Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Research Article

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

T. Chlubna[1] (✉), T. Milet[1], and P. Zemčík[1]

Abstract Light field rendering belongs to image-based rendering methods that do not use 3D models but only images of the scene as the 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 the information about the depth in the scene, a 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 for focusing of light field scenes, based on statistical analysis of the pixel values contributing to final image, is proposed. Compared to existing techniques, this method does not need a precomputed or acquired depth information. Memory requirements and streaming bandwidth are reduced and real-time rendering is possible even when using a high resolution light field data, yielding visually satisfactory results. Experimental evaluation of the proposed method implemented on GPU is presented in this paper.

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

1 Introduction

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, ray-tracing, etc. Complexity of the scene, however, considerably affects the time necessary for 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 the image-based rendering methods does not depend on the scene's content.

Field of light in a scene can be ideally described with a function representing lightning information for each point in the space and each direction relative to this point. The scene then can be visually reconstructed using arbitrary camera position and orientation. For usage in computer science, such a continuous function would be 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 as illustrated in Figure 1. The input images sample the scene from expected viewing angles, which provides as much of visual information about the scene as possible. The higher number of such images is available, the better quality of the final render can be achieved. Storing more images, however, increases the space requirements in memory. The goal of light field rendering methods is to use only a sparse set of image samples while achieving the best visual quality of the rendered result. In practice, light field can be viewed as an extension of classic photography which allows 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 the problems connected with a novel view image reconstruction. This work is focused on the elimination of so-called out-of-focus areas in light field without additional information about the depth or 3D models of the scene. When using the approximation of light field by a set of images, the pixel values that are combined together in the final render have to be taken from the same spot in the scene. This kind of spatial

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

Manuscript received: 2014-12-31; accepted: 2015-01-30.

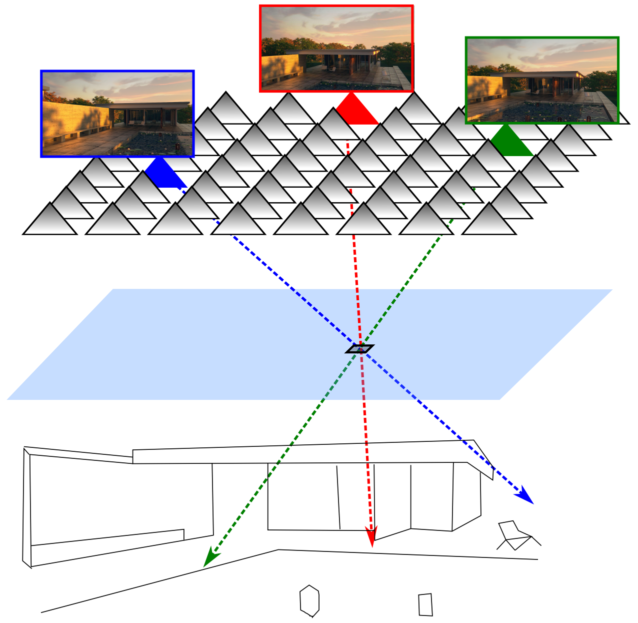

Fig. 1 Scene captured by a grid of cameras. Light field approximation consisting of the images from this grid can be used to reconstruct a novel view from any camera position outside the bounding volume of the scene. Three cameras are highlighted in the figure with rays coming through a pixel with the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

information for each pixel has to be estimated and used to achieve the correct focusing for the final image as shown in Figure 2. The proposed method is based on statistical analysis of the pixel values that are at the end combined, using shift-sum algorithm [2], into one pixel color in the resulting image. The method iterates over a range of focusing distances and stores the best distance for each pixel in a focus map. The best distance is chosen according to the minimal variance of the pixels, contributing to the interpolation process. A weighted shift-sum algorithm is used for the interpolation of the final image and for the pixel analysis. A novel view is synthesized using the generated focus map. With this method, visually acceptable all focused light field scenes can be rendered from the input set of images without further knowledge about the original scene.

2 Related work

The flow of light in a space can be described by a 7D plenoptic function $L = P(x, y, z, \theta, \varphi, t, \lambda)$ [1]. In terms of geometric optics, this function returns the light intensity (L) of an incoming ray to a point in 3D space (x, y, z) from a given direction (θ, φ). This value can change in time (t) and vary for each wavelength (λ). For practical usage, this function can be approximated by a 4D representation which is commonly referred as light field [24]. Let us assume that a scene is located between two parallel planes with a virtual camera outside. Rays coming from its center of projection intersect those two planes, producing one intersection point per plane. The intersection coordinates with camera plane $(\mathbf{st})$ are then used to decide which images from the input are being used for the final pixel interpolation and the coordinates from image plane $(\mathbf{uv})$ are used to get the correct pixels from the given images. Input images are typically captured in a regular grid that is mapped on the camera plane in 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 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 being in focus, intersection points on image plane need to be corrected, using depth of the scene. A simplified geometry of the scene can be used [14] as a scene surface approximation that would replace the planar image plane. Depth maps can also be used for the ray intersection correction and to enhance photographic effects such as depth of field [18]. The generalized concept of the two planes parametrization can be extended to support various light field shapes using two spheres, point and direction etc. [27]. Instead of plane intersection calculations, a shift-sum algorithm might be used for the interpolation [2], using a simple shifting of the input images and a summation of the corresponding pixels. Authors also proposed possible depth-aided user definitions of the focusing plane. Shift-sum algorithm is used as a part of the proposed method in this paper, not only for refocusing but also for the camera position change, using input image weights. The mutual orientation between the intersection point on the geometric proxy and the arbitrarily positioned input cameras can be used directly without regular grids to determine which pixels contribute to the result most [6]. An alternative approach to the two planes parametrization is a view dependent texture mapping on a simplified scene geometry where each polygon is associated with a part of the texture acquired from the light field. This texture might change according to the viewing angle of the virtual camera [11]. Finally a simple way to generate synthetic views from two neighbouring light field images is to use the optical flow aided interpolation [5].

