, to obtain the best possible quality [21]. Densely sampled epipolar-plane image reconstruction using the shearlet transform can be achieved by exploiting light field sparsity in the shearlet domain [22]. A way of finding the optimal sampling pattern for the light field reconstruction was published, defining a new sampling quality metric using symmetry constraints; it outperforms the maximized minimum distance and reduces the search space [23].

Deep learning approaches were also utilized to address the depth extraction [24, 25] and rendering [26, 27], based on a few reference images. An unsupervised approach working with planar light fields, using one network for disparity and one for occlusion map estimation, managed to yield results comparable to supervised approaches, overcoming the fully supervised methods' drawbacks [28].

The closest research to this paper is described in Ref. [29]. It first generates tens of differently focused views for a given viewpoint, using standard light field rendering methods. Areas in focus are then chosen [30] from the previously generated views and the final image is constructed from them. This approach, however, was demonstrated only on small resolution images with closely spaced cameras. It also uses multiple synthesis filters, exploiting the density of the Lytro dataset, which may not work well on sparse datasets. The method was further improved but it is still unusable for real-time rendering [31]. All-focus images can also be generated using high dynamic range light fields [32], where position, direction, and exposure time information is integrated in the light field model. Local focusing planes can be also estimated for each view or even for each triangle of the resulting viewing plane by simple minimization of least squares errors [33].

3 Light field focusing

The original light field rendering approach [3] and other derived methods support one focusing plane in which the image is constructed and focused. An effect similar to depth-of-field in classical photography is present in such an image. However, this effect is not always desired and an all focused image, as if captured by a pinhole camera, is often needed. To achieve this, each pixel of the image has to be focused to a different distance according to the scene geometry. Depth differences of scene geometry lead to a parallax effect. The apparent position of an object differs in each view; this can be described by a disparity map: the disparity of the object depends on its depth. Figure 3 shows such a scenario using two-plane parameterization. To correct the rays, depth or geometric information from the scene is necessary.

Even if a single focusing distance is enough to satisfy the user, the resulting out-of-focus effect is simply created by compositing the images on top of each other, resulting in block artifacts caused by the discrete light field representation: see Fig. 4. A sparser light field image grid representation leads to more visible block artifacts due to its inability to reconstruct the continuous light field. Again, depth or disparity information is needed to decide which

Fig. 3 Focusing distance effect in the two-plane light field parameterization (planes drawn as lines). Two rays coming from the virtual camera c intersect the scene's geometry at points a and b . The original sampling cameras c_i are evenly distributed along the st plane. New sampling rays are emitted from the closest cameras (c_5 , c_6 , and c_7) to a_{st} and b_{st} converging at the focusing distance on camera c 's ray vector. The image captured by the given sampling camera is projected onto the uv plane and the intersection points a'_{uv} , a''_{uv} and b'_{uv} , b''_{uv} determine which pixels are used in the final interpolation. The intersection points a'_{uv} , a''_{uv} demonstrate the correct situation where each camera ray intersects the geometry in a correct place. Points b'_{uv} , b''_{uv} simply intersect the uv plane, ignoring the geometry in the scene (rays sample the geometry in different places) which leads to a blurry image as shown in Fig. 4.

Fig. 4 Far left: original cube which is in focus. Inside left: ground truth out-of-focus cube, where defocusing is simulated using a Gaussian blur. Inside right, far right: defocusing generated by shifting the focusing plane using the two-plane method when using 8×8 and 4×4 light field grids respectively. Block artifacts become visible as grid dimensions decrease.

parts of the image should be filtered to simulate a smooth blurring effect.

4 Method

Our proposed rendering method consists of two steps. The first step generates a focus map in which each pixel contains a focusing value. The second image composition step uses the focus map values. In the result, each pixel has its own focusing value, eliminating the issues related to use of a single global focusing distance as shown in Fig. 2.