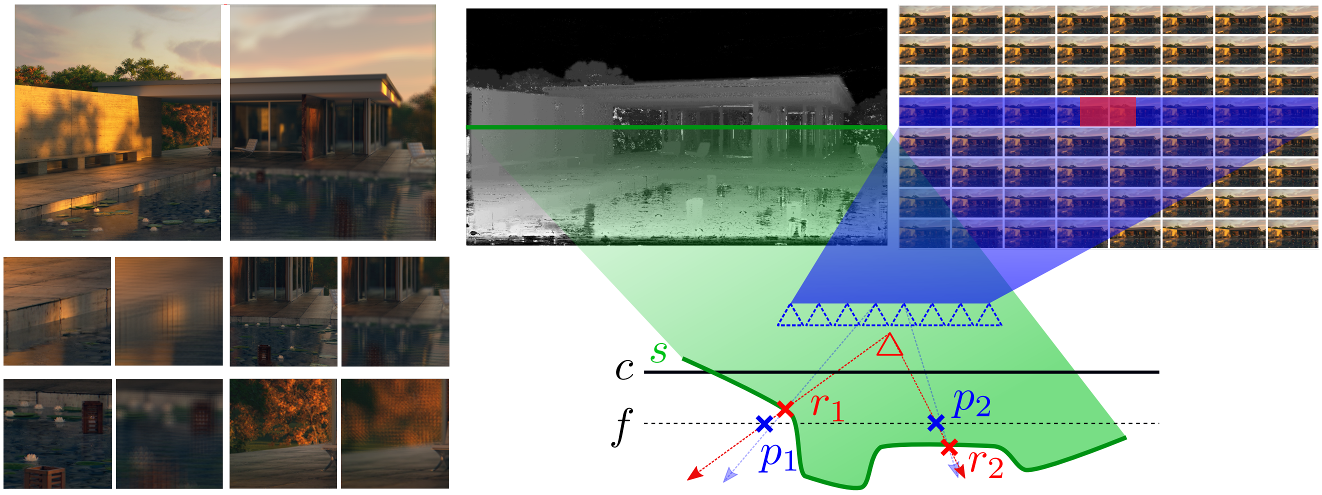

Fig. 2 Left: Comparison of fully focused light field image as a result of the proposed method with light field focused on single distance. Right: From the input set of images taken by camera grid, a new synthetic view of the scene is generated having every location in the scene focused as if captured by a pinhole camera. The proposed method performs real-time per-pixel focusing which solves the task of light field focusing without having 3D models of the scene available. Focusing distance values for each pixel are estimated and stored in a focus map which resembles disparity or depth map of the scene. This map is used to achieve a correct focusing of each pixel.

Because most of the rendering methods rely on the depth information, depth maps have to be estimated from the input images if they are not already available (using depth sensors on the capturing spot or obtaining them from synthetic scenes). A semi-global matching method was developed for a dense disparity estimation from rectified stereo images by searching for most similar pixel blocks between the images in predefined directions and search range [17]. The lowest cost (error) disparity is chosen in the end. In this way, disparity maps can be obtained from the light field images and filtered, resulting in a depth map approximation [3]. Optical flow based depth estimation methods using feature matching in images also exist but they are generally very slow [37]. 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 [20]. Graph Cuts method for the energy minimization for multi-camera scene reconstruction was proposed earlier as well [22]. Spatio-aware edge-aware filter can be used to estimate dense depth maps from first sparse phase which is faster than the whole dense optical flow calculation [9]. For datasets acquired by plenoptic cameras a depth-from-light-field technique exploiting symmetry property of the focal stack was proposed [26]. Another technique suitable for the plenoptic camera datasets uses spatial variance after angular integration of the epipolar image for defocus depth cues and angular variance for correspondence depth cues estimation [39]. Small radius matching windows can be used when having a lot of images in the light field datasets. Flat uniform regions which are not suitable for such approach can be analyzed in a lower resolution, leading to multi-resolution matching approaches [28]. The multi-resolution depth estimation can also be used when working with wide-baseline sparse datasets. The whole capturing and rendering pipeline using such approach with a point cloud projection based final image synthesis has already been proposed [32]. Using more cameras than the 4 × 4 proposed grid to achieve better rendering results in this pipeline might, however, negatively affect the 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 phase-shift theorem in the Fourier domain to estimate sub-pixel shifts [19]. The performance of depth or disparity estimation methods is in most cases not sufficient for real-time usage along with rendering. Optimized methods for light field data are also usually working well with the plenoptic camera data but not with the large baseline datasets. The 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 analysed in spectral domain by using image transformations, analyzing frequencies present in the images. One of the depth independent reconstruction methods exploits the sparsity in continuous Fourier domain to sample light field effectively, gaining the best possible quality [35]. Densely sampled epipolar-plane image reconstruction using shearlet transfrom can be achieved exploiting the light field sparsity in shearlet domain [40]. A way of finding the optimal sampling pattern for the light field reconstruction was published, defining a new sampling quality metric that outperforms the maximized minimum distance and reduces the search space, using symmetry constraints [34].

Both of the above-mentioned tasks are also addressed by deep learning approaches, be it the depth extraction [12, 30] or rendering based on few reference images [13, 15]. 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 full supervision methods' drawbacks [29].

The closest published research to this paper is All in-Focus View Synthesis from Under-Sampled Light Fields [38]. The method first generates tens of differently focused views for a given viewpoint, using standard light field rendering methods. Areas in focus are then chosen [36] from the previously generated views and the final image is constructed from them. This approach, however, was demonstrated only on small resolution images with a small distance between the cameras. It also uses multiple synthesis filters, exploiting the density of Lytro dataset, which might not work well on the sparse datasets. The method was further improved but it is still unusable for real-time rendering [23]. All-focus image can also be generated using high dynamic range light field [25], 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 square error [16].

3 Light field focusing

The original light field rendering [24] and other derived methods support one focusing plane where the image is constructed and focused. An effect similar to depth-of-field in classic 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 the scene geometry lead to a parallax effect. The apparent position of an object differs on each view which can be described by a disparity map. The amount of displacement of the object depends on its depth. Figure 3 shows the scenario where the two planes parametrization is used. To correct the rays, depth or geometry information from the scene is necessary.

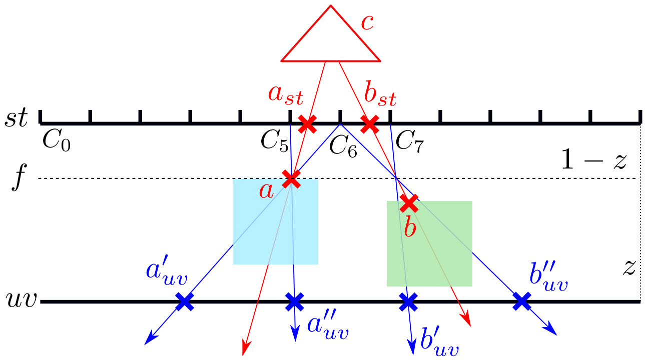

Fig. 3 The focusing distance effect in the two planes light field parametrization (planes drawn as lines). Two rays coming from the virtual camera c intersect the scene's geometry in points a and b. The original sampling cameras c_i are evenly distributed along the st plane. New sampling rays are emitted from the closest cameras (c_5, c_6 and c_7) to a_{st} and b_{st} converging at the focusing distance on the camera c ray vector. The image captured by the given sampling camera is projected on the uv plane and the intersection points a'_{uv}, a''_{uv} and b'_{uv}, b''_{uv} determine which pixels are taken into 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 are sampling the geometry in different places) which leads to blurry image as shown in Figure 4.

Even if one focusing distance is enough for the user, the resulting out-of-focus effect is simply created by composing the images on top of each other, resulting in block artifacts caused by the discrete light field representation as demonstrated in Figure 4. The sparser the light field image grid representation is, the higher is the amount of block artifacts due to the inability to reconstruct the continuous light field. Again, depth or disparity information would be necessary to decide which parts of the image should be filtered to simulate a smooth blur effect.

4 Proposed rendering method

The proposed method consists of two steps. The first step involves a focus map generation where each pixel of the map contains a focusing value. The second step is the final image composition using the focus map values. In the end, each pixel has its own focusing value, eliminating the single global focusing distance related

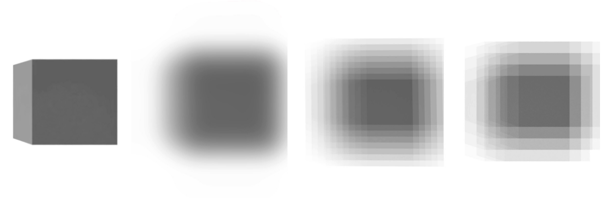

Fig. 4 The leftmost picture shows the original cube object which is in focus. The second picture is a ground truth out-of-focus cube where the defocusing is simulated using Gaussian blur filter. Next ones show the defocusing generated by shifting the focusing plane in two planes method when using 8 × 8 and 4 × 4 light field grid respectively. Block artifacts are visible with the decreasing grid dimensions.

issues as shown in Figure 2.

4.1 Weighted shift-sum algorithm

In the shift-sum algorithm [2], the output pixel is a result of a sum of pixels from different views (the input images from the camera grid). The resulting pixel values contain lightning 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. The pixels capturing the objects in the scene that share the same distance from the camera grid overlap in the resulting sum. Despite being from different views, their colors are similar or the same. To achieve the 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 the images, just those that are within a certain distance from the virtual view. The distance is defined globally, as the same value for all pixels. The algorithm is described by Equation 1 with the description of each variable in Table 1. One iteration of the algorithm is depicted in Figure 5.

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

4.2 Focus map

The generation of the focus map is based on a similar concept as the semi-global matching method [17], searching for the disparity value in a given range with the lowest cost. The weighted shift-sum algorithm is used to generate new views in various focusing distances. When the whole focusing range is iteratively scanned, each pixel is in a certain iteration in focus. The number of tested distances depends on how densely the focusing range is sampled and can be increased for wide 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 proven to be the most suitable metric in this case; generally the choice of color metric for given task is problematic [33]). The variance is calculated using the mean value from the set of colors of the contributing pixels in the shift-sum. At the end, the pixel with the lowest variance value 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 is recalculated in fragment shader. This way, the necessary bit depth of the map needs to cover just the number of the searched distances. This statistical analysis of the contributions to the final pixel value can determine whether the pixel is focused.

i	index of current input view
p_o	function computing the output pixel
p_i	pixel from ith view
n	number of input views
$\overline{c}$	coordinates in the output image
$\overline{o}_i$	offset between images in the grid
f	view shift (focus distance)
w_i	weight of the pixel

Tab. 1 Description of each variable used in Equation 1.

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 the Equation 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 Figure 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 ($\overline{c} + \overline{o}_i \cdot f$ from Equation 1, where the offset is a shift of the sampled image from the synthetic one). The focusing distance was previously acquired from the Algorithm 1. While in the focus map generation, a variance was the desired result of the summation, now the resulting color is used for the final image. In the presented algorithm, it is

```
Data: Grid of images, position of virtual camera,
      focusing bounds, focus step
Result: Focus map, focused pixel color
for $c = 0; c < focusMapPixels.size(); c++$ do
    variances = Array[focusLevels];
    colors = Array[focusLevels];
    for $i = 0; i < focusLevels; i++;$ do
        f = focusStart + i*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
```

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 (Equation 1) and returns the final color and variance of the colors, contributing in the summation, using Algorithm 2. This algorithm was generalized, returning also the focused pixel color. The image synthesis is, however, separated in the reference implementation.

possible to acquire the final image color directly but the focus map generation and the final image composition steps are separated, so each one can produce the result in a different resolution. In the 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. A better performance without significant quality loss can be achieved by the generation of the focus map in a lower resolution than the final image, as had been proven experimentally. That is the key element for a real-time usage. Sample result of both final image and focus map is shown on Figure 6.

5 GPU utilization scheme

The method can exploit massive parallelism available on GPU architectures. OpenGL was used for both rendering and GPGPU computations in the reference implementation. The focus map generation is performed in 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 a good GPU occupancy and memory access coherency, allowing an in-warp data transfer between the threads which is much faster than using the global or local memory. Each thread is computing one focusing distance (or more when denser search

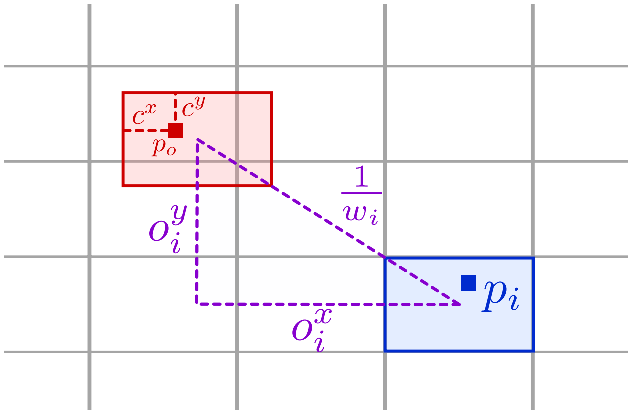

Fig. 5 One iteration of the shift-sum based image synthesis where a pixel from *ith* image is taken into the summation of the output pixel p_o. The red box depicts the new synthetic image, the purple lines are showing the offsets relative to the currently sampled image and the distance between the two images. The currently sampled pixel's weight (w_i) depends on the distance between the two images. Used symbols correspond to the Equation 1. The x and y superscript denotes the first and the second element of the vector variable.

is required), using the weighted shift-sum and the Welford's variance algorithm [42] (Algorithm 2) which improves the GPU occupancy by reducing the necessary number of registers. At the end, the minimal variance value within a warp is being 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, the missing pixels can be interpolated in texturing units if the resolutions of the result and the focus map differ. Figure 7 describes the work distribution on GPU.