4.1 Weighted shift-sum algorithm

In the shift-sum algorithm [1], each output pixel is the result of a sum of pixels from different views (the input images from the camera grid). The resulting pixel values contain lighting information (usually color). The pixels contributing to the result are shifted by an offset in the views depending on their position in the input grid (further images have to be shifted more than images closer to a chosen reference position in the grid). In this way, a single-focusing-distance image can be rendered. Pixels capturing objects in the scene that share the same distance from the camera grid overlap in the resulting sum. Despite taken from different views, their colors are similar or the same. To achieve a 3D effect when moving a virtual camera, weights can be used to prioritize views from the grid that are most relevant, according to the angle between a vector from the input view grid center to the virtual camera center and the view grid plane. It is not necessary to sample all images, just those that are within a certain distance of the virtual view. The distance is defined globally, and is the same value for all pixels. The calculation used is

$$p_o(\bar{c}, f) = \frac{\sum_{i=1}^n p_i(\bar{c} + \bar{o}_i \cdot f) w_i}{\sum_{i=1}^n w_i} \quad (1)$$

where i is the index of current input view, p_o is the

function computing the output pixel, p_i is the pixel from the i th view, n is the number of input views, $\bar{c}$ represents coordinates in the output image, $\bar{o}_i$ is the offset between images in the grid, f is the view shift (focus distance), and w_i is the weight of the pixel. One iteration of the shift-sum synthesis algorithm is depicted in Fig. 5.

4.2 Focus map

The generation of the focus map follows the semi-global matching method [10], searching for the disparity value in a given range with lowest cost. The weighted shift-sum algorithm is used to generate new views at various focusing distances. When the whole focusing range is iteratively scanned, each pixel is in focus in a certain iteration. The number of tested distances depends on how densely the focusing range is sampled and can be increased for large depth range scenes. For each pixel and each focusing distance, a variance is computed during summation based on Chebyshev distance between the pixel values (which was experimentally determined to be the most suitable metric here; generally the choice of color metric for a given task is problematic [34]). The variance is calculated relative to the mean value of the colors of pixels contributing to the shift-sum. The pixel with lowest variance is chosen, and the corresponding focusing distance is stored in the focus map. The whole process is outlined in Algorithm 1. The distance is simply stored as the index of the focusing step, and the final focusing value

Fig. 5 One iteration of shift-sum based image synthesis: a pixel from the i th image is summed into the output pixel p_o . The red box depicts the new synthetic image; the purple lines show the offsets relative to the currently sampled image and the distance between the two images. The currently sampled pixel's weight w_i depends on the distance between the two images. Notation is as in Eq. (1). Superscripts x and y denote components of a vector variable.

Algorithm 1 Focus map estimation, iterating over a range of focusing distances and choosing the value with minimal variance from the shift-sum phase. Function `shiftSum` uses the shift-sum algorithm (Eq. (1)) and returns the final color and variance of the colors contributing to the summation, using Algorithm 2. This algorithm was generalized, returning also the focused pixel color. Image synthesis is, however, separated in the reference implementation.

Data: Grid of images, position of virtual camera, focusing bounds, focus step

Result: Focus map, focused pixel color

```

for  $c = 0$ ;  $c < \text{focusMapPixels.size()}$ ;  $c++$  do
    variances = Array[focusLevels];
    colors = Array[focusLevels];
    for  $i = 0$ ;  $i < \text{focusLevels}$ ;  $i++$  do
         $f = \text{focusStart} + i \times \text{focusStep}$ ;
        pixel = shiftSum(camPos, f, c, grid);
        variances[i] = pixel.variance;
        colors[i] = pixel.color;
    end
    focusMapPixels[c] = variances.indexOfMin();
    pixelColor = colors[focusMapPixels[c]];
end

```

is recalculated in a fragment shader. This way, the necessary bit depth of the map needs to allow for just the number of distances searched. This statistical analysis of contributions to the final pixel value can determine whether a pixel is focused.