6 Evaluation

The purpose of the first experiment was to determine which color distance metric would be the most suitable one when computing the variance from the resulting color summation. The overall quality was measured in the second experiment, evaluating how good visual results can this method reach. The third experiment had been carried out to find out 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 had been performed to decide how many images from the input grid need to be sampled and the last experiment analyzed how the camera grid parameters of the dataset

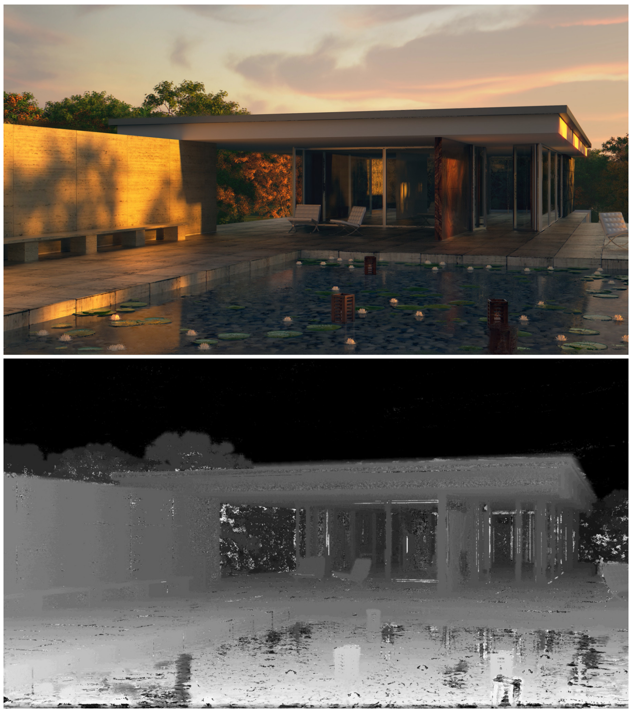

Fig. 6 Focused result using the *Pavilion* dataset with a correspondent focus map. The focus map contains estimated focusing value for each pixel of the final image. The map resembles depth or disparity map for the given synthetic view because the focusing values depend on the distance of the pixel from the camera.

```
Data: Stream of pixel values
Result: Estimated variance
n = 0;
mean = 0;
$m_2$ = 0;
for pixel in input do
    n++;
    delta = pixel-mean;
    distance = pixelDistance(pixel, mean);
    mean += delta/n;
    $m_2$ += distance*pixelDistance(pixel, mean);
end
$m_2$ /= n-1;
```

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

affect the quality of the results of the proposed method.

Datasets used in the experiments captured with camera array come from Stanford light field archives [41], light field captured by the plenoptic camera Lytro Illum belongs to EPFL Light-field dataset [31] and synthetic dataset was rendered on Barcelona Pavilion scene which is available at the Blender demo files page [8]. Only one Lytro dataset was used because the distance between Lytro views is very small due to its capturing mechanism 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, creating the 3D viewing effect. In all experiments, a ground truth center view from the original dataset was chosen as a reference and it was compared using SSIM and PSNR metrics to a new synthetic view rendered by the proposed method. *Pavilion* dataset was used for the performance tests because its resolution is big enough and reflects the commonly used FullHD video standard. One dataset is enough for performance tests because the computation time of the proposed method depends only on its parameters and dataset resolution and not on the content of the input images. All experiments were executed on a machine equipped with Nvidia GeForce RTX 2070 GPU and Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz CPU, running Arch Linux.

6.1 Color distance metric

The variance computation phase in the proposed algorithm requires a pixel color value distance metric to decide how much do two pixels differ in terms of color similarity. The right choice of the metric depends on various aspects such as expected color range, type of images or a final use-case. The first measurement was the comparison of various RGB color distance metrics to find out which one would yield the best visual quality results for light field datasets as showed in Figure 8. The quality differences were not significant but computational complexity of the metrics differed and might negatively affect the performance (e.g. DeltaE). Chebyshew metric was chosen for further experiments because of high-quality results and computational simplicity.

6.2 Overall quality

For each dataset (Figure 9), the best initial focusing level and search step was manually found and the resulting images were compared to the reference. Final visual quality is evaluated in Figure 10. The images are focused in all parts, but interpolation artifacts are visible in the problematic parts such as around thin edges or near similarly colored areas. A detailed look at the interpolation artifacts is captured in Figure 11. *Bunny* dataset contains only diffuse material and is clearly separated from the black background; therefore, the reconstruction had minimum artifacts.

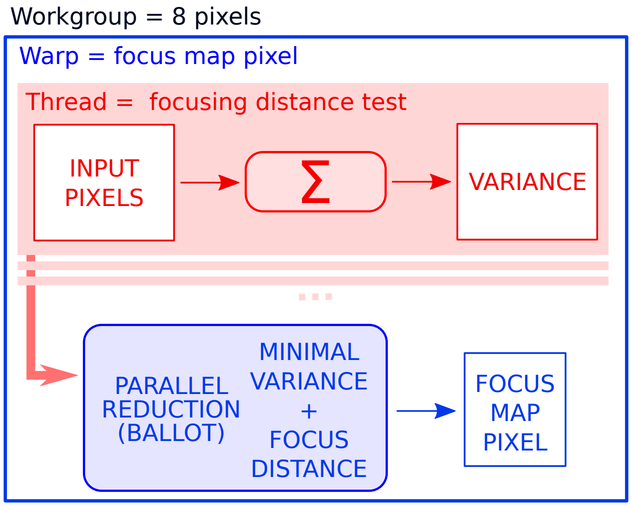
Workgroup = 8 pixels
Warp = focus map pixel
Thread = focusing distance test
INPUT PIXELS
Σ
VARIANCE
...
PARALLEL REDUCTION (BALLOT)
MINIMAL VARIANCE + FOCUS DISTANCE
FOCUS MAP PIXEL

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 a minimal variance in the focus map. Because the workload is divided into warp sized elements no global or local synchronization is needed.

Even though *Chess* dataset contains a lot of reflections the chessboard pattern along with a relatively small distance between the views improved the quality of the result. *Buldozer* contains a lot of small details that are clearly separated from the yellow construction of the model which again causes higher variance values when mixing nearby pixels. *Lego* dataset is filled with a single color area where for example on the wall in the back small edges or details are hard to detect, and the pixels interpolated from surrounding area might 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 are containing subtle noise that negatively affects the evaluation. *Pavilion* contains both big similar colored areas and complex objects with many details but the distance between cameras is a bit bigger to allow more freedom when moving the virtual camera. Figure 12 shows the elapsed times of focus map generation and final compositing of pixels from each dataset.

6.3 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 can be roughly compared to depth or disparity map estimation algorithms. Table 2 is an indicative overview of computation times of this stage.

A side-by side visual quality comparison with state of the art methods is shown in Figure 19. Methods that are capable of producing the synthetic view directly from the images were chosen for the evaluation. The proposed method outperforms other similar approaches. View reconstruction on *Bunny* dataset, using the biggest competitor, the shearlet approach [40], measured on GeForce GTX Titan X takes 5 s which is unsuitable for real-time rendering.

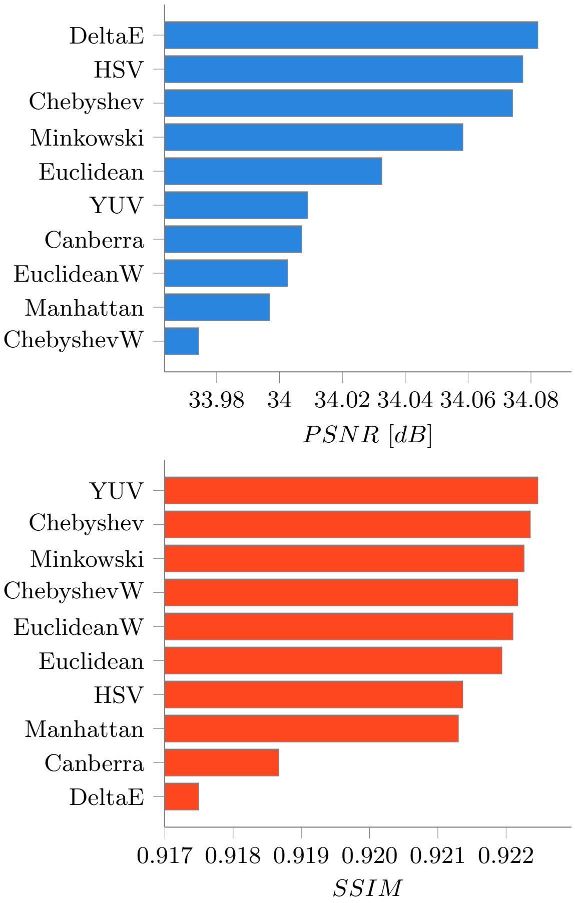

Fig. 8 A comparison of the RGB color distance metrics for the pixel similarity test during the variance computation phase. The *W* suffix at metric name stands for weighted metrics. Average results from all tested datasets are presented.

The proposed method does not reach the same visual quality as newer learning based methods [29] when measured on the same dataset that was used in the original paper, but slightly outperforms older methods [21] (indirect comparison on *Kitchen* and *Museum* datasets, difference about 1 dB [29]). The proposed method, however, does not depend on the training process.

Measurements show that the new proposed method is comparable to the other published algorithms in terms of visual quality, reaching performance suitable for real-time rendering. Rendering times can be further

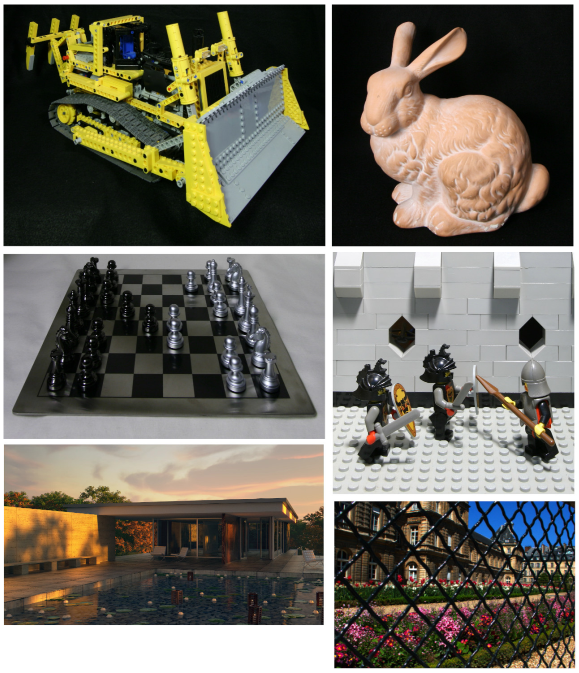

Fig. 9 Reference views from each dataset used in the experiments. From top left: Buldozer, Bunny, Chess, Lego, Pavilion, Lytro.

Method	Architecture	Resolution	Time
Proposed	RTX 2070	1920 × 1080 × 64	18 ms
[4]	Tesla C2050	640 × 480 × 2	16 ms
[3]	E3-1245 V2	541 × 376 × 9	1.5 s
[20]	i7 2.8GHz	512 × 512 × 49	13 min
[9]	i7-6700k	512 × 512 × 49	0.8 s
[10]	Quadro M1000M	1920 × 1080 × 45	1.58 s

Tab. 2 Overview of computation times of state of the art depth or disparity estimation methods from light fields.

improved by slight reduction of visual quality as shown in Figure 13.

6.4 Focus map resolution

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

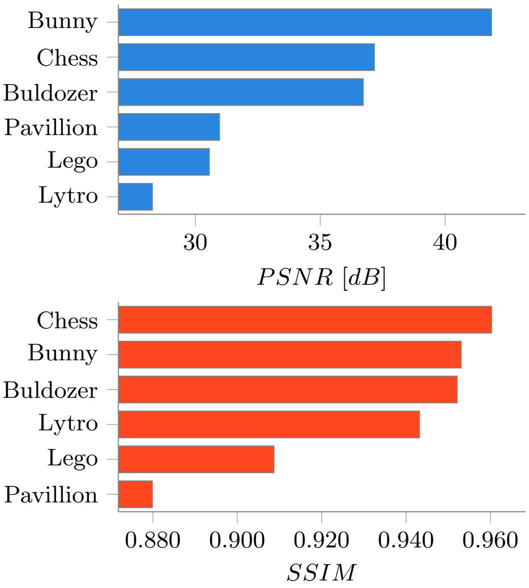

Fig. 10 Best results when rendering a new view from each dataset compared to the ground truth. Rendering settings were manually adjusted to reach the best visual quality. Some of the results are shown in Figures 2, 6, 11, 14 and 19.

6.5 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 given range does not affect the visual quality significantly and slows down the computation unnecessarily as shown in the Figure 16. 32 samples proved to be an optimal choice for most of the datasets. The most significant difference in quality was measured on *Pavilion* dataset which has the biggest depth range which is the only case where denser searching is necessary, especially when the objects in the scene are linearly distributed over the whole depth range.

6.6 Camera grid sample radius

The experimental results shown in Figure 17 show how many images need to be sampled when getting the pixel values for the resulting pixel sum. The plots show that sampling window in the input grid gives optimal results when having radius about 2 grid views wide. The value might slightly differ, depending on the dataset. Wider radius leads to more texture reads and excessive memory access which slows down the computation most. The sample distance is a radius of a circle with virtual camera position as its center. Surrounding images from the grid in distance from zero to the sampling distance radius are taken into account during the interpolation. When the radius is too wide, images from distant places in the grid might

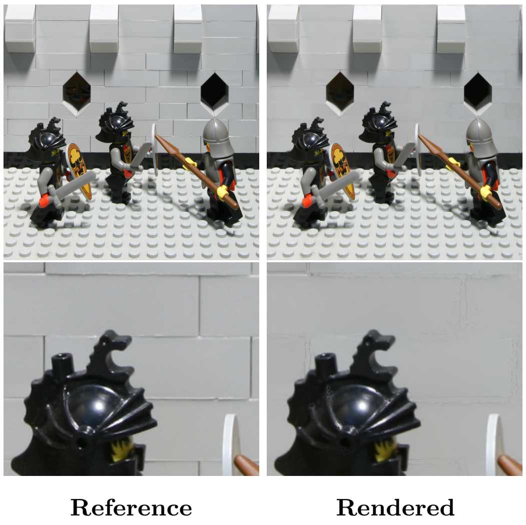

Fig. 11 Reference images are placed in the left column. Right column contains rendered reconstructed images with zoomed detail of interpolation artifacts caused by incorrect focusing level estimation in the affected pixels.