4.3 Final image synthesis

The same shift-sum algorithm is used for the final image synthesis. Each pixel of the output image is computed according to Eq. (1), mixing pixels from images in the grid that are within the defined distance from the new synthetic view. Each pixel is interpolated as depicted in Fig. 5 and the coordinates of the sampled pixels are computed by adding the relative offset of the new view and the currently sampled image from the grid, multiplied by focusing distance from the focus map to the currently computed pixel coordinates ($\bar{c} + \bar{o}_i \cdot f$ from Eq. (1), where the offset is the shift of the sampled image from the synthetic one). The focusing distance was previously determined in Algorithm 1. During focus map generation, a variance was the desired result of the summation, and now the resultant color is computed for the final image. In the given algorithm, it would be possible to acquire the final image color directly, but the focus map generation and the final image composition steps are separated, so that each one can produce a result at a different resolution. In image synthesis, only the outer loop over all pixels of the image is necessary, performing only the shift-sum with a focus value taken from the focus map.

Experiments demonstrate that better performance without significant quality loss can be achieved by generating the focus map at a lower resolution than the final image. Doing so is key to real-time usage. Examples of final image and focus map are shown in Fig. 6.

Fig. 6 Focused result using the Pavilion dataset, and a corresponding focus map. The latter contains an estimated focusing value for each pixel of the final image. The map resembles a depth or disparity map for the given synthetic view, as the focusing values depend on the distance of each pixel from the camera.

5 GPU utilization scheme

Our method can exploit the massive parallelism of GPU architectures. OpenGL was used for both rendering and GPGPU computations in the reference implementation. The focus map is generated using a compute shader. Each warp (32 threads on NVIDIA cards) is assigned to one pixel. Each workgroup consists of 8 neighbouring pixels. This scheme offers good GPU occupancy and memory access coherency, allowing in-warp data transfer between threads which is much faster than using global or local memory. Each thread computes one focusing distance (or more when denser search is required), using the weighted shift-sum and the Welford's variance algorithm [35] (see Algorithm 2) which improves GPU occupancy by reducing the number of registers needed. At the end,

Algorithm 2 Welford’s method for computing online variance in one pass, adjusted to pixel values (RGB colors in reference implementation). This algorithm is used in the shift-sum, analyzing new color values coming into the summation.

Data: Stream of pixel values
Result: Estimated variance

```

n = 0;
mean = 0;
m2 = 0;
for each pixel in input do
    n++;
    delta = pixel - mean;
    distance = pixelDistance(pixel, mean);
    mean += delta/n;
    m2 += distance × pixelDistance(pixel, mean);
end
m2 /= n-1;

```

the minimal variance value within a warp is found using parallel reduction with ballot operation. In the fragment shader, a surface representing the light field is rendered using the weighted shift-sum algorithm again, this time with the correct focusing values from the previously generated focus map. The focus map and the input images are stored as textures; therefore, missing pixels can be interpolated in texturing units if the resolutions of the result and the focus map differ. Figure 7 shows the work distribution on a GPU.

Fig. 7 Work distribution on the GPU for focus map generation. The compute shader analyses the input images, going through the focusing range and saving the focusing value with minimal variance in the focus map. Because the workload is divided into warp sized elements, no global or local synchronization is needed.

6 Evaluation

6.1 Preliminaries

The purpose of the first experiment was to determine which color distance metric would be most suitable for computing the variance from the resulting color

summation. The overall visual quality of results of our method was assessed in the second experiment. The third experiment aimed to determine the trade-off between performance and visual quality when reducing the focus map dimensions. The fourth experiment, similarly to the previous one, investigated the optimal depth of the resulting focus map. The fifth experiment was performed to decide how many images from the input grid need to be sampled, and the final experiment analyzed how camera grid parameters of the dataset affect the quality of the results of the proposed method.