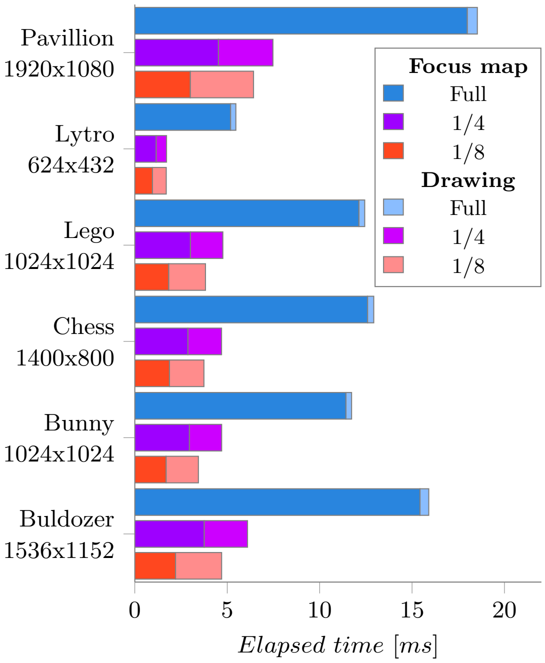

Fig. 12 Elapsed time of the focus map generation and drawing which depends on the focus map and resulting image resolution respectively. Full, 1/4 and 1/8 sized focus map is used in these measurements. Drawing time slightly increases when using smaller focus map most likely due to coordinates interpolation in texturing units due to resolution mismatch.

add unwanted ghosting artifacts in the final result, forcing the algorithm to use views that are showing the scene from a different angle than expected.

6.7 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 as showed in Figure 18. The camera setup used in the scene can be viewed in Figure 15. With an increasing space between cameras or decreasing field of view (increasing focal length), the differences between views increase and the interpolation is more prone to visual artifacts. On the other hand, the more different the camera positions of view cones are, the more freedom is gained for the virtual camera. This issue can be solved with denser sampling [7], providing more views in the grid, increasing its dimensions. This, however, leads to higher memory or bandwidth requirements.

7 Conclusion

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, compared to 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 in low enough times suitable for interactive applications while maintaining a good visual quality of the result. The method uses a simple statistical analysis of the colors contributing to each pixel of the final result. Each resulting pixel value can be computed independently on the rest of the image without an excessive memory access. The proposed principle is general enough to be used with every commonly used light field representation or parametrization. This research also revealed important information about the relation between the visual quality and computation time when adjusting parameters of the interpolation shift-sum algorithm. Massive parallelism of GPU allows this method to run in real-time even on 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 the current plenoptic cameras.

Visual artifacts are visible in the current version of the proposed method. They are caused by wrong focusing distances for the given pixels. The statistical method might fail if the tested pixel is blurred in a way that the resulting variance is actually lower than the one obtained by analyzing the correct focusing

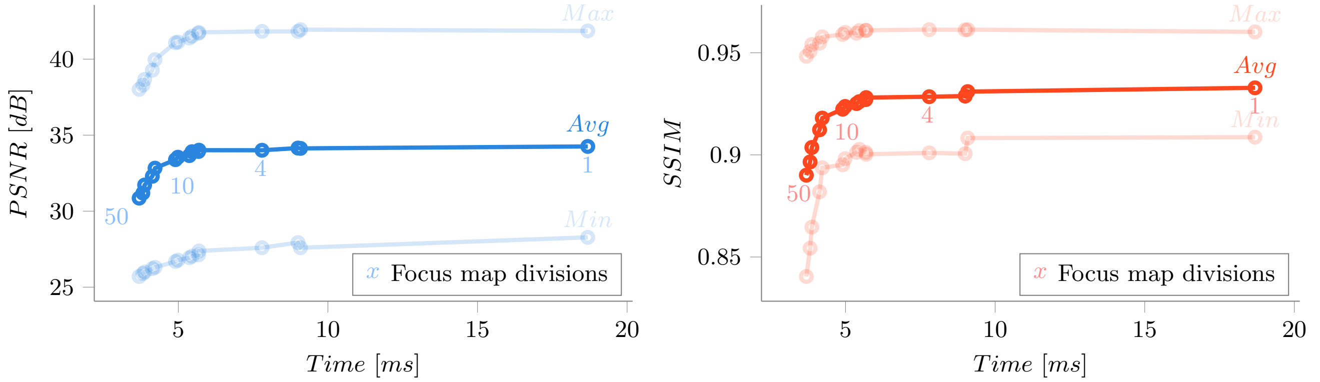

Fig. 13 Relation of visual quality, computation time, and amount of focus map dimensions division. The results are averaged from all tested datasets.

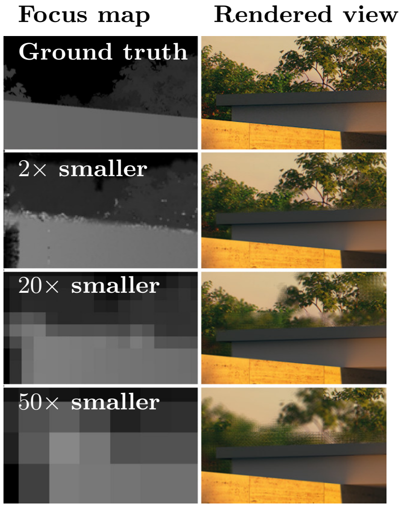

Fig. 14 Focusing artifacts caused by low resolution focus map. The first image shows reference image with generated depth map. The other ones are results with focus map dimensions divided by 2, 20 and 50 with the used focus maps.

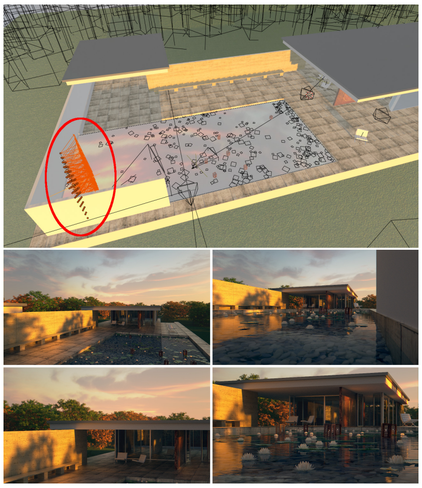

Fig. 15 The size of the grid (red circle) in *Pavilion* scene and the value of field of view was animated and resulting reconstruction quality was measured. Two views from the corners of the grid using 25mm focal length are placed in the middle and 55mm ones at the bottom. The difference between views is bigger in the 55mm version.

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 minimums might be a way to select a better focusing value.