Datasets used in the experiments included those captured with a camera array from Stanford light field archives [36], light fields from EPFL captured by a Lytro Illum plenoptic camera [37], and synthetic dataset rendered of the Barcelona Pavilion scene available from the Blender demo files page [38]. Only one Lytro dataset was used because the distance between Lytro views is very small: its capturing mechanism is based on a special lens creating multiple close views. While it is an ideal dataset for refocusing, it has a very limited ability to move the virtual camera to create a 3D viewing effect. In all experiments, a ground truth center view from the original dataset was chosen as a reference and it was compared to a new synthetic view rendered by the proposed method, using SSIM and PSNR metrics. The Pavilion dataset was used for the performance tests because its resolution is sufficient to reflect the commonly used FullHD video standard. One dataset is enough for performance testing because the computation time of the proposed method depends only on its parameters and dataset resolution and not on the data content. All experiments were executed on a machine equipped with an Nvidia GeForce RTX 2070 GPU and a 3 GHz Intel Core i5-8500 CPU @, running Arch Linux.

6.2 Color distance metric

The variance computation phase of the proposed algorithm requires a pixel color value distance metric to decide how much two pixels differ in terms of color similarity. The right choice of metric depends on various aspects such as expected color range, type of image, and final use-case. We first compared various RGB color distance metrics to find out which one yields the results with best visual quality for light field datasets: see Fig. 8. The quality differences were small but computational effort required by the

Fig. 8 Comparison of RGB color distance metrics for the pixel similarity test during the variance computation phase. W in a metric name stands for weighted metrics. Average results over all tested datasets are presented.

metrics differed, with the potential to negatively affect performance (e.g., DeltaE). The Chebyshev metric was chosen for further experiments because of its high-quality results and computational simplicity.

6.3 Overall quality

For each dataset (see Fig. 9), the best initial focusing level and search step were manually found and the resulting images were compared to the reference. Final visual quality is evaluated in Fig. 10. The images are focused in all parts, but interpolation artifacts are visible in problematic areas such as around thin edges or near similarly colored areas. Details of interpolation artifacts are captured in Fig. 11. The Bunny dataset contains only diffuse material and is clearly separated from the black background; therefore, the reconstruction had few artifacts. Even though the Chess dataset contains many reflections, the chessboard pattern along with a relatively small distance between views improved the quality of the result. The Bulldozer contains

Fig. 9 Reference views from each experimental dataset. From top left: Bulldozer, Bunny, Chess, Lego, Pavilion, Lytro.

many small details that are clearly separated from the yellow construction of the model which again causes higher variance values when mixing nearby pixels. The Lego dataset is filled with a single color area, where for example on the wall at the back small edges or details are hard to detect, and the pixels interpolated from surrounding area can yield lower variance. The distance between Lytro cameras is small so the result was expected to be better, but due to the technical drawbacks of the camera, the input images contain subtle noise that negatively affects the results. The Pavilion contains both large similarly colored areas and complex objects with many details, but the distance between cameras is somewhat greater, allowing more freedom when moving the virtual camera. Figure 12 shows the time taken for focus map generation and final compositing of pixels from each dataset.

6.4 Comparison to other methods

An accurate performance comparison to state of the art methods is complicated due to different methodology and outputs. The proposed method generates focus map for the new synthetic view used in the rendering stage. The process is roughly

Fig. 10 Best results when rendering a new view from each dataset compared to the ground truth. Rendering settings were manually adjusted for best visual quality. Some of the results are shown in Figs. 2, 6, 11, 14, and 19.

Fig. 11 Left: Reference images. Right: Rendered reconstructed images. Below: close-up detail of interpolation artifacts caused by incorrect focusing level estimates in affected pixels.

comparable to depth or disparity map estimation. Table 1 provides an indicative overview of computation time of this stage.

A side-by-side visual quality comparison with state of the art methods is provided in Fig. 19. Methods that are capable of producing the synthetic view directly from images were chosen for the evaluation. The proposed method outperforms other similar approaches. View reconstruction on the Bunny

Fig. 12 Time taken for focus map generation and drawing, which depends on the focus map and output image resolution respectively. Full, 1/4, and 1/8 sized focus maps were considered. Drawing time slightly increases when using smaller focus maps most likely due to coordinate interpolation in texturing units due to resolution mismatch.

Table 1 Computation time of state of the art depth or disparity estimation methods from light fields

Alg.	Architecture	Resolution	Time
Ours	RTX 2070	1920 × 1080 × 64	18 ms
[40]	Tesla C2050	640 × 480 × 2	16 ms
[11]	E3-1245 V2	541 × 376 × 9	1.5 s
[13]	i7 2.8 GHz	512 × 512 × 49	13 min
[15]	i7-6700k	512 × 512 × 49	0.8 s
[41]	Quadro M1000M	1920 × 1080 × 45	1.58 s

dataset, using the strongest competitor, the shearlet approach [22], takes 5 s on a GeForce GTX Titan X, which is unsuitable for real-time rendering.

The proposed method does not reach the same visual quality as newer learning based methods [28] when measured on the same dataset that was used in the original paper, but slightly outperforms older methods [39] (indirect comparison on Kitchen and Museum datasets, difference about 1 dB [28]). The proposed method, however, does not depend on any training process.

Measurements show that our new method is comparable to other published algorithms in terms of visual quality, yet reaches performance suitable for real-time rendering. Rendering time can be further improved by slight reduction of visual quality as shown in Fig. 13.

Fig. 13 Relation of visual quality, computation time, and amount of focus map dimension division. The results are averaged from all tested datasets.

6.5 Focus map resolution

One of the key features of the proposed method is the separation of focus map generation from interpolation of the final result. Figure 13 shows how reduction of focus map size affects the computation time and visual quality of the final image. Surprisingly, the quality does not decrease rapidly even with significant focus map downscaling. In certain cases, the quality even improves because some areas with incorrect focusing levels are smoothed by the filtering caused by resizing. However small map sizes can assign the same focusing level to nearby objects that might not lie at the same distance, causing out-of-focus artifacts as shown in Fig. 14.

6.6 Focus range search density

Bit depth of the focus map affects how accurate the focusing distance is. Increasing the number of search samples when iterating over the focusing distances in the given range does not affect the visual quality significantly and slows down the computation unnecessarily, as shown in the Fig. 16. 32 samples proved to be an optimal choice for most datasets. The most significant difference in quality was measured on the Pavilion dataset which has the biggest depth range, so denser searching is necessary, especially as objects in the scene are linearly distributed over the whole depth range.

6.7 Camera grid sample radius

The experimental results in Fig. 17 show how many images need to be sampled when acquiring pixel values for the resulting pixel sum. The plots show that the optimal sampling window in the input grid has a radius about 2 grid views wide, slightly differing with dataset. A wider radius leads to more texture

reads and excessive memory access which slows down the computation. The sample distance is the radius of a circle with the virtual camera position at its center. Surrounding images from the grid with distance from zero to the sampling distance are taken into account during interpolation. If the radius is too large, images from distant places in the grid can add unwanted ghosting artifacts to the final result, forcing the algorithm to use views that show the scene from a different angle to that expected.

Fig. 14 Focusing artifacts caused by low resolution focus maps. Top: generated depth map and reference image. Below: focus maps with resolutions divided by 2, 20, and 50, and generated results.

6.8 Camera grid parameters

The Pavilion dataset was used to measure the relation between visual quality of the reconstructed view and distance between cameras with various focal lengths. The distance between cameras, field of view, total depth range in the scene, and position of the camera grid in the scene affect the quality of the resulting reconstruction: see Fig. 18. The camera setup used in the scene is shown in Fig. 15. With increasing distance between the cameras or decreasing field of view (increasing focal length), the differences between

Fig. 15 Top: the size of the grid (red oval) in the Pavilion scene and the value of field of view were animated and the resulting reconstruction quality was measured. Centre, below: two views from the corners of the grid using 25 and 55 mm focal length respectively. The difference between the views is bigger in the 55 mm version.

Fig. 16 Variation of overall quality of the result with density of search of focusing range. Quality metric values are averaged over all tested datasets.

views increase and interpolation is more prone to visual artifacts. On the other hand, the more different the camera positions and view cones are, the more freedom is gained for the virtual camera. This issue can be overcome with denser sampling [42], providing more views in the grid, increasing its dimensions. This, however, leads to higher memory or bandwidth requirements.