As a future work, additional experiments with focus map filtering will be carried out. Preliminary tests showed that median filter might be used to denoise the map slightly, improving the visual quality. Resulting focus map can be also used to simulate additional photographic effects such as depth of field. It is necessary to define focusing range for each dataset manually. The searching bounds might be estimated automatically based on another statistical analysis detecting 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.

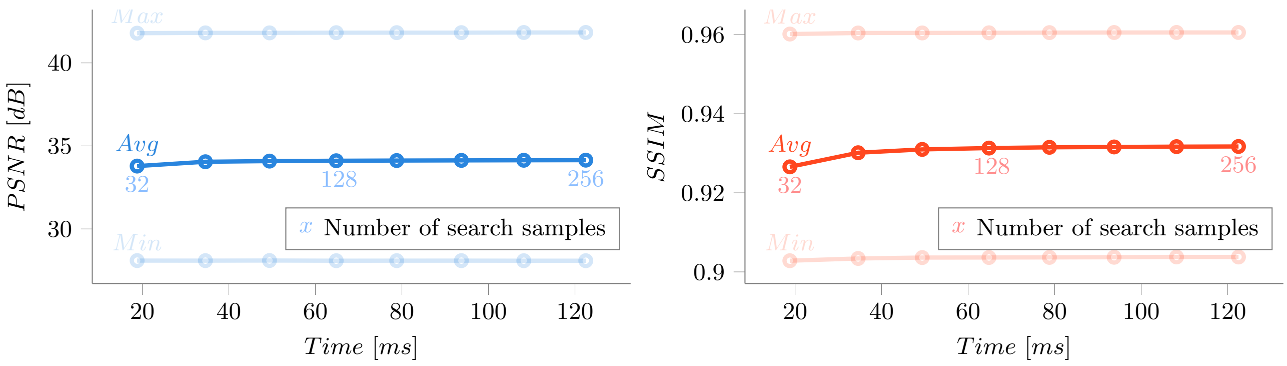

Fig. 16 The plot shows how is the overall quality of the result affected when searching the focusing range more densely. The quality metric values are average results from all tested datasets.

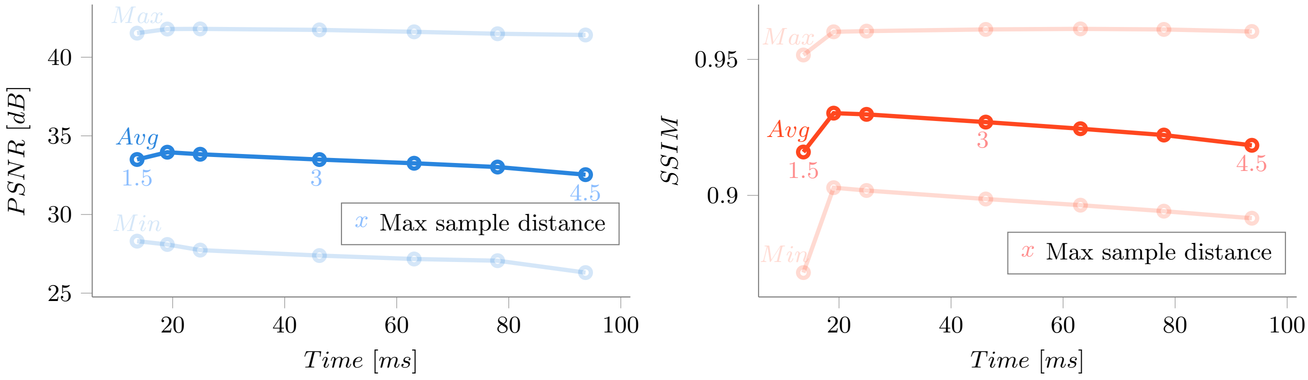

Fig. 17 Maximal sample distance parameter and its relation to both visual quality of the result and computation time. The results are average results from all tested datasets.

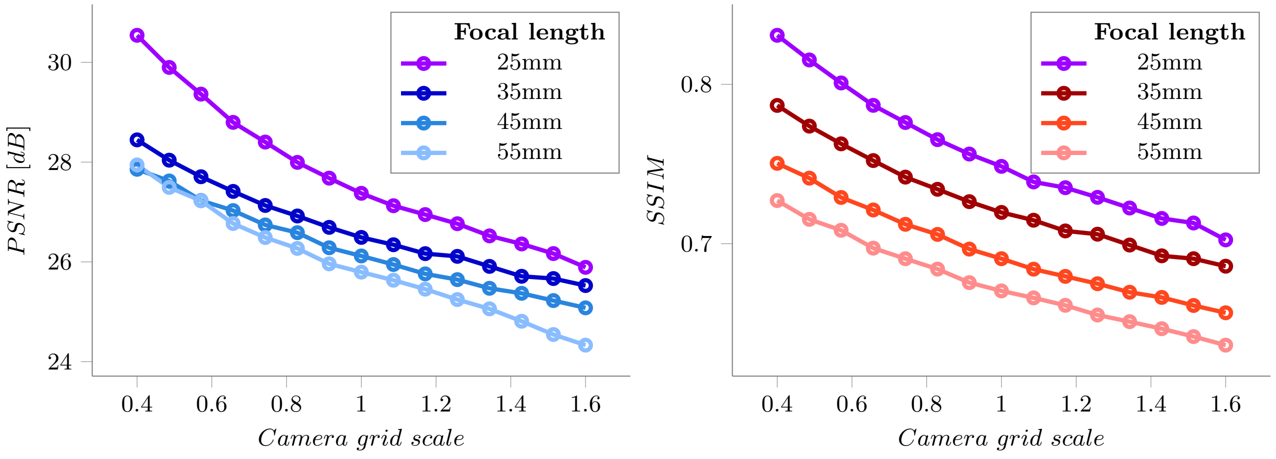

Fig. 18 The camera grid contains 8x8 cameras and is initially 2m wide in the scene-space. The first visible surface is about 1m far from the grid, and the furthest visible spot excluding the sky is about 90m away. The camera grid is uniformly scaled up and down to change the distance between cameras.