7 Conclusions

The task of light field focusing was addressed in this research, resulting in a novel method for per-pixel analysis and rendering of synthetic light field views. This method, unlike the state of the art, does not require precomputed or exported depth or scene geometry information, which also reduces memory and bandwidth requirements and computes the resulting view quickly enough to be suitable for interactive applications while providing results of good visual quality. The method uses a simple statistical analysis of the colors contributing to each pixel in the final result. Each resulting pixel value can be computed independently of the rest of the images without excessive memory access. The proposed principle is general enough to be used with every commonly used light field representation or parameterization. This research also revealed important information about the relation between visual quality and computation time when adjusting parameters of the interpolation shift-sum algorithm. The massive parallelism of GPUs allows this method to run in real time even for high resolution datasets, corresponding to current video standards. This method also works on datasets with larger distances between the input views than from datasets acquired with current plenoptic cameras.

Fig. 17 Maximal sample distance parameter and its relation to both visual quality of the result and computation time. Results are again averaged over all tested datasets.

Fig. 18 The camera grid contains 8×8 cameras and is initially 2 m wide in scene-space. The first visible surface is about 1 m from the grid, and the furthest visible spot excluding the sky is about 90 m away. The camera grid is uniformly scaled up and down to change the distance between cameras.

Fig. 19 Visual comparison with other state of the art methods: Vagharshakyan [22], Shi [21], Brox [9], and Ni [28], rendering a new synthetic view. Our method outperforms other general methods but does not reach the same quality as learning based methods, trained on the specific dataset. The results of direct methods are almost identical; differences from the proposed method are below 1 dB. Small blurring artifacts a few pixels large are visible around certain details in all cases. Our method produces the sharpest result. New learning based methods produce better results in parts of the image with thin and reflective objects, but depend on training with the dataset. Reflections and thin details can cause problems in our method, when comparing pixel colors from different views.

Visual artifacts are visible in the current version of the proposed method, and are caused by incorrect focusing distances for the affected pixels. The statistical method can fail if the pixel under test is blurred in a way that the resulting variance is lower than the correct focusing distance. This can happen

when thin edges or small details are surrounded by similarly colored areas. The global minimum of the variance does, therefore, not always lead to the best result. An analysis of the variance values and local minima may be a way to select a better focusing value.

In future work, additional experiments with focus map filtering will be carried out. Preliminary tests showed that median filtering may be used to denoise the map slightly, improving the visual quality. The resulting focus map can also be used to simulate additional photographic effects such as depth of field. Currently, it is necessary to define a focusing range for each dataset manually. It is possible that the search bounds could be estimated automatically based on another statistical analysis of the overall amount of blur in the scene.

Acknowledgements

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science, LQ1602.

References

- [1] Alain, M.; Aenchbacher, W.; Smolic, A. Interactive light field tilt-shift refocus with generalized shift-and-sum. *arXiv preprint* arXiv:1910.04699, 2019.
- [2] Landy, M.; Movshon, J. A. The plenoptic function and the elements of early vision. In: *Computational Models of Visual Processing*. MIT Press, 3–20, 1991.
- [3] Levoy, M.; Hanrahan, P. Light field rendering. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 31–42, 1996.
- [4] Gortler, S. J.; Grzeszczuk, R.; Szeliski, R.; Cohen, M. F. The lumigraph. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 43–54, 1996.
- [5] Isaksen, A.; McMillan, L.; Gortler, S. J. Dynamically reparameterized light fields. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 297–306, 2000.
- [6] Lin, Z. C.; Shum, H. Y. A geometric analysis of light field rendering. *International Journal of Computer Vision* Vol. 58, No. 2, 121–138, 2004.
- [7] Buehler, C.; Bosse, M.; McMillan, L.; Gortler, S.; Cohen, M. Unstructured lumigraph rendering. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 425–432, 2001.
- [8] Debevec, P.; Yu, Y. Z.; Borshukov, G. Efficient view-dependent image-based rendering with projective texture-mapping. In: *Rendering Techniques '98. Eurographics*. Drettakis, G.; Max, N. Eds. Springer Vienna, 105–116, 1998.
- [9] Brox, T.; Bruhn, A.; Papenberger, N.; Weickert, J. High accuracy optical flow estimation based on a theory for warping. In: *Computer Vision - ECCV 2004. Lecture Notes in Computer Science, Vol. 3024*. Pajdla, T.; Matas, J. Eds. Springer Berlin Heidelberg, 25–36, 2004.
- [10] Hirschmuller, H. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 30, No. 2, 328–341, 2008.
- [11] Anisimov, Y.; Wasenmüller, O.; Stricker, D. Rapid light field depth estimation with semi-global matching. In: *Proceedings of the IEEE 15th International Conference on Intelligent Computer Communication and Processing*, 109–116, 2019.
- [12] Sun, D. Q.; Roth, S.; Black, M. J. Secrets of optical flow estimation and their principles. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2432–2439, 2010.
- [13] Jiang, X.; Pendu, M. L.; Guillemot, C. Depth estimation with occlusion handling from a sparse set of light field views. In: *Proceedings of the 25th IEEE International Conference on Image Processing*, 634–638, 2018.
- [14] Kolmogorov, V.; Zabih, R. Multi-camera scene reconstruction via graph cuts. In: *Computer Vision — ECCV 2002. Lecture Notes in Computer Science, Vol. 2352*. Heyden, A.; Sparr, G.; Nielsen, M.; Johansen, P. Eds. Springer Berlin Heidelberg, 82–96, 2002.
- [15] Chen, Y.; Alain, M.; Smolic, A. Fast and accurate optical flow based depth map estimation from light fields. *arXiv preprint* arXiv:2008.04673v1, 2020.
- [16] Lin, H. T.; Chen, C.; Kang, S. B.; Yu, J. Y. Depth recovery from light field using focal stack symmetry. In: *Proceedings of the IEEE International Conference on Computer Vision*, 3451–3459, 2015.
- [17] Tao, M. W.; Hadap, S.; Malik, J.; Ramamoorthi, R. Depth from combining defocus and correspondence using light-field cameras. In: *Proceedings of the IEEE International Conference on Computer Vision*, 673–680, 2013.
- [18] Neri, A.; Carli, M.; Battisti, F. A multi-resolution approach to depth field estimation in dense image arrays. In: *Proceedings of the IEEE International Conference on Image Processing*, 3358–3362, 2015.
- [19] Sabater, N.; Boisson, G.; Vandame, B.; Kerbiriou, P.; Babon, F.; Hog, M.; Gendrot, R.; Langlois, T.; Bureller, O.; Schubert, A. Dataset and pipeline for multi-view light-field video. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1743–1753, 2017.