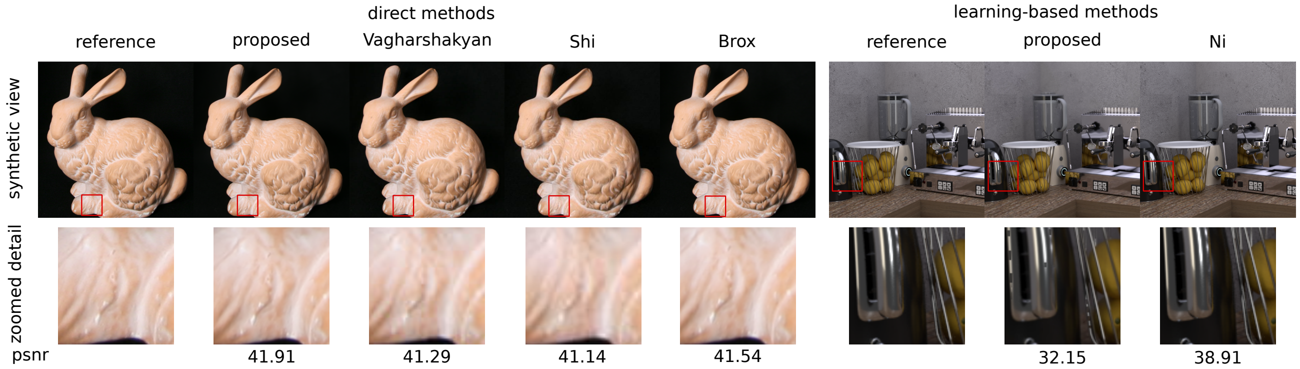

Fig. 19 Side-by-side visual comparison with other state of the art methods, rendering a new synthetic view. The proposed method outperforms other general methods but does not reach the same quality as learning based methods, trained on the specific dataset. The results of direct methods do not show any significant differences and are almost identical with difference below 1 dB from the proposed method. Slight, few pixels large, blur artifacts are visible around certain details in all cases. The proposed method produces the sharpest result. New learning based methods produce better results in parts of the image with thin and reflective objects. They, however, depend on the training process and dataset. Reflections and thin details can cause problems in the proposed method, when comparing pixel colors from different views. The proposed method was compared to Vagharshakyan [40], Shi [35], Brox [5] and Ni [29].

References

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[2] M. Alain, W. Aenchbacher, and A. Smolic. Interactive light field tilt-shift refocus with generalized shift-and-sum. *ArXiv*, abs/1910.04699, 2019.

[3] Y. Anisimov, O. Wasenmüller, and D. Stricker. Rapid light field depth estimation with semi-global matching. *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 109–116, 2019.

[4] C. Banz, H. Blume, and P. Pirsch. Real-time semi-global matching disparity estimation on the gpu. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 514–521, 2011.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. volume 3024, pages 25–36, 01 2004.

[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*, SIGGRAPH '01, page 425–432, New York, NY, USA, 2001. Association for Computing Machinery.

[7] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 307–318, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[8] H. Cheggour. Blender demo files - barcelona pavillion.

[9] Y. Chen, M. Alain, and A. Smolic. Fast and accurate optical flow based depth map estimation from light fields. In *Irish Machine Vision and Image Processing Conference (IMVIP)*, 2017.

[10] A. Chuchvara, A. Barsi, and A. Gotchev. Fast and accurate depth estimation from sparse light fields. *IEEE Transactions on Image Processing*, 29:2492–2506, 2020.

[11] P. Debevec, Y. Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In G. Drettakis and N. Max, editors, *Rendering Techniques '98*, pages 105–116, Vienna, 1998. Springer Vienna.

[12] D. Eigen, C. Puhrsch, and R. Fergus. 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 - Volume 2*, NIPS'14, page 2366–2374, Cambridge, MA, USA, 2014. MIT Press.

[13] S. M. A. Eslami, D. Jimenez Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[14] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 43–54, New York, NY, USA, 1996. Association for Computing Machinery.

[15] X. Han, H. Laga, and M. Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2019.

[16] B. Heigl, R. Koch, M. Pollefeys, J. Denzler, and L. Van Gool. Plenoptic modeling and rendering from

image sequences taken by a hand-held camera. 08 2000.

[17] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, Feb 2008.

[18] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 297–306, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[19] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. S. Kweon. Accurate depth map estimation from a lenslet light field camera. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1547–1555, 2015.

[20] X. Jiang, M. L. Pendu, and C. Guillemot. Depth estimation with occlusion handling from a sparse set of light field views. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 634–638, Oct 2018.

[21] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. Graph.*, 35(6), Nov. 2016.

[22] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. volume 2352, 12 2001.

[23] A. Kubota, K. Takahashi, K. Aizawa, and T. Chen. All-focused light field rendering. In *Rendering Techniques*, 2004.

[24] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery.

[25] C. Li and X. Zhang. High dynamic range and all-focus image from light field. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 7–12, 2015.

[26] H. Lin, C. Chen, S. B. Kang, and J. Yu. Depth recovery from light field using focal stack symmetry. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3451–3459, 2015.

[27] Z. Lin and H.-Y. Shum. A geometric analysis of light field rendering. *Int. J. Comput. Vision*, 58(2):121–138, July 2004.

[28] A. Neri, M. Carli, and F. Battisti. A multi-resolution approach to depth field estimation in dense image arrays. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3358–3362, 2015.

[29] L. Ni, H. Jiang, J. Cai, J. Zheng, H. Li, and X. Liu. Unsupervised Dense Light Field Reconstruction with Occlusion Awareness. *Computer Graphics Forum*, 38(7):425–436, 2019.

[30] J. Peng, Z. Xiong, D. Liu, and X. Chen. Unsupervised depth estimation from light field using a convolutional neural network. In *2018 International Conference on 3D Vision (3DV)*, pages 295–303, Sep. 2018.

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

[32] N. Sabater, G. Boisson, B. Vandame, P. Kerbiriou, F. Babon, M. Hog, R. Gendrot, T. Langlois, O. Bureller, A. Schubert, and V. Allié. Dataset and pipeline for multi-view light-field video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1743–1753, 2017.

[33] A. T. Sanda Mahama, A. S. Dossa, and P. Gouton. Choice of distance metrics for rgb color image analysis. *Electronic Imaging*, 2016:1–4, 02 2016.

[34] D. C. Schedl and O. Bimber. Optimized sampling for view interpolation in light fields with overlapping patches. In *Proceedings of the 39th Annual European Association for Computer Graphics Conference: Short Papers*, EG, page 17–20, Goslar, DEU, 2018. Eurographics Association.

[35] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand. Light field reconstruction using sparsity in the continuous fourier domain. *ACM Trans. Graph.*, 34(1), Dec. 2015.

[36] K. Sugita, T. Naemura, H. Harashima, and K. Takahashi. Focus measurement on programmable graphics hardware for all in-focus rendering from light fields. In *Virtual Reality Conference, IEEE*, page 255, Los Alamitos, CA, USA, mar 2004. IEEE Computer Society.

[37] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, June 2010.

[38] K. Takahashi, A. K. T. Naemura, and T. Naemura. All in-focus view synthesis from under-sampled light fields, 2003.

[39] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *2013 IEEE International Conference on Computer Vision*, pages 673–680, 2013.

[40] S. Vagharshakyan, R. Bregovic, and A. Gotchev. Light field reconstruction using shearlet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):133–147, Jan 2018.

[41] V. Vaish and A. Adams. The (new) stanford light field archive. *Computer Graphics Laboratory, Stanford University*, 6(7), 2008.

[42] A. B. P. Welford and B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, pages 419–420, 1962.

T. Chlubna 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 the 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 the M.Sc. degree. His research is focused on light field and shadow rendering.

P. Zemčík is dean and Full Professor of 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.