- [20] Jeon, H. G.; Park, J.; Choe, G.; Park, J.; Bok, Y.; Tai, Y. W.; Kweon, I. S. Accurate depth map estimation from a lenslet light field camera. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1547–1555, 2015.
- [21] Shi, L. X.; Hassanieh, H.; Davis, A.; Katabi, D.; Durand, F. Light field reconstruction using sparsity in the continuous Fourier domain. *ACM Transactions on Graphics* Vol. 34, No. 1, Article No. 12, 2014.
- [22] Vagharshakyan, S.; Bregovic, R.; Gotchev, A. Light field reconstruction using shearlet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 40, No. 1, 133–147, 2018.
- [23] Schedl, D. C.; Bimber, O. Optimized sampling for view interpolation in light fields with overlapping patches. *Computer Vision and Image Understanding* Vol. 168, 93–103, 2018.
- [24] Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2, 2366–2374, 2014.
- [25] Peng, J.; Xiong, Z.; Liu, D.; Chen, X. Unsupervised depth estimation from light field using a convolutional neural network. In: *Proceedings of the International Conference on 3D Vision*, 295–303, 2018.
- [26] Eslami, S. M. A.; Jimenez Rezende, D.; Besse, F.; Viola, F.; Morcos, A. S.; Garnelo, M.; Ruderman, A.; Rusu, A. A.; Danihelka, I.; Gregor, K. et al. Neural scene representation and rendering. *Science* Vol. 360, No. 6394, 1204–1210, 2018.
- [27] Han, X. F.; Laga, H.; Bennamoun, M. Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence* doi: 10.1109/TPAMI.2019.2954885, 2019.
- [28] Ni, L. X.; Jiang, H. Y.; Cai, J. F.; Zheng, J. M.; Li, H. F.; Liu, X. Unsupervised dense light field reconstruction with occlusion awareness. *Computer Graphics Forum* Vol. 38, No. 7, 425–436, 2019.
- [29] Takahashi, K.; Kubota, A.; Naemura, T. All in-focus view synthesis from under-sampled light fields. In: *Proceedings of the 13th International Conference on Artificial Reality and Telexistence*, 2003.
- [30] Sugita, K.; Takahashi, K.; Naemura, T.; Harashima, H. Focus measurement on programmable graphics hardware for all in-focus rendering from light fields. In: *Proceedings of the IEEE Virtual Reality*, 255–256, 2004.
- [31] Kubota, A.; Takahashi, K.; Aizawa, K.; Chen, T. All-focused light field rendering. In: *Proceedings of the 15th Eurographics Workshop on Rendering Techniques*, 235–242, 2004.
- [32] Li, C.; Zhang, X. High dynamic range and all-focus image from light field. In: *Proceedings of the IEEE 7th International Conference on Cybernetics and Intelligent Systems and IEEE Conference on Robotics, Automation and Mechatronics*, 7–12, 2015.
- [33] Heigl, B.; Koch, R.; Pollefeys, M.; Denzler, J.; Gool, L. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. In: *Mustererkennung 1999. Informatik aktuell*. Förstner, W.; Buhmann, J. M.; Faber, A.; Faber, P. Eds. Springer Berlin Heidelberg, 94–101, 1999.
- [34] Sanda Mahama, A.; Dossa, A.; Gouton, P. Choice of distance metrics for RGB color image analysis. *Electronic Imaging* Vol. 2016, No. 20, 1–4, 2016.
- [35] Welford, B. P. Note on a method for calculating corrected sums of squares and products. *Technometrics* Vol. 4, No. 3, 419–420, 1962.
- [36] Vaish, V.; Adams, A. The (new) stanford light field archive. 2008. Available at <http://lightfield.stanford.edu/>.
- [37] Rerabek, M.; Ebrahimi, T. New light field image dataset. In: *Proceedings of the 8th International Conference on Quality of Multimedia Experience*, 2016.
- [38] Cheggour, H. Blender demo files - barcelona pavillion.
- [39] Kalantari, N. K.; Wang, T.-C.; Ramamoorthi, R. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics* Vol. 35, No. 6, Article No. 193, 2016.
- [40] Banz, C.; Blume, H.; Pirsch, P. Real-time semi-global matching disparity estimation on the GPU. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 514–521, 2011.
- [41] Chuchvara, A.; Barsi, A.; Gotchev, A. Fast and accurate depth estimation from sparse light fields. *IEEE Transactions on Image Processing* Vol. 29, 2492–2506, 2020.
- [42] Chai, J. X.; Chan, S. C.; Shum, H. Y.; Tong, X. Plenoptic sampling. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 307–318, 2000.

T. Chlubna is a Ph.D. student and member of the Graph@FIT Group in the Department of Computer Graphics and Multimedia, Faculty of Information Technology, Brno University of Technology, Czech Republic where he received his M.Sc. degree. His research is focused on light field rendering and

computational geometry.

T. Milet is a Ph.D. student and member of the Graph@FIT Group at Department of Computer Graphics and Multimedia, Faculty of Information Technology, Brno University of Technology, Czech Republic where he received his M.Sc. degree. His research is focused on light field and shadow rendering.

P. Zemčík is Dean and Full Professor of the Faculty of Information Technology, Brno University of Technology, Czech Republic. He is also a member of the Graph@FIT Group at Department of Computer Graphics and Multimedia. His interests include computer vision and graphics algorithms, acceleration of algorithms, programmable hardware, and also applications.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.